



UNIVERSITÀ DEGLI STUDI DI SALERNO

STUDI DI SCIENZE MATEMATICHE,
FISICHE E NATURALI

ITADINFO 2025

Terzo Convegno Italiano sulla Didattica dell'Informatica

A cura di
Maria Angela Pellegrino e Veronica Rossano

Collana Scientifica dell'Università di Salerno

Studi di Scienze Matematiche, Fisiche e Naturali



UNIVERSITÀ DEGLI STUDI DI SALERNO
DIPARTIMENTO DI INFORMATICA



APS
Programma
il Futuro



Progetto Nazionale "Informatica"
del Piano Lauree Scientifiche



consorzio
interuniversitario
nazionale
per l'informatica

[ITADINFO]

3° CONVEGNO ITALIANO SULLA DIDATTICA DELL'INFORMATICA

Salerno, 03-05 ottobre 2025

CON IL SOSTEGNO DI

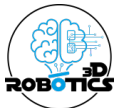


Fondazione
CRT



Progetto
DIDEROT

SPONSOR ESPOSITORI



THINK CLOUD

CON IL PATROCINIO DI



I contributi qui pubblicati sono stati revisionati dal Comitato Scientifico del Convegno con *single blind peer-review*.

© 2025 Ledizioni LediPublishing
Via Boselli, 10 – 20136 Milano – Italy
www.ledizioni.it
info@ledizioni.it

ITADINFO 2025. Terzo Convegno Italiano sulla Didattica dell'Informatica

Impaginazione, editing e revisione del presente volume a cura di Maria Angela Pellegrino e Veronica Rossano

I contenuti del presente volume sono pubblicati con la licenza Creative Commons 4.0 International Attribution-NonCommercial-ShareAlike.



Alcuni diritti sono riservati

Prima edizione: ottobre 2025

ISBN cartaceo: 9791256005123
ISBN eBook: 9791256005130
ISBN PDF online: 9791256005147

Copertina e progetto grafico: ufficio grafico Ledizioni

Informazioni sul catalogo e sulle ristampe: www.ledizioni.it

Le riproduzioni a uso differente da quello personale potranno avvenire, per un numero di pagine non superiore al 15% del presente volume, solo a seguito di specifica autorizzazione rilasciata da Ledizioni.

Indice

Prefazione	13
------------	----

RELAZIONI INVITATE

Educare al pensiero algoritmico: riflessioni e proposte per il primo ciclo <i>Marta Crippa, Violetta Lonati, Luca Lamanna, Laura Branchetti</i>	19
Le <i>Notional machine</i> per l'apprendimento della programmazione, anche nell'era dell'AI generativa <i>Michael Lodi, Simone Martini</i>	29

RACCONTI DI ESPERIENZE SUL CAMPO E RELAZIONI SCIENTIFICHE PER L'INFANZIA E LA PRIMARIA

Coding per tutti i gusti: come creare confusione negli allievi più piccoli <i>Gaetano Impoco</i>	39
Robo Amici della Terra <i>Daniela Troia</i>	43
Il linguaggio dei cristalli: simboli e narrazione per l'informatica nella scuola primaria <i>Guglielmo Abbruzzese, Luciana Mattei, Maurizia Di Fiore, Luca Forlizzi</i>	55
Coding, logica e movimento nella scuola dell'infanzia: un percorso laboratoriale tra carta, corpo e digitale <i>Mariarosaria Cianelli</i>	67
Attività didattiche di programmazione a partire dai giochi dell'informatica Bebras <i>Martina Palazzolo</i>	75
Programmare la parità: la robotica educativa per l'inclusione di genere <i>Laura Cesaro, Emanuele Mengatti</i>	83

RACCONTI DI ESPERIENZE SUL CAMPO E RELAZIONI SCIENTIFICHE PER LA PRIMARIA E LA SECONDARIA DI PRIMO GRADO

“Io Studio Informatica”. Una piattaforma basata sul modello UMC <i>Enrica Gentile, Paola Plantamura, Francesco Scarsi, Francesca Pia Trivisani</i>	93
---	----

Percorsi formativi di accompagnamento per docenti sul coding, machine learning e l'intelligenza artificiale generativa applicata alla didattica <i>Giovanni Silvestro, Marta Sanz Manzanedo</i>	103
--	-----

Il <i>Server</i> : proposta per un percorso didattico <i>Daniele Catozzella</i>	113
--	-----

Il potenziamento di tecnologia per promuovere la didattica dell'informatica <i>Nicola Mendolia</i>	117
---	-----

Coding e creatività per lo sviluppo del pensiero computazionale nel primo ciclo: un'esperienza didattica all'IC Praia a Mare <i>Emilia Ciccia, Teresa Agrello, Elena Castiglioni, Rosangela Papa, Marilena Salsano</i>	123
---	-----

RACCONTI DI ESPERIENZE SUL CAMPO E RELAZIONI SCIENTIFICHE PER LA SECONDARIA DI SECONDO GRADO

Machine Learning Unplugged e Python: un'esplorazione pratica <i>Francesco Picca</i>	143
--	-----

Insegnare informatica nell'era dell'IA: pratiche, sfide e prospettive <i>Paolo Ciancarini, Marcello Missiroli, Giancarlo Succi</i>	153
---	-----

Hackathon a scuola: un'esperienza didattica per lo sviluppo delle competenze non cognitive <i>Laura Castellana, Giuseppe Mastrandrea</i>	159
---	-----

Il modello tra problema e soluzione: un percorso tra competizione e consapevolezza <i>Fabrizio Marinelli, Gionata Massi</i>	167
--	-----

Codice che "pensa". Modelli ed algoritmi di intelligenza artificiale con Python per la didattica STEAM <i>Flavia Giannoli</i>	173
--	-----

Tre concetti chiave per un corso sul Machine Learning alle scuole superiori <i>Nicola Dalla Pozza, Emanuele Scapin</i>	179
---	-----

NextPyter: piattaforma collaborativa e open-source per la didattica dell'informatica <i>Linda Burchiellaro, Francesco Fuenzu, Claudia Canali</i>	185
---	-----

Dalla scuola allo Spazio: attività di coding con Python e Sense HAT per apprendere l'Informatica <i>Giuseppe Fiamingo</i>	191
--	-----

Educare all'Informatica con SteamCity <i>Manon Ballester, Mauro D'Angelo, Sébastien Nedjar, Maria Angela Pellegrino</i>	201
--	-----

Prototipo di un gioco unplugged per insegnare l'IA generativa: comprendere costi, bias e sostenibilità	209
<i>Leonardo Martino, Gennaro Iaccarino, Maddalena Braccisi</i>	

CriptoLab... alla scoperta della crittografia. Dalla teoria degli algoritmi al coding crittografico	217
<i>Francesco Paolo Caforio</i>	

Didattica del prompting: "Salviamo il pianeta"	225
<i>Luca Basteris, Maria Cristina Daperno, Alessandra Vassallo, Valerio Milano</i>	

Storie sonore di donne nella Scienza: narrazione, musica, coding e Intelligenza Artificiale	239
<i>Ersilia Pagano</i>	

Insegnare Informatica con progetti autentici: il monitoraggio delle acque del fiume Aterno	245
<i>Luca Forluzzi, Maria Luzzi, Giovanna Melideo, Giovanna Patrizio, Alessia Verticchio</i>	

Generazione digitale, ma non consapevole: giovani e IA fra percezioni e pratiche	251
<i>Laura Cesaro, Giovanni Doderò</i>	

Liceo Classico Digitale: proposta contenuti disciplinari e prime sperimentazioni	259
<i>Simone Cuconato, Kristian Reale</i>	

Aula Nova: Progettazione delle lezioni di Informatica nella scuola secondaria di secondo grado con i LLM	263
<i>Alexandra Shbeykina, Gabriele De Vito, Andrea De Lucia</i>	

RACCONTI DI ESPERIENZE SUL CAMPO E RELAZIONI SCIENTIFICHE PER OGNI ORDINE E GRADO

Vibe Coding: un paradigma emergente per la programmazione assistita da IA nella didattica	269
<i>Andrea Canesi, Marco Canesi</i>	

LABORATORI PER L'INFANZIA E LA PRIMARIA

ScratchJr: un laboratorio di coding per la scuola dell'infanzia e primaria	277
<i>Federica Luzzi</i>	

Introduzione al Machine Learning alla scuola primaria	279
<i>Maria Cristina Carrisi, Sara Vergallo, Mirko Marras</i>	

Salta, balla e rotola con il coding e l'AI	281
<i>Luca Basteris, Maria Cristina Daperno</i>	

LABORATORI PER LA SECONDARIA DI PRIMO GRADO

Blockchain@School <i>Lorenzo Guasti, Maria Angela Pellegrino</i>	285
La crittografia? È un gioco! <i>Arianna Boldi, Sara Capecechi, Iary Davidson, Marco Viola</i>	287

LABORATORI PER LA SECONDARIA DI SECONDO GRADO

Dalle foglie alle radici: imparare il Debugging dalle sue componenti fondamentali <i>Gabriele Pozzani, Tullio Vardanega</i>	291
Laboratorio “La Natura nel Computer” <i>Claudio Mirolo</i>	293
Cos'è e come funziona un Sistema Operativo? Scopriamolo con il ristorante Brachetti <i>Renzo Davoli</i>	295
Cosa si cela dietro ai pixel: un gioco per creare un gioco <i>Calogero Carlino, Luca Fortizzzi</i>	297
Sentiamo l'AI attraverso Musica e Suoni <i>Giorgio Delzanno, Giovanna Guerrini, Guido Vallarino</i>	299

POSTER PER LA PRIMARIA

La rappresentazione del dato “Immagine”: progettazione e sperimentazione di un percorso per il primo ciclo scolastico <i>Annunziata Marra</i>	303
--	-----

POSTER PER LA PRIMARIA E LA SECONDARIA DI PRIMO GRADO

Giochiamo con il linguaggio binario <i>Anna Rita Colella</i>	317
Scopri, crea e impara! <i>Paola Di Mizio</i>	323
Moon Camp: esplorare la Fisica con le Mani <i>Giuliana Gelsomino</i>	331
FOSTEM per orientamento e sensibilizzazione all'Informatica in scuole secondarie di primo grado in Campania <i>Valentina Casola, Cristina d'Alessandro, Christian Esposito</i>	335

Informatica nella Natura: ambienti aumentati per l'orientamento verticale e la promozione delle competenze digitali <i>Alessia Galli, Lara Rollo</i>	343
---	-----

POSTER PER LA SECONDARIA DI SECONDO GRADO

Educare all'Intelligenza Artificiale: un percorso di conoscenza e consapevolezza <i>Giacomo Salvi</i>	351
Robot, Logica e Creatività: l'esperienza "LucarelliRobot" tra STEM e futuro <i>Silvio Dell'Oste</i>	357
Educational Promptization: una nuova prospettiva per l'insegnamento dell'Informatica nella scuola secondaria <i>Erica Perseghin</i>	365
"Chi ha paura di un LLM?": un approccio psico-didattico all'integrazione di un LLM nelle lezioni di informatica al triennio <i>Michele Iacobellis</i>	375
GEM FLEXNAO: Tecnologia in movimento per il benessere del domani <i>Veronica Caricchi, Matteo Bettini, Nikolai Carnaghi, Amir Shams Eddin, Saverio Venturi, Ejuxchin Hoxha, Astik Gautam, Letizia Galletta</i>	383
Programmazione C++ applicata alla stampa 3D: un'esperienza interdisciplinare per l'apprendimento dei cicli iterativi nella scuola superiore <i>Sonia Guerzi, Federico Florit</i>	389
UNIUD Game Jam: Esperienza Formativa e di Orientamento attraverso lo Sviluppo di Videogiochi <i>Biagio Tomasello, Alessandro Forgiarini, Massimiliano Pascoli, Christian Corrà, Fabio Buttussi</i>	395
Verso una didattica della comunicazione con l'AI <i>Francesco Lombardi</i>	401
Una raccolta di modelli in linguaggio Python per la didattica delle applicazioni di ML <i>Giuliana Barberis</i>	407
Dal concept al gioco: una testimonianza sull'insegnamento della programmazione attraverso il game making nei coding camp ibridi <i>Corrado Cristallo, Alessandro Pagano, Veronica Rossano</i>	421
Servizi di Rete <i>Daniela Decembrino</i>	427

POSTER PER OGNI ORDINE E GRADO

EduPKG: un'Ontologia per la didattica STEM	433
<i>Paolo Campanelli, Alessandro Marcelletti, Barbara Re</i>	

VIDEO DIVULGATIVI

ST-EMotionaLearning	443
<i>Francesco Mario Pio Damiani, Daniela Troia</i>	
Il linguaggio segreto dei computer: 0 e 1!	445
<i>Maria Grazia Giannoccaro</i>	

Prefazione

Il convegno italiano sulla Didattica dell'Informatica (ITADINFO) si propone come un punto d'incontro tra il mondo della scuola e quello della ricerca per favorire lo scambio di esperienze, pratiche e visioni sull'insegnamento dell'informatica come disciplina scientifica. ITADINFO, giunto nel 2025 alla sua **terza edizione**, si è svolto a **Salerno**, consolidando il suo ruolo di spazio aperto, partecipativo e inclusivo, dedicato a chi promuove la cultura informatica in ogni ordine e grado scolastico.

L'edizione 2024, tenutasi a Genova, ha dimostrato l'efficacia del formato basato su laboratori interattivi, relazioni scientifiche e racconti dal campo. L'interesse e la partecipazione crescenti hanno confermato il bisogno diffuso di strumenti, formazione e confronto per rendere l'informatica accessibile, significativa e didatticamente sostenibile. ITADINFO 2025 ha raccolto questa eredità offrendo un programma ancora più ricco e variegato, capace di rappresentare le diverse anime della comunità educante.

Il programma di quest'anno si articola in **quattro sezioni principali**:

- **Relazioni scientifiche e racconti di esperienze**, che restituiscono pratiche didattiche sperimentate, attività interdisciplinari, riflessioni teoriche e percorsi strutturati per l'insegnamento dell'informatica;
- **Poster**, che arricchiscono lo spazio comune del convegno con una galleria di idee, prototipi, attività in corso e iniziative per la formazione e l'orientamento;
- **Laboratori formativi**, dedicati alla sperimentazione diretta di percorsi e strumenti, per supportare i docenti nella progettazione autonoma di attività in classe.
- Brevi **video divulgativi** per informare su temi dell'informatica, con esempi pratici, illustrazioni e testimonianze, così da rendere accessibili argomenti complessi a un pubblico ampio.

L'intero programma è attraversato da temi trasversali che testimoniano l'evoluzione delle esigenze educative e l'urgenza di una didattica dell'informatica aggiornata, critica e consapevole. L'intelligenza artificiale, la modellazione della realtà attraverso strumenti concettuali e computazionali, e l'integrazione di approcci educativi che combinano strumenti analogici e digitali rappresentano alcune delle direttrici che guidano le attività proposte. In questo contesto, l'informatica si conferma non solo come disciplina autonoma, ma anche come chiave di lettura per comprendere e agire nel mondo, promuovendo lo sviluppo del pensiero critico, creativo e computazionale. I contributi selezionati offrono un prezioso repertorio di attività, strumenti e riflessioni, rappresentando un punto di riferimento sia per chi si avvicina ora all'informatica come disciplina scolastica, sia per chi la insegna da anni e cerca nuove strade per farla vivere in classe. ITADINFO 2025 rinnova l'impegno a essere una comunità aperta, che valorizza il sapere dei docenti, la ricerca educativa e le sinergie tra scuola, università e territorio. A tutti i partecipanti, relatori e relatrici, un sentito ringraziamento per aver reso questo incontro una nuova tappa verso una didattica dell'informatica sempre più consapevole, condivisa e inclusiva.

*Veronica Rossano
Maria Angela Pellegrino*

Atti del convegno

ITADINFO 2025

Terzo Convegno Italiano sulla Didattica dell'Informatica

03-05 ottobre 2025

Laboratorio CINI Informatica e Scuola,
Università degli Studi di Salerno e APS Programma il Futuro

Comitato scientifico del convegno

Veronica Rossano, *Università degli Studi di Bari "Aldo Moro" (Presidente)*
Guglielmo Abbruzzese, *Università di Roma "Tor Vergata"*
Agnese Addone, *Università degli Studi di Salerno*
Giuliana Barberis, *Liceo Scientifico "Maria Curie" Pinerolo*
Claudia Canali, *Università di Modena e Reggio Emilia*
Sara Capecchi, *Università di Torino*
Antonella Carbonaro, *Università di Bologna*
Giorgio Delzanno, *Università di Genova*
Agnese Del Zozzo, *Università di Trento*
Valentina Di Noi, *Università di Torino*
Filomena Ferrucci, *Università degli Studi di Salerno*
Luca Forlizzi, *Università dell'Aquila*
Ilenia Fronza, *Università di Bolzano*
Maurizia Gai, *Istituto Comprensivo Vivaldi-Murialdo di Torino*
Enrica Gentile, *Università di Bari*
Giovanna Guerrini, *Università di Genova*
Gennaro Iaccarino, *I.I.S.S. "G. Galilei" di Bolzano, docente comandato presso la Dir. Istr. e Formazione italiana di Bolzano*
Luca Lamanna, *Università di Milano*
Michael Lodi, *Università di Bologna*
Violetta Lonati, *Università di Milano*
Sabrina Mantaci, *Università di Palermo*
Simone Martini, *Università di Bologna*
Giovanna Melideo, *Università dell'Aquila*
Claudio Mirolo, *Università di Udine*
Mattia Monga, *Università di Milano*
Alberto Montresor, *Università di Trento*
Anna Morpurgo, *Università di Milano*
Enrico Nardelli, *Università di Roma "Tor Vergata"*
Martina Palazzolo, *Istituto Comprensivo Ilaria Alpi di Milano*
Fabio Palomba, *Università degli Studi di Salerno*
Maria Angela Pellegrino, *Università degli Studi di Salerno*
Marcello Sarini, *Università di Milano – Bicocca*
Emanuele Scapin, *ITT Giacomo Chilesotti di Thiene*
Vittorio Scarano, *Università degli Studi di Salerno*
Ugo Solitro, *Università di Verona*
Tullio Vardanega, *Università di Padova*

Comitato organizzatore del convegno

Presidenza e relazioni istituzionali

Enrico Nardelli, *Università di Roma "Tor Vergata" e direttore del Laboratorio CINI*

Filomena Ferrucci, *Università degli Studi di Salerno*

Vittorio Scarano, *Università degli Studi di Salerno*

Programma scientifico e Atti del convegno

Maria Angela Pellegrino, *Università degli Studi di Salerno*

Veronica Rossano, *Università di Bari*

Comunicazione

Michael Lodi, *Università di Bologna*

Sponsorizzazioni

Fabio Palomba, *Università degli Studi di Salerno*

Tullio Vardanega, *Università di Padova*

Ufficio stampa e comunicazione

Reputation Agency

Francesco Lacchia, *APS Programma il Futuro*

Filomena Ferrucci, *Università degli Studi di Salerno*

Fabio Palomba, *Università degli Studi di Salerno*

Maria Angela Pellegrino, *Università degli Studi di Salerno*

Vittorio Scarano, *Università degli Studi di Salerno*

RELAZIONI INVITATE

Educare al pensiero algoritmico: riflessioni e proposte per il primo ciclo

Marta Crippa¹, Violetta Lonati¹, Luca Lamanna¹, Laura Branchetti¹

¹ Università degli Studi di Milano, Milano, Italia

marta.crippa2@studenti.unimi.it, lonati@di.unimi.it,

luca.lamanna@unimi.it, laura.branchetti@unimi.it

Abstract

Le nuove “Indicazioni Nazionali per il curriculum – Scuola dell’infanzia e Scuole del primo ciclo di istruzione” introducono nuovi obiettivi specifici e competenze attese nell’ambito delle discipline STEM, e in particolare della matematica, che fanno riferimento agli algoritmi. Questo contributo intende aprire una riflessione sul significato e sulle implicazioni didattiche del concetto di algoritmo.

1. Introduzione

Con la parola “algoritmo” in informatica si intende una descrizione precisa e non ambigua (in riferimento a uno specifico esecutore) di una procedura che si presta ad essere automatizzata [6]. Una volta espresso come programma (usando un linguaggio di programmazione) un algoritmo può essere eseguito da un agente “meccanico”, come un computer, o un dispositivo digitale in genere; gli algoritmi consentono dunque di automatizzare lo svolgimento di compiti o la risoluzione di problemi. Nella società contemporanea, gli algoritmi svolgono un ruolo sempre più pervasivo e strutturante, influenzando dinamiche economiche, culturali e sociali. Essi sono alla base dei sistemi informativi che regolano numerosi aspetti della vita quotidiana. In ambito formativo ciò implica la necessità di sviluppare – fin dal primo ciclo scolastico – competenze che vadano oltre l’uso strumentale delle tecnologie, a partire proprio dalla comprensione della natura e delle logiche degli algoritmi e dallo sviluppo del cosiddetto *pensiero algoritmico*, definito in [5] come l’insieme di abilità necessarie a progettare e analizzare algoritmi.

2. Il pensiero algoritmico nella scuola del primo ciclo

Nel contesto scolastico italiano, il pensiero algoritmico è strettamente connesso alla pratica del *problem solving*, intesa in senso ampio come attività cognitiva complessa che richiede l’elaborazione di strategie per affrontare situazioni non risolubili tramite applicazione meccanica di procedure note [4]. Lockwood e colleghi [7], da una prospettiva principalmente matematica, descrivono il pensiero algoritmico come una forma di pensiero logico e organizzato che consente, anche grazie a una visione d’insieme delle risorse disponibili, di scomporre obiettivi complessi

in sequenze ordinate di passi, identificando quindi il pensiero algoritmico come una forma di pensiero comune sia alla matematica che all'informatica. Tuttavia, nella pratica scolastica, il concetto di algoritmo è tradizionalmente accostato sin dalla scuola primaria (cf. [9, pag 50]) alla sua accezione più originaria e operativa, ovvero come esecuzione meccanica di procedimenti appresi a memoria. Questo riflette un'idea procedurale della conoscenza [2] che non favorisce una reale attivazione del pensiero algoritmico e può invece portare studenti e studentesse a svolgere attività algoritmiche senza comprendere i significati sottostanti né attivare forme di pensiero critico.

Un primo passo verso il superamento di questa concezione si ritrova nelle nuove *Indicazioni Nazionali per il curricolo – Scuola dell'infanzia e Scuole del primo ciclo di istruzione* [8]. In tale documento viene esplicitamente riconosciuto il ruolo centrale degli algoritmi – e, più in generale, dell'informatica – nella società contemporanea. In particolare, gli algoritmi sono più volte citati, direttamente o in maniera implicita, tra le competenze attese e gli obiettivi specifici di apprendimento nell'ambito dell'*Educazione integrata STEM*, e più specificatamente della matematica. Coerentemente con la natura prescrittiva ma non operativa del documento, le Indicazioni si limitano tuttavia a delineare finalità generali, senza entrare nel merito delle scelte didattiche e metodologiche necessarie per tradurre tali obiettivi in pratiche educative concrete.

Il presente contributo si propone di avviare una riflessione critica su come il tema degli algoritmi e dello sviluppo del pensiero algoritmico possano essere affrontati nella pratica didattica, proponendo al contempo esempi concreti di attività didattiche e situazioni-problema (sia in contesto matematico che in contesto informatico) finalizzate allo sviluppo del pensiero algoritmico. L'obiettivo è offrire agli insegnanti strumenti concettuali e operativi per integrare in modo consapevole e significativo il lavoro sugli algoritmi all'interno del curricolo, in una prospettiva che valorizzi la connessione tra matematica, informatica e competenze trasversali di *problem solving*.

3. Eseguire algoritmi appresi o progettare e riflettere su algoritmi?

L'ideazione, la progettazione e l'analisi di algoritmi richiede capacità analitiche, flessibilità cognitiva, riflessione sui vincoli e sulle strategie, oltre che consapevolezza degli obiettivi e dei possibili errori. Si tratta quindi di attività creative, riflessive e complesse che non corrispondono all'immagine procedurale che, ad esempio, viene restituita dalla trattazione in molti libri di testo - come si può osservare nell'estratto proposto in Figura 1, tratto da un libro di matematica per la scuola secondaria di primo grado [1] e riguardante l'algoritmo di Euclide. Qui l'algoritmo è presentato come metodo da applicare per calcolare il massimo comune divisore tra due interi; se ne propone un'applicazione di esempio sui numeri 72 e 42; lo si formula infine in maniera schematica per prepararne l'implementazione in un linguaggio di programmazione visuale. Il testo propone l'algoritmo come già progettato (anzi, "scoperto") da altri, e viene fornito pronto da "imparare". Si trascurava invece ogni considerazione sul perché l'algoritmo consenta di ottenere il risultato voluto (*correttezza* dell'algoritmo), sulla sua generalità, o su quali sue proprietà lo rendano interessante rispetto ad altri metodi per calcolare il MCD.

58 L'algoritmo di Euclide per il MCD

Il signor Euclide ha scoperto un semplice algoritmo per trovare il massimo comun divisore tra due numeri:

- diminuisci il più grande dei due numeri sottraendogli il più piccolo
- procedi così **fino a quando** i due numeri diventano uguali, quello sarà il valore del MCD

■ ESEMPIO

Calcola il MCD di 72 e 42 con l'algoritmo di Euclide.

72 e 42 diventano: 30 e 42,

poi 30 e 12,

poi 18 e 12,

poi 6 e 12

infine 6 e 6. Risposta: il MCD tra 72 e 42 è 6.

Possiamo scrivere l'algoritmo in Scratch. Il seguente blocco calcola il MCD tra due numeri e lo mette nella variabile **MCD**:

- crea variabile **MCD**
- definisci **calcolaMCD(m,n)**:

```

• ripeti fino a quando m=n:
  • se m>n allora porta m a (m-n)
  • altrimenti porta n a (n-m)
• porta MCD a m

```

SOTTOLINEA IL PIÙ
PICCOLO A OGNI
PASSAGGIO!



Figura 1: L'algoritmo di Euclide in un libro di testo per la secondaria di primo grado [1, Pag. 384]

4. Problemi specifici e problemi generali: l'algoritmo come soluzione di un problema generale

Nella pratica scolastica e nei libri di testo, con la parola “problema” spesso ci si riferisce a un quesito del quale si conoscono alcuni elementi (i dati) a partire dai quali si devono calcolare altri elementi¹. La soluzione di questo tipo di problemi è una risposta (di solito numerica), ottenuta mediante calcoli. Cambiando i dati numerici di un problema, se ne ottengono delle varianti che, condividendone la struttura, possono naturalmente essere risolte alla stessa maniera. Possiamo quindi distinguere tra *problemi specifici*, con dati fissati, e *problemi generali*, in cui i dati possono assumere un qualsiasi valore, tra quelli accettabili in relazione al problema. Ad esempio, ‘Calcolare

¹ Questa definizione di “problema” è molto limitata. Come evidenzia D’amore in [3], non si può infatti fornire una definizione univoca del concetto senza considerare opportuni elementi di contorno e contesto. In [9], i problemi sono identificati come “questioni autentiche e significative, legate alla vita quotidiana, e non solo esercizi a carattere ripetitivo o quesiti ai quali si risponde semplicemente ricordando una definizione o una regola”. Una parte significativa nella risoluzione di problemi così intesi riguarda la comprensione della questione in esame, l’identificazione e la caratterizzazione delle entità coinvolte e delle loro relazioni, nonché la loro modellazione e formalizzazione - anche mediante rappresentazioni diverse. Qui non ci occupiamo di questo aspetto fondamentale del *problem-solving*, e ci riferiamo invece a problemi che siano già stati formalizzati.

il MCD tra due numeri' è un problema generale, mentre 'Calcolare il MCD di 72 e 42' è un problema specifico (cf. Figura 1).

In Figura 2 mostriamo un esempio tratto dalle prove INVALSI 2010 di matematica per il grado 5, che richiede di risolvere tre problemi specifici, varianti di uno stesso problema generale, riassumibile come segue: 'Indica, per una tavoletta di costo e peso qualsiasi, se costa meno o più della media'.

D10. In media il costo di una tavoletta di cioccolata da 100 g è 1,50 euro.

Indica, per ciascuna tavoletta descritta nella tabella, se costa meno o più della media.

		Costa meno della media	Costa più della media
a.	Una tavoletta da 200 g che costa 3,40 euro	☐	☐
b.	Una tavoletta da 125 g che costa 3 euro	☐	☐
c.	Una tavoletta da 500 g che costa 3,50 euro	☐	☐

Figura 2: Quesito 10 delle prove INVALSI 2010 per il grado 5

Il compito può essere affrontato in modi diversi. Ai due estremi possiamo identificare questi due approcci:

- risolvere ciascuno dei 3 problemi specifici indipendentemente, senza sfruttare il fatto che hanno una struttura comune.
- affrontare il problema generale comune ai tre problemi specifici, identificando un modo di procedere (generale!) che possa poi essere applicato a ciascuno dei 3 casi (cf. ad esempio Figura 3).

È naturalmente possibile, inoltre, adottare approcci intermedi, per esempio risolvendo dapprima uno dei tre problemi specifici e adattando poi la strategia risolutiva agli altri due casi.

Data una tavoletta di costo e peso qualsiasi,

- 1. calcola quanto costano 100g della tavoletta**
- 2. confronta il valore ottenuto con 1,50€**
 - **se è maggiore allora rispondi "Costa più della media"**
 - **altrimenti rispondi "Costa meno della media"**

Figura 3: Un esempio di algoritmo (procedura generale) che risolve il problema (generale) di Figura 2

L'approccio B è tipico della prospettiva informatica, in cui si considerano solitamente problemi generali, che vengono chiamati *problemi computazionali* [10]. A differenza che per un problema specifico (la cui soluzione è tipicamente un numero), per un informatico risolvere un problema computazionale significa ideare un algoritmo, ovvero una procedura che fornisca

automaticamente la risposta (*output*) ad ogni problema specifico che si ottiene fissando il valore dei dati (*input*). In questo senso l'algoritmo non risulta concepito come *strumento* per risolvere problemi specifici (come tipico nella visione esclusivamente procedurale: ad esempio, applico acriticamente l'algoritmo di Euclide per trovare il MCD di 72 e 42) ma costituisce in sé una *soluzione al problema generale*.

Esercizi come quelli di Figura 2 consentono di riflettere sui diversi approcci e sui relativi vantaggi e svantaggi: ad esempio, con l'approccio A si possono sfruttare, caso per caso, eventuali proprietà specifiche dei dati per facilitare i conti aritmetici, mentre l'approccio B fornisce un metodo che si presta ad essere eventualmente riutilizzato, senza modifiche, per altri casi specifici. È proprio questa caratteristica di generalità a rendere particolarmente significativa la possibilità di automatizzare la procedura, mediante un *programma*.

5. Come si esprime un algoritmo?

La differenza tra algoritmi e programmi è che questi ultimi sono formulati in un *linguaggio di programmazione*, che dunque prevede una precisa sintassi per descrivere i passi da compiere. Gli algoritmi, invece, sono solitamente formulati in linguaggio naturale (come nell'esempio di Figura 3) o in *pseudo-codice* (come in Figura 1). In ogni caso, uno degli aspetti chiave nella concezione informatica degli algoritmi è la necessità di esplicitare un *interprete/esecutore*, ovvero un agente computazionale capace di compiere determinate operazioni elementari (e non altre): è proprio il riferimento all'interprete e al suo *insieme di primitive* che consente di poter formulare un algoritmo senza ambiguità (da questo punto di vista, risulta infatti assai problematico l'utilizzo della "ricetta di cucina" come esempio paradigmatico di algoritmo). Ogni programma, d'altra parte, incarna (rendendolo eseguibile automaticamente) un algoritmo; in questo senso è facile immaginare come le attività di programmazione (spesso identificate con l'etichetta *coding*) possano contribuire allo sviluppo del pensiero algoritmico. Tuttavia, nella programmazione intervengono anche gli aspetti legati proprio all'esistenza di regole linguistiche/formali a cui adeguarsi, e quindi è facile che l'attenzione si sposti principalmente su questi aspetti.

In Figura 4 mostriamo un quesito a cavallo tra il pensiero algoritmico e la programmazione, proposto dal Bebras² dell'Informatica nel 2018. Senza usare terminologia specifica, il quesito di fatto presenta un semplice interprete (il robot) capace di eseguire 3 tipi di istruzioni, ciascuna identificata da una cifra o da una lettera dell'alfabeto. Questo insieme di istruzioni (con la loro semantica) costituisce il linguaggio di programmazione del robot e ciascuna delle opzioni di risposta è un programma scritto in questo linguaggio: per comprendere il quesito è necessario dunque innanzitutto comprendere questo linguaggio di programmazione, che tuttavia è sufficientemente semplice da consentire agli alunni di rimanere concentrati sugli aspetti algoritmici. Uno dei modi per rispondere al quesito è quindi impersonare il robot e simulare l'esecuzione di ciascuno dei quattro programmi, per verificare quale realizza il compito assegnato.

Il quesito stimola necessariamente attività di pensiero algoritmico: non si può prescindere dal ragionare su sequenze di istruzioni e passi elementari. Non ci sono invece algoritmi da apprendere, né algoritmi già appresi da applicare acriticamente (cf. Sezione 3); al contrario, il quesito sollecita il pensiero strategico e una riflessione linguistica e comunicativa sul significato di "dare istruzioni".

2 <http://bebras.it>

Mattoncini colorati



Un robot ha il compito di spostare e impilare dei mattoni colorati.
La posizione del mattone è definita mediante una lettera che indica la posizione del mattone (o, più precisamente, del suo lato sinistro).
Il robot è capace di spostare solo un mattone alla volta.

La tabella sotto mostra tutte le istruzioni che un robot è capace di eseguire:

Istruzione	Azione
1	prendi il mattone
0	deposita il mattone
una lettera da A a H	sposta il braccio nella posizione indicata dalla lettera

Il robot nella figura deve spostare i mattoni impilati a sinistra in modo che si trovino alla fine nella posizione indicata sulla destra.

Quale delle seguenti sequenze di istruzioni permette di eseguire il compito assegnato?

☐ **A1GOA1H1A1FO**

☐ **A1FOA1GOA1EO**

☐ **A1FOA1GOA1HO**

☐ **A10FA10GA10E**

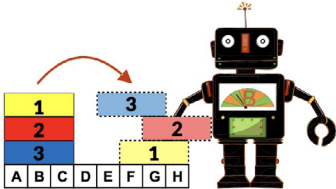


Figura 4: Il problema dei mattoncini colorati - Bebras dell'Informatica (id: 2018-MY-05)

6. Algoritmi per descrivere processi, fenomeni, sistemi

Fissata una precisa situazione di partenza o uno specifico insieme di dati iniziali (il cosiddetto *input*), un algoritmo determina una precisa sequenza di azioni/passi/eventi (tecnicamente, in informatica si usa il termine *processo*), che sarà uguale ad ogni esecuzione, a parità di input (gli algoritmi sono, cioè, *deterministici*). Dunque, gli algoritmi possono essere visti anche come uno strumento per descrivere processi, o più in generale fenomeni o sistemi che evolvono nel tempo per passi discreti; in questo senso il pensiero algoritmico si rivela essere uno strumento di pensiero utile nell'analisi e nella modellazione di processi, fenomeni e sistemi.

Il problema della lumaca proposto in Figura 5 si presta a essere affrontato secondo modalità risolutive profondamente diverse. Ad un estremo troviamo un approccio “meccanico”, che potrebbe portare a una semplificazione “aritmetica” del problema, ad esempio applicando acriticamente una formula del tipo ‘metri saliti meno metri scesi’ per ottenere l’avanzamento giornaliero, da moltiplicare quindi fino a raggiungere o superare l’altezza del muro, senza interrogarsi sulla coerenza del calcolo rispetto alla situazione descritta (es. cosa accade al sedicesimo giorno, quando la lumaca raggiunge per la prima volta la cima del muro) e producendo un risultato numericamente errato.

Una lumaca si trova alla base di un muro alto 20 metri. Ogni giorno sale di 5 metri ma, durante la notte, scivola in giù di 4.

In quanti giorni la lumaca raggiungerà la cima del muro?

Figura 5: Il problema della lumaca

In un approccio orientato alla modellizzazione, invece, il percorso di risoluzione inizia con un'attenta analisi del contesto: l'alunno rappresenta la situazione, identifica gli elementi rilevanti, formula ipotesi sull'evoluzione del fenomeno. Tramite la sua simulazione, almeno parziale, il processo può essere visualizzato e compreso. Il pensiero algoritmico diventa quindi strumento utile alla comprensione, alla modellazione, all'astrazione.

7. Sequenza ordinata di passi elementari: una definizione riduttiva

La definizione di algoritmo con cui abbiamo iniziato questo articolo – *una descrizione precisa e non ambigua (in riferimento a uno specifico esecutore) di una procedura che si presta ad essere automatizzata* – è più complessa di quella che si utilizza nel senso comune, in cui l'algoritmo è definito più semplicemente come una *sequenza di istruzioni da eseguire passo-passo*. Il compito presentato in Figura 6 può costituire l'occasione per mettere in discussione questa concezione riduttiva di algoritmo.

Il quesito descrive il sistema con cui vengono spostati dei sacchi in un magazzino, formulando di fatto un algoritmo (anche qui: non da imparare per poi applicare, cf. Sezione 3), e richiede di riflettere sull'algoritmo stesso e sui suoi effetti. L'algoritmo, tuttavia, qui è formulato tramite *regole*, piuttosto che tramite *istruzioni*; queste regole determinano comunque in modo univoco (per ciascuna delle situazioni specifiche che si possono verificare – ovvero il numero, l'ordine e il peso dei sacchi da spostare) il modo con cui i sacchi verranno spostati, e cioè la *sequenza* di azioni che sono prescritte e che andranno compiute se si segue l'algoritmo. Per rispondere al quesito è necessario identificare questa sequenza di azioni e simularne l'esecuzione, tenendo traccia di ciò che succede.

È interessante anche notare che, in questo quesito, non si pone l'attenzione tanto sul compito che viene realizzato o sul risultato finale che si ottiene (il cosiddetto *output*), ma sul processo e in particolare su ciò che avviene durante questo processo, cf. Sezione 6.

8. Conclusioni

L'analisi proposta ha messo in luce come il pensiero algoritmico rappresenti una risorsa cognitiva significativa per affrontare problemi in modo strategico, riflessivo e creativo. Un approccio centrato sulla progettazione e l'analisi di algoritmi (come può essere quello tipico dell'informatica), piuttosto che sulla loro sola applicazione, consente di valorizzare le potenzialità trasversali del pensiero algoritmico, in connessione con le competenze chiave del *problem solving*.

Perché questa visione possa concretizzarsi nella pratica scolastica, è necessario superare l'impostazione procedurale ancora dominante nei libri di testo e nelle consuetudini didattiche. Le attività presentate in questo contributo mostrano come, anche in contesti scolastici del primo ciclo, sia possibile proporre esperienze significative che mirano a sviluppare la capacità di analizzare situazioni, formalizzare strategie risolutive, riflettere sui processi e non solo sui risultati. In questo senso, educare al pensiero algoritmico contribuisce a promuovere una forma di alfabetizzazione critica e consapevole rispetto alla tecnologia e ai suoi effetti nella società.

Sacchi nell'ascensore



Nel corridoio accanto all'ascensore ci sono alcuni sacchi allineati alla parete. Su ogni sacco è scritto il peso in kg.



L'ascensore serve per trasportare i sacchi in un magazzino. L'ascensore porta al massimo 100 kg e viene fatto partire non appena il suo carico supera gli 80 kg.

Per caricarlo, si prende il primo sacco della fila (quello più vicino all'ascensore) e lo si mette nell'ascensore, tranne quando il sacco fa superare il limite di 100 kg; in questo caso il sacco viene messo oltre l'ascensore a formare via via una nuova fila che parte dal fondo del lato opposto del corridoio.

Quando tutti i sacchi della fila sono stati spostati, si continua nello stesso modo considerando la fila formata dall'altro lato del corridoio.

Dopo che tutti i sacchi sono stati trasportati in magazzino, quale delle seguenti affermazioni è vera?

V	F	Una volta l'ascensore ha trasportato esattamente 100 kg.
V	F	Non si è mai formata una fila a destra dell'ascensore.
V	F	La prima volta l'ascensore ha trasportato 94 kg.
V	F	Ci sono voluti cinque viaggi per trasportare tutti i sacchi.

Figura 6: Sacchi nell'ascensore - Bebras dell'Informatica (id: 2016-CZ-02a)

Ringraziamenti

Lavoro supportato dal Piano Nazionale di Ripresa e Resilienza (PNRR), Missione 4, Componente 2, Investimento 1.1, Bando n. 104 pubblicato il 2/2/2022 dal Ministero dell'Università e della Ricerca (MUR), finanziato dall'Unione Europea – NextGenerationEU – Titolo del progetto “Learning Informatics” – CUP E53D23007720006 – Decreto di assegnazione del finanziamento n. 959 adottato il 22/04/2022 dal Ministero dell'Università e della Ricerca (MUR).

Bibliografia

- [1] Clara Bertinetto, Arja Metiäinen, Johannes Paasonen, and Eija Voutilainen. Contaci! - Vol. 3. Zanichelli, 2019.
- [2] David W. Braithwaite and Lauren Sprague. Conceptual knowledge, procedural knowledge, and metacognition in routine and nonroutine problem solving. *Cognitive Science*, 45, 2021.
- [3] Bruno D’Amore. Il problema di matematica nella pratica didattica. Digital Docet, 2014.
- [4] Karl Duncker and L. Less. On problem solving. *Psychological Monographs*, 58(5):1–113, 1945.
- [5] Gerald Futschek. Algorithmic thinking: the key for understanding computer science. In Roland T. Mittermeir, editor, *Informatics education – The bridge between using and understanding computers*, pages 159–168, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.
- [6] R. R. Korfhage. Algorithm. In Anthony Ralston, Edwin D. Reilly, and David Hemmendinger, editors, *Encyclopedia of Computer Science*, pages 36–38. John Wiley and Sons, Inc, 2003.
- [7] Elise Lockwood, Autumn Asay, Anna DeJarnette, and Matt Thomas. Algorithmic thinking: An initial characterization of computational thinking in mathematics. In M. B. Wood, E. E. Turner, M. Civil, and J. A. Eli, editors, *Proceedings of the 38th Annual Meeting of the North American Chapter of the International Group for the Psychology of Mathematics Education*, pages 1588–1595. The University of Arizona, Tucson, 2016.
- [8] MIM. Indicazioni nazionali per il curricolo – scuola dell’infanzia e scuole del primo ciclo di istruzione, 7 luglio 2025. Disponibile all’indirizzo https://www.mim.gov.it/documents/20182/0/INDICAZIONI_NAZIONALI_7_7_2025.pdf/63802aed-f9f4-dd6e-f427-c45aa9222f31?version=1.0&et=1751893295452 (ultima consultazione: 28 luglio 2025).
- [9] MIUR. Indicazioni nazionali per il curricolo della scuola dell’infanzia e del primo ciclo d’istruzione, 2012. Disponibile all’indirizzo https://www.miur.gov.it/documents/20182/51310/DM+254_2012.pdf (ultima consultazione: 28 luglio 2025).
- [10] C. Papadimitriou. *Computational complexity*. Pearson, 1993.

Le *Notional machine* per l'apprendimento della programmazione, anche nell'era dell'AI generativa

Michael Lodi¹, Simone Martini¹

¹ Dipartimento di Informatica - Scienza e Ingegneria Alma Mater Studiorum - Università di Bologna
Bologna, Italia. Laboratorio CINI Informatica e Scuola

{michael.lodi, simone.martini}@unibo.it

Abstract

Imparare a programmare richiede di acquisire conoscenze di tipo sintattico, concettuale e strategico. La comprensione concettuale del comportamento dei programmi durante l'esecuzione è fondamentale per un apprendimento efficace. Questo tipo di conoscenza si basa sulla costruzione di modelli mentali adeguati. Gli insegnanti possono utilizzare una (o più) “notional machine”, strumenti pedagogici che in modo semplificato ma preciso spiegano il funzionamento della macchina che esegue un programma scritto in un certo linguaggio. In assenza di una formazione esplicita, gli studenti sviluppano spontaneamente tali modelli, rischiando però misconcezioni. Per favorire lo sviluppo di modelli mentali accurati, sono cruciali attività di visualizzazione dell'esecuzione del codice, che includono tecniche di tracciamento manuale e strumenti automatici, e attività di lettura e comprensione del codice. L'avvento di strumenti di intelligenza artificiale generativa capaci di scrivere codice rende ancor più urgente che gli studenti padroneggino una robusta conoscenza concettuale: solo così potranno valutare criticamente e comprendere i programmi generati automaticamente, individuandone limiti e potenzialità.

1. Introduzione

Imparare a programmare richiede diversi tipi di conoscenza [1, 12]. In particolare, possiamo individuare tre categorie principali.

La prima è la conoscenza **sintattica**, cioè la capacità di utilizzare correttamente la sintassi di un linguaggio di programmazione. Questa conoscenza riguarda, ad esempio, come si scrive un determinato costrutto nel linguaggio scelto.

La seconda categoria è la conoscenza **concettuale**, che consiste nel comprendere la dinamica di un programma in esecuzione. In altre parole, significa sapere come funzionano realmente i costrutti e in che modo essi determinano l'esecuzione del programma (es. quante volte un ciclo verrà eseguito, quando termina, che effetto ha la modifica di un elemento di una lista, e così via...).

La terza categoria è la conoscenza **strategica**, ovvero la capacità di applicare sia la conoscenza sintattica che quella concettuale per risolvere problemi nuovi o raggiungere obiettivi

specifici. Si tratta, per esempio, di sapere quando utilizzare un particolare costrutto, perché sceglierlo rispetto ad altri e, più in generale, come affrontare e risolvere un problema utilizzando la programmazione.

Sebbene la sintassi costituisca una difficoltà tipicamente solo nella fase iniziale dell'apprendimento della programmazione (abbastanza presto chi è alle prime armi riesce a scrivere programmi sintatticamente corretti – ma spesso semanticamente insensati), la maggior parte dei libri di testo si concentra sulla conoscenza sintattica (es. il suo indice è strutturato sulla base dei costrutti che vengono insegnati in ciascun capitolo). Meno attenzione è posta alla conoscenza concettuale (tema di cui tratteremo in questo articolo) e ancor meno alla conoscenza strategica.

2. Le *Notional machine*

Tutti noi creiamo modelli mentali (semplificati) di ciò che ci circonda, che servono per spiegarci come funziona qualcosa in situazioni diverse e fare previsioni. Si possono avere modelli mentali di noi stessi, di un'altra persona, di oggetti, di sistemi. Grazie a questi modelli, possiamo eseguire nella nostra mente delle simulazioni. Queste simulazioni possono porre l'oggetto del modello mentale in contesti diversi, o, per usare una metafora informatica, istanziare alcune variabili del modello per simulare mentalmente cosa accadrebbe in una specifica condizione. Queste simulazioni impiegano la limitata memoria di lavoro che l'essere umano ha a disposizione, e dunque non possono includere troppi stati o variabili. Ovviamente, le simulazioni fatte sui nostri modelli mentali sono *fallibili*, nel senso che non sempre predicono correttamente la realtà [15].

Sappiamo che un programma scritto in un linguaggio di programmazione è eseguito da una certa macchina: può essere una macchina fisica, ma, più probabilmente, una macchina astratta, cioè un modello concettuale che descrive precisamente “un qualsiasi insieme di strutture dati e di algoritmi che permettano di memorizzare ed eseguire programmi” [6]. Come spesso accade in Informatica, per i linguaggi ad alto livello abbiamo solitamente a che fare con una gerarchia di macchine astratte, che “scendono” fino ad arrivare alla macchina fisica - che davvero “farà i conti” usando elettroni e transistor [10]. Già da tempo la ricerca sull'apprendimento della programmazione sostiene che sia comunque indispensabile comprendere almeno il funzionamento del livello “immediatamente inferiore” [2, 17].

Chi apprende deve quindi formarsi un modello mentale della macchina che sta imparando a controllare, nel suo ruolo di esecutrice di programmi (scritti nel linguaggio studiato). Il modello deve essere sufficientemente astratto, ma preciso, per permettere di spiegarsi correttamente come i programmi sono eseguiti.

Le studentesse e gli studenti si formeranno il modello in ogni caso. Se non viene insegnato loro esplicitamente, se lo costruiranno inferendo esclusivamente da ciò che vedono: la sintassi e gli esempi di programmi che vengono presentati loro. Specialmente nei linguaggi ad alto livello moderni, molto leggibili dagli esseri umani, numerosi dettagli sul funzionamento a tempo di esecuzione sono nascosti. Questa “astrazione” dai livelli inferiori è ovviamente un bene, ma può portare a modelli inaccurati e incompleti - che causano delle incomprensioni, chiamate in letteratura *misconcezioni*. Ad esempio, vedendo solo esempi in cui le variabili hanno sempre nomi di una sola lettera, chi apprende potrà pensare che nomi più lunghi rappresentino parole

1 Esistono progetti didattici che, con un approccio bottom-up, provano a risalire l'intera gerarchia di astrazione: uno tra tutti è il famoso “From NAND to Tetris” (<https://www.nand2tetris.org/>).

riservate (in analogia a quelle che impara, come *for*, *if*, *def*); potrà inoltre pensare che l'esecuzione di un *while* termini nel momento esatto in cui la guardia diventa negativa, e non quando il controllo “ripassa” dalla guardia per rivalutarla dopo aver eseguito il corpo del ciclo. La letteratura ha documentato centinaia di queste misconcezioni [3].

Per limitare le misconcezioni, è importante che gli insegnanti facciano esplicito riferimento a una o più *notional machine* (NM)². La letteratura ha proposto diverse definizioni (si vedano ad esempio [16, 7, 5, 13]), ma in generale possiamo dire che si tratta di un dispositivo pedagogico (uno “scaffolding”) che l'insegnante fornisce allo studente affinché egli maturi un proprio modello dell'esecutore [13] adeguato a “ragionare sul comportamento di un programma e scrivere programmi che funzionano” [7].

Possiamo informalmente pensare a una *notional machine* come a un modello semplificato dell'esecutore che l'insegnante fornisce. La *notional machine* non deve essere necessariamente fedele a quel che realmente succede dietro le quinte (nella macchina astratta e/o nella macchina fisica), purché raggiunga i suoi obiettivi (far sì che gli studenti si spieghino correttamente il comportamento osservabile di un programma e riescano a scriverne uno corretto). Esistono *notional machine* diverse per scopi didattici e per livelli di astrazione diversi³ (es. che rappresentano la programmazione ad oggetti come uno scambio di messaggi oppure come la manipolazione di riferimenti ad oggetti in memoria, e così via).

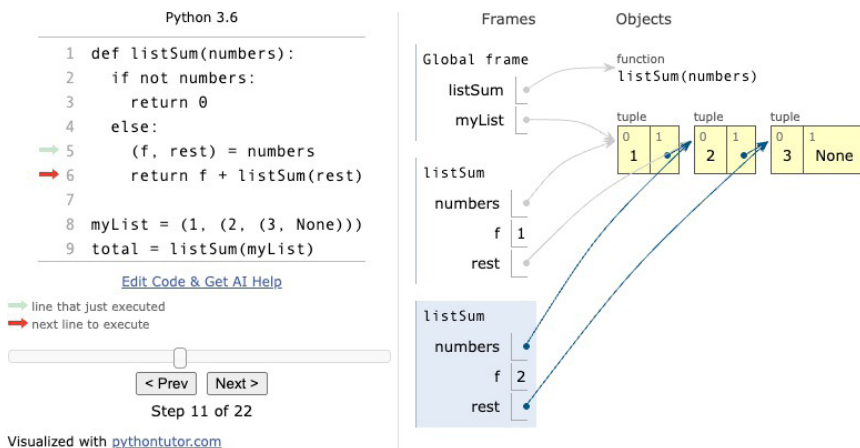


Figura 1: *Pythontutor.com*

2 L'uso di *notional* pare indicare il fatto che sia “semplificata” e “solo parzialmente vera” [5]; in questo senso, e per mantenere esplicito il suo intento pedagogico, si potrebbe tradurre come “macchina ideale”, “macchina pedagogica” o anche “macchina astratta didattica”. Un'alternativa è “macchina concettuale” per legarsi alla conoscenza concettuale di cui abbiamo parlato nell'introduzione (a quanto pare di recente anche in inglese qualcuno utilizza “conceptual machine” [5]). Un problema di questa traduzione è, però, che si può confondere la *notional machine* con il “modello concettuale”, che solitamente indica invece un modello estremamente preciso e fedele alla realtà [5].

3 Una collezione di *notional machine*, distinte in “rappresentazioni della macchina” e “analogie” è disponibile a <https://notionalmachines.github.io/>

3. “Rappresentare” la *notional machine*

Una delle tecniche più utilizzate per favorire la creazione di un modello mentale *adeguato* a comprendere e scrivere programmi è quella di rendere *visibile* ciò che visibile non è: ciò che accade a *runtime* quando un programma viene eseguito. La tecnica più ovvia è quella di disegnare / annotare alla lavagna lo stato interno quando si traccia l'esecuzione di un programma (utilizzando ad esempio rettangoli etichettati per tenere traccia dei valori legati ai nomi, o usando frecce per riferimenti, o una “pila” per rappresentare i frame). È importante chiedere a chi apprende di svolgere anche attivamente questo tipo di tracciatura su un foglio di carta: se ciò non avviene, potrebbero interpretare i disegni solo come uno strumento di spiegazione, e non un’abitudine che devono costruirsi loro stessi [4]. Esistono poi strumenti automatici, che permettono di visualizzare lo stato interno eseguendo passo passo un programma. Uno dei più famosi è Python Tutor⁴ – che, a dispetto del nome, può visualizzare l'esecuzione in moltissimi linguaggi moderni (Figura 1). Ad una granularità più fine (e quindi più adatto per programmi brevi, iniziali) è il debugger di Thonny⁵, un IDE didattico per Python. Il debugger (Fig. 2) permette di visualizzare il risultato della valutazione delle singole espressioni in Python, il passaggio dei parametri (con il legame tra formali e attuali) in caso di chiamata di una funzione, e così via.

Funzioni analoghe sono presenti negli IDE per Java BlueJ⁶ e Greenfoot⁷ (anche per il linguaggio “a frame” Stride⁸).

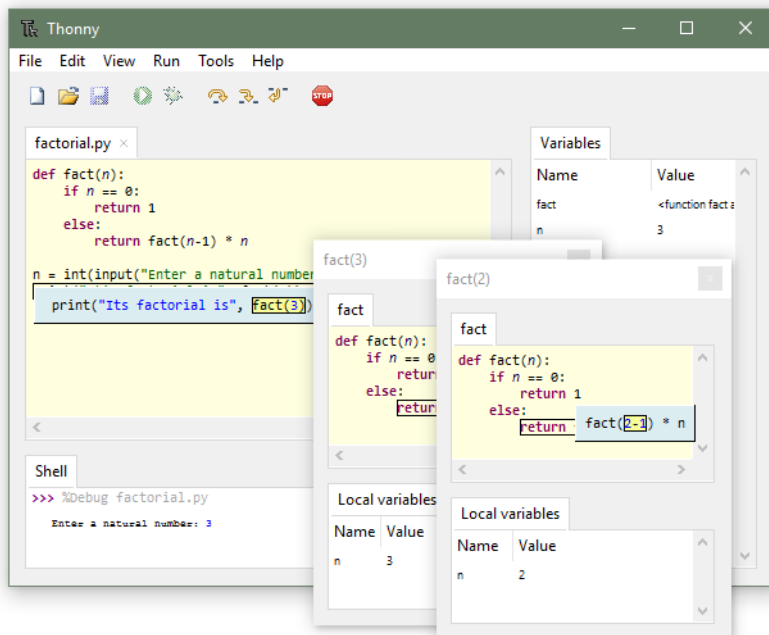


Figura 2: Thonny.org

4 <https://pythontutor.com/>

5 <https://thonny.org/>

6 <https://bluej.org/>

7 <https://www.greenfoot.org/home>

8 <https://stride-lang.net/>

Anche nei linguaggi a blocchi sono presenti semplici strumenti: quando si crea una variabile in Scratch⁹ e nei suoi derivati, automaticamente il suo nome e valore corrente vengono visualizzati nello Stage (Fig. 3). In Snap!¹⁰ è possibile eseguire “passo passo” uno script facendo sì che il blocco eseguito in quel momento “si illumini”, di modo da collegare il blocco all’effetto grafico che produce (Fig. 3).

Un aspetto cruciale per la creazione di modelli mentali adeguati è l'utilizzo di attività di *program comprehension* [8], in cui non si chiede di scrivere codice, ma di leggerlo, analizzarlo, comprenderlo (ma anche completarlo, riordinarlo, ecc.): un esplicito allenamento per la conoscenza concettuale. Metodologie di scaffolding per l'introduzione alla programmazione quali UMC [9] e PRIMM [14] si basano tra gli altri proprio su questo tipo di attività.

4. *Notional machine* nell'era dell'AI generativa

Gli strumenti di AI generativa quali gli LLM esibiscono ogni giorno di più capacità di scrivere programmi corretti, impostare progetti, trovare e correggere errori sulla base di prompt scritti in linguaggio naturale. La macchina che interpreta tali prompt può essere vista come una “black box” statistica, allenata su una mole di dati a predire “la prossima parola”, spesso con elementi di randomicità.

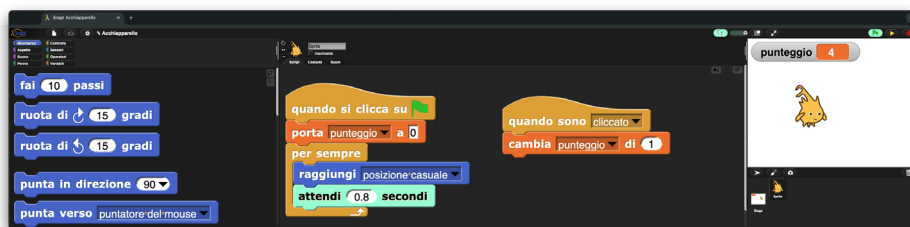


Figura 3: In linguaggi come Scratch e Snap!, il nome con relativo valore delle variabili è solitamente visualizzato sullo stage. In Snap!, è possibile eseguire *step-by-step* i blocchi di istruzioni. L’istruzione correntemente eseguita è illuminata di azzurro (in questo caso: *attendi 0.8 secondi*)

Al giorno d’oggi, l’output è ancora - nella maggior parte dei casi - un programma scritto in un linguaggio di programmazione ad alto livello (es. Python). Riprendendo i tre tipi di conoscenza di cui abbiamo parlato nell’introduzione, questi programmi sono quasi sempre corretti sintatticamente. Invece, anche per la natura stocastica delle reti neurali alla base dei moderni sistemi, talvolta contengono errori importanti a livello concettuale e strategico. Dunque, almeno per il momento, è importante (anzi, *più importante* di prima) che chi apprende acquisisca la conoscenza concettuale, formandosi un modello mentale adeguato a leggere e comprendere un programma scritto da una intelligenza artificiale – per poterne valutare ad esempio la correttezza, l’efficienza, l’opportunità. Se un LLM è in grado di scrivere programmi, occorre che chi li utilizza per questo scopo sappia leggerli e comprenderli, anche per motivi etici, visto che sono i

9 <https://scratch.mit.edu/>

10 <https://snap.berkeley.edu/>

programmatore umani (e non le IA) ad avere la responsabilità sociale e giuridica dei programmi che, con qualsiasi strumento prodotti, rendono disponibili [11].

In futuro però, potrebbe esserci un vero e proprio “step-up” nel livello di astrazione: il prossimo linguaggio di programmazione potrebbe “essere l’inglese”. In tale scenario, il prompt diventa il nuovo programma, e l’output potrebbe essere direttamente un comportamento o un artefatto computazionale. Non è ancora chiaro quale sia il livello “immediatamente inferiore” alla programmazione via prompt: la macchina che media tra questi due livelli non è più una macchina deterministica tradizionale, ma un modello statistico complesso. Sarà quindi necessario chiarire i meccanismi interni rilevanti da esplicitare per favorire la comprensione. Gli insegnanti dovranno avvalersi di nuove, e molto diverse, notional machine: su questo la ricerca è solo agli albori. La sfida per la didattica dell’informatica nei prossimi anni sarà dunque duplice: da un lato, potenziare l’insegnamento della conoscenza concettuale tradizionale, per garantire una comprensione dei programmi; dall’altro, aiutare gli studenti a comprendere e governare sistemi generativi non deterministici.

Ringraziamenti

Ringraziamo Violetta Lonati per i preziosi suggerimenti alla stesura della versione finale.

Fondi

Il lavoro di M. Lodi è supportato dallo Spoke 1 “FutureHPC & BigData” del Centro Nazionale di Ricerca in “High Performance Computing, Big Data and Quantum Computing” (ICSC) finanziato da MUR Missione 4 Componente 2 Investimento 1.4: Potenziamento strutture di ricerca e creazione di campioni nazionali di R&S (M4C2-19) - Next Generation EU (NGEU). Il lavoro di S. Martini è parzialmente finanziato da: (a) SERICS (PE00000014) nell’ambito del Piano Nazionale di Ripresa e Resilienza del MUR, finanziato dal progetto “NextGenerationEU” dell’Unione Europea; (b) INdAM GNSAGA; (c) supporto finanziario ricevuto nell’ambito del Piano Nazionale di Ripresa e Resilienza, Missione 4, Componente 2, Investimento 1.1, Bando No. 104, pubblicato il 2/02/2022 dal Ministero dell’Università e della Ricerca, finanziato dal progetto “NextGenerationEU” dell’Unione Europea, titolo del progetto “Learning Informatics” – CUP E53D23007720006, decreto di assegnazione del finanziamento No. 959 adottato il 22/04/2022 dal Ministero dell’Università e della Ricerca

Bibliografia

- [1] Piraye Bayman and Richard E. Mayer. Using conceptual models to teach BASIC computer programming. *Journal of Educational Psychology*, 80(3):291–298, September 1988.
- [2] Mordechai Ben-Ari. Constructivism in computer science education. *Journal of computers in Mathematics and Science Teaching*, 20(1):45–73, 2001.
- [3] Luca Chiodini, Igor Moreno Santos, Andrea Gallidabino, Anya Taffiovich, André L. Santos, and Matthias Hauswirth. A curated inventory of programming language misconceptions. In *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1, ITiCSE '21*, page 380–386, New York, NY, USA, 2021. Association for Computing Machinery.
- [4] Kathryn Cunningham, Sarah Blanchard, Barbara Ericson, and Mark Guzdial. Using tracing and sketching to solve programming problems: Replicating and extending an analysis of what students draw. In *Proceedings of the 2017 ACM Conference on International Computing Education Research, ICER '17*, page 164–172, New York, NY, USA, 2017. Association for Computing Machinery.
- [5] Sally Fincher, Johan Jeuring, Craig S. Miller, Peter Donaldson, Benedict du Boulay, Matthias Hauswirth, Arto Hellas, Felienne Hermans, Colleen Lewis, Andreas Mühling, Janice L. Pearce, and Andrew Petersen. Notional machines in computing education: The education of attention. In *Proceedings of the Working Group Reports on Innovation and Technology in Computer Science Education, ITiCSE-WGR '20*, page 21–50, New York, NY, USA, 2020. Association for Computing Machinery.
- [6] Maurizio Gabbrielli and Simone Martini. *Linguaggi di programmazione. Principi e paradigmi*. McGraw-Hill, Milano, 2011.
- [7] S. Grover. *Computer Science in K-12: An A-To-Z Handbook on Teaching Programming*. Edfinity, 2020.
- [8] Cruz Izu, Carsten Schulte, Ashish Aggarwal, Quintin Cutts, Rodrigo Duran, Mirela Gutica, Birte Heinemann, Eileen Kraemer, Violetta Lonati, Claudio Mirolo, and Renske Weeda. Fostering program comprehension in novice programmers - learning activities and learning trajectories. In *Proceedings of the Working Group Reports on Innovation and Technology in Computer Science Education, ITiCSE-WGR '19*, page 27–52, New York, NY, USA, 2019. Association for Computing Machinery.
- [9] Irene Lee, Fred Martin, Jill Denner, Bob Coulter, Walter Allan, Jeri Erickson, Joyce Malyn-Smith, and Linda Werner. Computational thinking for youth in practice. *ACM Inroads*, 2(1):32–37, February 2011.
- [10] Michael Lodi and Simone Martini. Algoritmi, programmi e linguaggi. *Nuova Secondaria*, Anno XLIII, 2025/2026. To appear.
- [11] Simone Martini. Teaching programming in the age of generative AI. In *Proceedings of the 2024 Conference on Innovation and Technology in Computer Science Education V. 1, ITiCSE 2024*, pages 1–2, New York, NY, USA, 2024. Association for Computing Machinery.
- [12] Tanya J. McGill and Simone E. Volet. A conceptual framework for analyzing students' knowledge of programming. *Journal of Research on Computing in Education*, 29(3):276–297, 1997.

- [13] Bhagya Munasinghe, Tim Bell, and Anthony Robins. Computational thinking and notional machines: The missing link. *ACM Trans. Comput. Educ.*, 23(4), December 2023.
- [14] Sue Sentance and Jane Waite. PRIMM: Exploring pedagogical approaches for teaching text-based programming in school. In *Proceedings of the 12th Workshop on Primary and Secondary Computing Education, WiPSCE '17*, page 113–114, New York, NY, USA, 2017. Association for Computing Machinery.
- [15] Juha Sorva. Visual program simulation in introductory programming education. PhD thesis, Aalto University, 2012.
- [16] Juha Sorva. Notional machines and introductory programming education. *ACM Trans. Comput. Educ.*, 13(2), July 2013.
- [17] Matti Tedre, Tapani Toivonen, Juho Kahila, Henriikka Vartiainen, Teemu Valtonen, Ilkka Jormanainen, and Arnold Pears. Teaching machine learning in k–12 classroom: Pedagogical and technological trajectories for artificial intelligence education. *IEEE Access*, 9:110558–110572, 2021.

**RACCONTI DI ESPERIENZE SUL CAMPO
E RELAZIONI SCIENTIFICHE**

PER L'INFANZIA E LA PRIMARIA

Coding per tutti i gusti: come creare confusione negli allievi più piccoli

Gaetano Impoco¹

¹ I.T.T. “Emanuele Morselli”, Gela (CL)

gaetano@impoco.it

Abstract

Le attività di *coding* sembrano uguali nella sostanza o comunque così sono percepite dagli insegnanti, in special modo negli ordini scolastici dell’infanzia e della primaria. Al contrario, le proposte differiscono sensibilmente e il fatto che gli insegnanti, e spesso anche chi ne cura la formazione, non ne siano consapevoli causa confusione negli allievi. Questo contributo intende mettere in evidenza queste differenze e, a titolo di esempio, mostra una possibile modifica di una di queste proposte per allinearne la struttura a quella di attività più diffuse. L’obiettivo è suggerire alla comunità scientifica e ai formatori di organizzare le attività e chiarirne i presupposti, in modo tale che gli insegnanti possano scegliere consapevolmente.

1. Introduzione

Nei primi segmenti di scuola, soprattutto nell’infanzia, il *pensiero computazionale* viene veicolato attraverso attività di “coding”, inteso come pianificazione di un percorso fatto di passi atomici da far eseguire ad un esecutore. Con i più piccoli si privilegia l’uso del corpo e, man mano, si introduce l’uso di piattaforme digitali quali quella de *L’ora del codice* [6].

La semplice riflessione presentata in queste pagine nasce da alcune chiacchierate con insegnanti di scuola dell’infanzia. In particolare, durante un corso di formazione ci è stata posta una domanda in merito alla difficoltà di comprensione delle istruzioni di ciò che viene comunemente definito *coding unplugged* da parte di alcuni bambini. Le insegnanti riferivano che alcuni allievi interpretano istruzioni quali “gira a destra” ruotando correttamente sul posto, altri invece si muovono lateralmente ruotando o – a volte – senza ruotare affatto.

Analizzando le attività che le insegnanti in questione svolgono con i propri studenti abbiamo notato come il materiale fornito e le attività stesse proponessero interpretazioni del comando “gira” a volte come rotazione sul posto, altre come movimento laterale. Riteniamo possibile, inoltre, che questa ambiguità (meglio: discordanza) di interpretazione sfugga ai formatori e forse anche a chi progetta e propone queste attività. Per questo ci siamo convinti che sia necessario ordinare il materiale e rendere esplicita l’interpretazione da dare al comando “gira”, così da consentire agli insegnanti di scegliere opportunamente attività che diano messaggi univoci ai propri allievi, evitando ambiguità interpretative.

In questo contributo cerchiamo di chiarire meglio i contorni della questione, facendo riferimento anche alle attività de *L’ora del codice* [6], e mostriamo come siamo intervenuti per “emendare” la nota proposta *CodyFeet* [1] per allinearne il messaggio a quello del *Labirinto classico* [4].

2. Interpretazioni

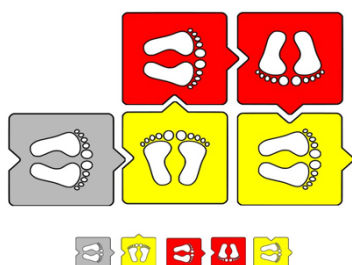
Gli insegnanti dei primi gradi che vogliano avvicinare i propri allievi al “coding” possono scegliere tra diverse proposte e una gran quantità di materiale. Anche tra le fonti più autorevoli, tuttavia, le proposte si differenziano significativamente per l’interpretazione dei comandi di movimento. Vogliamo qui mettere a confronto tre attività rappresentative (Figura 1).



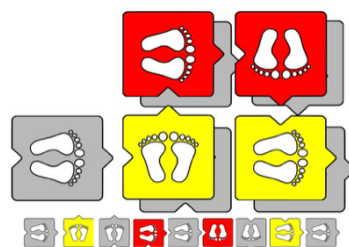
(a) *Labirinto classico*



(b) *Labirinto: Sequenze*



(c) *CodyFeet*



(d) *CodyFeet modificato*

Figura 1: Lo stesso percorso rappresentato con modelli interpretativi diversi

1.1 Rotazione sul posto

All’interno della proposta de *L’ora del codice* [6], il *Labirinto classico* [4] è una delle attività introduttive al coding più efficaci e, probabilmente, tra le più usate in classe. Questo percorso si basa su un’interpretazione dei blocchi “gira a ...” come una rotazione sul posto di 90°. Se pensiamo ad un gioco in cui il bambino replica i movimenti usando il proprio corpo, attività tipica nella scuola dell’infanzia, di fatto stiamo offrendo all’“esecutore” un insieme di tre comandi: **avanti**, **sinistra** e **destra**, questi ultimi interpretati come rotazioni del bambino su sé stesso, rispettivamente sul fianco sinistro o sul destro.

2.2 Spostamento di lato

Diverso è il caso di attività quali Labirinto: Sequenze [5] e Guerre Stellari [3]. Nella prima si usano i concetti di **nord**, **sud**, **est** e **ovest**. Nella seconda, in maniera del tutto equivalente, si usano **sposta in alto**, **sposta in basso**, **sposta a destra** e **sposta a sinistra**. In questi casi, l'interpretazione dei comandi si distacca dal soggetto esecutore e va ancorata al tavolo di gioco. La descrizione stessa dei comandi differisce da quella sopra, apparentemente senza ambiguità. Tuttavia, il bambino che osserva il gioco può identificare nord con **avanti**, **ovest** con **sinistra** e **est** con **destra**. A questo punto si crea un cortocircuito che rischia di generare confusione quando ci si sposta dallo schermo ad una griglia sul pavimento.

2.3 Punti di vista

Perché accade questo? La nostra ipotesi è che il bambino non riesca a distinguere i due punti di osservazione cui i sistemi di regole si riferiscono. Le regole {**avanti**, **sinistra**, **destra**}, infatti, si riferiscono al punto di vista dell'esecutore, mentre l'insieme {**alto**, **basso**, **destra**, **sinistra**} è legato al punto di vista dell'osservatore che "legge" la sequenza, solidale con il piano di gioco. Questi due punti di osservazione apparentemente coincidono al computer¹, dove il bambino adotta il punto di vista dell'esecutore solo immaginandolo, ma questa sovrapposizione non avviene quando si usa il corpo. Siamo convinti che la confusione dei messaggi nasca da questo disallineamento in ciò che il bambino percepisce. Detto diversamente, in qualche modo stiamo alimentando una *misconcezione*.

2.4 Il caso CodyFeet

CodyFeet [1] è una proposta didattica nella quale il "programma" è costituito da tessere colorate, con opportuni incastri, che vengono posizionate sul pavimento per tracciare un percorso. L'esecutore entra in una casella, legge il comando e lo esegue spostandosi nella casella indicata. Il programma e il percorso, dunque, coincidono. Le tessere *operative* sono di tre tipi: **rotazione a sinistra** (gialla), **rotazione a destra** (rossa), **non ruotare** (grigia). Le rotazioni sono sul posto e si intendono di 90°. Diversamente dal *Labirinto classico*, però, ogni tessera ha un avanzamento implicito nella direzione indicata, rispettivamente a sinistra, a destra o in avanti (per questo l'assenza di rotazione ha perfettamente senso e non indica un *no-operation*).

In quest'attività il punto di vista è nuovamente quello dell'esecutore. Tuttavia, c'è una stratificazione di operazioni (ruota e vai avanti) in un solo comando (la tessera) che rischia di costituire un ulteriore livello di ambiguità poiché aggiunge involontariamente complessità senza che questa sia resa chiara ed esplicita.

2.5 Quale modello?

Quale interpretazione scegliere? Domanda insidiosa alla quale si sarebbe tentati di rispondere con un laconico "dipende". Ci viene però in aiuto un'altra semplice considerazione: per l'apprendimento del pensiero computazionale nell'infanzia e nella primaria vengono usate indifferentemente (e in maniera intercambiabile) attività di coding e di robotica educativa.

¹ In molte rappresentazioni di cui i bambini sono consapevoli (compresi i segnali stradali) la freccia verso l'alto indica avanti.

Se dunque vogliamo mantenere la coerenza tra le proposte didattiche è preferibile usare interpretazioni che siano allineate con quelle usate in robotica, ovvero equivalenti al *Labirinto classico*.

2.6 “Emendare” CodyFeet

Riallineare *CodyFeet* al *Labirinto classico* è semplice se si accetta di ridurre ogni tessera ad un comando: le tessere colorate indicano solo la rotazione sul posto, senza movimento, e la tessera grigia torna ad assumere il significato di **avanti**. Per consentire l'equivalenza diretta tra percorso e programma basta stratificare le tessere. L'esecutore raccoglie (*fetch*²) una tessera e la esegue. Se si tratta di una rotazione, nella stessa casella ne troverà un'altra sotto quella precedente che gli chiederà, per esempio, di andare avanti³.

Questa modifica è stata proposta ad alcune insegnanti e probabilmente sarà sperimentata durante il prossimo anno scolastico.

3. Conclusioni

La breve esemplificazione in questo contributo, per quanto rapida e poco approfondita, ha inteso sollevare una questione di ambiguità relativamente alle attività di coding proposte per la scuola dell'infanzia e per la primaria. Più precisamente, il problema di come la diversificazione dei meccanismi sottostanti tali attività di cui non sono consapevoli gli insegnanti e spesso nemmeno chi ne cura la formazione, rischia di ricadere sugli allievi nella forma di messaggi dissonanti che alimentano confusione.

In calce alla discussione, abbiamo presentato un esempio di come una di queste proposte possa essere allineata ad altre più comuni con una modifica che, pur sostanziale dal punto di vista formale, è di semplice utilizzo per gli insegnanti.

Bibliografia

- [1] CodyFeet. Coding per l'infanzia. <https://codemooc.org/codyfeet/>
- [2] CodyFeet, CodyColor e CodyRoby a confronto. <https://codemooc.org/codyfeet-codycolor-codyroby/>
- [3] Guerre Stellari. <https://programmailfuturo.it/come/ora-del-codice/guerre-stellari>
- [4] Il Labirinto. <https://programmailfuturo.it/come/primaria/vecchie-lezioni-tecnologiche/corso-rapido/il-labirinto/>.
- [5] Labirinto: Sequenze. <https://programmailfuturo.it/come/primaria/vecchie-lezioni-tecnologiche/corso-1/lezione-4>.
- [6] Ora del Codice. <https://programmailfuturo.it/come/ora-del-codice>.

² L'equivalenza tra le operazioni di raccolta e di *fetch* è dell'autore stesso di *CodyFeet* [2].

³ Con questo metodo si risolve anche il problema degli incroci del percorso su sé stesso ma si rinuncia alla correttezza per costruzione [2] data dagli incastri.

Robo Amici della Terra

Daniela Troia¹

¹ IC G. Rodari – D. Alighieri – G. Marconi

dnltr71@gmail.com

Abstract

Il racconto descrive un'esperienza realizzata con due classi quarte di scuola primaria. Il progetto ha proposto un percorso laboratoriale ispirato alla robotica educativa e alla programmazione a blocchi, finalizzato all'introduzione dei concetti fondamentali dell'informatica. Attraverso l'utilizzo del robot Codey Rocky, integrato in un contesto narrativo significativo, sono stati esplorati i concetti di evento, istruzione, sequenza, ciclo e condizione. L'intero progetto è stato pianificato secondo i traguardi e gli obiettivi di apprendimento presenti nel curriculum digitale della scuola, costruito secondo il DigComp2.2 e le indicazioni suggerite dalla Proposta di Indicazioni Nazionali per l'insegnamento dell'Informatica nella scuola (CINI). L'esperienza dimostra come sia possibile proporre un percorso di informatica rigoroso anche nella scuola primaria, partendo dal fare, dall'osservazione e dalla riflessione condivisa.

1. Introduzione

L'introduzione del pensiero computazionale nella scuola primaria voluto dal Piano Nazionale Scuola Digitale e meglio specificato dai Nuovi Scenari del 2018, ha trovato una sua più precisa definizione grazie al DigComp 2.2, che ha introdotto esplicitamente la programmazione.

Le linee guida STEM hanno ulteriormente chiarito il senso del pensiero computazionale nella scuola, dedicandone un paragrafo e ribadendo il valore dell'informatica come scienza non riducibile a semplici attività di coding.

Secondo la proposta del CINI, l'informatica deve essere introdotta nella scuola primaria come disciplina autonoma, strutturata intorno a concetti fondamentali che permettano agli studenti di comprendere la natura computazionale della realtà.

Il progetto "Robo amici della Terra" ha strutturato il percorso formativo lungo gli obiettivi generali indicati dal documento del CINI, assunti nel curriculum digitale della scuola:

- comprendere e applicare principi e concetti fondamentali della disciplina;
- applicare ai problemi gli strumenti e i metodi dell'informatica;
- analizzare e risolvere problemi costruendone rappresentazioni formali e definendo soluzioni algoritmiche, espresse mediante la programmazione informatica (oggi detta anche "coding");
- valutare l'opportunità di usare o meno tecnologie digitali nella soluzione dei problemi;
- utilizzare in modo consapevole, responsabile, competente, fiducioso e creativo le tecnologie digitali.

ed ha utilizzato la robotica educativa come laboratorio attivo per la sperimentazione e l'astrazione dei concetti.

2. Il progetto

Il percorso si inserisce nell'ambito delle attività relative al curriculum trasversale previsto nel PTOF ed ha come tematica "Territorio ed Ambiente" con particolare attenzione al problema dell'inquinamento atmosferico e all'obiettivo 11 dell'Agenda 2030 "Città e comunità sostenibili". L'esperienza ha visto coinvolti 40 alunni di classe quarta con pregresse esperienze di coding: gli alunni avevano già sperimentato l'uso della programmazione visuale a blocchi con Code.org e Scratch 3.0. Le attività relative all'esperienza si pongono in continuità e sono tese ad un ulteriore approfondimento dei concetti di programmazione, applicandoli alla robotica educativa con l'utilizzo del software Mblock 5.5 e di Codey Rocky. Idea centrale, la possibilità di utilizzare la robotica educativa per simulare possibili interventi a favore dell'ambiente.

2.1 Ambiente di apprendimento: riferimenti pedagogici e metodologici

"Pavimenti bassi, soffitti alti" (S. Papert) hanno caratterizzato un ambiente di apprendimento altamente inclusivo in cui il ricorso iniziale a progetti semplici (pavimenti bassi), è stato affiancato da attività gradualmente più complesse (soffitti alti) in cui sono stati realizzati momenti di peer tutoring. Centrato su fondamenti pedagogici di stampo costruzionista e su metodologie come problem based learning e project based learning, il progetto ha reso le bambine e i bambini protagonisti attivi del processo di apprendimento, portandoli ad ipotizzare e testare ipotesi risolutive a problemi reali. L'integrazione della robotica educativa ha dato concretezza a tale processo in quanto ha richiesto la costruzione di artefatti fisici e digitali funzionali alla risoluzione del problema.

2.2 Il percorso

Si è deciso di usare il film Wall-E come scenario del percorso per la sua duplice attinenza al progetto:

- La tematica, l'inquinamento ambientale;
- L'identità del protagonista, un robotino impegnato nella salvaguardia dell'ambiente.

Il percorso si è articolato in 3 grandi fasi:

1° Fase: Il segreto di Wall-E, introduzione alla robotica;

2° Fase: Amici di Wall-E, analisi degli artefatti fisici e virtuali;

3° Fase: Il sogno di Wall-E, dal problema al progetto.

2.3 Descrizione delle fasi

1° FASE: IL SEGRETO DI WALL-E

Nella prima fase, si è chiesto ai gruppi di dare risposte in merito alla natura di Wall-E, interrogandosi su:

- cosa fosse Wall-E, quali compiti dovesse svolgere e come potesse realizzarli;
- cosa sia un robot e come funziona.

I gruppi hanno risposto tramite form (Fig.1), e si è giunti alla conclusione che Wall-E è un robot e **il suo segreto è che "nasconde un programma grazie al quale può svolgere il suo compito"**.

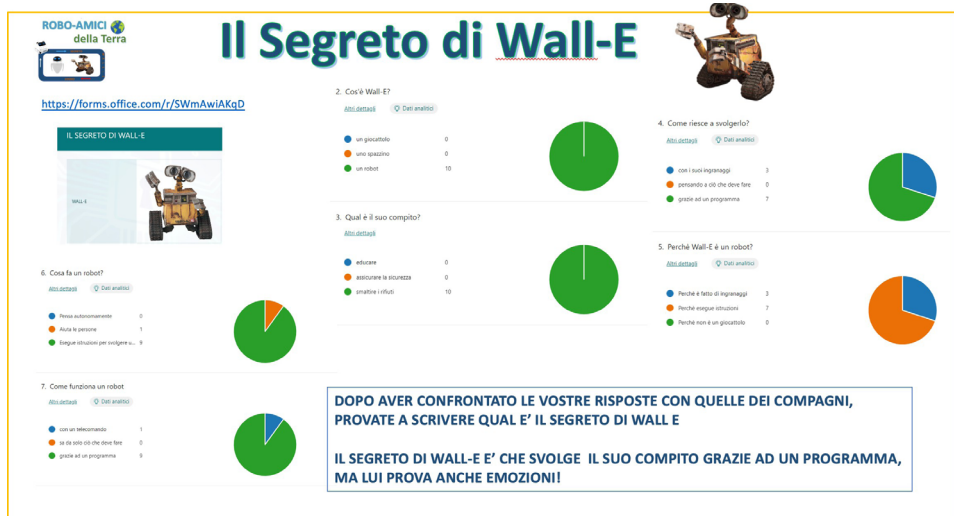


Figura 1: Brainstorming

2° FASE: AMICI DI WALL-E

Si è quindi passati ad esplorare gli artefatti fisici e virtuali: Codey Rocky e Mblock 5.5. Si è partiti dall'analisi dell'hardware: Codey Rocky è stato messo a confronto con Wall-E, per coglierne analogie e differenze fisiche. Ciascun gruppo ha avuto a disposizione un robottino: si è lasciato il tempo di analizzarlo e provare a formulare ipotesi in merito alle sue parti e relative funzioni, infine schematizzate. (Fig.2)

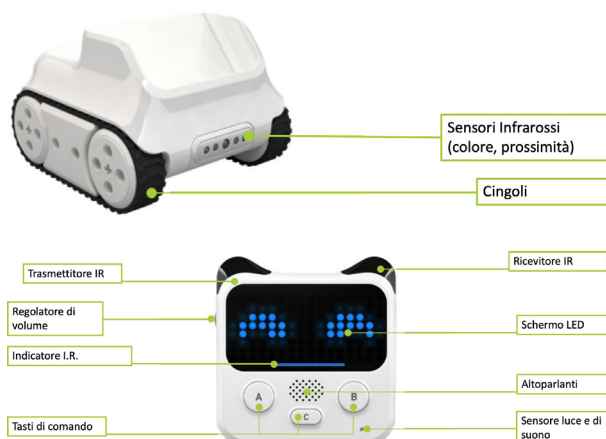


Figura 2: analisi dell'hardware

È seguita la programmazione di Codey Rockey in modo che l'immagine del precedente progetto, diventasse un'animazione. (Fig.5)

Wall E deve piantare un albero: quali azioni sono necessarie?



Figura 5: Sequenze

- **Cicli e Condizioni:** costruire la nozione di automazione dei processi. Gli alunni hanno osservato che la stessa sequenza poteva essere ripetuta e hanno utilizzato il blocco “ripeti X volte” per rendere più efficiente il programma. Questa attività ha permesso di comprendere che un ciclo è un costrutto che consente la ripetizione di un insieme di istruzioni, riducendo la complessità del codice. (Fig.6)

Wall-E deve piantare alberi lungo una strada di 20 metri alla distanza di 5m: quante volte dovrà ripetere la sequenza di azioni?

Un **CICLO** è un pezzo di codice che il robot ripete più volte

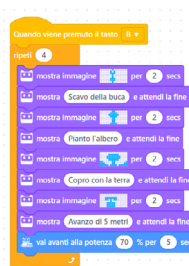
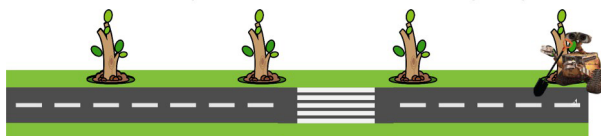


Figura 6: Cicli

Si è quindi passati alle istruzioni condizionali. Partendo dalla situazione narrativa “Wall-E guarda fuori e decide se prendere l’ombrello”, si è introdotto il costrutto “SE...ALLORA”, applicato poi all’interazione con sensori e alla comunicazione.

Gli alunni hanno programmato i robot in modo che reagissero a condizioni ambientali. (Fig.7) Hanno così appreso che le istruzioni condizionali consentono ai programmi di prendere decisioni, adattando il comportamento a seconda dei dati in ingresso.

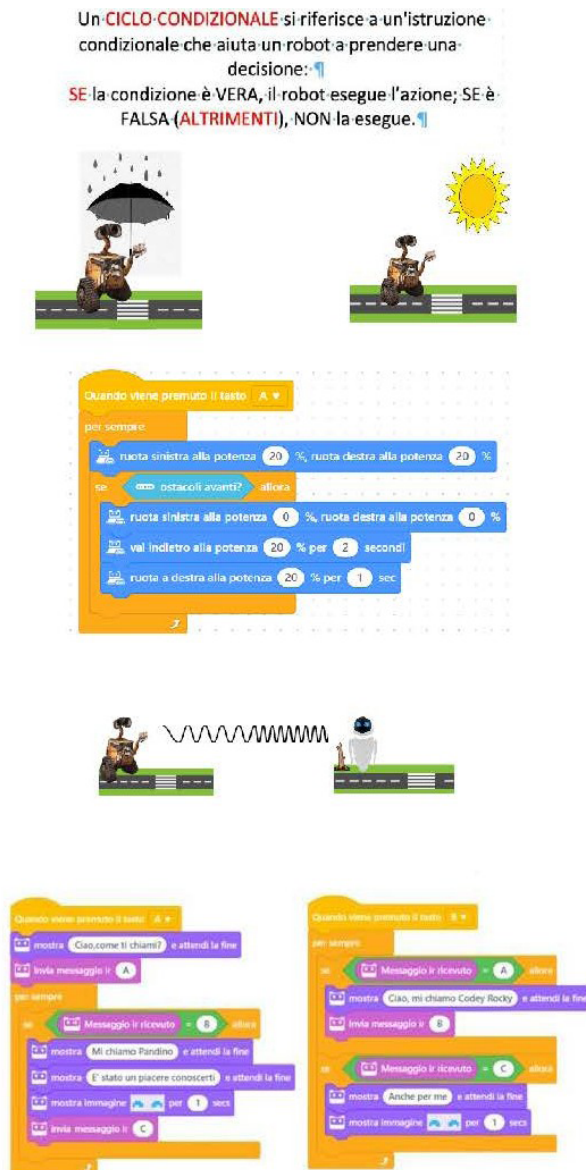


Figura 7: Condizioni

3° STEP IL SOGNO DI WALL-E

Il percorso si è concluso con un compito di realtà: progettare un ambiente urbano capace di risolvere il problema dell'inquinamento, grazie all'utilizzo della tecnologia e dell'informatica.

I bambini hanno progettato una città green simulata, immaginando soluzioni di mobilità sostenibile con robo-bus elettrici e piste ciclabili intelligenti.

- **Programmazione di robo-bici e robo-bus sensore di distanza:** I piccoli Codey Rocky hanno simulato la libera circolazione su piste ciclabili dedicate attraverso l'uso del sensore di prossimità. (Fig.8) e del robo-bus (Fig.9) usando la programmazione per il follow line.



Figura 8: Robo-bici e sensori

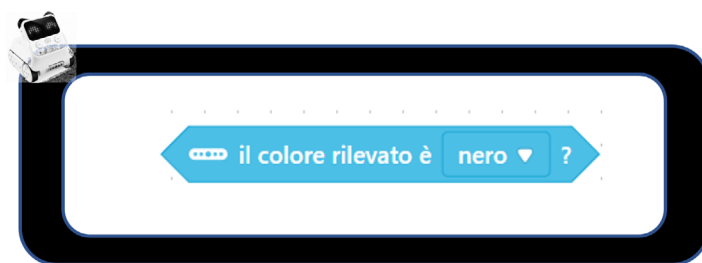


Figura 9: Robo-bus e follow line

- **Programmazione della robo-app e del robo-semaforo.** Codey Rocky senza base sono stati programmati per simulare sia cellulari dotati di app per la richiesta di fermata lungo la traiettoria del bus (Fig.10), che semafori (fig.11), connessi al robo-bus attraverso trasmettitore e ricevitore IR.



Figura 10: Trasmettitore e ricevitore IR

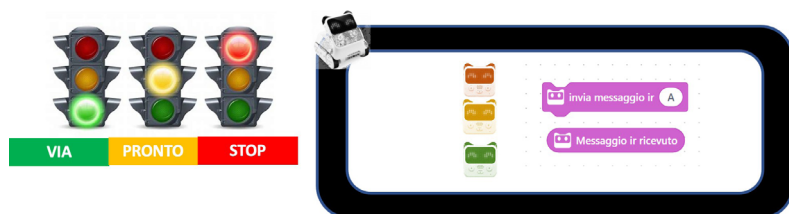


Figura 11: Trasmettitore e ricevitore IR

- **Costruzione del modello di città green e test sulla funzionalità.** Le attività di sperimentazione delle programmazioni sono state affiancate dalla costruzione di un modello di città green con materiale di riciclo: case e gallerie realizzate con scatole di cartone, alberi con cartoncini, fiori con tecnica e carta per origami, basi di compensato, tempere e nastro isolante nero per le strade e la traiettoria del robo-bus. (Fig.12)



Figura 12: Costruzione dello scenario

Testando via via con i robottini la funzionalità delle programmazioni (Fig.13)



Figura 13: Primi test

3. Verifica e Valutazione

La valutazione ha avuto un carattere essenzialmente formativo, centrata sul processo e finalizzata a rendere i bambini consapevoli degli errori commessi e delle strategie attivate per risolverli. A tal fine, per ciascuna fase è stato previsto un momento di metacognizione (Fig. 14) da parte di ciascun gruppo su quanto realizzato, da condividere con gli altri gruppi e funzionale all'autovalutazione.

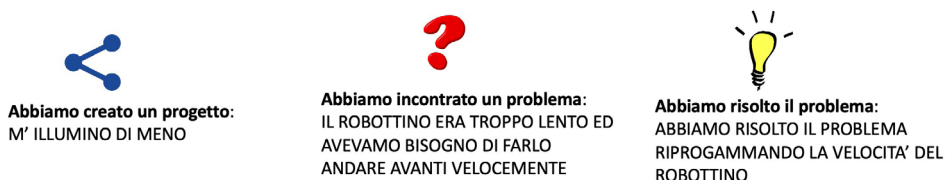


Figura 14: Metacognizione

Al termine del percorso i gruppi hanno completato a più mani una scheda di valutazione finale del progetto (Fig.15):



COSA ABBIAMO IMPARATO: Abbiamo imparato a programmare Codey Rocky e a risolvere i problemi che abbiamo incontrato svolgendo i vari progetti che riguardavano i sensori, le emozioni...

COSA CI E' PIACIUTO DI PIU': Ci è piaciuto di più il progetto finale perché abbiamo dipinto, creato origami e per la prima volta abbiamo fatto una così lunga programmazione.

QUESTO PROGETTO CI E' SERVITO A: riflettere su quello che facciamo e a pensare ai passi necessari da compiere per risolvere i problemi.

Figura 15: Valutazione

Il progetto infine è stato presentato alle famiglie: i bambini hanno raccontato tutte le fasi, mostrato e spiegato le programmazioni ai genitori, raccontando le difficoltà incontrate, le strategie messe in atto, i successi raggiunti. (Fig.16)



Figura 16: Presentazione del progetto

4. Conclusione

L'esperienza "RoboAmici della Terra" ha evidenziato la validità di introdurre l'informatica già nella scuola primaria, andando oltre una visione riduttiva del coding come attività creativa o come esperienza tecnico-operativa priva di riferimenti concettuali strutturati.

Il percorso ha permesso agli alunni di esplorare in modo progressivo e rigoroso alcuni concetti fondanti dell'informatica, attraverso situazioni autentiche e contestualizzate. I bambini hanno compreso che un robot è un sistema che esegue istruzioni, sviluppando così una rappresentazione chiara dell'algoritmo come modello di azione e di pensiero.

Tale percorso si è inserito all'interno del curriculum digitale della scuola, che prevede una specifica sezione dedicata allo sviluppo del pensiero computazionale e della programmazione, elaborata proprio sulla base delle indicazioni fornite dal CINI. Ciò ha garantito coerenza pedagogica e progressione verticale degli apprendimenti, promuovendo l'informatica come disciplina di senso fin dai primi anni di scuola.

Un'introduzione precoce e ben orientata dell'informatica può permettere ai bambini di sviluppare capacità logiche, analitiche e progettuali, favorendo una comprensione più profonda della realtà tecnologica che li circonda. L'informatica, intesa come scienza del pensiero computazionale, diventa così uno strumento per leggere, interpretare e trasformare il mondo, offrendo agli studenti le basi culturali per agire consapevolmente nella società digitale.

Bibliografia

- [1] Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc.
- [2] Mitchel Resnick, 2018, *Come i bambini. Immagina, crea e condividi. Coltivare la creatività con il Lifelong Kindergarten del MIT*. Trento. Eickson
- [3] Michael Lodi, Simone Martini, Enrico Nardelli. Abbiamo davvero bisogno del pensiero computazionale? *Mondo Digitale*, 2017, 72, pp.1-15. fffhal-01656340f
- [4] *Robot & scuola. Guida per la progettazione, la realizzazione e la conduzione di un Laboratorio di Robotica Educativa (LRE)* di Giovanni Marciàno - Ed. Hoepli
- [5] DigComp 2.2, il quadro delle competenze digitali per i cittadini <https://repubblicadigitale.gov.it/portale/-/digcomp-2.2-il-quadro-delle-competenze-digitali-per-i-cittadini>
- [6] MIUR.(2018).Indicazioni Nazionali e Nuovi Scenari. miur.gov.it/documents/20182/0/Indicazioni+nazionali+e+nuovi+scenari/
- [7] Raccomandazione del Consiglio del 22 maggio 2018 relativa alle competenze chiave per l'apprendimento permanente.
- [8] Proposta di Indicazioni Nazionali per l'insegnamento dell'Informatica nella Scuola <https://www.consortio-cini.it/images/Proposta-Indicazioni-Nazionali-Informatica-Scuola-numerata.pdf>
- [9] Piano Nazionale Scuola Digitale, La scuola digitale - MIUR (istruzione.it)

Sitografia

[1] Programma il Futuro - Code.org - ProgrammaIlFuturo.it

[2] <https://ide.mblock.cc/>

[3] <https://www.innovationforeducation.it/academy/cose-steam-on-board-la-piattaforma-per-fare-coding-a-distanza-con-mblock-e-google-classro>

Il linguaggio dei cristalli: simboli e narrazione per l'informatica nella scuola primaria

Guglielmo Abbruzzese¹, Luciana Mattei², Maurizia Di Fiore³, Luca Forlizzi⁴

¹ Università degli Studi di Roma "Tor Vergata", Lab. Naz. Informatica e Scuola (CINI)

abbruzzese@axp.mat.uniroma2.it

² I.C. Giuliano/Goldoni di Latina

luciana.mattei@icgiuliano.edu.it

³ I.C. Giuliano/Goldoni di Latina

maurizia.difiore@icgiuliano.edu.it

⁴ Università degli Studi dell'Aquila, Lab. Naz. Informatica e Scuola (CINI)

luca.forlizzi@univaq.it

Abstract

L'esperienza descritta ha mostrato come, anche nei primi anni della scuola primaria, sia possibile introdurre concetti chiave dell'informatica attraverso attività fondate su narrazione, manipolazione concreta e rappresentazione visuale. Il *linguaggio dei cristalli*, inserito in una cornice narrativa coinvolgente e supportato da dispositivi fisici e digitali, è stato recepito con naturalezza dagli alunni e ha favorito l'acquisizione intuitiva di strutture proprie dei linguaggi formali. Questo articolo viene proposto nelle intenzioni come report di esperienza didattica, ma l'osservazione in classe ha consentito anche di calibrare un protocollo replicabile e condiviso, in grado di adattarsi ai diversi livelli cognitivi, oltre che fornire indicazioni utili per rilevare e colmare eventuali lacune nei prerequisiti delle attività. I risultati incoraggiano a proseguire nella direzione di un'informatica precoce orientata alla comprensione strutturata e al problem solving, integrando elementi di computazione fin dalle fasi iniziali del curriculum scolastico. Le evidenze raccolte costituiranno la base per il consolidamento e l'estensione della sperimentazione nelle fasi successive del progetto PRIN "PINI - Apprendimento dell'Informatica".

1. Introduzione

Fino alle recenti indicazioni del MiM [19]¹ gli insegnanti non hanno avuto ampie opportunità di riflessione pubblica sul ruolo dell'informatica nella scuola [2], pur potendo contare sul supporto strutturato di organizzazioni internazionali come CSTA [12], Informatics Europe [15], Code [11] e in Italia il CINI [10] e Programma il Futuro [27]. Nella scuola primaria, le attività di informatica sono solitamente realizzate con robotica educativa e strumenti come Scratch e Code.org, talvolta ispirate alle linee guida CINI [9], più spesso erroneamente interscambiate con le

1 Ancora in fase di pubblica discussione alla stesura del presente articolo.

competenze digitali [20]. Aspetti formali dei linguaggi per la rappresentazione e il trattamento dell'informazione (dagli studi e classificazioni di Chomsky sui linguaggi [7][8]) restano confinati a livello universitario, sebbene esistano approcci graduali e visuali per studenti dai 12 anni [6][14][16]. Alcune teorie cognitive suggeriscono che:

- concetti astratti come gli automi sono accessibili dopo gli 11-12 anni [25];
- il costruttivismo favorisce un approccio concreto, assistito e graduale [29];
- il game-based learning facilita la comprensione attraverso ambienti visuali.

Pur con tali premesse, Piaget stesso non esclude che già nella fase pre-operatoria (6-8 anni) i bambini possano utilizzare simboli [24], aprendo la strada a un approccio pre-formale basato su rappresentazioni visuali e ludiche. Questo collima con le “powerful ideas” di Papert [22] e le esperienze documentate da Bers [4][3], suggerendo la possibilità di sviluppare un'intuizione computazionale su “regole” e “stati” ancor prima di un loro inquadramento formale.

In questo quadro teorico, il presente lavoro documenta un'esperienza pilota svolta presso la scuola primaria “C. Goldoni” (I.C. “G. Giuliano” di Latina, 10 insegnanti, 12 classi) finalizzata a sviluppare, attraverso l'impiego di linguaggi formali e concetti di informatica, le competenze teorico-pratiche necessarie per sostenere una valutazione delle capacità di problem solving e di ragionamento logico condotta attraverso un puzzle. L'attività si inserisce nel progetto PRIN “PINI – Apprendimento dell'Informatica”² e mira al conseguimento dei seguenti obiettivi di apprendimento, definiti in [9]: O-P3-P-1, O-P3-P-2 (ambito Algoritmi); O-P3-D-1, O-P3-D-2, O-P5-D-1, O-P5-D-2 (ambito Dati e Informazioni).

Sebbene concepito come resoconto di una esperienza didattica, l'osservazione in aula durante le diverse sessioni di lavoro ha permesso di definire un protocollo replicabile e standardizzato [1], flessibile rispetto ai vari livelli cognitivi e capace di offrire indicazioni utili per individuare e compensare eventuali carenze nei prerequisiti delle attività.

2. L'attività

Nella fase preliminare del progetto è stato utilizzato il puzzle digitale “Rimbalzi” [23], prototipo sviluppato come applicazione web e validato con le insegnanti per adeguarne la proposta alle fasi di sviluppo cognitivo [25]; tale approccio per definire sequenze didattiche a livelli basate sullo sviluppo anziché banalmente sulla progressione di difficoltà.

2.1 Il gioco

Il gioco³ utilizza una matrice quadrata (2x2-4x4), da ciascun bordo possono essere lanciate “palline/raggi di luce”: queste, attraversano le celle (contenenti al massimo un ostacolo), fuoriescono da un bordo opposto – eventualmente deviate da ostacoli interni (linee a $\pm 45^\circ$) che causano deviazioni di 90° o 270°), con l'obiettivo di dedurre gli ostacoli analizzando le traiettorie di ingresso/uscita (Figura 1).

² Progetto di Rilevante Interesse Nazionale (PRIN), finalizzato a valutare i benefici trasversali della programmazione su altri domini - tuttora in corso durante la redazione di questo documento.

³ Sono stati considerati elementi di difficoltà intrinseca per tarare le sequenze didattiche.

3.1 La storia

L'attività *unplugged* “Il Cristallo Magico e il Labirinto delle Luci” [17] è stata contestualizzata in un universo fantastico (ispirato a Harry Potter) dove giovani “maghetti fotonici” apprendono i principi di navigazione in un labirinto con cristalli (ostacoli) dotati di un *linguaggio* simbolico da decifrare per trovare l'uscita.

3.2 Il labirinto delle luci

Progettazione e utilizzo. L'analogia con l'ottica geometrica ha ispirato la creazione di un prototipo fisico composto da:

1. *griglia 3x3*, realizzata con materiali da imballaggio su base di cartone (Fig. 2);
2. *sistema di specchi* (10x15 cm), innestabili a $\pm 45^\circ$ lungo guide diagonali;
3. *puntatore laser* (“bacchetta magica”), per tracciare percorsi luminosi attraverso fori sagomati ai bordi della griglia.

Una seconda *griglia dei percorsi* (Figura 3) in compensato, integrava, con:

- punti di ingresso/uscita etichettati (a-n);
- ganci per ricostruire traiettorie con fili di lana.

Applicazione didattica. I dispositivi sono stati utilizzati strategicamente:

- come supporto concreto per bambini con difficoltà di visualizzazione spaziale;
- in abbinamento a sequenze didattiche strutturate (§3.3);
- per tradurre i principi astratti in esperienza tattile e visiva.

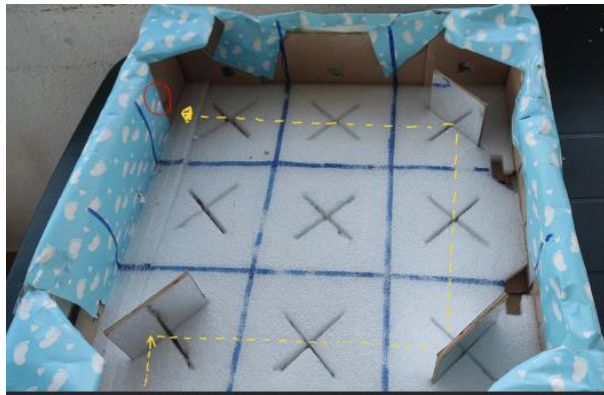


Figura 2: Il labirinto delle luci

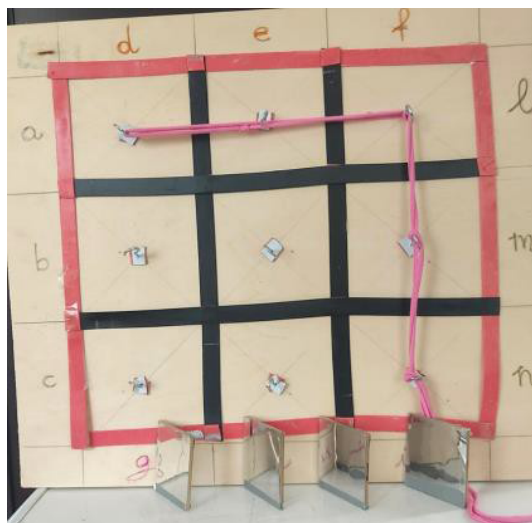


Figura 3: Gli ostacoli innestabili (cristalli) e la griglia dei percorsi

3.3 Il linguaggio dei cristalli⁴

Approccio ai linguaggi. L'attività ha introdotto intuitivamente concetti informatici legati ai linguaggi [30], con duplice obiettivo:

1. evidenziare analogie coi linguaggi naturali (e formali);
2. far emergere regole e strutture: attraverso esperienze concrete.

L'approccio, volutamente non teorico, ha privilegiato:

- collegamenti interdisciplinari;
- riferimenti all'esperienza linguistica quotidiana;
- scoperta guidata di pattern e relazioni.

Razionali didattici. Per diventare “*maghetti fotonici*” gli studenti apprendono il *linguaggio dei cristalli* (ostacoli/specchi), strutturato in modo graduale:

1. Approccio ai linguaggi
 - *Classi 4^a e 5^a*: Riflessione su analogie col linguaggio naturale, richiamando:
 - **simboli** (lettere di un alfabeto);
 - **parole** (lettere concatenate);
 - **frasi** (parole concatenate, soggette a regole grammaticali).
 - *Obiettivo*: Far intuire che tali strutture sono universali, proprie non solo dell'Italiano [30].
 - *Classi 2^a e 3^a*: Focus su *simboli* e *significato*, senza approfondire gli aspetti grammaticali.
- a. Lessico minimo per le traiettorie. Per evitare errori legati alla *confusione speculare* [26], sono state introdotte 4 parole-chiave che diventeranno la base del linguaggio:

4 Questa metodologia è stata perfezionata nel corso delle attività introduttive svolte con le varie classi, motivo per cui non è descritta nella progettazione iniziale allegata al §3.4.

- “Su”, “Giù”, “Destra”, “Sinistra”

- Simboli degli ostacoli:

Zin (“ / ”) e **Zum** (“ \ ”)

Ogni combinazione tra ostacolo e direzione genera una *regola di riflessione* (Figura 4).

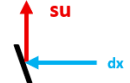
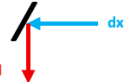
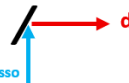
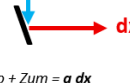
Alfabeto (simboli)	Zin “/”	Zum “\”
Parole (comandi)		
“Su”	 Da $sx + \text{Zin} = \text{in alto}$	 Da $dx + \text{Zum} = \text{in alto}$
“Giù”	Da $dx + \text{Zin} = \text{in basso}$  giù	Da $sx + \text{Zum} = \text{in basso}$  giù
“Destra”	Dal basso + Zin = a dx  basso	Dall'alto + Zum = a dx  alto
“Sinistra”	 sx Dall'alto + Zin = a sx	Dal basso + Zum = a sx  sx

Figura 4: Il linguaggio dei cristalli

Sinonimi e automazione.

- Se gli studenti conoscono il concetto di *sinonimo* (in genere, dalla 3^a), si rimarca che:
 - “ $sx + zin$ ” = “ $dx + zum$ ” = “ su ” (stessa traiettoria).
- Dopo aver familiarizzato con le logiche espresse, la maggior parte degli alunni ne ha applicato le regole senza necessità di consultazione preventiva.

Introduzione di automi semplificati (classi 4^a+). Oltre alla *griglia dei percorsi* fisica del labirinto (Figura 3), ne è stata introdotta una versione simbolica (Figura 5) con gli *obiettivi didattici*:

- astrazione progressiva dal concreto (labirinto fisico → diagrammi);
- preparazione a concetti di **computazione** (automi, sintassi).

Per favorire la comprensione dei percorsi rappresentati a destra in Figura 6 sono stati informalmente introdotti i concetti di:

- Contenuto della cella (**stato**):
 - vuota $\emptyset$, oppure popolata con ostacolo (**Zin** / o **Zum** \).

- Passaggio da una cella all'altra (**transizione**):
 - Qual è il meccanismo di passaggio da una cella all'altra? In base alla direzione in ingresso sulla cella, e alla presenza di eventuale ostacolo.
- Le regole di transizione sono rappresentate come frecce tra celle, modellate come automi a stati finiti semplificati.

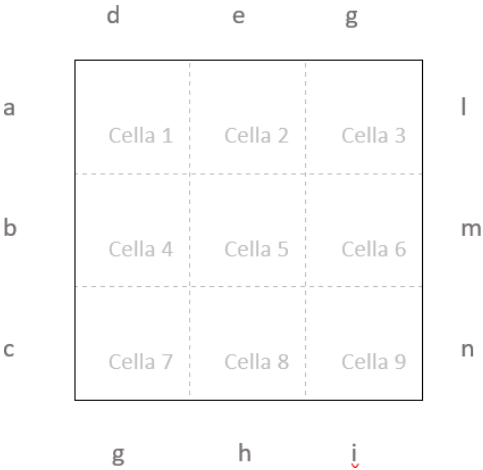


Figura 5: Griglia dei percorsi virtuale (vuota)

Esempio applicato in classe. In Figura 6 si riporta la sequenza di 3 lanci su griglia con configurazione nota, tradotti in percorsi *grafici* (freccie) e *regole di transizione* (es.: “ingresso da SINISTRA con ZIN → devia in SU”). Esempi di tale tipologia sono stati utilizzati nelle classi durante le attività didattiche per introdurre i formalismi ritenuti congrui in base all’età.

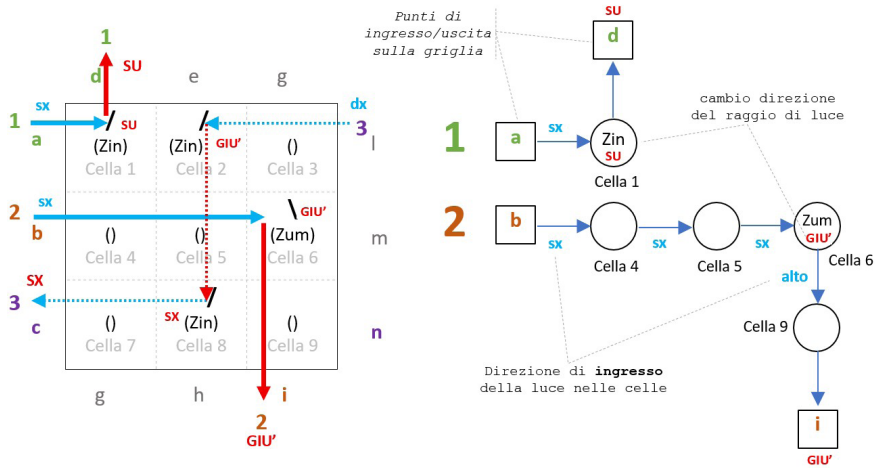


Figura 6: Esempio di rappresentazione percorsi con grafi

Esercizi su carta raffigurante la *griglia dei percorsi virtuale* con ostacoli sono stati proposti a tutte le classi. Con la seguente modalità, il discente:

1. tenta una soluzione su carta o LIM (lato sinistro Figura 6);
2. prova a visualizzare in caso di difficoltà sulla griglia dei percorsi fisica (Fig. 3);
3. imposta sul labirinto delle luci la configurazione degli ostacoli e prova la soluzione col laser (Figura 2);
4. se l'attività è svolta in laboratorio, imposta gli ostacoli sull'app online e prova la congruenza della stessa configurazione/soluzione;
5. solo per le classi quarte (/quinte), prova a rappresentare i percorsi in griglia virtuale con i formalismi riportati nel lato destro della Figura 6).

3.4 Sequenza didattica e attività in classe

L'attività, introdotta attraverso la narrazione descritta nel §3.1, è stata inizialmente realizzata (e ottimizzata a valle delle riflessioni tra docenti) con una classe seconda (Figura 7). Il report di dettaglio dell'intera attività è qui documentata [1] e resa disponibile per una comprensione puntuale dei passaggi didattici realizzati.

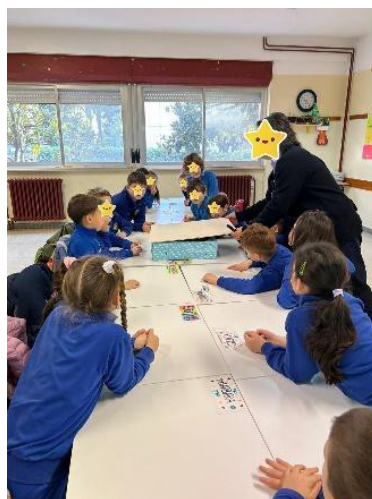


Figura 7: Attività in classe con labirinto e griglia dei percorsi

La prima esecuzione ha richiesto 5 ore totali (inclusi spostamenti/pause), di cui:

- 2h30m: con labirinto fisico e griglia;
- 1h40m: con versione digitale del gioco.

A valle di ogni sessione con ciascuna classe il protocollo è stato tarato sulla base delle note riportate nella quinta colonna della Figura 8 ("Note e Osservazioni"), dalle quali sono scaturiti i potenziali miglioramenti per ciascuna delle fasi progettate.

Tabella 1: Stralcio dalle attività iniziali

Attività del 18/12/2024 - Appunti - Rev. 18/12/2024

Classe: 2X

Breve descrizione della classe: 2X, 18 studenti (10 maschi e 8 femmine). L'insegnante la introduce come "una classe con tanti bambini con neurotipie che però sono brillanti nelle loro bizzarrie".

ID	TEMPI	ATTIVITA'	P/U	I/G	OBIETTIVI/DESCRIZIONE	NOTE E OSSERVAZIONI	MIGLIORAMENTI
		(P=Plugged, U=Unplugged)					
#01	00:00:00 (05:00)	Disposizione banchi	U	G	L'insegnante invita a disporre i banchi al centro della classe, formando un rettangolo con due file di banchi contigui, con studenti a coppie uno di fronte all'altro.	Il tavolone "unplugged" risulta un po' troppo lungo e a tratti dispersivo	Studiare una configurazione che minimizzi la distanza massima tra due studenti, tipo circolare
#02	00:05:00 (05:00)	Presentazioni	U	G	Ciascuno studente si presenta col suo nome. La figura esterna ([GA]) viene introdotta lasciando intendere che possa rappresentare qualcosa in più di quello che sembra, e che custodisca qualche segreto.	Viene chiesto agli studenti se hanno idea di cosa sia l'informatica. Alcune risposte: "una cosa tecnologica", "i computer",	
#03	00:10:00 (05:00)	Racconto storia	U	G	La storia elaborata, "Il Cristallo Magico e il Labirinto delle Luci", viene letta e recitata per intero dall'insegnante rispettando la stesura iniziale. Vengono introdotti i personaggi principali (il mago buono, il mago cattivo, gli aspiranti maghetti). I bambini vanno a identificarsi con gli aspiranti maghetti che devono muoversi con la magia nel labirinto delle luci. Si lascia sottintendere che il formatore esterno ([GA]) potrebbe essere il mago del labirinto.	La storia, più che troppo lunga, va definita meglio nella parte centrale, prestando maggiore attenzione anche nella prima parte agli aspetti di chiarezza ed alla scelta dei termini da utilizzare. Riesce comunque a creare la giusta atmosfera e interesse sul come andrà ad evolversi l'attività	Il lavoro di fino sul linguaggio specifico va effettuato a più riprese: risulta opportuno selezionare solo le parti rilevanti, minimizzando le informazioni utili. Si risolve di estrarre/elaborare nel frattempo un insieme di parole/espressioni chiave da utilizzare nell'ambientazione e nella narrazione.
#04	00:15:00 (10:00)	Introduzione labirinto delle luci	U	G	La scatola rappresentante il labirinto delle luci viene finalmente scoperta e mostrata, nella curiosità generale, al centro del tavolone.	L'associazione tra mappatura su tabellone e ingressi/uscite su griglia di gioco risulta molto	Il tabellone di compensato è idoneo a contenere la scatola del gioco le

Evoluzione dei tempi:

- *Classi seconde*:
 - Riduzione da 5h → 3h per l'ultima classe coinvolta;
 - Integrazione dei concetti rifiniti (§3.3.) già nelle ultime due sessioni.
- *Classi terze*: tempi più brevi (maggiore reattività degli alunni);
- *Classi quarte/ (quinte)*: 2 ore sufficienti per raggiungere gli obiettivi.

Punti chiave:

1. Adattamento progressivo basato su feedback (riduzione tempi tra prima e ultima sessione).
2. Differenziazione per età:
 - *Seconde*: Approccio più graduale;
 - *Terze+*: Maggiore autonomia → tempistiche compresse;
3. Documentazione sistematica (figure 7-8) per replicabilità.

3.5 Strategie di osservazione e valutazione adottate

Come già dichiarato, l'attività si proponeva di valutare eventuali mancanze degli alunni in merito ai prerequisiti individuati, e di colmarli entro i tempi di svolgimento previsti. Le osservazioni sono state condotte basandosi sulla consolidata conoscenza degli studenti da parte delle insegnanti integrata con la risposta degli stessi alle varie situazioni prospettate loro attraverso l'uso degli strumenti unplugged e plugged. Essendo l'obiettivo la massimizzazione della comprensione degli aspetti teorici e tecnologici, sono stati investiti sui bambini tempi differenti a seconda delle necessità individuali; tale approccio ha garantito anche l'accessibilità ai giochi a studenti con bisogni educativi particolari.

Con tale precisazione risulta chiaro che l'approccio empirico e inclusivo non ha reso necessari strumenti di valutazione eccessivamente formalizzati; ci è stato possibile individuare la seguente distribuzione delle competenze acquisite e dei comportamenti degli alunni -condivisa con le insegnanti- che con tolleranze minime può essere considerata valida per tutte le classi (con classi medie di 20 alunni circa):

Tabella 2. Distribuzione media dei risultati delle classi

Etichetta	Percentuale	Tipologia studente
<i>Lightning</i>	5-10%	Acquisisce immediatamente i concetti veicolati e sviluppa strategie per risolvere problemi anche più complessi di quelli presentati
<i>Centered</i>	60-80%	Comprende nei tempi e nei modi previsti i concetti veicolati attraverso il gioco
<i>Misguided</i>	10-15%	Presenta alcune misconcezioni legate al funzionamento degli specchi, che vengono più o meno agevolmente corrette con l'utilizzo combinato degli strumenti unplugged e plugged
<i>Mentorable</i>	0-5%	Non raggiunge pienamente gli obiettivi con le attività proposte, e mostra qualche debolezza della quale si dovrà tenere conto nelle fasi successive del lavoro ⁵

Prima di essere tarata in base alle varie età, l'attività era inizialmente progettata in 2 moduli da 2 ore onde evitare stanchezza e potenziale perdita di attenzione delle classi, che però si sono dimostrate interessate e reattive alle variazioni previste nelle varie fasi dello svolgimento. Uno alla volta, un passo alla volta, e salvo situazioni particolari tutti hanno completato il percorso fin dove proposto; con tempi diversi e in qualche caso con comprensibile e necessaria pazienza, ma il risultato è stato generalmente soddisfacente e positivamente sorprendente anche a giudizio delle stesse insegnanti.

4. Considerazioni finali

Il successo dell'attività ha confermato il potere della narrazione come leva pedagogica: calati nel ruolo di "maghetti fotonici", gli alunni hanno affrontato con entusiasmo e curiosità sfide che, pur nella loro semplicità apparente, veicolano i primi rudimenti dell'informatica formale. La metafora del labirinto delle luci si è rivelata uno strumento potente per attivare immaginazione, pensiero logico e capacità di astrazione, trasformando ostacoli luminosi e traiettorie deviate in occasioni concrete per ragionare su regole, stati e transizioni. Il linguaggio dei cristalli, costruito passo dopo passo attraverso simboli e percorsi, ha permesso agli studenti di entrare in contatto con strutture profonde del pensiero computazionale, senza mai perdere il senso del gioco e della scoperta. Questa esperienza suggerisce che, se opportunamente guidati, anche i più piccoli possono iniziare a parlare la lingua dell'informatica – non come tecnica astratta, ma come racconto che si fa conoscenza. La strada è aperta: le evidenze raccolte costituiranno la base per il consolidamento e l'estensione della sperimentazione nelle fasi successive del progetto, portando le luci del labirinto ad altri bambini e insegnanti.

Ringraziamenti

All'I.C. “**G. Giuliano**” di Latina per la collaborazione, in particolare la DS (dott.ssa Ramacci) e le maestre Ida Borrelli, Silvia Cannizzaro, Lorena Checcacci, Lina Ciervo, Annalisa D'Amico, Maurizia Di Fiore, Romina Mattarelli, Luciana Mattei, Vanessa Scuoch, Carmen Tabellini per la loro impagabile professionalità, disponibilità e dedizione.

Al Prof. Enrico Nardelli, per aver ispirato le attività descritte e aver messo a disposizione la versione prototipale dell'app “Rimbalzi”.

Fondi

Si riconosce il supporto finanziario ricevuto nell'ambito del Piano Nazionale di Ripresa e Resilienza (PNRR), Missione 4, Componente 2, Investimento 1.1, Avviso n. 104 pubblicato il 2.2.2022 dal Ministero dell'Università e della Ricerca (MUR), finanziato dall'Unione Europea – NextGenerationEU – Titolo del Progetto ‘Learning Informatics’ – CUP E53D23007720006 – Decreto di Concessione della Sovvenzione n. 959 adottato il 22/04/2022 dal Ministero dell'Università e della Ricerca (MUR).

Bibliografia

- [1] Abbruzzese, G. (2025). ITADInfo25_Shared. Tratto da Shared Drive: <https://drive.google.com/file/d/1TQWgpY2tul5wCXNHkUxUNST4U8qYq0bS/view?usp=sharing>
- [2] Abbruzzese, G., Celestino, T., & De Vitis, A. (2023). Epistemologia, etica ed estetica dell'informatica: insegnamento tra tecnologie e mondo digitale. *Nuova Secondaria Ricerca*, 2, 431- 450.
- [3] Bers, M. (2018). Coding as a Playground. Programming and Computational Thinking in the Early Childhood Classroom. Eye On Education, ISBN 9780367900502.
- [4] Bers, M., Flannery, L., Myers, E., & Sullivan, A. (2014, 03). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education*, 72, 145-157. doi: 10.1016/j.compedu.2013.10.020
- [5] Bruner, J. (1990). Acts of meaning. London: Harvard University Press.
- [6] Castro, F. (2022). Exploring the Use of Finite-State Machines and Game Creation to Teach Computational Thinking in Middle Schools. In I. ‘. Education (Ed.). 2, p. 618. Dublin, Ireland: ITiCSE 2022. doi: 10.1145/3502717.3532137
- [7] Chomsky, N. (1956). Three models for the description of language. Cambridge, Massachusetts: Massachusetts Institute of Technology.
- [8] Chomsky, N. (1959). On Certain Formal Properties of Grammars. *INFORMATION AND CONTROL*, 2, pp. 137-167.
- [9] CINI. (2018). Documenti. Retrieved from Informatica e Scuola: <https://www.consortio-cini.it/index.php/it/lab-informatica-e-scuola/documenti>

- [10] CINI. (2025). CINI home page. Retrieved from Consorzio Interuniversitario Nazionale per l'Informatica: <https://www.consorzio-cini.it>
- [11] Code.org. (2025). Code.org. Retrieved from Code.org: <https://studio.Code.org/home>
- [12] CSTA. (2025). Computer Science Teachers Association. Retrieved from Computer Science Teachers Association: <https://csteachers.org/>
- [13] Egan, K. (1989). *Teaching as Story Telling: An Alternative Approach to Teaching and Curriculum in the Elementary School*. The University of Chicago Press.
- [14] Forlizzi, L., Melideo, G., & Sebastiano, A. (2024). Percorsi per l'introduzione degli automi nel contesto didattico scolastico. In S. Capecchi, & B. Catania (Ed.), *ITADInfo 2024* (pp. 52-57). Genova: Genova University Press, e-ISBN: 978-88-3618-287-9.
- [15] IE. (2025). Informatics Europe. Retrieved from Informatics Europe: <https://www.informatics-europe.org/>
- [16] Isayama, D., Ishiyama, M., & Relator, R. (2016, 09 29). Computer Science Education for Primary and Lower Secondary School Students: Teaching the Concept of Automata. *ACM Transactions on Computing Education*, 17, 1-28. doi:10.1145/2940331
- [17] labinformaticaescola. (2025, 05 23). Il cristallo magico e il labirinto delle luci. Retrieved from labinformaticaescola.it: <https://labinformaticaescola.it/storia>
- [18] Lodi, M. (2020). Informatical Thinking. *Olympiads in Informatics*, 113-132.
- [19] MiM. (2025, 06 06). Nuove indicazioni per la scuola dell'infanzia e primo ciclo di istruzione 2025. Retrieved from Ministero dell'Istruzione e del Merito: <https://www.mim.gov.it/-/pubblicato-il-testo-delle-nuove-indicazioni-per-la-scuola-dell-infanzia-e-primo-ciclo-di-istruzione-materiali-per-il-dibattito-pubblico>
- [20] MiM. (2025). Piano Nazionale Scuola Digitale. Retrieved from MiM www.mim.gov.it/scuola-digitale
- [21] Nardelli, E. (2020). *La rivoluzione informatica*. Roma: Themis.
- [22] Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books.
- [23] Papert, S. (1998). Does Easy Do It? Children, Games and Learning. *Game Developer Magazine*, 5(6), pp. 88-89.
- [24] Piaget, J. (1951). *Play, dreams and imitation In childhood*. London, Great Britain: Routledge, pag. 288, ISBN 0415-21005-4.
- [25] Piaget, J., & Inhelder, B. (1966). *La psicologia del bambino*. Giunti.
- [26] Piaget, J., & Inhelder, B. (1976). *La rappresentazione dello spazio nel bambino*. Giunti Barbera.
- [27] PiF. (2025, 06 06). Programma il Futuro. Retrieved from Programma il Futuro: <https://programmailfuturo.it/>
- [28] Salehi, S., Toorawa, R., Wang, K., & Wieman, C. (2020). Can Majoring in Computer Science Improve General Problem Solving Skills? *SIGCSE '20*, 156-161
- [29] Vygotsky, L. (1978). *Mind in society: The development of higher psychological processes*. Harvard: Harvard University Press.
- [30] Vygotsky, L. (2008). *Pensiero e linguaggio*. Firenze: Giunti. (Prima edizione 1934).
- [31] Wikipedia. (n.d.). *Ottica Geometrica*. Retrieved 05 23, 2025, from Wikipedia: [Wikipedia: \[https://it.wikipedia.org/wiki/Ottica_geometrica\]\(https://it.wikipedia.org/wiki/Ottica_geometrica\)](https://it.wikipedia.org/wiki/Ottica_geometrica)

Coding, logica e movimento nella scuola dell'infanzia: un percorso laboratoriale tra carta, corpo e digitale

Mariarosaria Cianelli¹

¹ Associazione no-profit PERLATECNICA, Maddaloni (Caserta)

cianellirosaria@gmail.com

Abstract

In un contesto in cui l'informatica assume sempre più rilevanza sin dalla prima infanzia, il progetto ha coinvolto un gruppo di 12 bambini di 5 e 6 anni in un percorso articolato di 30 ore presso una scuola dell'infanzia casertana. Attraverso una metodologia laboratoriale, ludico-esperienziale e multimodale, i bambini sono stati guidati all'esplorazione del pensiero computazionale attraverso attività unplugged, strumenti digitali e robotica educativa. In particolare, i bambini si sono cimentati in attività motorie di orientamento spaziale, scomposizione di problemi e individuazione di sequenze di comandi per la loro risoluzione, esplorazione di meccanismi causa-effetto, familiarizzazione con la programmazione mediante i moduli di code.org, e l'utilizzo di Blue-Bot per la comprensione del concetto di esecutore. Le attività proposte sono strettamente interconnesse e mirano a stimolare gli studenti a risolvere problemi reali mediante la progettazione e codifica di semplici algoritmi. I risultati osservati hanno evidenziato un miglioramento nell'autonomia operativa, nella capacità di pianificazione condivisa e soprattutto nella comprensione dei concetti logico-spaziali. L'esperienza ha confermato l'efficacia del laboratorio come strumento pedagogico per introdurre il pensiero computazionale e il coding nella scuola dell'infanzia, valorizzando la creatività, la consapevolezza logico-spaziale e l'apprendimento cooperativo.

1. Introduzione

Il progetto *Coding, logica e movimento nella scuola dell'infanzia* si è svolto presso un istituto casertano ed è stato elaborato su una durata complessiva di 30 ore. Il progetto ha coinvolto 12 bambini di 5 e 6 anni, in un percorso laboratoriale di avvicinamento al coding e allo sviluppo del pensiero computazionale, attraverso attività unplugged e digitali. L'intervento nasce dalla necessità di promuovere l'educazione all'informatica in modo precoce, sfruttando l'approccio ludico-esperienziale per stimolare logica, creatività e cooperazione. Il percorso è stato articolato secondo il principio del *learning by doing* [6, 2], alternando attività pratiche, giochi di ruolo, costruzioni creative, esercizi motori e interazioni con strumenti digitali e robot educativi. Le attività proposte hanno l'obiettivo di introdurre concetti base funzionali alla programmazione e al pensiero computazionale come sequenze, orientamento spaziale e meccanismi causa-effetto.

2. Descrizione del progetto

Il percorso ha previsto attività diversificate e multimodali, finalizzate allo sviluppo di specifiche competenze informatiche, distinte per aree di apprendimento secondo il quadro di riferimento proposto in [4]:

- **Dati e Informazioni:** i bambini hanno imparato a rappresentare dati utilizzando simboli convenzionali per esprimere concetti a loro familiari, come ad esempio l'uso di frecce per indicare direzioni di movimento.
- **Algoritmi:** attraverso attività pratiche, hanno sperimentato la scomposizione di problemi complessi in sottoproblemi più semplici, affrontabili singolarmente con i simboli a disposizione.
- **Programmazione:** è stata sviluppata la consapevolezza del concetto di sequenza di istruzioni, guidando i bambini nella creazione di semplici programmi.

Giochi con comandi direzionali, percorsi motori e logici per comprendere movimenti nello spazio (destra, sinistra, avanti, indietro) usando sedie e cerchi. La prima attività ha proposto giochi con comandi direzionali, organizzati sotto forma di percorsi motori e logici. Tra le esperienze più coinvolgenti, spicca l'attività "Io Robot". Attraverso il gioco abbiamo introdotto il concetto di Coding, ovvero la creazione di una sequenza di istruzioni ordinate che il bambino-programmatore elabora per guidare il bambino-robot. I bambini possono utilizzare un insieme limitato di comandi (destra, sinistra, su, giù codificati mediante apposite frecce). I bambini, organizzati in coppie, assumono ruoli distinti: uno interpreta il robot esecutore del codice, mentre l'altro svolge la funzione di programmatore (come visibile in Figura 1). Per rendere più coinvolgente e divertente il gioco, il bambino-robot viene adornato con piccoli accessori, come fili di feltro, mollette o pinzette colorate, che lo rendono più simile a un automa, come visibile in Figura 2. Questo dettaglio creativo aumenta il livello di immedesimazione e la dimensione ludica dell'esperienza.



Figura 1: Fase istruttoria del bambino-robot



Figura 2: Personalizzazione del bambino-robot

Il programmatore ha il compito di guidare il robot attraverso un percorso fatto di cerchi o sedie, che riproduce fedelmente quello rappresentato nelle schede di lavoro, visibili in Figura 3. Ogni istruzione ha un collegamento diretto tra input (il comando dato) e output (il movimento eseguito dal robot). I bambini possono osservare in tempo reale gli effetti della sequenza di comandi scritta in precedenza, mentre viene letta dal “programmatore” ed eseguita dal “Robot”.

L'obiettivo dell'attività è far arrivare il robot da un punto A a un punto B, evitando eventuali ostacoli: in questo modo si introduce in modo concreto e intuitivo il concetto di sequenza, nell'area della programmazione. Gli altri bambini, disposti in cerchio attorno al tracciato, seguono lo svolgimento dell'attività, intervenendo in caso di errore, inserendo così il concetto di correzione dell'errore e ruolo del *debugger*. Ad esempio, il robot, eseguendo le istruzioni, non raggiunge l'obiettivo o urta un ostacolo. Il gruppo collabora per correggere il comando e trovare la sequenza corretta. Questa fase mira a sviluppare il pensiero computazionale, inteso come la capacità di scomporre un problema in sottoproblemi, riconoscere schemi ricorrenti, creare sequenze logiche (algoritmi), testare soluzioni e correggere gli errori [8].

Inoltre, per rafforzare il coinvolgimento, sono state usate delle schede accompagnate da una breve storia per trasformare l'esercizio in un'avventura. Questo espediente narrativo [3,6] si è rivelato molto efficace nel mantenere alta l'attenzione e verificare il grado di apprendimento dei partecipanti nella creazione di sequenze di istruzioni e della comprensione della relazione causa-effetto, nella previsione del comportamento del robot in base al comando ricevuto.

A conclusione della parte logica, i bambini sono stati coinvolti in un momento creativo in cui hanno potuto colorare i disegni presenti sulla scheda, come in Figura 4. Alcuni hanno preferito concentrarsi sugli sfondi, altri si sono dedicati agli elementi del percorso. Questo momento rilassante, spesso richiesto spontaneamente, si è rivelato un prezioso spazio espressivo. Infine, come un piccolo rito personale, ogni bambino ha disegnato sul retro della scheda un robottino di fantasia, frutto della propria immaginazione. Ogni creazione, unica nella sua realizzazione, ha permesso ai bambini di personalizzare e rendere ancora più proprio il lavoro svolto.



Figura 3: Schede didattiche



Figura 4: Momento creativo

Introduzione al pensiero computazionale con strumenti digitali e esercizi *drag-and-drop*.

Un momento chiave del percorso è stato l'utilizzo di Code.org, una piattaforma progettata per introdurre i bambini ai concetti fondamentali della programmazione tramite un linguaggio visuale e attività di *drag-and-drop*. Questa fase si è inserita come naturale prosecuzione dell'attività precedente, continuando a stimolare la capacità di scomporre i problemi e costruire sequenze logiche di comandi per risolverli. L'elemento innovativo è rappresentato dall'introduzione di simboli (istruzioni) che si identificano nella "rappresentazione del dato", introducendo la capacità di esprimere istruzioni in modo comprensibile per un esecutore non umano una competenza fondamentale in vista dell'interazione con Blue-Bot, prevista nella fase successiva. I bambini hanno imparato a costruire semplici sequenze di istruzioni trascinando blocchi di comando sullo schermo, come "Vai avanti", "Gira a destra", ed a verificare in tempo reale l'effetto delle loro scelte. In caso di risultati errati, i bambini hanno potuto sperimentare la correzione degli errori. L'attività è stata introdotta posizionando i bambini in cerchio, per creare un clima di collaborazione e scoperta condivisa (Figura 5). Inizialmente, gli esercizi sono stati svolti insieme, guidati dall'insegnante, stimolando l'aiuto reciproco e il confronto tra pari. Questo approccio ha favorito un apprendimento guidato e graduale, secondo il principio dello *scaffolding* [7]: l'adulto supporta il bambino nel momento del bisogno, lasciandogli progressivamente più autonomia per potersi mettersi alla prova.

Questa modalità si è rivelata estremamente efficace perché l'attività è risultata stimolante e divertente, trasformando l'apprendimento in un'esperienza coinvolgente. Oltre allo sviluppo delle prime competenze di programmazione, i bambini hanno mostrato maggiore fiducia in sé stessi, superando la frase "non lo so fare" e dando vita ad una crescente motivazione intrinseca e desiderio di mettersi in gioco con entusiasmo.

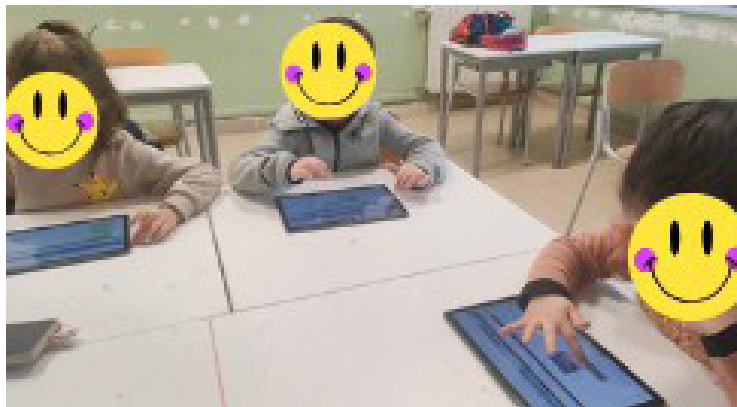


Figura 5: Bambini durante l'attività Code.org

Esperienze con robotica educativa grazie a Blue-Bot¹, stimolando la collaborazione e la pianificazione condivisa.

La fase finale del progetto ha introdotto i bambini alla robotica educativa attraverso l'utilizzo di Blue-Bot, un piccolo robot programmabile pensato per l'infanzia, che ha rappresentato un potente strumento per insegnare loro che per raggiungere un obiettivo (es. far arrivare Blue-Bot alla casella desiderata), serve una serie ordinata di comandi. Questa attività condivide con le fasi precedenti l'obiettivo di apprendere che l'algoritmo è “una sequenza ordinata e finita di passi elementari per risolvere un compito” con la differenza che l'esecutore è rappresentato anche da strumenti automatici, come Blue-Bot

Le attività sono iniziate con la programmazione di percorsi su tappeti fisici e, successivamente, anche virtuali. I bambini hanno avuto l'opportunità di interagire con l'applicazione di Blue-Bot, come in Figura 6, sfruttando anche la funzione vocale per associare comandi e simboli, potenziando così la memorizzazione spaziale. I percorsi iniziali erano semplici e guidati, ma con il progredire delle lezioni sono stati proposti tracciati via via più complessi, pensati per sviluppare la capacità di anticipazione e di costruzione di sequenze logiche.

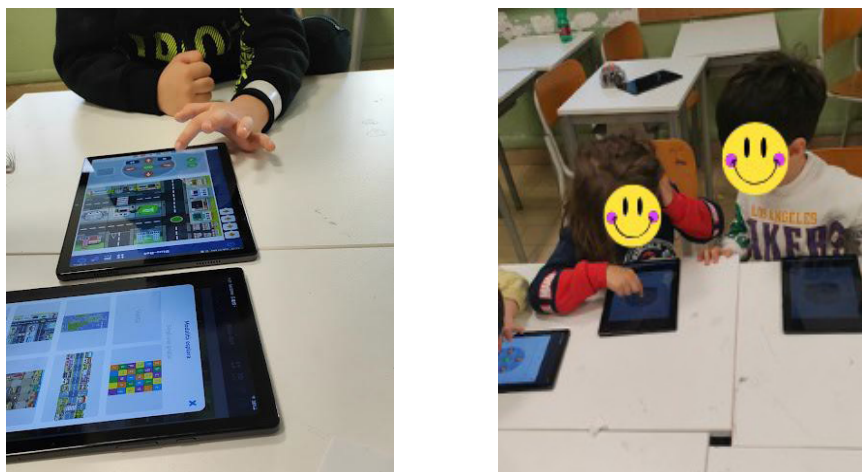


Figura 6: Configurazione del percorso e programmazione di Blue-Bot

Come ulteriore attività volta a verificare quanto appreso dai bambini, sono stati realizzati due percorsi identici: uno sul banco e uno riprodotto sul pavimento. Il percorso sul banco è destinato al Robot Blue-Bot, quello sul pavimento al “bambino-robot” (Figura 7), sfruttando la tecnica a specchio [5]. A turno, un bambino svolge il ruolo di programmatore, programmando il comportamento sia del bambino-robot che di Blue-Bot creando una simbiosi tra il robot umano e quello digitale. Entrambi i percorsi erano delimitati dagli stessi elementi cromatici, associando il colore verde per la destra ed il giallo a sinistra. L'attività è stata ripetuta per tutti i bambini assegnando di volta in volta un nuovo obiettivo ed facendo ruotare i ruoli. Terminata l'esecuzione della sequenza codificata dal programmatore, tutti i bambini assumevano il ruolo di *debugger*, correggendo la sequenza di comandi per migliorare l'accuratezza del percorso.

¹ <https://www.robotstore.it/Blue-Bot>



Figura 7: Bambino programma il Blue-Bot e la bambina-robot

3. Osservazioni durante l'esperienza

Nel corso delle 30 ore di attività, è emersa l'importanza di alternare proposte differenti per mantenere viva l'attenzione e la motivazione dei bambini. La ripetitività di un'unica modalità, in particolare nei momenti meno dinamici, portava talvolta a un calo di interesse, specialmente nei bambini più vivaci. Per questo motivo, ogni incontro è stato strutturato in tre fasi principali: un momento iniziale con una scheda logica accompagnata da una breve narrazione, una pausa con attività fisica o di movimento ispirata alla programmazione (come balli o percorsi corporei), e infine una fase conclusiva digitale, solitamente svolta con i tablet e le attività proposte da Code.org. L'articolazione modulare delle attività si è rivelata efficace nel favorire un equilibrio tra apprendimento e coinvolgimento. La varietà delle proposte ha permesso ai bambini di vivere ogni fase con entusiasmo, affrontando le sfide con crescente autonomia e motivazione.

L'ambiente laboratoriale ha favorito la sperimentazione, la collaborazione e lo sviluppo di abilità trasversali: seguire istruzioni, ragionare in sequenza, cooperare in coppia o in gruppo. I bambini si sono mostrati curiosi, disponibili a mettersi in gioco e aperti a ricevere feedback, interiorizzando gradualmente le logiche sottese alle attività di coding proposte.

La composizione del gruppo, 12 bambini, di cui 10 maschi e 2 femmine, ha messo in luce alcune differenze comportamentali significative. Le bambine si sono rivelate inizialmente più metodiche, concentrate e attente ai dettagli, ma sensibili all'atteggiamento dei compagni: tendevano infatti a perdere interesse se notavano disattenzione o confusione nel gruppo. Mostravano però anche un forte senso di responsabilità e autonomia, cercando di recuperare eventuali ritardi senza richiedere interventi esterni, seppur necessitando a volte di maggiore supporto emotivo nei momenti di difficoltà. I maschi, al contrario, hanno mostrato un approccio più impulsivo ed esecutivo: portavano a termine le attività con entusiasmo, ma talvolta in modo affrettato e meno preciso, soprattutto nella costruzione dei percorsi e nella coloritura. Tuttavia, la

loro partecipazione è stata intensa, con un evidente spirito competitivo e desiderio di avanzare rapidamente. Queste osservazioni ci hanno permesso di personalizzare l'approccio didattico, bilanciando attività riflessive e dinamiche in funzione delle diverse attitudini, con l'obiettivo di rispettare i differenti stili di apprendimento e mantenere elevata la partecipazione di tutti.

L'attività ha generato un forte entusiasmo, favorendo la partecipazione collettiva e lo spirito di gruppo. Ad esempio, durante l'attività "Io Robot", i bambini osservatori hanno partecipato attivamente supportando proattivamente l'attività del bambino-programmatore. L'interesse suscitato è stato tale che quasi tutti i bambini hanno espresso il desiderio di sperimentare entrambi i ruoli, manifestando curiosità e voglia di mettersi alla prova. Il percorso educativo ha previsto anche l'impiego di schede progressive per il tracciamento dei percorsi logici: ogni incontro si apriva con una nuova scheda, calibrata secondo il livello di difficoltà raggiunto dai bambini, e pensata per portarli gradualmente verso sfide sempre più complesse.

L'intero percorso ha confermato l'efficacia di una metodologia attiva e laboratoriale anche nella fascia della scuola dell'infanzia. L'integrazione tra gioco, tecnologia e apprendimento ha permesso ai bambini di sviluppare competenze logiche, motorie e relazionali in modo naturale e stimolante. Questa esperienza dimostra come sia possibile introdurre il coding fin dalla scuola dell'infanzia, utilizzando strumenti semplici ma significativi, in un ambiente dove l'apprendimento avviene attraverso il gioco, l'esplorazione e la relazione con l'altro.

La varietà delle attività, la cura nella progettazione dei materiali e l'attenzione alle dinamiche del gruppo hanno creato un contesto favorevole all'apprendimento profondo e al coinvolgimento emotivo. I risultati osservati sono stati positivi: maggiore autonomia nell'interazione con le tecnologie, comprensione più chiara dei concetti spaziali e computazionali di base, miglioramento nella capacità di *problem solving* e nella collaborazione.

Bibliografia

- [1] Catani, Gabriele. "Scrivere una fiaba col coding: una sperimentazione didattica con Scratch". Italiano a Scuola, 2(1), 171–200 (2020) <https://doi.org/10.6092/issn.2704-8128/10895>
- [2] Deluigi, Rosita and Romanazzi, Grazia. "L'educazione esperienziale come "aiuto alla vita" dai servizi educativi per l'infanzia all'università. Dialoghi aperti con il metodo di Maria Montessori tra l'Università di Macerata e Chiaravalle" In Education Sciences & Society (2023) <https://doi.org/10.3280/ess2-2023oa16667>
- [3] Ferrero, Daniele. "La struttura della fiaba secondo la morfologia di Propp". https://www.labirintoermetico.com/03Fiabe/struttura_fiaba_secondo_propp.htm
- [4] Forlizzi, Luca, et al. "A core informatics curriculum for Italian compulsory education." International Conference on Informatics in Schools: Situation, Evolution, and Perspectives. Cham: Springer International Publishing, (2018) <https://www.mat.uniroma2.it/~nardelli/publications/ISSEP-18-curriculum.pdf>
- [5] Maj, Erika. "Specchi, robot, codici e simmetria" <https://blog.matematica.deascuola.it/articoli/concetti-di-simmetria-assiale-e-centrale> (2024). Accessed: 2025-06-10
- [6] Pezzano, Teodora. "Dewey e i modelli della scuola d'infanzia". In Pedagogia più Didattica, Vol. 5(1). (2019) <https://rivistedigitali.erickson.it/pedagogia-piu-didattica/archivio/vol-5-n-1/dewey-e-i-modelli-della-scuola-d-infanzia>

- [7] Puntambekar, Sadhana. “Distributed Scaffolding: Scaffolding Students in Classroom Environments”. *Educational Psychology Review* 34, 451–472 (2022). <https://doi.org/10.1007/s10648-021-09636-3>
- [8] Wing, Jeannette M. “Computational thinking.” *Communications of the ACM* 49.3 (2006): 33-35 <https://dl.acm.org/doi/pdf/10.1145/1118178.1118215>

Attività didattiche di programmazione a partire dai giochi dell'informatica Bebras

Martina Palazzolo¹

¹ Istituto Comprensivo Ilaria Alpi di Milano

martina.palazzolo@icilariaalpi.edu.it

Abstract

I giochi dell'Informatica Bebras impegnano gli studenti su quesiti che presentano reali situazioni affrontate dalla scienza informatica. Presentiamo qui due percorsi di didattica dell'informatica che sono stati sviluppati a partire dal gioco “Il tessuto” della gara del 2015 e dal gioco “Cuore geometrico” del 2022. Le proposte didattiche fanno lavorare gli studenti su diverse attività coinvolte nella programmazione:

- eseguire un programma, scrivere un programma, collaudare e fare debugging per il primo gioco “Il tessuto”;
- usare sequenze di operazioni anche incomplete e poco precise (astrazioni dei comandi) per progettare un disegno a partire da semplici figure geometriche come proposto nel gioco “Cuore geometrico” e quindi usare un semplice linguaggio di programmazione per realizzare lo stesso disegno con il software LibreLogo.

Le attività dimostrano largo coinvolgimento degli alunni che hanno messo in campo abilità trasversali quali collaborazione, risoluzione di problemi tramite confronto con i pari, approccio costruttivo e critico verso gli errori, creatività e competenza. In entrambe le attività gli studenti hanno dimostrato alla fine di saper lavorare “da programmatori” realizzando programmi eseguibili da un compagno (nel primo percorso) o da una macchina (nel secondo percorso).

1. Introduzione

I giochi dell'informatica Bebras [1] nascono nel 2004 in Lituania per divulgare i principi dell'informatica intesa come scienza. Ad oggi partecipano alla comunità Bebras paesi di tutti i continenti [2] [3]. Delegato per l'Italia è il gruppo ALaDDIn (Laboratorio di Didattica e Divulgazione dell'Informatica) del Dipartimento di Informatica dell'Università Statale di Milano. La gara si svolge nel mese di novembre attraverso la piattaforma Bebras [1] che mette a disposizione anche le soluzioni e le spiegazioni dei giochi dopo la gara e per gli anni successivi.

I giochi offrono spunti sulla scienza informatica [4] che possono essere approfonditi con attività didattiche mirate. In questo resoconto presentiamo i percorsi realizzati a partire da due giochi Bebras: il “Tessuto” e “Cuore geometrico”. I percorsi sono stati realizzati in anni differenti, ciascuno su una classe prima della scuola secondaria di primo grado. I percorsi sono nati con l'obiettivo di far comprendere ai ragazzi alcuni concetti della programmazione, in accordo con la Proposta di Indicazioni Nazionali per l'insegnamento dell'Informatica nella

scuola [5]. Nel documento tra i traguardi per lo sviluppo delle competenze al termine della scuola primaria troviamo:

- T-P-1 l'allievo comprende che un algoritmo descrive una procedura che si presta ad essere automatizzata in modo preciso e non ambiguo;
- T-P-2 comprende come un algoritmo può essere espresso mediante un programma scritto usando un linguaggio di programmazione;
- T-P-3 legge e scrive programmi strutturalmente semplici;
- T-P-4 spiega usando il ragionamento logico perché un programma strutturalmente semplice raggiunge i suoi obiettivi;
- T-P-5 inizia a riconoscere la differenza tra l'informazione e i dati;
- T-P-6 esplora la possibilità di rappresentare dati di varia natura (numeri, immagini, suoni,...) mediante formati diversi, anche arbitrariamente scelti.

I percorsi descritti, a nostro avviso, presentano esempi di attività didattiche che possono concorrere al raggiungimento di tali traguardi. Essi affrontano una parte dell'informatica, quella dedicata alla programmazione, traendo ispirazione dai giochi dell'informatica Bebras. Le attività proposte sono laboratoriali in modo che gli studenti possano essere attivamente partecipi del proprio apprendimento. Abbiamo utilizzato il metodo di lavoro descritto in *Allenatori Bebras* [6] dove gli studenti imparano concetti dell'informatica rielaborando e producendo materiali che poi utilizzano con i pari oppure con i compagni più giovani tramite compiti autentici.

Il primo percorso che utilizza il gioco “Il tessuto” vede gli studenti impegnati in diverse attività coinvolte nella programmazione: viene richiesto loro di eseguire una serie di comandi di cui devono comprendere sintassi e semantica, poi sono chiamati a scrivere un programma utilizzando il linguaggio imparato e infine collaudare e fare debugging dei programmi scritti dai compagni. In questo percorso gli studenti arrivano a scrivere programmi semplici (vedi T-P-3) realmente eseguibili da operatori umani. Vista la natura del gioco imparano anche a comprendere che immagini (nel loro caso i disegni) possono essere rappresentati in un formato arbitrariamente scelto (vedi T-P-6).

Il secondo percorso, invece, inizia dal gioco “Cuori geometrici” che invita gli studenti a ragionare attraverso le cosiddette astrazioni dei comandi, ovvero descrizioni di ciò che un comando deve poter fare se eseguito, senza il rigore formale del linguaggio di programmazione, in modo da capire come potrebbe essere impostato un programma, prima di scriverlo (vedi T-P-4). Si chiede poi agli studenti di scrivere un programma per realizzare il disegno progettato dopo aver imparato i primi comandi del linguaggio di programmazione LibreLogo (vedi T-P-2).

2. Percorso ispirato al gioco Bebras “Il tessuto”

2.1 Caratteristiche del gioco

Il quesito chiede agli studenti di eseguire la sequenza di comandi che viene data a un telaio per produrre tessuti con motivi geometrici. Un tipo di comando codifica il numero di righe e di colonne di una griglia e un altro tipo di comando specifica in quale casella della griglia inserire una figura geometrica data e in quelle consecutive. Gli studenti hanno trovato difficile capire il gioco e solo un gruppo è riuscito a risolverlo correttamente. La parte che ha messo più in difficoltà i ragazzi è stata la codifica della griglia e l'individuazione del punto nella griglia in cui inserire le figure geometriche. Una volta chiarito il procedimento abbiamo ideato un percorso che permettesse loro di prendere dimestichezza con questo tipo di codifica.

Tessuto (max 6 punti)



Un telaio produce tessuti con motivi geometrici, tramite alcuni comandi.

Il comando

2	≡	5
---	---	---

disegna una griglia di caselle con due righe e cinque colonne.

Il comando

2	▲	1	3
---	---	---	---

Inserisce in una griglia due triangoli nelle caselle a partire da quella posizionata nella prima riga e nella terza colonna, procedendo da sinistra verso destra.

Dopo aver eseguito i due comandi indicati otteniamo il seguente motivo

		▲	▲	

Eseguite ora la sequenza di comandi:

3	≡	4	
2	●	3	1
3	●	1	1
1	■	2	4

Che motivo si ottiene?
Cliccate nelle caselle e selezionate i simboli giusti.

Figura 1: Il gioco originale

2.2 Fasi del percorso

Il percorso si articola in 3 attività consecutive, ognuna con obiettivi specifici relativi all'attività di programmazione.

Prima Attività – Fase di comprensione della sintassi e semantica del linguaggio di programmazione. Abbiamo chiesto agli studenti di risolvere il quesito “Il tessuto” fornito in formato cartaceo. Molti studenti hanno avuto bisogno di chiarimenti da parte dell’insegnante.

Seconda Attività – Fase di verifica della comprensione attraverso la produzione di un programma. Abbiamo chiesto agli studenti di produrre un nuovo disegno inventato da loro usando la stessa codifica del gioco. Alcuni studenti sono partiti ideando la griglia e il disegno e quindi codificando la sequenza di comandi necessaria per ottenere il disegno scelto. Altri invece sono partiti dalla stesura della sequenza di comandi da cui poi hanno ricavato il disegno. Questo tipo di attività ha permesso loro di fare esperienza diretta su cosa significa scrivere un programma (una sequenza di comandi) ed eseguire un programma in modo da ottenere il risultato voluto (il disegno finale).

Terza attività – Fase di certificazione dei programmi. Abbiamo chiesto agli studenti di eseguire i programmi di altri compagni senza sapere in anticipo quale sarebbe stato il disegno codificato. In questa attività i ragazzi hanno avuto modo di fare esperienza con il processo di esecuzione di un programma ed eventualmente identificare e correggere gli errori. Alcuni studenti hanno eseguito correttamente il programma ma hanno ottenuto un disegno differente da quello che l'autore del programma voleva realizzare. In questo caso è stato chiesto agli autori e

ai certificatori di confrontarsi, ragionare insieme e trovare l'errore (che poteva essere dell'autore del programma o dei certificatori). In pochi casi è stato necessario l'intervento dell'insegnante per risolvere il caso.

2.3 I lavori realizzati dagli studenti e conclusioni

I lavori che vi proponiamo indicano i passaggi chiave che dimostrano come le diverse attività permettano agli studenti di imparare con la pratica e il confronto a:

- scrivere un programma a partire da una codifica che imparano attraverso il gioco;
- eseguire il programma seguendo la codifica data;
- valutare la correttezza del programma confrontando il risultato atteso con il proprio.

Nella figura 2 si vede un lavoro piuttosto complesso con una griglia 10x10 in cui vengono posizionati, per ottenere il muso di un gatto stilizzato, cerchi marroni o blu, triangoli rosa e quadrati marrone chiaro o marrone scuro. Il programma, vedi la tabella a sinistra, è corretto fino alle ultime quattro righe. Alla quarta riga dal fondo il colore del cerchio è marrone mentre l'autore, nell'esecuzione al centro, l'ha disegnato blu essendo l'occhio del gatto. Il certificatore, disegno a destra, ha eseguito correttamente il comando. Il certificatore ha identificato e segnalato un altro errore, quello che si trova nelle ultime due righe in cui il cerchio blu e il triangolo dovrebbero sovrapporsi nell'esecuzione del codice.

Nella figura 3 vediamo il codice che prevede una griglia 5x5, in cui diventano più elaborati i segni grafici da riportare nella griglia stessa, tanto che il certificatore ha avuto difficoltà a riprodurli, come si vede nella sua correzione (il codice indicato dalla freccia rosa).

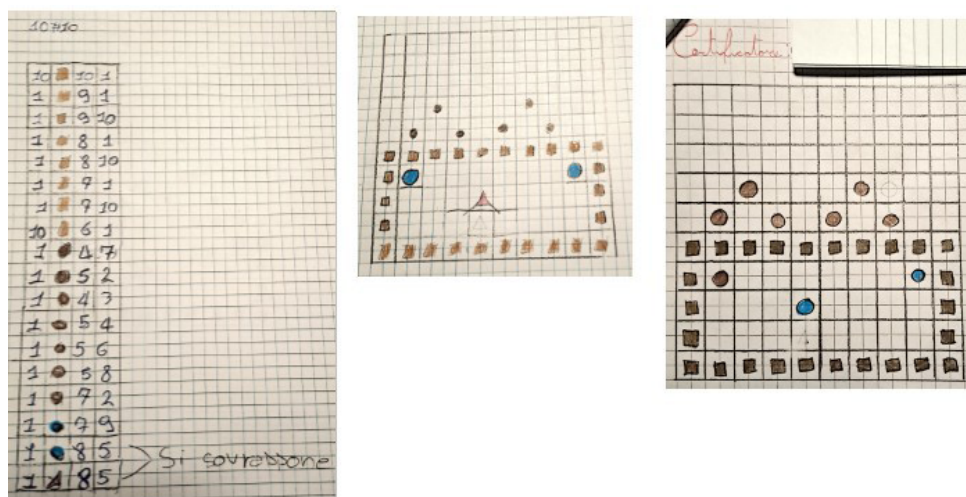


Figura 2: Lavoro con codice (a sinistra), esecuzione da parte dell'autore del codice (in centro) e risultato dell'esecuzione di un certificatore (a destra)

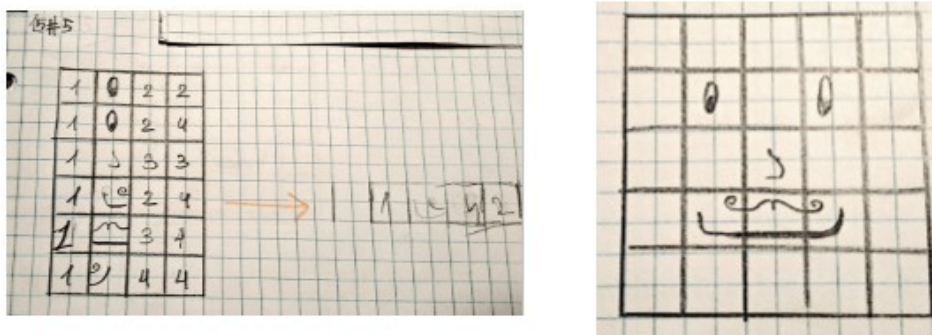


Figura 3: Lavoro con codice (a sinistra) e correzione da parte del certificatore disegno (a destra)

3. Percorso ispirato al gioco Bebras “Cuore geometrico”

Cuore geometrico



Tina si diverte a creare figure. Ha iniziato con un cerchio e un quadrato disposti così:




Tina ha creato questo cuore usando solo le seguenti azioni sulle figure:

- RUOTARE: ruota la figura (come vuoi)
- SPOSTARE: sposta la figura (dove vuoi)
- DUPLICARE: crea una copia della figura (nella stessa posizione)



Quale tra queste sequenze di azioni ha usato Tina per creare il cuore?

Figura 4: Il gioco originale

3.1 Caratteristiche del gioco

La protagonista del gioco, Tina, crea un cuore a partire da un cerchio e un quadrato utilizzando solo 3 azioni: RUOTARE, SPOSTARE e DUPLICARE. Si chiede agli studenti di scegliere quale tra differenti sequenze di azioni date ha usato Tina per creare il cuore. Gli studenti hanno trovato interessante la possibilità di creare un nuovo disegno, e il cuore in particolare, a partire da figure semplici utilizzando alla fine solo 3 azioni.

3.2 Fasi del percorso

Prima Attività

Fase di comprensione dell'uso dei comandi RUOTARE, SPOSTARE e DUPLICARE per ottenere il cuore come prima figura e poi per ottenere altri disegni a scelta. Agli studenti viene consegnato il quesito in formato cartaceo, gli viene chiesto di realizzare con cartoncino alcuni quadrati e alcuni cerchi e ruotarli e spostarli sul banco in modo da ottenere la forma del cuore e poi altre forme a scelta. Gli studenti, ottenuta la giusta sequenza di azioni per ottenere la figura desiderata devono riportarla nel proprio quaderno come promemoria che potranno riprendere nelle fasi successive del percorso. Gli studenti hanno apprezzato molto questa fase e soprattutto la possibilità comporre e scomporre attraverso semplici azioni nuove figure.

Seconda attività

Fase di apprendimento dei comandi e della sintassi di un linguaggio di programmazione per ottenere un cerchio, un quadrato e le tre azioni ruotare, spostare, duplicare. Abbiamo spiegato agli studenti i comandi e la sintassi del linguaggio LibreLogo nella suite Libreoffice [7], in particolare i comandi CIRCLE, SQUARE, LEFT, RIGHT, FORWARD e BACK. Gli studenti hanno lavorato nel laboratorio di informatica con LibreLogo scrivendo il programma per riprodurre il disegno desiderato al computer. Ottenere sullo schermo il disegno voluto grazie alla sequenza corretta dei comandi imparati (vedi T-P-3) ha dato loro grande soddisfazione. Gli studenti hanno lavorato con interesse e creatività.

Terza attività

Fase di applicazione di quanto imparato per insegnarlo/condividerlo con i compagni più giovani. Abbiamo chiesto agli studenti di preparare delle schede o più in generale alcuni materiali a scelta che avrebbero utilizzato durante il raccordo con le classi quinte della primaria e durante l'open day per far vedere ai bambini come programmare con LibreLogo. I materiali prodotti in questa fase sono molto creativi, colorati e curati. La parte formale di scrittura del programma proposto è sicuramente da migliorare e abbiamo pensato per l'anno prossimo di fare usare loro una stampa del codice piuttosto che la sua scrittura a mano. I programmi scritti a mano comunque sono corretti e danno il disegno desiderato.

3.3 I lavori realizzati dagli studenti e conclusioni

Nella figura 5 si vedono i lavori di due gruppi. A sinistra gli studenti hanno lavorato alla realizzazione di una busta da lettera che si apre e al cui interno hanno inserito un foglietto con il codice per realizzare il disegno della busta stessa. Nella realizzazione del codice hanno mantenuto proporzioni e colori della busta originale. Oltre al comando CIRCLE hanno utilizzato anche il comando RECTANGLE e disegnato i bordi con i comandi FORWARD e RIGHT.

A destra invece gli studenti hanno realizzato, utilizzando solo cerchi con diametro differente, il musetto di un orsetto con gli occhi neri. Hanno quindi utilizzato solo il comando `CIRCLE` e la variabile lunghezza del diametro spostando la posizione dei cerchi con i comandi `FORWARD`, `LEFT` e `RIGHT`.

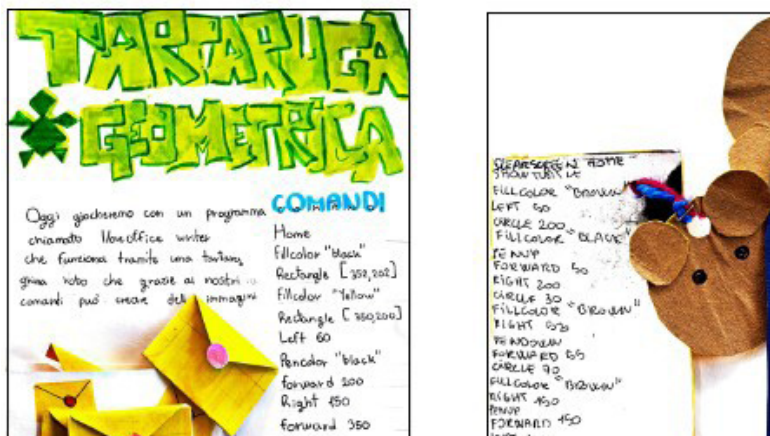


Figura 5: Lavori preparati da due gruppi: a sinistra una busta delle lettere e a destra il muso stilizzato di un orsetto

Nella figura 6 si vede a destra la copertina della cartelletta che hanno usato durante l'open day in cui presentano il manufatto e un disegno del quadrifoglio. Al centro si vede la copertina di un materiale che hanno ritenuto importante fornire ai bambini, ovvero una griglia in modo da valutare soprattutto le rotazioni necessarie. A destra si vede una busta con i quadrifogli realizzati con il cartoncino verde: hanno usato un quadrato (che non si vede) dove vendono incollati i 4 cerchi verdi. Nel codice il quadrato è presente e orienta il movimento dei 4 cerchi per ottenere il quadrifoglio.



Figura 6: Il Lavoro del quadrifoglio

4. Conclusioni

Le attività previste in questi due percorsi hanno dimostrato largo coinvolgimento degli alunni che hanno messo in campo abilità trasversali quali collaborazione, risoluzione di problemi tramite confronto con i pari, approccio costruttivo e critico verso gli errori, creatività e competenza. In entrambe le attività gli studenti hanno dimostrato alla fine di saper lavorare “da programmatori” realizzando programmi eseguibili da un compagno (nel primo percorso) o da una macchina (nel secondo percorso).

Bibliografia

- [1] bebras.it <https://www.bebas.org/countries.html>
- [2] V. Dagienė and G. Stupuriene. Informatics education based on solving attractive tasks through a contest. IFIP-KEYCIT 2014, p. 51-62, 2014
- [3] A. Calcagni, V. Lonati, D. Malchiodi, M. Monga, A. Morpurgo. Promoting Computational Thinking Skill: Would you use this Bebras Task? ISSEP 2017 pagg. 102 – 113 LNCS 11913, Springer 2019
- [4] <https://www.consorzio-cini.it/images/Proposta-Indicazioni-Nazionali-Informatica-Scuola-numerata.pdf>
- [5] C. Bellettini, V. Lonati, M. Monga, A. Morpurgo, M. Palazzolo. Situated Learning with Bebras Tasklets. ISSEP 2019 pagg. 225 – 239 LNCS 11913, Springer 2019
- [6] <https://it.libreoffice.org/>

Programmare la parità: la robotica educativa per l'inclusione di genere

Laura Cesaro¹, Emanuele Menegatti¹

¹ Università degli Studi di Padova

laura.cesaro@phd.unipd.it, emanuele.menegatti@unipd.it

Abstract

La persistente sottorappresentazione femminile nelle discipline STEM, in particolare nell'informatica, solleva interrogativi urgenti su come rendere questi ambiti più accessibili e inclusivi. Questo contributo esplora il potenziale della robotica educativa come leva per promuovere la partecipazione delle ragazze, superare stereotipi di genere e innovare la didattica dell'informatica. A partire dalla prospettiva costruzionista e dall'analisi di progetti europei e nazionali, il paper individua tre ambiti prioritari: metodologie didattiche attive e collaborative, integrazione curricolare sistematica e strategie specifiche per l'inclusione di genere. L'analisi è supportata da evidenze raccolte in una sperimentazione condotta nella scuola primaria. I risultati suggeriscono che la Robotica Educativa, se progettata consapevolmente, può contribuire a costruire ambienti di apprendimento più equi, motivanti e significativi, favorendo una più ampia partecipazione femminile alle discipline STEM.

1. Introduzione

La ridotta presenza femminile nelle discipline STEM, in particolare nell'informatica e nella robotica, rappresenta una sfida significativa per il sistema educativo. Nonostante l'aumento delle iniziative in tal senso, la disparità di genere persiste, non per mancanza di interesse o abilità, ma per l'influenza di fattori culturali e sociali che si acquiscono con l'ingresso a scuola [1]. Come osservavano Turkle e Papert già nel 1991 [2], ciò che spesso allontana le ragazze da questi ambiti non è la difficoltà dei contenuti in sé, ma piuttosto il conflitto tra il loro approccio alla conoscenza, spesso più relazionale e contestualizzato, e un modello culturale e didattico dominante che tende a privilegiare approcci rigidi, astratti o fortemente competitivi.

La robotica educativa (RE), con la sua natura laboratoriale e creativa, offre un contesto significativo per attività coinvolgenti, collaborative e motivanti. Se progettata secondo adeguati principi di equità e inclusione, la RE può contribuire allo sviluppo del pensiero computazionale nelle studentesse e a un loro approccio più consapevole e partecipato all'informatica. Questo contributo riflette su tre dimensioni chiave per delineare un approccio alla progettazione di esperienze didattiche attento alle differenze di genere, facendo riferimento a pubblicazioni istituzionali e a progetti europei centrati sull'inclusione delle ragazze come cornice teorica di riferimento [3-11]:

1. Le metodologie;
2. l'integrazione curricolare delle attività;
3. le strategie progettuali mirate a favorire l'inclusione di genere.

Le considerazioni teoriche sono supportate da evidenze osservate nel corso di un'esperienza didattica realizzata in un contesto curricolare nella scuola primaria, che verrà presentata nella parte conclusiva del lavoro.

2. Metodologie didattiche

L'efficacia della RE nel promuovere apprendimenti significativi e una partecipazione inclusiva è direttamente correlata alle metodologie didattiche impiegate. I principi dell'Universal Design for Learning [12] mostrano come pratiche educative flessibili che offrono molteplici mezzi di coinvolgimento, rappresentazione ed espressione, rendano l'apprendimento accessibile e motivante per tutti gli studenti. Per quanto riguarda la robotica, in particolare, diverse ricerche e progetti europei indicano l'importanza di approcci costruzionisti, cooperativi e orientati al problem-solving, che risuonano con le "4 P" di Mitchel Resnick: *Projects, Peers, Passion e Play*, elementi chiave per ambienti di apprendimento stimolanti e significativi [13]. Gli approcci metodologici selezionati incoraggiano diverse modalità di partecipazione, valorizzano il lavoro cooperativo e creano le condizioni per esprimersi e partecipare attivamente, contribuendo in questo modo alla riduzione di ridurre barriere cognitive, sociali e culturali.

1. **Apprendimento Costruzionista e "Learning by Making".** L'approccio costruzionista di Papert indica nel robot un "oggetto con cui pensare" che stimola la creatività e promuove il pensiero computazionale attraverso la manipolazione attiva [14]. La dimensione laboratoriale favorisce un apprendimento per prove ed errori, dove concretezza e riflessione si intrecciano, superando il modello rigido e formalizzato che spesso allontana le ragazze dalla tecnologia [1].
2. **Problem-Based Learning (PBL).** Il PBL si concretizza in sfide progettuali autentiche che richiedono progettazione, costruzione e programmazione collaborative [5,6]. Questa metodologia stimola il pensiero critico, favorisce la collaborazione per soluzioni condivise e sostiene la motivazione intrinseca attraverso obiettivi concreti e osservabili.
3. **Inquiry-Based Learning.** L'approccio inquiry-based trasforma la classe in un laboratorio di ricerca dove gli studenti formulano ipotesi, progettano esperimenti con i robot, raccolgono dati e discutono soluzioni. L'integrazione tra robotica e inquiry favorisce apprendimenti profondi, collegati sia alle competenze scientifiche che metacognitive [9].
4. **Approccio Cooperativo.** Le attività di robotica, se progettate efficacemente, promuovono interdipendenza positiva e riconoscimento reciproco. In un'ottica di inclusione di genere, si rivelano fondamentali strategie quali: sviluppare le abilità di team working, proporre attività collaborative anziché competitive, valorizzare la narrazione e la dimensione sociale del problema, dare rilievo alla progettazione estetica e comunicativa del robot, e sostenere l'autoefficacia con feedback mirati.

3. Robotica educativa e curriculum

Perché la RE possa esprimere appieno il suo potenziale trasformativo nell'apprendimento e promuovere una reale inclusione, è fondamentale che essa sia integrata strutturalmente nel curriculum scolastico, superando la frammentazione delle attività occasionali e la natura opzionale dei progetti extracurricolari [7]. Per realizzare questo obiettivo, è cruciale agire su diversi aspetti:

1. **Integrazione curricolare sistematica.** La robotica si presta naturalmente all'utilizzo trasversale, coinvolgendo materie come matematica, scienze, tecnologia, arte e lingua [7]. Questa dimensione interdisciplinare favorisce una molteplicità di approcci e contribuisce a superare le gerarchie implicite tra le discipline.
2. **Continuità e verticalizzazione del curriculum.** È essenziale costruire percorsi che accompagnino studenti e studentesse a partire dalla scuola dell'infanzia fino alla scuola secondaria, con attività via via più complesse. La robotica permette questa gradualità: dalle attività unplugged e di manipolazione nei primi anni, alle progettazioni avanzate nelle scuole superiori. Questa continuità aiuta a consolidare le competenze e a contrastare l'autoesclusione precoce delle ragazze legata a percezioni stereotipate della tecnologia [7].
3. **Formazione docenti.** Integrare la robotica nel curriculum richiede di investire nella formazione non solo tecnica, ma soprattutto metodologica e pedagogica. I progetti più efficaci includono percorsi blended, laboratori, documentazione di buone pratiche e supporto in classe [6]. Strumenti di autovalutazione, come quelli dell'European Institute for Gender Equity [8], possono inoltre aiutare gli insegnanti a riflettere sui propri approcci didattici, sulle dinamiche di genere e sulla qualità dell'inclusione.
4. **Ambienti di apprendimento flessibili.** Un nodo importante riguarda l'organizzazione didattica e logistica. Le pratiche più avanzate promuovono l'idea di una robotica diffusa e integrabile in vari contesti, grazie a kit mobili, materiali modulari e tecnologie accessibili. In tal modo, la robotica si configura come uno strumento trasversale per ripensare il curriculum, innovando metodologie e ambienti di apprendimento in chiave sistemica e sostenibile.

4. Strategie per la partecipazione e l'inclusione delle ragazze

La bassa partecipazione femminile nelle discipline STEM dipende da fattori culturali e didattici che richiedono strategie specifiche per mitigare il ruolo di stereotipi, l'assenza di modelli o l'uso di approcci didattici competitivi o tecnicisti. Per contrastare ciò, la RE, se proposta a tutti e progettata con attenzione al genere, può diventare uno strumento efficace di inclusione e riequilibrio [6,8,15]. A tal fine, è fondamentale implementare approcci mirati che intervengano su diverse leve:

1. **Contrasto agli stereotipi:** è importante evitare associazioni tra robotica e contenuti "maschili" (robot da combattimento, sfide di velocità), e proporre piuttosto problemi aperti e contesti significativi anche sociali, verificare la neutralità nei materiali e controllare il linguaggio usato [8].
2. **Inclusione attiva delle ragazze:** è fondamentale rafforzare l'autoefficacia grazie a feedback positivi, dare visibilità ai successi, riconoscere i contributi individuali e offrire modelli femminili nelle STEM. Non si tratta di "semplificare" le attività per le ragazze o di modellarle sui loro stereotipi, ma di moltiplicare linguaggi e modi di partecipare, rendendo visibili e legittimi approcci diversi. [2, 6, 8, 11, 15].

3. **Contesti non competitivi:** situazioni competitive tendono ad attrarre maggiormente i maschi, mentre le ragazze partecipano a gare e sfide principalmente quando hanno già forte interesse tecnologico, o si sentono abbastanza capaci.
4. **Design inclusivo dei robot:** per promuovere un interesse più equilibrato e paritario nelle discipline STEM è importante tener conto del design dei robot scelti, in modo che possano attrarre sia studenti che studentesse e che evitino di rafforzare gli stereotipi di genere [16].

5. Un’esperienza sul campo

In una sperimentazione condotta a Padova, 86 alunni di quattro classi quinte della scuola primaria hanno partecipato a un percorso di robotica educativa della durata complessiva di 10 ore, strutturato in contesti cooperativi e non competitivi. Le attività si sono svolte durante l’orario curricolare, con sessioni quotidiane di due ore distribuite nell’arco di una settimana.

L’approccio didattico adottato si è fondato su un sistema autoregolato costituito da mini-task progressivi. Gli studenti, organizzati in coppie omogenee per genere, hanno lavorato in autonomia su un obiettivo descritto e illustrato su una carta fisica. Una volta completato un compito, potevano richiedere la carta successiva, con attività di complessità crescente. La possibilità di procedere al proprio ritmo ha favorito un aumento dell’autonomia e dell’engagement, sostenuti anche alla contestualizzazione delle attività in situazioni legate alla vita quotidiana.

I dati raccolti mostrano che le studentesse hanno valutato l’esperienza in modo leggermente più positivo rispetto ai compagni maschi. Le medie delle valutazioni (su una scala da 1 a 5) sono riportate di seguito (Tabella 1).

Questi risultati suggeriscono che l’inclusione di attività robotiche all’interno del curricolo scolastico può favorire un accesso più equo alle discipline STEM, mitigando il fenomeno dell’autoesclusione femminile, spesso più frequente nelle proposte extracurricolari.

Tra le domande finali, una chiedeva agli studenti di confrontare il proprio interesse per la robotica prima e dopo il percorso. Il 68% delle studentesse ha dichiarato che l’attività è risultata più interessante di quanto si aspettasse. Questo dato, raccolto in un contesto scolastico obbligatorio, appare particolarmente rilevante: suggerisce che, in assenza di un’opportunità curricolare, molte ragazze non si sarebbero spontaneamente avvicinate alla robotica. La robotica educativa emerge così come un’esperienza in cui si intrecciano competenze tecniche, relazioni interpersonali e valori educativi, contribuendo a contrastare stereotipi di genere.

	Mi è piaciuto poco – tanto (da 1 a 5)	È stato difficile-facile (da 1 a 5)	È stato noioso- interessante (da 1 a 5)	È stato faticoso- divertente (da 1 a 5)
Femmine	4,79	3,44	4,50	4,29
Maschi	4,65	3,26	4,35	4,22

Tabella 1: Valutazione dell’esperienza

Una differenza significativa è emersa anche nella scelta del modello di robot da costruire (Tabella 2). Le coppie femminili hanno mostrato una marcata preferenza per Starter Bot (15 su

19), un robot dal design biomimetico, mentre diverse coppie maschili hanno optato per MTA-Bot (11 su 23), caratterizzato da un aspetto più tecnico e meccanico. Questi dati suggeriscono che il design dei robot educativi può influenzare il coinvolgimento: proporre robot percepiti come più “amichevoli” o simili a esseri viventi può aumentare l’attrattiva e l’accessibilità per un pubblico più ampio.

Scelta del robot	F+F	F+M	M+M	Totale
 Starter Bot	15	1	12	28
 MTA-Bot	4	0	11	15

Tabella 2: Scelta del robot

Infine, alla domanda “Ti piacerebbe continuare?”, il 97% delle studentesse e il 92% degli studenti ha risposto affermativamente. La leggera differenza a favore delle ragazze rafforza l’idea che un approccio didattico inclusivo, proposto in un contesto curricolare non competitivo, possa promuovere il coinvolgimento femminile nelle attività di robotica educativa. L’esperienza mostra come la robotica, se progettata con attenzione all’equità, possa diventare uno strumento efficace per avvicinare le studentesse alle STEM, riducendo barriere culturali e superando stereotipi anche impliciti ancora presenti.

6. Conclusioni

La RE offre un approccio pratico e creativo all’informatica, permettendo agli studenti di sviluppare pensiero computazionale e logica in un contesto concreto e meno astratto. Le ricerche evidenziano che, se integrata con metodologie intenzionali e attenzione al design, la robotica può efficacemente contrastare gli stereotipi di genere e aumentare la partecipazione femminile nelle discipline STEM. È cruciale considerare anche il curriculum implicito della scuola: le modalità di proposta della robotica (dove, quando, come, con quali risorse) possono rafforzare o decostruire gli stereotipi. L’assenza di riferimenti al pensiero computazionale e all’educazione digitale nei documenti ufficiali rischia inoltre di emarginare simbolicamente le tecnologie. Integrare la robotica, perciò, significa ridefinire le priorità formative e culturali della scuola, promuovendo un’innovazione che sia inclusiva, riflessiva e consapevole.

Questa analisi non è esaustiva: si propone come una ricognizione iniziale e invita a future indagini sul campo per valutare l’impatto delle strategie adottate su partecipazione, motivazione e apprendimenti.

L'integrazione di metodologie costruzioniste, ambienti cooperativi, design inclusivo e approcci curriculari sistematici crea legittimità per chi non si sente "portato" per l'informatica. Lavorare sull'inclusione di genere significa migliorare la qualità dell'insegnamento per tutti, rendendolo più ricco, equo e aderente alla diversità di ciascuno.

La sfida non è solo tecnologica ma anche pedagogica: progettare intenzionalmente esperienze che valorizzino diversità di approcci, contrastino stereotipi e creino opportunità di partecipazione autentica per studentesse e studenti. Solo così la RE potrà contribuire a "programmare la parità" nelle discipline STEM.

Bibliografia

- [1] Martinot, P., Colnet, B., Breda, T. et al. , «Rapid emergence of a maths gender gap in first grade» *Nature*, 2025.
- [2] Turkle, S., & Papert, S., «Epistemological pluralism and the revaluation of the concrete.» in *Constructionism*, Ablex Publishing, 1991, pp. 161-191.
- [3] «No Gender Gap» [Online]. Available: <https://www.scuoladirobotica.it/en/progettoeu/no-gender-gap/>.
- [4] «Girls Discover Robots» [Online]. Available: <https://roberta-home.de/en>
- «RoboGirls» [Online]. Available: <https://robogirls.eu/en>.
- [5] Bagattini, D., Miotti, B., *Lavorare sul genere a scuola con coding e robotica educativa*, Roma: Carocci Edizioni, 2022.
- [6] Nulli, G., Miotti, B., Di Stasio, M., *Robotica educativa e coding: strumenti per la trasformazione del curricolo*, Roma: Carocci Edizioni, 2022.
- [7] EIGE, «Toolkits and Guides» [Online]. Available: <https://eige.europa.eu/publications-resources/toolkits-guides>.
- [8] Gaudiello, I., & Zibetti, E. , *Learning robotics, with robotics, by robotics: Educational robotics.*, New York: John Wiley & Sons, 2016.
- [9] Malvezzi, M.; Alimisis, D.; Moro M., «Education in & with Robotics to Foster 21st-Century Skills» in *Edurobotics 2021*, 2021.
- [10] Directorate-General for Education, Youth, Sport and Culture (European Commission), «Addressing the gender gap in STEM education across educational levels - Analytical report» 2024. [Online]. Available: <https://op.europa.eu/en/publication-detail/-/publication/5c74b478-3ffe-11ef-865a-01aa75ed71a1/language-en>.
- [11] CAST, «The UDL Guidelines» [Online]. Available: <https://udlguidelines.cast.org/>.
- [12] M. Resnick, *Come i bambini. Immagina, crea, gioca e condividi. Coltivare la creatività con il Lifelong Kindergarten del MIT.*, Feltrinelli, 2018.
- [13] S. Papert, *Mindstorms -- Children, Computers and Powerful Ideas*, Basic Books, Inc. Publishers, 1980.
- [14] OECD, «The ABC of Gender Equality in Education» 2015. [Online]. Available: https://www.oecd.org/en/publications/2015/03/the-abc-of-gender-equality-in-education_g1g51025.html.

- [15] Cesaro, L., Franceschini, A., Badaloni, S., Menegatti, E., Rodà, A., «Gender Biases in Robots for Education» in Proceedings of the 3rd Workshop on Bias, Ethical AI, Explainability and the role of Logic and Logic Programming co-located with the 23rd International Conference of the Italian Association for Artificial Intelligence (AIxIA 2024), 2024.

**RACCONTI DI ESPERIENZE SUL CAMPO
E RELAZIONI SCIENTIFICHE**

PER LA PRIMARIA E LA SECONDARIA DI PRIMO GRADO

“Io Studio Informatica”. Una piattaforma basata sul modello UMC

Enrica Gentile¹, Paola Plantamura¹, Francesco Scarci¹, Francesca Pia Travisani¹

¹ Dipartimento di Informatica

Università degli Studi di Bari “Aldo Moro”

paola.plantamura@uniba.it

Abstract

La didattica dell'informatica è un tema molto discusso, ma ancora oggi non dispone di una metodologia didattica ben definita che sostenga i docenti nell'insegnamento. Affinché sia garantito un apprendimento trasversale e completo, è fondamentale implementare delle metodologie didattiche sistematiche. Queste devono agevolare la trasmissione dei contenuti nell'ambito informatico, promuovendo un coinvolgimento attivo e consapevole nella progettazione e creazione di soluzioni tecnologiche. Per rispondere a questa esigenza, in questo lavoro presentiamo la piattaforma “Io Studio Informatica” rivolta a docenti e studenti delle scuole primarie e secondarie di primo e secondo grado che, basandosi sulla metodologia UMC (*Use-Modify-Create*), offre un approccio didattico strutturato e consente la gestione di percorsi di apprendimento organizzati, dinamici e versatili per l'insegnamento dell'informatica.

1. Introduzione

Nel sistema scolastico italiano l'informatica è ormai una componente essenziale e ampiamente diffusa che va oltre la sola programmazione e possiede un grande potenziale educativo sia come disciplina autonoma, per imparare a usare e a realizzare applicativi, sia come tecnologia dell'informazione, per manipolare e diffondere dati e informazioni in vari formati.

Infatti, come sottolineano Wing [16] e Nardelli [11], l'informatica riveste un ruolo fondamentale nella formazione degli studenti poiché è una disciplina scientifica che favorisce lo sviluppo del pensiero computazionale e che, inglobando pensiero critico, creatività, problem-solving e ragionamento logico, risulta applicabile nei diversi contesti della vita quotidiana [4].

Nel corso del tempo, la ricerca scientifica e pedagogica ha sviluppato diversi modelli per la didattica dell'informatica che, incentrati sulla promozione dell'apprendimento attivo e creativo, mettono insieme concetti teorici e pratici, contestualizzandoli in situazioni reali in cui il docente agisce come guida, accompagnando gli studenti nel processo di apprendimento.

Questo avviene, ad esempio, nella piattaforma Code.org, che, in collaborazione con l'iniziativa italiana del Ministero dell'Istruzione “Programma il Futuro”, ha cominciato a combinare la programmazione a blocchi con tutorial interattivi, introducendo il pensiero computazionale nelle scuole [8].

Le proposte dei moderni modelli didattici, in particolare nel campo della didattica dell'informatica, sono state inoltre arricchite con framework strutturati e innovativi che permettono di supportare i docenti nell'organizzazione delle lezioni. Tra questi possiamo richiamare il *Problem-Based Learning* [1], il *Substitution, Augmentation, Modification, Redefinition* [3], il *Predict, Run, Investigate, Modify, Make* [9] e l'*Engage, Explore, Explain, Elaborate, Evaluate* [14]. Oltre a questi, trova grande applicazione il modello UMC (*Use-Modify-Create*) [6] che, con tre semplici fasi progressive, permette agli studenti di passare da essere utenti di un prodotto a creatori. Oggetto del presente lavoro è la presentazione del prototipo della piattaforma "*Io Studio Informatica*" che, attraverso l'implementazione del modello UMC, guida docenti e studenti nello svolgimento delle attività didattiche durante le lezioni di informatica.

2. Metodologia UMC

Il modello UMC si distingue per la sua semplicità, flessibilità, efficacia ed efficienza nel supportare l'apprendimento progressivo e personalizzato in diversi ambiti didattici [2, 12, 13, 15]. Esso favorisce un livello costante di concentrazione e partecipazione attiva degli studenti, bilanciando la dimensione pratica e tecnologia della didattica con quella teorica dell'informatica e del pensiero computazionale [2, 7, 10].

A differenza di altri approcci, l'UMC offre un'organizzazione, strutturata in poche e semplici fasi, adatta per lo sviluppo della creatività e delle competenze di ragionamento logico e problem-solving, promuovendo così la comprensione dei concetti alla base dell'informatica e della sua applicazione pratica.

Il modello UMC, sviluppato nel 2011 da Lee *et al* [6], è particolarmente usato nell'insegnamento della programmazione informatica e del pensiero computazionale ma è applicabile anche in altri ambiti in cui è previsto un apprendimento attivo, creativo e laboratoriale. Il modello UMC guida l'apprendimento degli studenti attraverso tre fasi consecutive: Use, Modify, Create (Figura 1):

- Nella prima fase (*Use*) gli studenti utilizzano strumenti, software o risorse già esistenti. Esplorando, osservando e interagendo con prodotti o codici creati da altri ("Not Mine"), acquisiscono familiarità con concetti di base, interfacce e funzionalità fondamentali.
- Nella fase intermedia (*Modify*) gli studenti modificano, personalizzano e migliorano i prodotti o i codici analizzati precedentemente ("Not Mine") iniziando a sviluppare le competenze di problem-solving. Le modifiche possono variare in complessità, da semplici personalizzazioni a ristrutturazioni più articolate ("Mine").
- La fase finale (*Create*) prevede che gli studenti creino integralmente un nuovo progetto basandosi sulle conoscenze e competenze acquisite nelle fasi precedenti per sviluppare soluzioni originali e innovative ("Mine"). Quest'ultima fase incoraggia la creatività e l'applicazione pratica delle competenze acquisite puntando anche al miglioramento continuo del progetto finale creato.

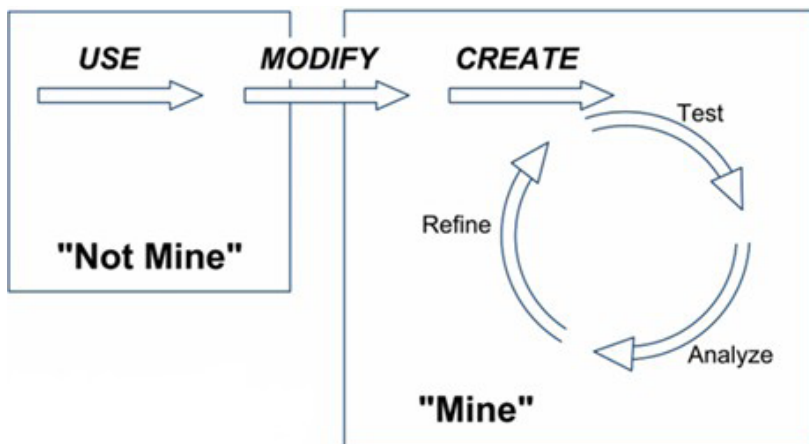


Figura 1: Metodologia UMC [6]

Nella fase *Create* è presente un ciclo interno che itera le fasi di *test* del progetto, di *analisi* dei risultati, per valutare quali funzionalità sono eseguite correttamente e cosa potrebbe essere migliorato, ed infine di *refine*, per apportare modifiche e perfezionamenti al progetto sulla base dell'analisi effettuata. In tal modo, il modello UMC permette di definire un percorso di apprendimento strutturato con l'obiettivo di sviluppare le competenze e le capacità di agire in autonomia passando dall'utilizzo di esempi già funzionanti alla creazione e miglioramento iterativo di nuove proposte [5].

Per la realizzazione della piattaforma “Io Studio Informatica” abbiamo quindi scelto il modello UMC in quanto fornisce un modello organizzativo chiaro, fondato su una visione modulare dell'inserimento del materiale didattico, e permette di far abituare progressivamente gli studenti a compiti sempre più difficili attraverso una serie di fasi con complessità graduale.

3. La piattaforma “Io Studio Informatica”

La piattaforma “Io Studio Informatica” si presenta come un ambiente guidato nel quale docenti e studenti possono collaborare allo svolgimento di laboratori di programmazione che sviluppano in modo sistematico il modello UMC.

Il docente è invitato a creare sessioni di approfondimento per ogni singolo argomento, denominate “lezioni”, ciascuna composta da cinque attività (Figura 2):

- USE
- TEORIA
- MODIFY
- TEST
- CREATE

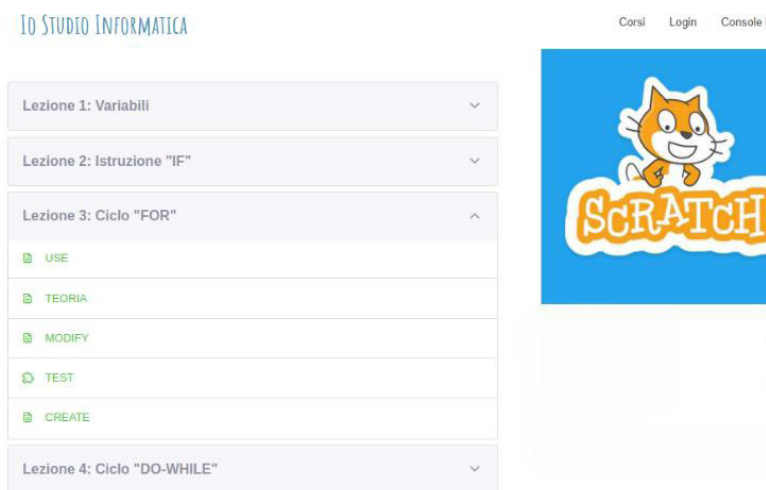


Figura 2: Organizzazione delle lezioni

L'ambiente è stato progettato per offrire non solo un sistema di inserimento dei contenuti, bensì anche un modello organizzativo chiaro, fondato su una visione modulare dell'inserimento del materiale didattico. La piattaforma "Io Studio Informatica" è strutturata in modo semplice e coerente, seguendo l'impostazione dei moderni portali dedicati all'erogazione di corsi online. L'interfaccia è progettata per essere intuitiva e facilmente navigabile, anche da utenti con competenze informatiche di base.

Durante la progettazione dei corsi, il docente può utilizzare il linguaggio di programmazione che meglio si adatta alle attività laboratoriali previste (Figura 3).

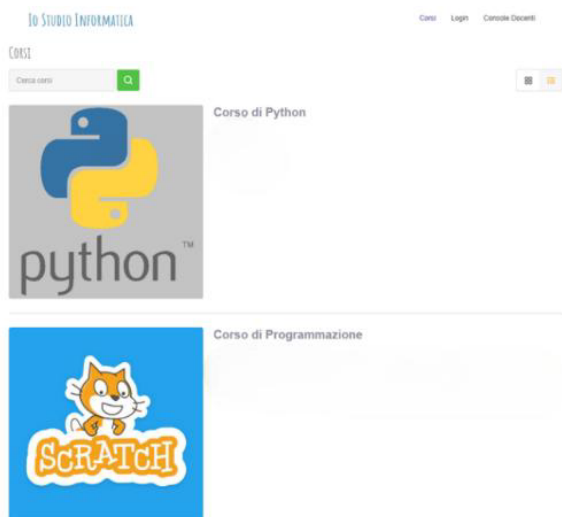


Figura 3: Corsi

Il docente avrà inoltre a disposizione una propria console gestionale dove poter gestire i corsi (Figura 4). Una volta creato il corso e fatto l'accesso, il docente ha la possibilità di creare le singole lezioni.

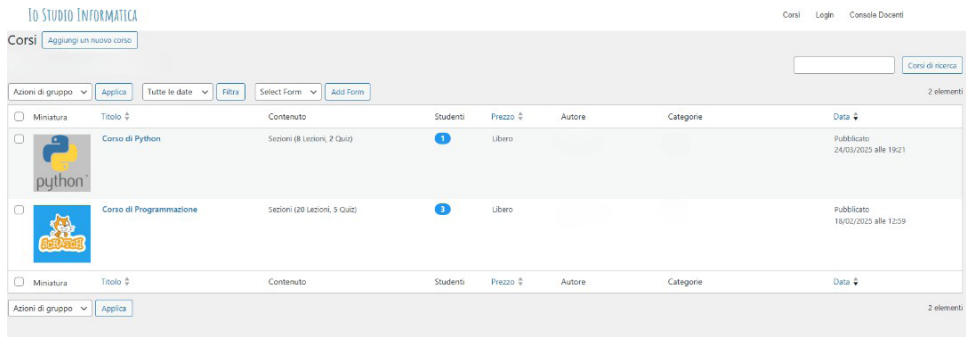


Figura 4: Console docente

La lezione creata apparirà automaticamente organizzata nelle attività richiamate dal modello UMC (Figura 5). Queste attività di base vengono inoltre integrate con un'attività di teoria, in cui il docente caricherà il pdf con le nozioni teoriche di base con cui lo studente può consolidare le sue conoscenze, ed un'attività di test, utile allo studente per verificare la comprensione dei contenuti. Il docente, a questo punto, dovrà preparare il progetto che intende presentare agli studenti utilizzando un ambiente di sviluppo scelto sulla base del linguaggio di programmazione oggetto della lezione. Nel nostro esempio abbiamo utilizzato Snap.Berkeley per la programmazione a blocchi (Figure 6,7,8) e Trinket per il linguaggio Python. Dopo di che potrà inserire:

- nell'**attività USE**, il link dell'anteprima dell'esecuzione del progetto preparato;
- nell'**attività MODIFY**, una descrizione della modifica da apportare con il link al progetto preparato nella modalità editing;
- nell'**attività CREATE**, una descrizione del compito da svolgere con il link all'ambiente di lavoro.

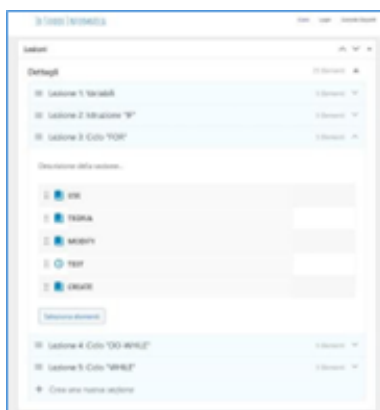


Figura 5: Creazione lezione Ciclo FOR

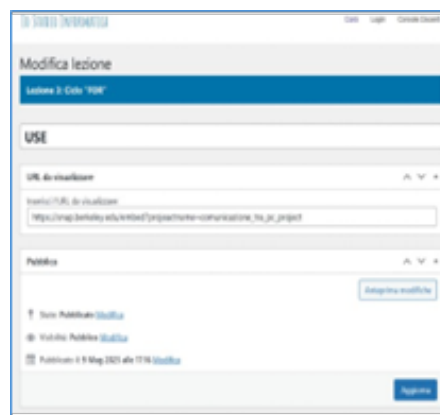


Figura 6: Creazione attività USE



Figura 7: Creazione attività MODIFY



Figura 8: Creazione attività CREATE

Lo studente, seguendo il modello UMC, svolgerà le attività didattiche organizzate come segue:

- **Attività USE**, in cui viene mostrato un esempio completo dell'argomento in questione offrendo una visione immediata e concreta del risultato finale (Figura 9).
- **Attività TEORIA**, in cui viene proposto un contenuto che illustra allo studente le nozioni teoriche per contestualizzare l'argomento della lezione (Figura 10).
- **Attività MODIFY**, in cui lo studente esplora nel dettaglio la struttura e il codice dell'esempio realizzato. Inoltre, ha la possibilità di modificarlo in base alle indicazioni fornite dal docente, sperimentando e comprendendo attivamente il funzionamento del progetto (Figura 11).
- **Attività TEST**, in cui sono presenti domande a risposta multipla. Ogni domanda ha un massimo di tre risposte e solo una sarà quella corretta. Una volta terminato il test, il sistema evidenzierà l'esito delle scelte e, in caso di errore, indicherà quella corretta (Figura 12).
- **Attività CREATE**, in cui, seguendo la traccia fornita dal docente, lo studente sarà invitato a creare un nuovo progetto (Figura 13).



Figura 9: Attività USE



Figura 10: Attività TEORIA



Figura 11: Attività MODIFY



Figura 12: Attività TEST



Figura 13: Attività CREATE

4. Conclusioni

La piattaforma “Io Studio Informatica”, utilizzando il modello UMC, ha un forte potenziale didattico nell’ambito della formazione in Informatica.

Al fine di valorizzare l’insegnamento dell’informatica, abbiamo infatti ritenuto opportuno adottare la metodologia UMC per l’organizzazione delle unità didattiche in quanto essa offre un supporto trasversale, concreto e sistematico al processo educativo.

L’uso di questa piattaforma mira a facilitare la fruizione dei contenuti informatici, teorici e pratici, da parte degli studenti, sia delle scuole primarie che delle secondarie di primo e secondo grado, poiché permette loro di assumere in maniera crescente conoscenze e competenze, passando da semplici utilizzatori di un prodotto software, a ideatori e creatori di nuovi.

Bibliografia

- [1] Bashith, A., & Amin, S. The Effect of Problem-Based Learning on EFL Students' Critical Thinking Skill and Learning Outcome. *Al-Ta'lim Journal*, 24(2), 93-102, 2017. <https://doi.org/10.15548/jt.v24i2.271>
- [2] Capecchi, S., Lodi, M., Lonati, V., & Sbaraglia, M. Castle and Stairs to Learn Iteration: Co-designing a UMC Learning Module with Teachers. In *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1* (pp. 222-228), 2023, June. <https://doi.org/10.1145/3587102.3588793>
- [3] Crompton, H., & Burke, D. Apprendimento mobile e opportunità pedagogiche: una revisione sistematica configurativa della ricerca PreK-12 utilizzando il framework SAMR. *Computers & Education*, 156, Articolo 103945, 2020. <https://doi.org/10.1016/j.compedu.2020.103945>
- [4] Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C., & Woollard, J. *Computational Thinking. A Guide for Teachers*. Computing at School, 2015.
- [5] Franklin, D., Coenraad, M., Palmer, J., Eatinger, D., Zipp, A., Anaya, M., White, M., Pham, H., Gökdemir, O., & Weintrop, D. An Analysis of Use-Modify-Create Pedagogical Approach's Success in Balancing Structure and Student Agency. In *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education* (pp. 14-24), 2020. <http://dx.doi.org/10.1145/3372782.3406256>
- [6] Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W. C., Erickson, J., Malyn-Smith, J., & Werner, L. Computational Thinking for Youth in Practice. *ACM Inroads*, 2(1), 32-37, 2011. <https://doi.org/10.1145/1929887.1929902>
- [7] Lytle, N., Cateté, V., Boulden, D., Dong, Y., Houchins, J., Milliken, A., ... & Barnes, T. Use, Modify, Create: Comparing Computational Thinking Lesson Progressions for STEM Classes. In *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education* (pp. 395-401), 2019, July.
- [8] Ministero dell'Istruzione e Code.org. Programma il Futuro. Disponibile su: <https://programmmailfuturo.it>
- [9] Montresor, A., & Glow Association. *Sperimentazione del metodo PRIMM nell'insegnamento della programmazione*. Università di Trento, 2023.
- [10] Mselle, L. J., & Twaakyondo, H. M. The impact of Memory Transfer Language (MTL) on reducing misconceptions in teaching programming to novices. *International Journal of Machine Learning and Applications*, 1(1), 2012. <https://doi.org/10.4102/ijmla.v1i1.3>
- [11] Nardelli, E. Informatica nella scuola: disciplina fondamentale e trasversale, ovvero "di cosa parliamo quando parliamo di pensiero computazionale". *Scienze e Ricerche Magazine*, 47, 36-40, 2017.
- [12] Nardelli, E., Lacchia, F., Davoli, R., Lodi, M., Sbaraglia, M., Rossano, V., Gentile, E., Lonati, V., Monga, M., Morpurgo, A., Forlizzi, L., Melideo, G., Capecchi, S., Fronza, I., & Vardanega, T. Iterazione di apprendimento per i gradi 2-3: Puzzle vs. UMC in Code.org. Negli Atti del 54° Simposio tecnico ACM sull'educazione informatica V. 2 (SIGCSE 2023). Association for Computing Machinery, New York, NY, USA, 1368, 2023. <https://doi.org/10.1145/3545947.3576312>

- [13] Rachmatullah, A., Mayhorn, C. B., & Wiebe, E. N. The effects of prior experience and gender on middle school students’ computer science learning and monitoring accuracy in the Use-Modify- Create progression. *Learning and Individual Differences*, 86, 101983, 2021. <https://doi.org/10.1016/j.lindif.2021.101983>
- [14] Siwawetkul, W., & Koraneekij, P. Effect of 5E instructional model on mobile technology to enhance reasoning ability of lower primary school students. *Kasetsart Journal of Social Sciences*, 2018.
- [15] Vieira, C., Gómez, R. L., Gómez, M., Canu, M., & Duque, M. Implementing Unplugged CS and Use- Modify-Create to Develop Student Computational Thinking Skills: A Nationwide Implementation in Colombia. *Educational Technology & Society Journal*, 26(3), 155-175, 2023. [https://doi.org/10.30191/ETS.202307_26\(3\).0012155](https://doi.org/10.30191/ETS.202307_26(3).0012155)
- [16] Wing, J. M. Computational Thinking. *Communications of the ACM*, 49(3), 33-35, 2006.

Percorsi formativi di accompagnamento per docenti sul coding, machine learning e l'intelligenza artificiale generativa applicata alla didattica

Giovanni Silvestro¹, Marta Sanz Manzanedo²

¹ EFT Toscana - IC Gambera di Pisa

giovanni.silvestro1@scuola.istruzione.it

² EFT Toscana – Liceo Francesco Cecioni di Livorno

marta.sanzmanzanedo@scuola.istruzione.it

Abstract

Nel contesto delle attività delle Équipe Formative Territoriali (EFT) della Toscana, svoltesi nel biennio 2023–2025 in attuazione del Piano Nazionale Scuola Digitale e del PNRR, il presente contributo descrive due laboratori condotti con classi della scuola secondaria di primo grado, in stretta collaborazione tra il docente corsista e il formatore. I laboratori, inseriti nel percorso “DigiLab”, hanno previsto un accompagnamento diretto in aula, configurandosi come esperienze significative di co-progettazione didattica e formazione in situazione. Il primo laboratorio ha coinvolto gli studenti nell’addestramento di un modello di *Machine Learning*, tramite il software *mBlock5* e il robot educativo *Codey Rocky*, per la classificazione dei rifiuti in un contesto di robotica educativa orientata alle STEM. Il secondo ha sperimentato l’uso di chatbot basati su intelligenza artificiale generativa per il potenziamento delle competenze comunicative in lingua straniera, integrando lo sviluppo di test interattivi realizzati con linguaggi web (HTML, CSS e JavaScript) guidati da prompt generativi. Entrambi gli interventi hanno favorito lo sviluppo delle competenze digitali degli studenti, offrendo modelli replicabili di didattica laboratoriale e interdisciplinare con tecnologie emergenti.

1. Introduzione

Durante il biennio 2023-2025, le Équipe Formative Territoriali (EFT) hanno rappresentato il fulcro della trasformazione digitale delle scuole italiane, in perfetta sintonia con le linee guida del Piano Nazionale Scuola Digitale (PNSD) e con gli investimenti del Piano Nazionale di Ripresa e Resilienza (PNRR). Rispetto agli altri Poli formativi presenti in piattaforma Scuola Futura e finanziati con fondo PNRR, le EFT si sono contraddistinte per le attività di accompagnamento nelle scuole del territorio mediante richieste di “help desk” effettuate direttamente dai docenti e dirigenti scolastici. In Toscana, questo modello operativo centrato sul supporto costante dei docenti nelle loro classi, ha avuto la massima espressione nei laboratori “blended” dal titolo DigiLab che si svolgono regolarmente dall’A.S. 2023/24. In questo articolo, mostriamo in particolare due laboratori di accompagnamento ai docenti svolti con le loro classi sul

coding e la robotica per le discipline STEM e un altro sull'intelligenza artificiale generativa per addestrare chatbot per le lingue e per generare codice con linguaggi HTML e JavaScript al fine di realizzare Test di valutazione avviabili su browser.

Nel primo laboratorio sono stati inseguiti obiettivi dell'area 3 del DigComp 2.2 [1] "Creazione di contenuti digitali", specificamente nella competenza 3.4 "Programmazione", e nell'area DigComp numero 5 "Risolvere problemi", con un focus sulla competenza 5.3 "Utilizzare in modo creativo le tecnologie digitali".

Nel secondo laboratorio ci si è focalizzati nell'area di competenza 2 del DigComp 2.2 "Comunicazione e collaborazione", esplorando in particolare la competenza 2.1 "Interagire con gli altri attraverso le tecnologie".

2. Laboratori DigiLab sulle metodologie applicate alla didattica digitale e potenziati dall'informatica e l'IA.

I percorsi formativi DigiLab si sono posti l'obiettivo di fornire ai docenti di qualsiasi disciplina della Toscana, strumenti operativi semplici e prontamente utilizzabili nella pratica quotidiana. In linea con i progetti nazionali EFT, questo workshop ha favorito la transizione digitale nelle scuole e, in particolare, lo sviluppo delle competenze digitali degli studenti attraverso attività di accompagnamento a supporto della formazione dei docenti, che è l'area DigCompEdu [2] di riferimento del corso. Una caratteristica fondamentale del DigiLab è stata la possibilità, per i docenti partecipanti, di essere accompagnati nell'esperienza in aula direttamente dai componenti dell'EFT Toscana, previo accordo e disponibilità. Questo approccio pratico sul campo ha permesso al team di formatori, di affiancare gli insegnanti e le loro classi in attività concrete, trasformando i concetti teorici in esperienze di apprendimento significative per gli studenti.

3. Laboratorio sull'addestramento di Machine Learning per la programmazione di robot da attivare per spostare elementi didattici significativi

Facendo riferimento alla Proposta di Indicazioni Nazionali per l'insegnamento dell'Informatica nella Scuola [3], gli alunni coinvolti delle classi seconde dell'I.C. Gamera di Pisa, hanno raggiunto diversi traguardi come il "saper selezionare gli strumenti digitali più appropriato rispetto ai propri obiettivi espressivi" (competenza O-M-R-2 nell'Ambito creatività digitale) e "saper scrivere programmi che usano l'annidamento di cicli e selezioni" (competenza O-M-P-2 dell'ambito programmazione).

L'obiettivo disciplinare dell'insegnante della classe, invece, era quello di far classificare i rifiuti agli alunni, al fine di addestrare un modello di *machine learning* in grado di guidare un robot nello smistamento corretto degli stessi. Un ulteriore obiettivo consisteva nella riflessione critica sulla costruzione di un dataset di immagini adeguato, al fine di ridurre al minimo la presenza di bias. L'idea alla base del progetto era dimostrare come l'Intelligenza Artificiale e la robotica possano essere applicate in un contesto pratico e significativo, quale la raccolta differenziata.

Come mezzo didattico abbiamo utilizzato due strumenti principali: il software "*mBlock5*" [4] su un *notebook* dotato di telecamera esterna e il robot educativo *Codey Rocky* [5].

Codey Rocky è stato scelto perché è un robot educativo concepito per lo studio delle materie STEM adatto agli alunni degli Istituti Comprensivi. È composto da due parti: *Codey*, che è il controller programmabile e rimovibile che contiene oltre 10 moduli elettronici (inclusi sensori e un display LED espressivo), e *Rocky*, che è la base mobile che si sposta grazie a *Codey*. *Rocky* ha la capacità di evitare ostacoli, riconoscere colori e seguire linee. Inoltre è compatibile all'aggiunta di componenti Lego che gli alunni hanno utilizzato per trasformarlo in una sorta di “muletto sposta-rifiuti”.

Il software di programmazione utilizzato è “*mBlock5*” di *Makeblock*. Questo ambiente si basa sul linguaggio visuale tipo *Scratch 3.0* che lo rende molto intuitivo per i ragazzi grazie all'interfaccia a blocchi colorati che si trascinano fino a incastrarsi in una sequenza logica. La web app di coding “*mBlock5*” è particolarmente interessante perché supporta anche funzionalità di Intelligenza Artificiale e persino di *Internet of Things* (IoT).

Il cuore di questo laboratorio è stato l'utilizzo dell'estensione “*Machine Learning 2.0*” all'interno di “*mBlock5*”. Questo ci ha permesso di far creare agli studenti un modello di riconoscimento di immagini. In pratica, hanno insegnato al computer a distinguere diversi tipi di rifiuti.

Ecco i passaggi chiave che abbiamo seguito, che i ragazzi hanno svolto in prima persona:

- Prima di tutto, all'interno di “*mBlock 5*”, nella sezione “*Sprite*”, si è aggiunta l'estensione “*Machine Learning 2.0*” (Figura 1) all'area dei blocchi.
- Sulla nuova pagina di progetto, si è dato un nome al modello, ad esempio “Rifiuti Differenziata”.
- Questa è stata la fase più interattiva per i ragazzi (Figura 2): nella pagina per l'addestramento del modello, hanno dovuto aggiungere i dati. Abbiamo definito le diverse “classi” o categorie di rifiuti che volevamo che il modello riconoscesse, ad esempio “Plastica”, “Carta”, “Metallo”, “Organico”, cliccando su “aggiungi una classe” per ogni tipo. Per ciascuna classe, abbiamo usato la webcam collegata al PC per raccogliere diverse immagini di quel tipo di rifiuto. I ragazzi hanno partecipato attivamente mostrando gli oggetti (bottiglie, pezzi di carta, lattine, ecc.) da varie angolazioni, comprendendo l'importanza di avere molti dati diversi per un buon addestramento.

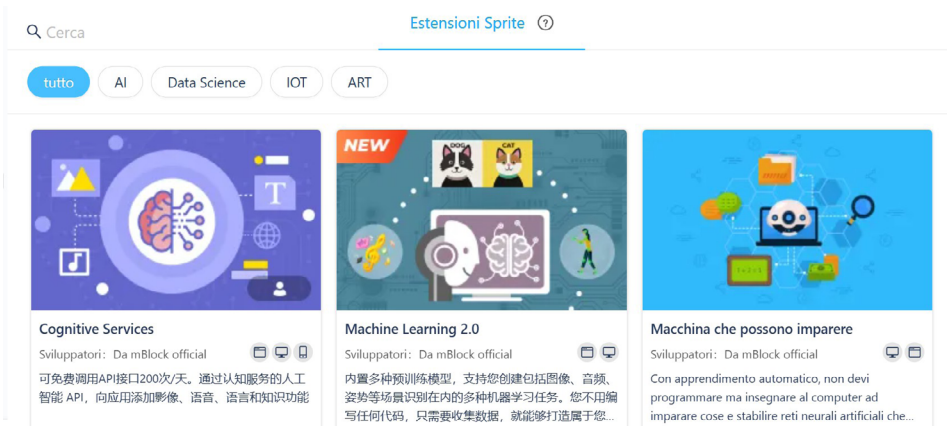


Figura 1: Estensione Machine learning 2.0

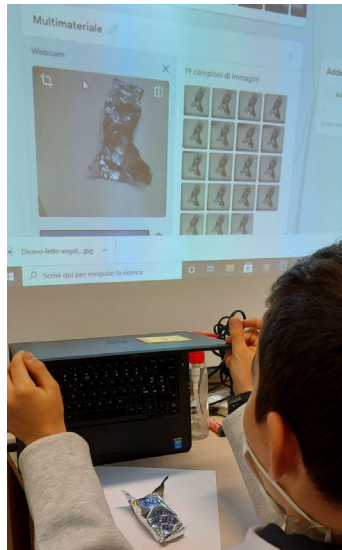


Figura 2: Fase di Addestramento

- Dopo aver raccolto i dati per tutte le classi definite, si è avviato l'addestramento del modello. La web app “mBlock”, in automatico, ha elaborato le immagini fornite e ha “imparato” a riconoscere i diversi tipi di rifiuti basandosi sui dati.
- Una volta completato l'addestramento, i ragazzi hanno potuto visualizzare in anteprima i risultati. Sul pannello di destra, appariva l'immagine catturata dalla telecamera e sotto, per ogni classe, la probabilità che l'oggetto riconosciuto appartenesse a quella categoria. Se i risultati non erano soddisfacenti, potevano aggiungere altri dati o modificare i parametri e riaddestrare il modello (Fig. 3).

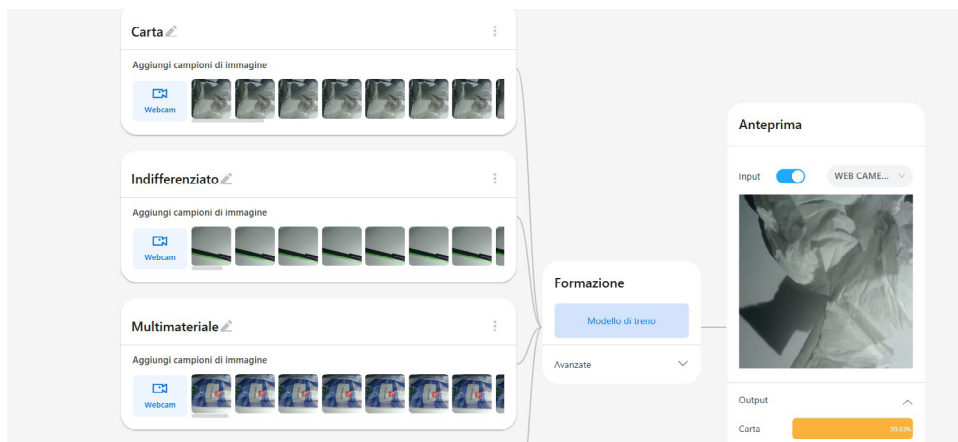


Figura 3: Collando sull'efficacia dell'addestramento con Machine learning 2.0

- Quando sono stati soddisfatti del modello addestrato, la web app in automatico ha integrato il modello creato nei blocchi di programmazione di “mBlock”, rendendo disponibili nuovi blocchi basati sull’AI.

A questo punto, gli alunni hanno potuto utilizzare i blocchi basati sull’AI per programmare il comportamento del robot. In sintesi hanno unito l’intelligenza artificiale addestrata su mBlock nell’estensione “Machine Learning 2” inserita nel coding a blocchi di messaggi inviati dall’area “sprite” all’area di coding del robot *Codey Rocky* che li riceve mediante i blocchi “Situazioni” mettendo in atto il movimento necessario a spostare i rifiuti nell’area dedicata alla raccolta differenziata.

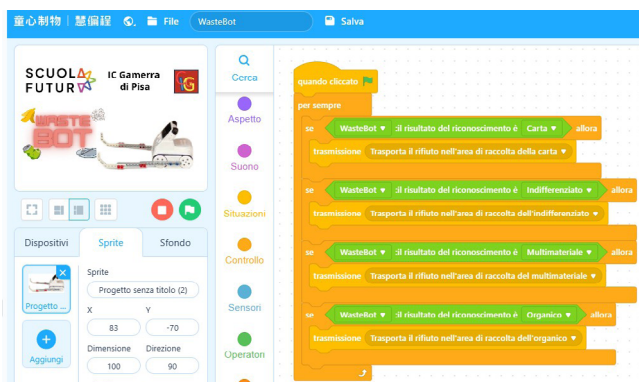


Figura 4: Coding nell’area Sprite per associare il riconoscimento rifiuti al messaggio da inviare al robot *Codey Rocky*

In pratica, utilizzando i blocchi di programmazione di mBlock, gli alunni hanno scritto il codice per *Codey Rocky*. Hanno usato blocchi condizionali (come “se... allora”) per dire al robot cosa fare in base al risultato del riconoscimento (Figura 4). Ad esempio: “Se il modello riconosce ‘Plastica’, allora invia il messaggio a *Codey Rocky* di muoversi in avanti e ruotare per un certo tempo fino a trascinare il rifiuto spostandolo verso l’area designata per la plastica.” (Figura 5).

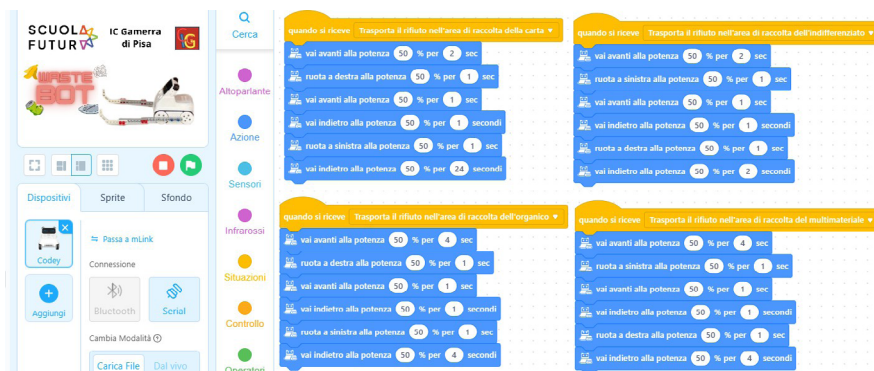


Figura 5: Ricezione del messaggio nell’area “Dispositivi”, per muovere il robot

Questo laboratorio ha permesso ai ragazzi di mettere letteralmente le mani in pasta con concetti di Intelligenza Artificiale, robotica e programmazione. Hanno migliorato le loro abilità di pensiero computazionale, logica e creatività, familiarizzando con tecnologie fondamentali per il futuro. È stato un modo concreto e divertente per capire come l'AI “vede” il mondo e come può interagire con esso tramite un robot. Il frutto del loro lavoro è stato mostrato da una rappresentanza degli alunni a due eventi regionali ma di importanza nazionale: Scuola Futura Campus Arezzo (Figura 6) che si è svolto dal 6 al 8 maggio nella città aretina e durante il Festival della Robotica che si è svolto a Pisa dal 9 all'11 maggio 2025 (Figura 7).



Figura 6: Scuola Futura Campus ad Arezzo



Figura 7: Festival della Robotica di Pisa

4. Laboratorio sul potenziamento delle lingue straniere tramite Chatbot preaddestrati dal docente e test valutativi ottenuti in codice HTML con applicativi basati sull'I.A. generativa.

Con l'avvento degli LLM sempre più docenti stanno scoprendo l'opportunità di creare esercizi di programmazione, spiegazioni di codice e materiali didattici personalizzati, facilitando la preparazione delle lezioni e l'adattamento ai bisogni degli studenti [6]. Una delle attività svolte ha riguardato l'uso di un chatbot (Figura 8), nello specifico “Mizou” [7], con gli alunni di una classe prima di scuola secondaria di primo grado. L'obiettivo era aiutare gli studenti a comprendere come interagire tramite diverse tecnologie digitali e quali strumenti di comunicazione sono appropriati in contesti specifici. L'utilizzo della web app non prevede l'utilizzo di account degli alunni e il loro utilizzo può essere verificato dall'account docente che ha creato e condiviso il chatbot.

Abbiamo lavorato a stretto contatto con la docente di spagnolo della classe, creando un'esperienza interdisciplinare. Durante l'attività (Figura 9), gli studenti, con l'aiuto dell'insegnante e del formatore esperto, hanno utilizzato il chatbot. Abbiamo esplorato esempi di come identificare i segnali che indicano se si sta comunicando con un essere umano o con un agente conversazionale basato sull'IA. È stato un'opportunità per discutere l'uso degli strumenti digitali per la comunicazione e la consapevolezza che i sistemi di IA, come i chatbot, mirano a fornire un'interazione simile a quella umana. Abbiamo anche toccato aspetti legati all'interazione e al feedback che si può dare a un sistema di intelligenza artificiale per orientare le sue risposte future.

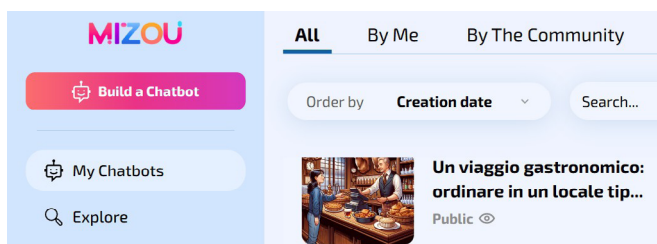


Figura 8: Chatbot addestrato dal docente per simulare il ruolo di cameriere di un paese estero



Figura 9: Fase di accompagnamento in classe del Digilab per le lingue straniere con chatbot

Nella parte di formazione dei docenti è stata mostrata una modalità semplificata per generare un test di valutazione finale dell'attività svolta mediante l'uso dell'app Gemini (modello 2.5 Flash). L'idea è nata da utilizzi didattici con l'HTML senza IA attuata in altri contesti e con altre applicazioni [8]. Per realizzare il test interattivo di spagnolo [9], abbiamo utilizzato tre tecnologie fondamentali che lavorano insieme all'interno del browser web:

- **HTML (HyperText Markup Language)**

L'HTML è il linguaggio di base per costruire la struttura di qualsiasi pagina web.

Con l'HTML abbiamo definito la presenza di tutti gli elementi visibili del test: il titolo principale, le istruzioni per l'alunno, i campi di input dove l'alunno può digitare il suo nome, cognome, classe e l'indirizzo email del docente (Figura 10). Ogni domanda del quiz è stata inserita con le sue opzioni di risposta (gestite tramite un tipo di input chiamato "radio button" che permette di selezionare una sola opzione per domanda). Infine, l'HTML ha anche posizionato i pulsanti che l'alunno cliccherà per "Verificare Risposte" e "Ricominciare Test", e l'area dove verranno visualizzati i risultati.

- **CSS (Cascading Style Sheets) con Tailwind CSS**

Il CSS è il linguaggio utilizzato per lo stile e la presentazione visiva della pagina web. Se l'HTML definisce "cosa" c'è sulla pagina, il CSS definisce "come" appare. Abbiamo usato il CSS per impostare i colori di sfondo, i tipi e le dimensioni dei caratteri, la spaziatura tra gli elementi, l'allineamento del testo e dei blocchi di contenuto. Questo linguaggio ha permesso di rendere il test visivamente gradevole e facile da leggere. Abbiamo impiegato un framework CSS chiamato Tailwind CSS, che fornisce una serie di "classi" predefinite. Queste classi sono piccole istruzioni che, aggiunte direttamente nell'HTML, permettono di applicare rapidamente

stili come bordi arrotondati, ombreggiature e comportamenti responsivi, garantendo che il test si adatti bene a schermi di diverse dimensioni, dai computer ai tablet e smartphone.

• JavaScript

JavaScript è il linguaggio di programmazione che rende la pagina web dinamica e interattiva, gestendo la logica del test. Pensa a JavaScript come al “motore” che fa funzionare il test. Il suo ruolo è stato cruciale per diverse funzionalità:

- Generazione delle domande: invece di scrivere ogni domanda e opzione direttamente nell'HTML, JavaScript le preleva da una lista predefinita e le “costruisce” dinamicamente sulla pagina quando questa viene caricata (Figura 11).
- Gestione dell'interazione: quando l'alunno clicca sul pulsante “Verifica Risposte”, JavaScript entra in azione. Prende i valori inseriti nei campi (nome, cognome, classe, email) e controlla quali opzioni di risposta sono state selezionate per ogni domanda.
- Reset del test: infine, JavaScript gestisce anche la logica per “resettare” il test, cancellando tutte le selezioni e i risultati per permettere all'alunno di ricominciare.
- Calcolo e Feedback: JavaScript confronta le risposte dell'alunno con quelle corrette, calcola il punteggio finale e poi mostra un feedback dettagliato per ogni domanda, indicando se la risposta data era giusta o sbagliata, e qual era quella corretta in caso di errore. (Figura 12)

Questo laboratorio ha permesso agli studenti di approcciare l'IA in modo pratico e guidato, sviluppando la capacità di scegliere tecnologie digitali semplici per l'interazione e identificare adeguati mezzi di comunicazione in un dato contesto. Parallelamente, ha rafforzato la comprensione che le tecnologie digitali, inclusa l'IA, sono utilizzate in diversi contesti e possono influire su vari aspetti della vita, come la comunicazione e l'apprendimento delle lingue.

¡A comer! Il tuo test di spagnolo al ristorante

Questo test ti aiuterà a capire quanto sei pronto/a a ordinare il cibo e interagire con un cameriere in un ristorante spagnolo. Leggi attentamente ogni situazione e scegli la risposta che ti sembra più appropriata. ¡Mucha suerte!

Nome:

Cognome:

Classe:

Email del docente (per l'invio dei risultati):

Figura 10: Maschera iniziale

Domanda 1: Sei appena entrato/a nel ristorante. Il cameriere ti si avvicina e ti chiede: "¿Qué desea, señor/señorita?" Cosa rispondi per chiedere un tavolo?

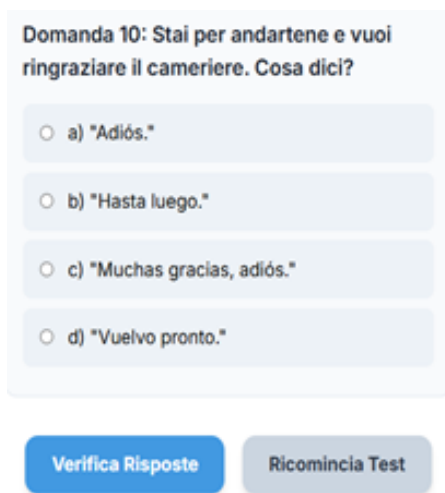
☒ a) "Quiero una mesa para dos, por favor."

☐ b) "Está bien, gracias."

☐ c) "¿Cuánto cuesta?"

☐ d) "No quiero nada."

Figura 11a: Domanda del test di spagnolo ottenuta dalla web app generata con l'IA

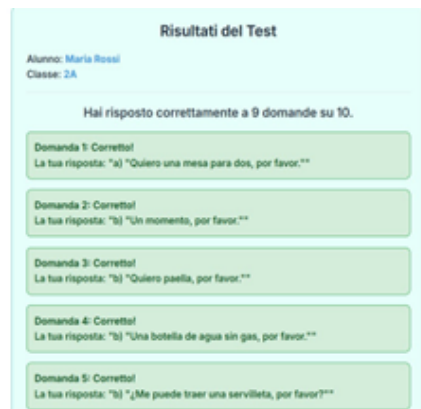


Domanda 10: Stai per andartene e vuoi ringraziare il cameriere. Cosa dici?

- ☐ a) "Adiós."
- ☐ b) "Hasta luego."
- ☐ c) "Muchas gracias, adiós."
- ☐ d) "Vuelvo pronto."

Verifica Risposte **Ricomincia Test**

Figura 11b: Domanda del test di spagnolo ottenuta dalla web app generata con l'IA



Risultati del Test

Alunno: Maria Rossi
Classe: 2A

Hai risposto correttamente a 9 domande su 10.

Domanda 1: Corretto!
La tua risposta: "a) "Quiero una mesa para dos, por favor.""

Domanda 2: Corretto!
La tua risposta: "b) "Un momento, por favor.""

Domanda 3: Corretto!
La tua risposta: "b) "Quiero pasta, por favor.""

Domanda 4: Corretto!
La tua risposta: "c) "Una botella de agua sin gas, por favor.""

Domanda 5: Corretto!
La tua risposta: "b) "Me puede traer una servilleta, por favor?""

Figura 12: Feedback restituito

5. Conclusioni

Queste esperienze sul campo, rese possibili dal formato pratico del corso DIGILab e dalla disponibilità degli insegnanti toscani ad aprirsi all'innovazione didattica, dimostrano l'efficacia dell'apprendimento attivo e laboratoriale. Affiancare i docenti nell'implementazione di attività che coinvolgono direttamente gli studenti in esperienze con l'IA, la programmazione e la robotica, non solo consolida la loro comprensione degli strumenti digitali ma favorisce anche lo sviluppo delle competenze digitali degli studenti in modo concreto e stimolante. Attraverso questi laboratori, gli studenti non si limitano a utilizzare la tecnologia, ma iniziano a comprenderne il funzionamento, a interagire con essa in modo critico e consapevole, e persino a crearla. L'integrazione di queste competenze è fondamentale per prepararli a partecipare pienamente alla società digitale.

Bibliografia

- [1] European Commission's Joint Research Centre. DigComp 2.1: The Digital Competence Framework for Citizens with eight proficiency levels and examples of use" (<http://europa.eu/!Yg77Dh>) https://www.agid.gov.it/sites/default/files/repository_files/digcomp2-1_ita.pdf
- [2] European Commission's Joint Research Centre. DigCompEDU Quadro di riferimento europeo per le competenze digitali dei docenti
- [3] CINI, GRIN, GII. Proposta di Indicazioni Nazionali per l'insegnamento dell'Informatica nella Scuola. Dicembre 2017
- [4] mBlock, web app di coding. <https://mblock.cc/>
Robot Codey Rocky Guida all'uso
- [5] Bonner, E., Lege, R., & Frazier, E. (2023). Large Language Model-Based Artificial Intelligence In The Language Classroom: Practical Ideas For Teaching. Teaching English With Technology. <https://doi.org/10.56297/bkam1691/wieo1749>.
- [6] Mizou - Chatbot addestrabile specifico per la scuola. <https://mizou.com/education>
- [7] Pereira, D., Lima, J., Jardim, R., Rocha, P., Santos, F., & Tarouco, L. (2019). HTML5 Authoring Tool to Support the Teaching-Learning Process. International Journal for Innovation Education and Research. <https://doi.org/10.31686/IJIER.VOL7.ISS2.1325>.
- [8] Esempio di Test di lingue per studenti di spagnolo generato con Google Gemini modello gratuito 2.5 Flash. <https://g.co/gemini/share/64a0a8a9ac31>

Il Server: proposta per un percorso didattico

Daniele Catozzella¹

¹ IC Frascolla Taranto – docente esperto esterno

daniele.catozzella@gmail.com

Abstract

Il documento propone un percorso didattico, a partire dal concetto di *server*, per promuovere in modo integrato l'insegnamento dell'informatica e della cittadinanza digitale nella scuola primaria e secondaria di primo grado. La proposta rappresenta una possibile risposta alle nuove Indicazioni Ministeriali e alle criticità del contesto scolastico italiano su questi temi che, ad oggi, hanno prodotto una frammentazione degli interventi didattici e una scarsa comprensione del ruolo chiave della disciplina informatica per studenti e studentesse. Il percorso, sperimentato in due laboratori PNRR, si articola in otto moduli e può essere adattato alle diverse esigenze didattiche. Il *Server* si presenta come uno strumento didattico trasversale e utile ai docenti del consiglio di classe per la progettazione di un curriculum coerente all'interno dell'anno scolastico.

1. Introduzione

La recente pubblicazione del Ministero dell'Istruzione e del Merito [1] della bozza delle nuove *Indicazioni ministeriali per la scuola dell'infanzia e per il primo ciclo di istruzione* resa disponibile per l'avvio del dibattito pubblico, segna per il contesto scolastico italiano una correzione di rotta urgente e necessaria sul tema delle competenze informatiche e gli obiettivi di apprendimento per studenti e studentesse. Le Indicazioni, che proseguono il loro iter sino alla piena applicazione, prevista per l'anno scolastico 2026/2027, prevedono, infatti, l'introduzione di specifici obiettivi di apprendimento relativi alla disciplina dell'informatica ed inquadrati all'interno delle materie curriculari di matematica e di tecnologia.

Gli obiettivi di apprendimento individuati riprendono il lavoro sviluppato dal CINI [2] con la *Proposta di Indicazioni Nazionali per l'insegnamento dell'Informatica nella Scuola e la Raccomandazione sul miglioramento dell'offerta di competenze digitali nell'istruzione e nella formazione* [3] che hanno affermato l'importanza della disciplina informatica nei percorsi didattici.

Le Indicazioni si inseriscono in un contesto scolastico italiano in cui, sino ad oggi, come confermato dai dati [4] [5] sono due gli elementi centrali con i quali ciascun protagonista della scuola, sia esso docente o studente, si è dovuto confrontare:

- la “confusione” tra competenze informatiche e competenze digitali con un mancato bilanciamento tra le competenze di cittadinanza digitale come indicate dal framework europeo DigComp 2.2 [6] adottate dalla scuola e le attività didattiche disciplinari dell'informatica,
- la disomogeneità e la frammentazione dell'azione didattica, data dal punto precedente, a livello nazionale sul tema delle competenze informatiche e digitali.

In questo quadro la sfida centrale per la scuola dei prossimi anni è quella di costruire un dialogo efficace, concreto e consapevole all'interno di un consiglio di classe per la realizzazione di percorsi didattici di apprendimento nelle classi in grado di unificare, riconoscendone le specifiche, informatica e cittadinanza digitale.

2. Un percorso didattico: il server

La proposta denominata “*Server*” ha l'obiettivo di rispondere alle criticità descritte nell'introduzione sullo scenario attuale delle competenze informatiche e digitali all'interno della scuola italiana. L'obiettivo della proposta è quello di: favorire l'apprendimento delle competenze informatiche e delle competenze di cittadinanza digitale attraverso un percorso didattico coerente in grado di supportare l'intero consiglio di classe nell'affrontare le logiche informatiche della rete, dei suoi processi e servizi affiancando tali apprendimenti alle riflessioni sul comportamento consapevole della rete richieste dalla cittadinanza digitale.

L'idea di utilizzare il *Server*, elemento hardware e software al centro del modello di rete *client/server*, nasce dalla volontà di proporre agli studenti una prospettiva di indagine, studio e intervento del funzionamento e delle attività in rete da un punto di vista altrimenti invisibile, e spesso sconosciuto, della rete.

In tal senso il *Server* diviene un contenitore didattico, a disposizione di tutti i docenti della classe in grado di legare tra loro le diverse lezioni e apprendimenti di un anno scolastico favorendo nel docente la collaborazione e il coordinamento con le altre discipline curriculari laddove ciascuna può giovare del *Server* come strumento didattico interno al quale lavorare.

3. La sperimentazione

Una piccola sperimentazione è stata effettuata all'interno di due laboratori inseriti nelle attività del PNRR sul tema Stem presso l'IC *Fruscolla* di Taranto dove ho svolto l'attività di docente esperto esterno. Il percorso si è sviluppato su otto incontri, di tre ore ciascuno realizzati rispettivamente con due gruppi di quindici studenti e studentesse di classi 4^e e 5^e primaria e di classi 1^a e 2^a di scuola secondaria di primo grado.

Nel corso delle lezioni i diversi temi e le diverse attività proposte sono stati sempre ricondotti all'idea di simulare, di osservare e svolgere le attività dal punto di vista di un *server* ovvero dal dietro le quinte della rete; nel gruppo di scuola primaria, inoltre, alcune attività sono state proposte in formato analogico per facilitarne la comprensione.

La prima lezione ha utilizzato il comando *tracert* (riga di comando) per proporre agli studenti di mappare le proprie esperienze online distinguendo collegamenti fisici, *server*, IP e browser e instradamento per rendere visibili gli hardware e software dietro un clic.

La seconda lezione, individuata l'infrastruttura della rete e il percorso dietro i servizi online, si è concentrata sul concetto di dato in rete, sul significato di identità digitale e su processi di raccolta dei dati all'interno dei servizi di rete.

Nella terza lezione si è affrontato il tema della sicurezza online attraverso: la password, l'autenticazione a due fattori e la lettura critica delle policy di alcuni servizi utilizzati dagli studenti (*gaming*).

La quarta lezione ha introdotto il concetto di algoritmo e di algoritmo in rete per osservare il comportamento nei siti web in particolare attraverso la profilazione e i cookie.

Nella quinta lezione si è proposto di analizzare i servizi online dal punto di vista economico, osservando quei meccanismi, in particolare nell'area del *gaming*, alla base del marketing.

La sesta lezione si è concentrata sull'informazione in rete proponendo di progettare fake news per comprenderne gli elementi distintivi e la dimensione umana che ne facilita la circolazione.

Le ultime due lezioni sono state, infine, dedicate ai confronti sul percorso sviluppato con particolare riferimento a ciò che ha spinto gli studenti a modificare alcune proprie abitudini online così da poterne trarre elementi narrativi nella progettazione realizzazione di un podcast finale.

Il podcast finale, progettato e realizzato dagli studenti, racconta il percorso lasciando emergere alcune raccomandazioni sull'utilizzo competente e sicuro del digitale rivolto ai genitori (primaria) e pari (scuola secondaria di I grado).

Il percorso proposto ha identificato specifici obiettivi di apprendimento sperimentando un'integrazione tra il *DigComp 2.2* e la *Proposta di Indicazioni Nazionali per l'insegnamento dell'Informatica nella Scuola* (CINI) indicata sinteticamente nella tabella.

	CINI	DigComp 2.2
	Ambito consapevolezza digitale O-P5-N-2	Area 1 - Alfabetizzazione su informazione e dati
Primaria	Ambito creatività digitale O-P3-N-1 Traguardi per lo sviluppo delle competenze T-P- 5 e 8	1.1 Area 2 - Comunicazione e collaborazione 2.2 e 2.5 Area 3 – Creazione di contenuti digitali
Secondaria di I grado	Traguardi per lo sviluppo delle competenze T-M-12. Ambito consapevolezza digitale O-M-N- 1 e 5	3.1-3.2-3.3-3.4 Area 4 Sicurezza 4.1-4.2 Area 5 Risolvere problemi 5.1-5.3

Tabella 1: Integrazione obiettivi di apprendimento

Un esempio di integrazione nel percorso è il tema della condivisione dei dati personali; in un primo momento, dal punto di vista informatico, ci si è chiesti: come viaggia il mio dato nella rete? Perché dovrei perderne il controllo?

Successivamente si è proseguito il lavoro sulle competenze di cittadinanza digitale come la privacy per poi riprendere le competenze informatiche nella gestione dei sistemi di sicurezza dal punto di vista informatico generando un circuito virtuoso in grado di unire competenze informatiche e cittadinanza digitale.

È stato somministrato un questionario all'avvio e al termine del percorso che ha mostrato un miglioramento delle competenze degli studenti sugli specifici temi delle lezioni.

4. Le prospettive

Il Server propone di considerare come prioritario, in questa opportunità di ricostruzione dell'impianto disciplinare della scuola del primo ciclo, il disegnare prospettive didattiche in grado di fornire a studenti e studentesse un quadro univoco della tecnologia, delle sue caratteristiche, del suo utilizzo.

In questa direzione l'obiettivo tracciato dal Manifesto di Vienna [7] nel 2019 resta un faro per chi si occupa di informatica e per chi si occupa di educazione digitale:

“L'educazione all'informatica e al suo impatto sociale devono iniziare il prima possibile. Gli studenti dovrebbero imparare a unire le competenze informatiche con la consapevolezza delle questioni etiche e sociali in gioco”

* Le risorse utilizzate nel laboratorio sono accessibili: <https://drive.google.com/file/d/1dsdi-mo8rxobh3astcxn9lv57z1kdudv/view?usp=sharing>

Bibliografia

- [1] MIM “Nuove indicazioni per la scuola dell'infanzia e primo ciclo di istruzione 2025”[Online]. Available: <https://www.mim.gov.it/-/pubblicato-il-testo-delle-nuove-indicazioni-per-la-scuola-dell-infanzia-e-primo-ciclo-di-istruzione-materiali-per-il-dibattito-pubblico>. [Accessed Maggio 2025]
- [2] Consorzio CINI “Proposta di Indicazioni Nazionali per l'insegnamento dell'Informatica nella Scuola.” 2017 [Online]. Available: <https://www.consorzio-cini.it/images/Proposta-Indicazioni- Nazionali-Informatica-Scuola-numerata.pdf> [Accessed Maggio 2025]
- [3] Commissione Europea (proposta del Consiglio) “Raccomandazione sul miglioramento dell'offerta di competenze digitali nell'istruzione e nella formazione” – 2023. [Online]. Available: <https://eur-lex.europa.eu/legal-content/IT/TXT/PDF/?uri=CELEX:52023DC0206>
- [4] MIM Osservatorio Scuola Digitale “Report Osservatorio Scuola Digitale (2022-2023)”. 2024. [Online]. Available: <https://scuoladigitale.istruzione.it/pnsd/osservatorio-scuola-digitale/> [Accessed Maggio 2025]
- [5] ISTAT “Le competenze digitali dei cittadini (2023)” 2024 [Online]. Available: <https://www.istat.it/comunicato-stampa/le-competenze-digitali-dei-cittadini-anno-2023/> [Accessed Maggio 2025]
- [6] “Digital Competence Framework for Citizens 2.2” 2022 [Online]. Available: <https://repubblicadigitale.gov.it/portale/-/da-oggi-il-digcomp-2.2-parla-italiano> [Accessed Maggio 2025]
- [7] AAVV Manifesto di Vienna 2019. Versione italiana a cura di Enrico Nardelli, Marco Aiello, Carlo Ghezzi, Francesco Ricci, Viola Schiaffonati, Oliviero Stock, Guglielmo Tamburrini. [Online]. Available: https://dighum.ec.tuwien.ac.at/wpcontent/uploads/2019/07/Vienna_Manifesto_on_Digital_Humanism_IT.pdf [Accessed Maggio 2025]

Il potenziamento di tecnologia per promuovere la didattica dell'informatica

Nicola Mendolia¹

¹Istituto Comprensivo Statale "Ignazio Buttitta" – Bagheria (PA)

nicola.mendolia@icsbuttitta.it

Abstract

Il contributo propone una riflessione sulle modalità di introduzione della didattica dell'informatica nella Scuola secondaria di primo grado, alla luce delle Nuove Indicazioni Nazionali per il primo ciclo di istruzione. Si tratta di un importante obiettivo, che si scontra con le difficoltà della scuola pubblica nel Mezzogiorno, dove il tempo prolungato rimane un miraggio. La nostra Scuola opera a Bagheria, in un contesto con elevato livello di povertà educativa, anche nella competenza digitale. Come sviluppare una formazione dell'informatica senza risorse e tempi aggiuntivi? Questo contributo racconta l'introduzione della didattica dell'informatica sfruttando l'opportunità di una cattedra di potenziamento di tecnologia.

1. Il potenziamento di tecnologia per l'introduzione dell'informatica nella scuola secondaria di primo grado

La L.n.107/2015 ha introdotto il concetto di “*potenziamento*”, con l'obiettivo di migliorare le competenze degli studenti, compresa quella digitale [8]. L'istituto scolastico “I. Buttitta” ha avuto assegnata una cattedra di potenziamento di tecnologia per la Scuola secondaria di primo grado. A partire dall'a.s. 2020-2021 è stato avviato un progetto sperimentale di ampliamento dell'offerta formativa sulla competenza digitale, utilizzando le 18 ore del potenziamento, distribuendole fra tre insegnanti, per realizzare due significative innovazioni:

1. l'istituzione di una sezione “*ad orientamento digitale*”, con un percorso strutturato per la didattica dell'informatica e la *media education*, caratterizzato dall'aggiunta di un laboratorio di potenziamento pomeridiano, per tutto il triennio;
2. l'organizzazione, per le restanti sezioni della scuola secondaria, di *laboratori di potenziamento per l'alfabetizzazione digitale e informatica*.

1.1 La sezione *ad orientamento digitale*

La sezione *ad orientamento digitale* [4], usufruisce per tutto il triennio di due ore aggiuntive curriculari per un laboratorio di informatica pomeridiano, con la seguente articolazione didattica:

- il primo anno le attività del laboratorio riguardano la formazione informatica di base e lo sviluppo delle competenze iniziali per un uso consapevole degli strumenti di comunicazione digitale; si introducono gli elementi basilari dell'informatica, mediante il coding, con la

realizzazione di prodotti finali, tra i quali, l'elaborazione/modifica di un semplice videogioco (Fig. 1);

- la classe seconda si dedica alla robotica educativa, mettendo in campo le competenze informatiche acquisite con il coding per la programmazione di prototipi reali;
- la classe terza svolge un corso di modellazione tridimensionale con software con programmazione a blocchi, che consente di realizzare semplici manufatti con la stampante 3D, sviluppando la creatività digitale.

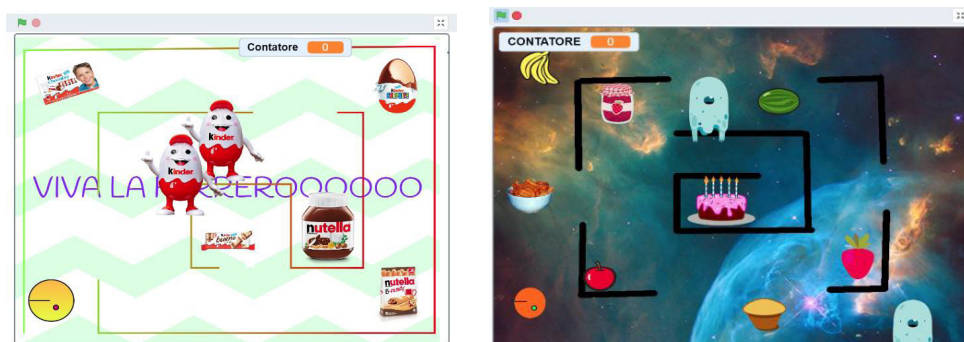


Figura 1: Rielaborazioni di un semplice videogioco ispirato a PacMan

1.2 I laboratori di potenziamento dell'informatica nelle altre sezioni della scuola secondaria di primo grado

Per gli altri otto corsi della scuola secondaria sono stati attivati laboratori di potenziamento pomeridiani [8], della durata di 20/30 ore ciascuno, articolati su due principali linee di azione:

- l'alfabetizzazione digitale per tutti gli alunni del primo anno;
- lo sviluppo del pensiero computazionale per gli ultimi due anni, con la proposizione di corsi di coding e robotica educativa.

I laboratori di alfabetizzazione digitale hanno riguardato sia le competenze digitali di base, sia il coding. Riguardo allo sviluppo delle competenze del pensiero computazionale, si è operato con percorsi più flessibili e limitati, in base alle adesioni manifestate dagli studenti. Quando è stato possibile, sono state sfruttate ulteriori risorse (PNSD, PON, PNRR) per ampliare ulteriormente l'offerta formativa sul digitale.

2. Aspetti metodologici

I percorsi di potenziamento sono stati strutturati su:

- apprendimento sperimentale e prevalentemente induttivo;
- realizzazione pratica di prodotti significativi;
- utilizzo di processi didattici di *debugging* e apprendimento per errori.

È ormai assodato che la costruzione della conoscenza diventa più efficace in un contesto dove il soggetto che apprende è impegnato nella costruzione di prodotti significativi, oltre che nella mera trasmissione di conoscenze, soprattutto nel primo ciclo di istruzione: si tratta di lavorare su “*artefatti cognitivi*”, oggetti o dispositivi che facilitano l'apprendimento nell'esperienza [1]. Si è scelto, inoltre, di lavorare sull'errore, inteso come opportunità per stimolare l'apprendimento in modo creativo, dalla ricerca e definizione dell'incongruenza, verso la sua soluzione, procedendo per successivi aggiustamenti. Con tale stratagemma si è provato a sviluppare negli studenti lo spirito di osservazione personale ed una riflessione più critica, stimolando il *problem solving*, evitando il rischio di un approccio ripetitivo nella semplice messa in pratica delle istruzioni di programmazione/montaggio.

Gli studenti, in assetto per piccoli gruppi, sono stati guidati nella “*caccia all'errore*”, per individuare incongruenze di montaggio, programmazione o funzionamento, elaborando soluzioni migliorative (Fig. 2).



Figura 2: Ambiti di risoluzione degli errori nel processo costruttivo della robotica

Di seguito alcuni esempi di applicazione della “*caccia all'errore*”.

- **Errori nelle istruzioni di montaggio**, per esempio individuando i pezzi mancanti di un prototipo di stampante xy (su foglio bidimensionale), che non funziona bene in quanto il foglio non avanza e la testina non si muove correttamente; gli alunni, una volta montato il modello, dovranno individuare, analizzando degli schemi di funzionamento, quali sono i meccanismi incompleti e inserire, per tentativi, i pezzi mancanti, pervenendo come risultato al corretto funzionamento (Fig. 3).

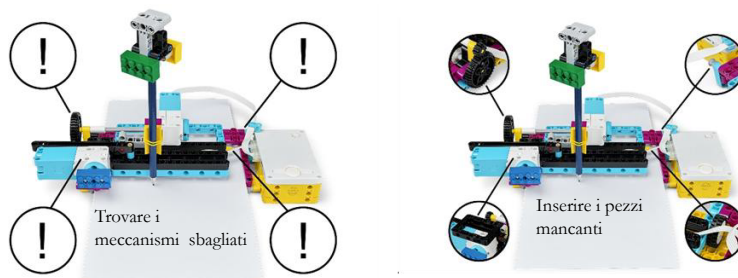


Figura 3: Ricerca degli errori di montaggio di una stampante xy

- **Errori nella programmazione**, in quanto il codice suggerito dal docente non è corretto e bisogna procedere al *debug*; per esempio, il codice per i sensori di distanza di un veicolo è errato e questo non si arresta correttamente davanti un ostacolo o non esegue correttamente le manovre. Gli alunni devono individuare quali sono i parametri di codice errati, schematizzando l'algoritmo e correggendo per tentativi gli errori, agendo sulla distanza per l'attivazione del sensore a infrarossi. Il risultato sarà il corretto movimento del mezzo (Fig. 4).

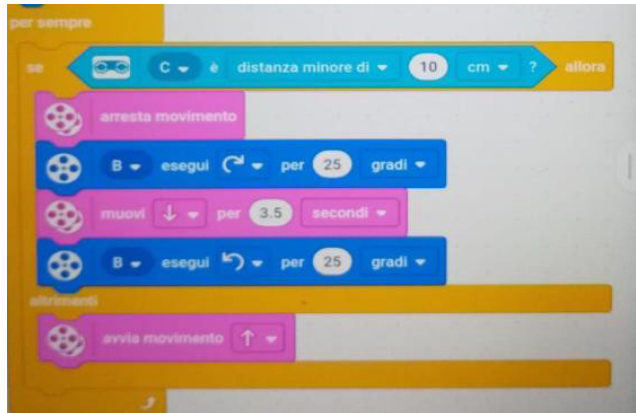
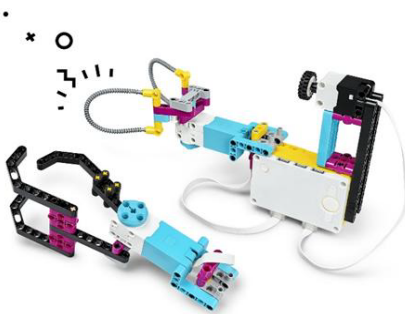
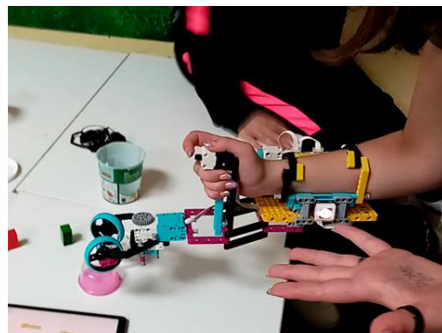


Figura 4: Ricerca degli errori di programmazione dei sensori di distanza di un veicolo

- **Revisione del modello realizzato**, in quanto non rispondente agli obiettivi prefissati; per esempio, dopo il montaggio di un braccio meccanico, vengono proposte diverse soluzioni di presa meccanica e con l'ausilio di schede didattiche con check list vanno eseguite diverse prove sulla presa di differenti oggetti. Il gruppo dovrà individuare la scelta più idonea e proporre una miglioria costruttiva al modello iniziale (Fig. 5).



Analisi e valutazione



Scelta e montaggio

Figura 5: Riprogettazione del modello di un braccio meccanico

3. Analisi dell'esperienza e obiettivi di miglioramento nel primo ciclo di istruzione

Dopo cinque anni di sperimentazione, è possibile formulare un bilancio dell'esperienza, per rivederne alcuni aspetti, in riferimento alle Nuove Indicazioni Nazionali [6] per l'informatica. Si possono individuare tre livelli di riflessione:

1. L'appropriatezza della metodologia;
2. le modalità di valutazione;
3. L'articolazione della didattica.

La metodologia adottata per il potenziamento si è basata sull'apprendimento laboratoriale, per cui, per esempio, il concetto di *istruzione condizionale* o di *ciclo* è stato prima sperimentato empiricamente in un esercizio di programmazione o in un laboratorio di robotica e solo successivamente individuato come categoria della programmazione. Tale approccio induttivo è stato efficace per gli alunni, per i quali, spesso, si è trattato della prima esperienza di didattica dell'informatica.

Nell'ottica dell'introduzione di un curriculum digitale verticale della scuola [2], che declini sin dalla scuola primaria gli obiettivi di apprendimento e le competenze attese per l'informatica [5], sarà opportuno ridefinire i contenuti anche per la scuola secondaria, con un approccio più strutturato, non solo sperimentale, integrando i laboratori di informatica pomeridiani con ulteriori ore nell'orario curriculare delle discipline STEM, in cui sviluppare meglio l'aspetto teorico.

Dal punto di vista della valutazione, fino ad ora si è proceduto in modo empirico rispetto al raggiungimento degli specifici obiettivi prefissati per i singoli laboratori. L'introduzione delle Nuove Indicazioni Nazionali [6] impone la definizione di specifiche griglie oggettive di valutazione, con riguardo agli obiettivi di apprendimento dell'informatica, tra cui, in particolare:

- ambito algoritmi: livelli di padronanza nella formulazione di algoritmi per semplici processi e capacità di rilevarne le possibili ambiguità, individuandone gli errori;
- ambito programmazione: livelli di padronanza delle funzioni base dell'informatica, quali sequenze, cicli, istruzioni condizionali, variabili;

L'articolazione della didattica è stata limitata dalla effettiva disponibilità oraria del potenziamento e di eventuali implementazioni con altre risorse, dando sempre precedenza alla copertura del fabbisogno dei laboratori della sezione ad orientamento digitale (6h settimanali). Riguardo alle restanti ore destinate ai laboratori di potenziamento di informatica (12h settimanali), queste non sempre sono state sufficienti per il raggiungimento degli obiettivi prefissati. Per esempio, per i laboratori del primo anno, quando si è presentata l'opportunità di sfruttare ulteriori risorse, si è arrivati a coprire la formazione di tutti gli alunni, con migliori risultati anche in termini di intensità della frequenza e di successo formativo, grazie alla presenza della figura del tutor. In altri casi, senza risorse aggiuntive o a causa di tagli dell'orario del potenziamento, la platea è stata ridimensionata.

Nella prospettiva di una maggiore strutturazione della didattica dell'informatica nel curriculum scolastico per tutte le sezioni della scuola secondaria, appare necessario che essa non sia più limitata al laboratorio pomeridiano, ma è importante che questa si integri nell'orario curriculare delle discipline STEM, in modo da rendere più stabili i percorsi di formazione, mediante la progettazione di UdA interdisciplinari dell'area STEM, articolate in una parte più teorica, in orario curriculare e una parte più sperimentale, nei laboratori di potenziamento pomeridiani.

- Gli obiettivi di miglioramento possono, dunque, sintetizzarsi in due aspetti:
- rendere **più interdisciplinare l'approccio alla didattica dell'informatica**, in conformità alle Linee Guida per le discipline STEM [7], progettando specifiche **UdA per l'informatica dell'area STEM**, che comprendano sia ore curriculari, che laboratori di potenziamento;
- sistematizzare in un **nuovo curriculum digitale verticale** il percorso della didattica dell'informatica per tutto il primo ciclo di istruzione, sulla base delle buone pratiche laboratoriali sin qui sviluppate, in un'ottica più strutturata, conformemente alle Nuove Indicazioni Nazionali [6].

Bibliografia

- [1] Papert S., *Mindstorms: Children, Computers, and Powerful Ideas*. New York: Basic books, 1980
- [2] Di Stasio M., Miotti B., Nulli G. *Robotica educativa e coding: strumenti per la trasformazione del curriculum*. 2022
- [3] Gentile E., Monga M. *Atti del primo convegno italiano sulla didattica dell'informatica*, Università degli Studi di Bari, 2023.
- [4] Domenica Roberta Mistretta. "Sperimentazione di un curriculum digitale", in *Atti del primo convegno italiano sulla didattica dell'informatica*, pagg. da 228 a 237, Università degli Studi di Bari, 2023.
- [5] Consorzio Interuniversitario Nazionale per l'Informatica. *Proposta di indicazioni nazionali per l'insegnamento dell'informatica nella scuola*. Roma, 2017
- [6] Ministero dell'Istruzione e del Merito. *Proposta di Nuove Indicazioni Nazionali 2025 per la Scuola dell'infanzia e il Primo ciclo di istruzione*. Roma, 2025
- [7] Ministero dell'Istruzione e del Merito. *Linee guida per le discipline STEM*, Allegato 1 al Decreto Ministeriale n.184. Roma, 2023
- [8] Legge 13 luglio 2015, n.107, *Riforma del sistema nazionale di istruzione*, art.1, comma 7: iniziative di potenziamento dell'offerta formativa, *Gazzetta Ufficiale* n.162, 15-07-2015.

Coding e creatività per lo sviluppo del pensiero computazionale nel primo ciclo: un'esperienza didattica all'IC Praia a Mare

Emilia Ciccia¹, Teresa Agrello¹, Elena Castiglioni¹, Rosangela Papa¹, Marilena Salsano¹

¹ Istituto Comprensivo Praia a Mare

emilia.ciccia@scuola.istruzione.it *teresa.agrello@icpraia.edu.it* *elena.castiglioni@icpraia.edu.it*

rosangela.papa@icpraia.edu.it *csic8au004@istruzione.it*

Abstract

Il percorso di coding sviluppato presso l'Istituto Comprensivo Praia a Mare nell'ambito del D.M. 65/2023 del PNRR ha rappresentato un'esperienza didattica innovativa e coinvolgente. Il progetto si è ispirato al modello "Coding in your classroom, now!" di Alessandro Bogliolo, con un approccio che unisce il coding unplugged e attività digitali visuali, ponendo al centro l'acquisizione del pensiero computazionale e la creatività degli studenti.

Lungo tutto il percorso, sono stati affrontati i contenuti propri della disciplina Informatica, come la comprensione delle strutture algoritmiche (sequenze, cicli, condizioni), l'uso consapevole di ambienti di programmazione a blocchi come Scratch, l'introduzione a concetti di variabili, input/output e debugging. Il progetto ha fatto riferimento agli obiettivi e traguardi di apprendimento delineati nella Proposta di Indicazioni Nazionali per l'insegnamento dell'Informatica nella scuola elaborata dal gruppo CIN.

Significativa è stata l'attività multidisciplinare in cui gli studenti hanno sperimentato forme alternative di comunicazione attraverso codici visivi. Utilizzando i pentamini, figure geometriche costituite da cinque quadrati uniti tra loro, gli alunni hanno inventato parole e costruito un vero e proprio alfabeto simbolico. Questa attività ha permesso di esplorare il valore comunicativo del codice, stimolando allo stesso tempo logica, creatività e collaborazione.

Il progetto ha favorito l'acquisizione di numerose competenze trasversali: capacità di risolvere problemi, lavoro in team, consapevolezza digitale, autonomia e metacognizione. Attraverso attività laboratoriali e una didattica inclusiva e centrata sullo studente, il percorso ha valorizzato l'espressività e la curiosità, rendendo l'informatica accessibile a tutti. La documentazione e i prodotti finali sono stati pubblicati sul sito dell'istituto, consolidando l'esperienza come buona pratica replicabile.

1. Introduzione

Nell'ambito del PNRR – D.M. 65/2023, l'Istituto Comprensivo di Praia a Mare ha avviato un articolato percorso laboratoriale di coding e pensiero computazionale destinato agli alunni della scuola primaria e secondaria di primo grado. Il progetto si è svolto nei plessi "Falcone" di Via Galilei e Via Manzoni, ed è parte dell'investimento previsto nella Missione 4 del PNRR – "Istruzione e ricerca" –, dedicato allo sviluppo di competenze STEM, digitali e di innovazione.

L'intero percorso è stato progettato e realizzato secondo il modello didattico proposto da Alessandro Bogliolo (2016; 2020) [1,2]: un approccio che valorizza il coding unplugged come porta d'ingresso alla comprensione dei meccanismi fondamentali dell'informatica e del linguaggio computazionale, rendendoli accessibili anche ai più piccoli, e stimolando il pensiero logico attraverso attività inclusive e creative. Tale modello contribuisce anche al potenziamento delle life skills, così come indicate dal bollettino OMS Skills for life n. 1(1992), e cioè tutte quelle abilità e competenze da apprendere per sapersi mettere in relazione con gli altri e per affrontare anche i problemi, le pressioni e gli stress della vita quotidiana. Il bollettino evidenzia anche che l'acquisizione di queste abilità deve essere favorita attraverso l'introduzione di specifici programmi nelle scuole o in altri luoghi deputati all'apprendimento. Anche il Piano Nazionale di Ripresa e Resilienza (PNRR), nel riconoscere comunque fra le cause di abbandono scolastico la mancata acquisizione delle competenze di base, suggerisce «attività per il potenziamento delle competenze trasversali [...] soprattutto del primo ciclo di istruzione».

Il progetto Coding dell'IC Praia è stato pensato proprio come programma per facilitare, oltre a competenze specifiche sul pensiero computazionale, scientifico e matematico, anche le abilità che vengono definite non cognitive e che sono state normate dalla Legge 19 febbraio 2025, n. 22.

Il documento dell'OMS così le definisce:

- **Decision making.** È la capacità di prendere decisioni consapevoli e costruttive nelle diverse situazioni della vita quotidiana, valutando alternative e conseguenze, e scegliendo in modo responsabile.
- **Problem solving.** Competenza che consente di affrontare in maniera efficace le problematiche della vita, analizzandole e trovando soluzioni adeguate. La capacità di risolvere i problemi riduce il rischio di stress e tensioni derivanti da difficoltà non gestite.
- **Creatività.** Abilità che contribuisce sia al decision making sia al problem solving, poiché consente di esplorare nuove idee, alternative e strategie innovative per affrontare le situazioni in modo flessibile e adattivo.
- **Pensiero critico (critical thinking).** Capacità di analizzare informazioni, situazioni ed esperienze in modo oggettivo e riflessivo, valutando vantaggi e svantaggi delle diverse opzioni disponibili. Questa competenza favorisce la presa di decisioni consapevoli e autonome, e supporta la promozione della salute, aiutando a riconoscere l'influenza di fattori esterni come le pressioni sociali o i mass media sui propri atteggiamenti e comportamenti.
- **Comunicazione efficace.** Comprende la capacità di esprimersi in modo chiaro e appropriato, sia verbalmente che non verbalmente, rispettando la cultura di riferimento e il contesto. Significa anche saper ascoltare attivamente, esprimere opinioni, bisogni, sentimenti e richieste di aiuto in modo congruo.
- **Relazioni interpersonali.** Abilità di interagire con gli altri in maniera positiva, costruendo e mantenendo rapporti significativi, necessari per il benessere psicosociale. Include anche la capacità di concludere relazioni in modo costruttivo, laddove necessario.
- **Autocoscienza (self-awareness).** Conoscenza di sé, del proprio carattere, dei punti di forza e di debolezza, dei desideri e dei bisogni personali. Costituisce un prerequisito fondamentale per una comunicazione efficace, per relazioni sane e per la comprensione empatica degli altri.
- **Empatia.** Capacità di comprendere sentimenti, emozioni e punti di vista altrui, anche quando diversi dai propri o provenienti da contesti non familiari. Favorisce relazioni sociali rispettose e l'accettazione della diversità etnica e culturale, promuovendo disponibilità

all'aiuto e alla cura. Gestione delle emozioni. Competenza che permette di riconoscere e comprendere le proprie emozioni e quelle degli altri, valutare l'impatto che hanno sui comportamenti e gestirle in modo appropriato rispetto al contesto.

- **Gestione dello stress.** Capacità di identificare le fonti di stress quotidiano e di controllarle attraverso modifiche nello stile di vita, nell'ambiente o tramite tecniche di rilassamento, in modo da prevenire ripercussioni negative sulla salute psico-fisica.

Queste abilità confluiscono in tre aree principali:

1. Area cognitiva (decision making e problem solving, pensiero critico e pensiero creativo);
2. Area relazionale (Capacità di comunicazione e di gestione delle relazioni interpersonali, auto-consapevolezza, empatia);
3. Area emotiva (capacità di gestire lo stress e le emozioni).

Si evince un nucleo comune che sostiene queste tre aree, «un sistema interattivo che contribuisce allo sviluppo e alla crescita della persona in modo sinergico e complementare» (Documento AIP sulla proposta di legge sulle competenze non cognitive nella scuola, 2022).

Si tratta di elementi connessi alla dimensione cognitiva, affettivo-emozionale e sociale e, come afferma Cottini (2022), «non sono scindibili dal pensare, operare e partecipare». Queste competenze, quindi, descrivono la capacità di usare conoscenze e abilità in modo efficace rispetto a compiti e contesti che possono risultare anche complessi, al fine di affrontarli e risolvere i problemi che possono presentarsi. Ricorrere allora alla definizione di “non cognitive”, riferendosi a queste abilità, può risultare fuorviante. La scuola, infatti, offre esperienze di apprendimento in chiave cognitiva ma, allo stesso tempo, promuove abilità sociali ed emotive; i docenti, nell'ambito del loro lavoro di istruzione ed educazione, stimolano metacognizione, autovalutazione, motivazione, autoefficacia, affermatività, abilità di comunicazione, capacità di affrontare le prove valutative senza stress, in definitiva quelle che il Ministero dell'Istruzione, all'interno del rapporto tecnico allegato al D.M. 139 del 2007, definisce “competenze chiave di cittadinanza” e che le Linee guida Percorsi per le Competenze trasversali e l'Orientamento (PCTO) del 2019 identificano come soft skills, definendole come «competenze trasversali e trasferibili attraverso la dimensione operativa del fare: capacità di interagire e lavorare con gli altri, capacità di risoluzione di problemi, creatività, pensiero critico consapevolezza e capacità di individuare le forme di orientamento e sostegno disponibili per affrontare la complessità e l'incertezza dei cambiamenti, preparandosi alla natura mutante dell'economia moderne e delle società complesse». Secondo l'Invalsi esse sarebbero complementari alle hard skills, riferendosi con questa definizione alle competenze tecnico-specialistiche che riguardano quanto si sa e come si fa; al contrario, le soft skills interesserebbero il chi si è [4].

Poiché alcune delle life skills contribuiscono ad incrementare il profitto scolastico e a ridurre dispersione e povertà educativa (con ricadute significative sulla qualità dell'apprendimento), allora possono essere educate trasversalmente attraverso la didattica delle abilità tecniche. Il loro insegnamento è di natura pragmatica e si basa sull'apprendimento per esperienza, sulla partecipazione attiva e sull'assunzione diretta di responsabilità; cooperative learning, peer tutoring, role playing, brainstorming e discussioni sono solo alcune delle metodologie che le favoriscono [5]. L'apprendimento è un'acquisizione attiva, che avviene attraverso la trasformazione e la strutturazione dell'esperienza (Bandura, 1977); il documento OMS precedentemente menzionato raccomanda la messa in pratica di metodologie attive ed esperienziali per la promozione delle life skills.

Anche se molte soft skills possono sembrare compiti e sfide tipici degli individui adulti, già in età precoce è possibile osservare l'emergenza di tali competenze e, di conseguenza, è possibile promuoverle già a partire dalla scuola dell'infanzia. Esiste sicuramente una elevata variabilità interindividuale, ma non è insolito riscontrare in bambini molto piccoli capacità di autonomia nel portare a termine un'attività senza la supervisione di un adulto, flessibilità cognitiva quando riescono a modificare le risposte in funzione delle richieste e dei contesti, creatività attraverso l'elaborazione di prodotti non attesi, ma anche intelligenza emotiva. I repentini passaggi fra didattica in presenza e a distanza nel periodo pandemico, per esempio, hanno richiesto elevate capacità di flessibilità e di adattamento da parte dei bambini, e questo fa capire quanto sia fuorviante definirle "abilità non cognitive", dal momento che le soft skills si fondano su quell'insieme di funzioni cognitive di controllo e regolazione dei processi mentali che costituiscono le Funzioni Esecutive (da ora FE) [6].

Oltre a FE di base, che comprendono inibizione, memoria di lavoro e flessibilità cognitiva, vi sono componenti di alto livello, funzioni più complesse che favoriscono la pianificazione, il problem solving e il ragionamento [7]. È evidente la sovrapposizione con le componenti delle soft skills e quanto queste ultime siano influenzate dalle FE: nel processo di negoziazione, ad esempio, un buon compromesso fra le parti si può raggiungere solo grazie alla capacità di inibire la tendenza a soddisfare i propri impulsi, se si è in grado di mantenere in memoria le diverse versioni della questione e se si ha la capacità di spostare l'attenzione sul punto di vista dell'altro.

Inoltre, le FE sono coinvolte nelle relazioni sociali e di regolazione emotiva; bassi livelli di competenze esecutive in età prescolare sono risultate altamente predittive di atti di bullismo in età scolare e di disregolazione dei comportamenti [8,9]

Le FE, quindi, promuovono e sostengono molte delle life o soft skills fondamentali per l'apprendimento che, a loro volta, possono essere stimulate e potenziate dalla scuola stessa attraverso attività specifiche.

Fra le attività di riconosciuta efficacia, l'informatica e l'educazione al pensiero computazionale attraverso il coding dimostrano un impatto notevole sugli studenti e sui loro apprendimenti per una serie di ragioni: utilizzano un linguaggio che è molto vicino alla generazione dei nativi digitali, assecondano naturalmente i diversi stili di apprendimento, stimolano l'interesse incrementando la motivazione e la perseveranza, inducono padronanza e senso di autoefficacia, potenziano le abilità relazionali, soprattutto quando le attività richiedono un'esecuzione partecipata e collaborativa.

Con il termine coding si indica l'utilizzo di strumenti e metodi di programmazione visuale a blocchi per favorire lo sviluppo del pensiero computazionale, ossia la capacità di individuare un procedimento costruttivo, costituito di passaggi non ambigui che aiutano la risoluzione di un problema complesso [2]. Sono evidenti, quindi, ricadute significative sulle abilità di problem solving e su quelle che implicano la creatività, e questo perché i passaggi, seppur posti in maniera algoritmica, non sono determinati e rigidi, ma possono essere condotti in maniera anche originale e imprevedibile. Indirettamente mette in gioco anche l'autoconsapevolezza e l'autocoscienza perché diventare consapevoli dell'esistenza di diverse categorie di problemi, problema step by step o problema di insight, permette di capire come affrontarli e trovare la migliore strategia di soluzione. Mentre nei problemi step by step si procede in maniera sequenziale, con un progressivo avvicinamento alla risposta, in quelli per insight bisogna assumere una prospettiva diversa e immaginare operazioni innovative. Per questo motivo, il problema per insight è strettamente collegato alla creatività. Essere consapevoli del tipo di problema da risolvere aiuta a prevedere la risoluzione, è la meta-capacità alla base del problem solving

[5]. Se il bambino approccia un problema con un atteggiamento non pertinente, perché non ne riconosce la natura, rischia di procedere per tentativi ed errori e finire in un vicolo cieco, senza possibilità di rivedere i passaggi e correggere gli errori commessi, in altre parole, senza possibilità di effettuare un debugging. Affrontare un percorso di coding, sia esso unplugged o con il supporto di devices, implica saper identificare il problema, analizzarlo individuandone gli obiettivi, riuscire a produrre alternative di soluzione per poi scegliere e applicare quella migliore, dimostrando anche abilità creative e di decision making; inoltre, la verifica dei risultati è immediata.

Il coding può costituire anche un'efficace stimolazione didattica per favorire l'emergere di idee creative, per sviluppare nuovi codici comunicativi ed espressivi, per produrre nuove linee narrative a partire dalla ristrutturazione di elementi dati [10].

2. Il Laboratorio Coding: partecipanti e obiettivi

2.1 Partecipanti

Il laboratorio di Coding si è svolto nell'ambito del D.M. 65/23 per un totale di 20 ore. Hanno partecipato 57 alunni (25 M e 32 F) delle classi terza, quarta e quinta della scuola primaria di tre plessi dell'Istituto Comprensivo di Praia a Mare, Falcone di Via Galilei, Via Verdi e Via Manzoni (Tab. 1). Gli incontri si sono svolti con cadenza regolare in orario extracurricolare e hanno visto un progressivo coinvolgimento degli alunni in attività sempre più articolate, finalizzate all'acquisizione di competenze digitali e al potenziamento del pensiero computazionale.

Distribuzione campione	3 ^a	4 ^a	5 ^a	Totale
Maschi	3	13	9	25
Femmine	4	20	8	32
Totale	7	33	17	57
Età media (DS)	8,2 (0,4)	9,4 (0,5)	10,5 (0,5)	

Tabella 1: descrizione dei partecipanti

Diverse le competenze in entrata degli alunni: la classe terza non aveva ancora sperimentato esperienze di coding, ad eccezione di qualche modalità unplugged; la classe quarta aveva una conoscenza di base della programmazione a blocchi ed aveva cominciato un percorso di informatica sul sito www.code.org, la classe quinta aveva completato l'intero percorso sullo stesso sito e continuava le attività di coding nel laboratorio di informatica della scuola durante le ore della disciplina Tecnologia.

Le attività sono state condotte da quattro docenti, di cui due con funzione di esperto e due con funzione di tutor.

2.2 Obiettivi

Obiettivi Educativi e Didattici

L'intero percorso si è basato su metodologie didattiche attive e laboratoriali, volte a favorire l'apprendimento attraverso il gioco e la sperimentazione. Gli obiettivi principali perseguiti sono stati:

1. *Apprendere i concetti base dell'informatica.* Comandi, istruzioni, sequenze, cicli, condizioni; Utilizzo di strumenti di coding unplugged (senza computer) e coding visuale (Scratch, Blockly, Code.org).
2. *Consolidare competenze disciplinari.* Italiano: costruzione di testi sequenziali, istruzioni, storytelling; Matematica: calcoli, sequenze, logica, geometria (angoli, rotazioni); Geografia/scienze: creare mappe di percorsi o simulazioni; Lingue: utilizzo di comandi in lingua straniera.
3. *Acquisire linguaggi e metodi di programmazione base.* Organizzare istruzioni in modo ordinato e comprensibile; Usare blocchi o script per risolvere problemi o creare prodotti (storie animate, giochi).
4. *Prepararsi al pensiero computazionale avanzato.* Propedeutica al coding testuale e alla robotica educativa; Comprendere i principi della programmazione e della logica.

Life Skills e Competenze Trasversali

Oltre agli aspetti strettamente legati alla disciplina informatica, il progetto ha stimolato negli studenti competenze trasversali fondamentali per la crescita personale e relazionale. Tra queste:

- *Pensiero critico e problem solving.* Ogni esercizio, ogni sfida ha richiesto l'analisi di problemi, l'elaborazione di soluzioni, la capacità di affrontare gli errori come opportunità di miglioramento.
- *Collaborazione e lavoro di gruppo.* Molti progetti sono stati sviluppati in piccoli team, favorendo la condivisione dei compiti, l'ascolto reciproco e il rispetto dei ruoli.
- *Creatività e comunicazione.* La creazione di storie animate e giochi ha stimolato l'inventiva e ha offerto spunti per comunicare contenuti anche di altre discipline, come l'italiano o la matematica.
- *Consapevolezza digitale e cittadinanza attiva.* Nel corso del progetto si è parlato anche di uso etico e responsabile delle tecnologie, promuovendo il rispetto delle regole digitali e la protezione dei dati.
- *Resilienza e gestione dell'errore.* Incoraggiamento a vedere gli errori come opportunità di apprendimento e miglioramento.

Obiettivi DigComp 2.2 e DigicomEdu

Il DigComp 2.2 identifica 5 aree di competenza digitale: 1. Information and data literacy; 2. Communication and collaboration; 3. Digital content creation; 4. Safety; 5. Problem solving.

All'interno di questo framework il Coding rientra principalmente nell'*Area 3: Creazione di contenuti digitali*, in particolare nella competenza 3.4 – *Programmazione (Programming)*: capacità di pianificare e sviluppare una sequenza di comandi o istruzioni per un sistema informatico, un'app o un robot, al fine di eseguire compiti specifici o risolvere problemi.

Il DigCompEdu definisce le competenze digitali dei docenti in 6 aree: 1. Professional Engagement; 2. Digital Resources; 3. Teaching and Learning; 4. Assessment; 5. Empowering Learners; 6. Facilitating Learners' Digital Competence.

All'interno di questo secondo framework il Coding si colloca prevalentemente nella *Area 6*, competenza *6.4 Computational thinking*, che implica: supportare gli studenti nello sviluppo di strategie di risoluzione di problemi complessi attraverso algoritmi, programmazione e pensiero computazionale.

Traguardi di competenze

Seguendo i riferimenti della Proposta di Indicazioni Nazionali per l'insegnamento dell'Informatica nella Scuola, elaborata dal gruppo Informatica e Scuola del CINI, sono stati perseguiti i principali traguardi di sviluppo delle competenze per il primo ciclo, tra cui: riconoscere e utilizzare strutture algoritmiche fondamentali (sequenze, cicli, condizioni); progettare e realizzare procedimenti e semplici algoritmi, con o senza strumenti digitali; riflettere sul funzionamento dei codici, imparando a debuggare e a testare soluzioni; comprendere le relazioni tra dati, operazioni, input/output e interfacce.

3. Metodologie e attività

Il progetto ha consentito agli alunni di avvicinarsi alla disciplina informatica attraverso attività pratiche, graduali e coinvolgenti. La modalità di svolgimento è stata laboratoriale e inclusiva, seguendo un'ottica costruttivista dell'apprendimento. L'uso del coding unplugged ha permesso di coinvolgere anche gli alunni con maggiori difficoltà, grazie ad attività manipolative, visive e corporee. I docenti tutor e gli esperti hanno accompagnato i gruppi in un percorso graduale, basato sulla sperimentazione, sul gioco serio e sulla riflessione metacognitiva.

Le attività sono state documentate anche sul sito della scuola, dove è possibile visionare immagini, descrizioni e materiali didattici prodotti: (<https://www.icpraia.edu.it/pnrr-d-m-65-2023-competenze-stem-emultilinguistiche/coding-d-m-65-2023-pnrr/>).

3.1 Attività Svolte. Fase 1. Avviamento

Il percorso ha previsto un approccio graduale al pensiero computazionale, combinando attività *unplugged* e digitali per favorire la comprensione dei concetti fondamentali della programmazione.

Pixel Art

Il laboratorio è iniziato con la Pixel Art, un'attività che ha permesso agli alunni di comprendere il concetto di rappresentazione digitale delle immagini e di allenare il pensiero logico-sequenziale. È stato chiesto ai bambini di scrivere il proprio nome, o il diminutivo per motivi di spazio, all'interno di un rettangolo quadrettato 10 x 18 colorando i quadrati necessari. Successivamente, gli stessi hanno dovuto ricavare il codice da condividere con i compagni (fig. 1). I bambini di tutte le classi avevano già familiarità con attività di pixel art da eseguire seguendo un codice dato, ma questa modalità li ha indotti a operare una ricerca e di effettuare dei tentativi di soluzione step by step che permettessero loro di completare la richiesta in modo corretto. Questa fase ha favorito la concentrazione, la pazienza e la precisione, competenze trasversali utili in molteplici ambiti.

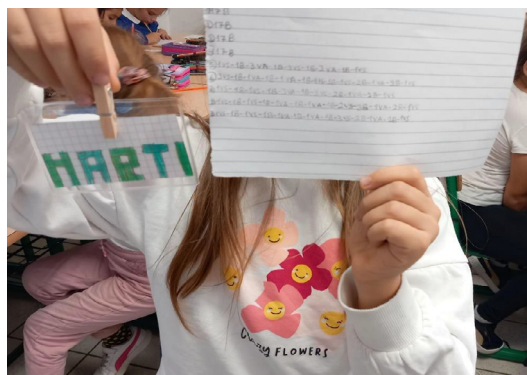


Figura 1: Scrittura del proprio nome come pixel art e condivisione del codice corrispondente.

Per permettere ai bambini della classe terza, meno esperti, di familiarizzare con logica e problem-solving, è stata introdotta un'attività consistente nel risolvere quesiti logici riguardanti la decodifica di sequenze per giungere ad una “parola nascosta”, il completamento di crucinumeri basati su operazioni matematiche, la predisposizione di situazioni problematiche partendo da un semplice racconto. Queste attività, oltre ad esercitare le capacità di problem solving, sono state predisposte secondo una strutturazione di cooperative learning per favorire, altresì, le abilità socio- relazionali (Fig. 2).

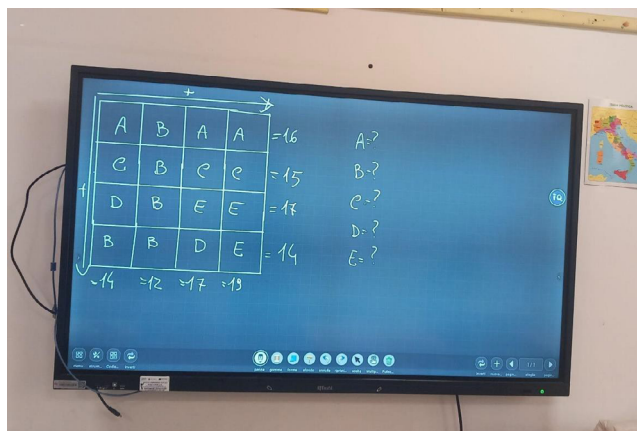


Figura 2: Attività di avviamento al problem solving: completamento di crucinumeri

3.2 Attività Svolte. Fase 2. Percorsi CodyFeet, CodyColor e CodyMaze

Le fasi successive del laboratorio di coding hanno previsto la suddivisione in squadre degli alunni. Le squadre sono state formate secondo criteri di equità, inserendo in ciascuna rappresentanti dei bambini di ogni classe e rispettando il più possibile la proporzione tra maschi

e femmine. Alle squadre così formate sono stati presentati dei percorsi di coding *unplugged* suggeriti dai testi di Alessandro Bogliolo (2016; 2020) [1,2]. Letteralmente il termine *unplugged* significa “senza collegamento alla rete elettrica” ma, poiché al giorno d’oggi la maggior parte dei dispositivi elettronici funziona senza alcun collegamento, ciò che caratterizza davvero questa modalità di coding è il gioco di ruolo, in cui due persone fungono, rispettivamente, da programmatore e da esecutore.

Gli alunni hanno acquisito le regole dei percorsi e i codici trasmessi dalle forme (freccie, quadrati, cerchi, piedi) e dai colori (grigio = avanti; rosso = gira a destra; giallo = gira a sinistra) e hanno prodotto da soli le carte necessarie per giocare (Fig. 3).



Figura 3: Preparazione delle carte CodyFeet

CodyFeet: Gli studenti hanno sperimentato la programmazione attraverso il movimento fisico, utilizzando le carte direzionali per guidare il percorso. CodyFeet è un vero linguaggio di programmazione che utilizza il gioco come modalità di apprendimento. Durante le prime sessioni del laboratorio sono state sperimentate le attività *Passo a due* e *Tiro alla fune*. La prima consiste in una sorta di lotta tra due giocatori di due squadre diverse per restare più a lungo possibile sulla scacchiera. Ogni giocatore riveste il ruolo di Roby (il robottino) che deve essere guidato dal programmatore della propria squadra, Cody, il quale suggerisce i passaggi per arrivare al punto di arrivo con il numero maggiore di comandi. Partendo dai rispettivi punti di partenza, individuati in modo simmetrico sulla scacchiera, ogni giocatore deve disporre a turno le proprie tessere cercando di bloccare la squadra avversaria, intralciandone il percorso con il proprio. La squadra vincitrice è quella che riesce ad aggiungere altre tessere quando l'altra ha la strada bloccata. (Fig. 4)



Figura 4: Attività unplugged Passo a due del CodyFeet.

Nell'attività *Tiro alla fune* le due squadre condividono il punto di partenza (al centro del lato Sud della scacchiera, C5) e il percorso, ma ciascuna cercherà di portarlo dalla propria parte uscendo dalla parte sinistra della scacchiera (A1 o B1) o da quella destra (D1 o E1). (Fig. 5) Queste attività hanno rafforzato la capacità di seguire istruzioni, la pianificazione e la coordinazione spaziale.

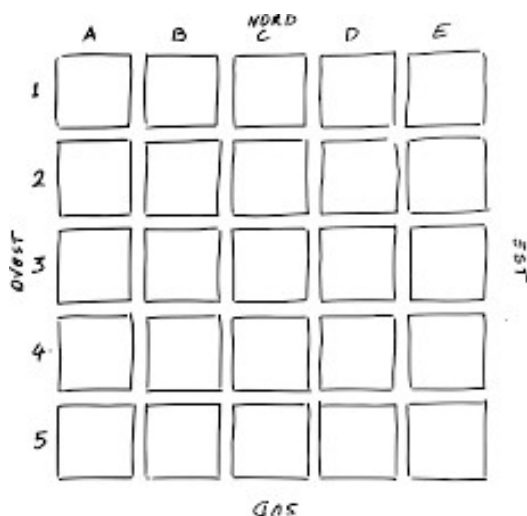


Figura 5: Scacchiera per le attività unplugged.

CodyColor: il repertorio di attività di questo percorso è simile a quello di CodyFeet ma senza impronte o incavi e protuberanze lungo i bordi. Si hanno a disposizione semplici quadrati colorati nei modi convenzionali precedentemente indicati, a cui si aggiungono due tessere che denotano il punto di partenza, un triangolo azzurro, e un punto di arrivo, un cerchio azzurro. Gli alunni hanno realizzato e plastificato un grande numero di tessere che hanno poi utilizzato per le attività. Venti tessere sono state disposte sopra la scacchiera e, sulla base dei colori, i partecipanti si sono mossi per affrontare un percorso verso l'uscita. È stato stabilito un criterio di gioco, e cioè che non fosse possibile procedere step by step, ma solo dopo che il bambino-programmatore aveva già predisposto esternamente alla scacchiera il numero di tessere necessarie per far compiere tutti i passi (Fig. 6). L'attività ha rafforzato la capacità di interpretare ed eseguire sequenze di istruzioni, utilizzando il codice colore per guidare i movimenti. Inoltre, ha favorito il riconoscimento di schemi logici e l'approccio alla programmazione condizionale.



Figura 6: Attività unplugged CodyColor.

CodyMaze: questa attività è stata particolarmente gradita ai bambini poiché veniva introdotta la componente del debugging, e cioè la correzione dell'errore a seguito della scoperta di una casella "esplosiva". È stata creata una griglia sul pavimento in cui le celle contenevano un codice QR che, se scansionato, forniva un'istruzione, un quiz logico o, condizione da evitare, l'immagine disegnata di una bomba che decretava la fine del gioco per la squadra. Lo studente che impersonava il "robot" doveva seguire le istruzioni usando comandi base (es. AVANTI, GIRA A SINISTRA, SE...ALLORA, RIPETI). L'obiettivo era raggiungere la cella finale (target) risolvendo problemi, evitando ostacoli o errori di programmazione (Fig. 7). Attraverso questo strumento, che prevedeva, appunto, tessere con QR Code invece delle immagini, i bambini hanno lavorato su logica, strategia e problem-solving, interagendo con sfide di crescente difficoltà.



Figura 7: Attività unplugged CodyMazy.

3.3 Attività Svolte. Fase 3. Attività Digitali - Code.org

Il percorso si è concluso con Fondamenti di Informatica Corso Rapido sul sito www.code.org. È stata creata una classe virtuale con i nomi degli alunni partecipanti al laboratorio e sono state differenziate le attività in base al livello di competenza e all'età. Gli studenti di quinta, infatti, avevano già affrontato le attività e i livelli del percorso e, quindi, hanno lavorato su quelli più difficili, che comprendevano le funzioni e le istruzioni condizionali, mentre gli alunni non esperti hanno iniziato con la programmazione a blocchi di Angry Bird, del Collezionista, dell'Artista e della Contadina e il labirinto con il debugging. Un aggancio importante con la disciplina Matematica è stato fornito proprio dall'Artista che ha, permesso di realizzare dei disegni utilizzando e personalizzando i blocchi con le misurazioni corrette; in questo modo gli alunni hanno imparato a costruire dei poligoni programmando la giusta ampiezza degli angoli, hanno indirettamente rinforzato una conoscenza di alcuni elementi matematici (cosa sono gli angoli, a quanto corrisponde la somma degli angoli interni di un triangolo, quanto misura un angolo retto, piatto, giro, ecc.) e, attraverso un ragionamento induttivo, sono giunti alla conclusione che la circonferenza è, in realtà, molto simile a un poligono formato da 360 angoli di 1 grado (Fig. 8). Queste attività hanno in qualche modo migliorato le competenze logiche e di pensiero critico degli studenti e li hanno aiutati a capire come applicare principi generali a situazioni specifiche.

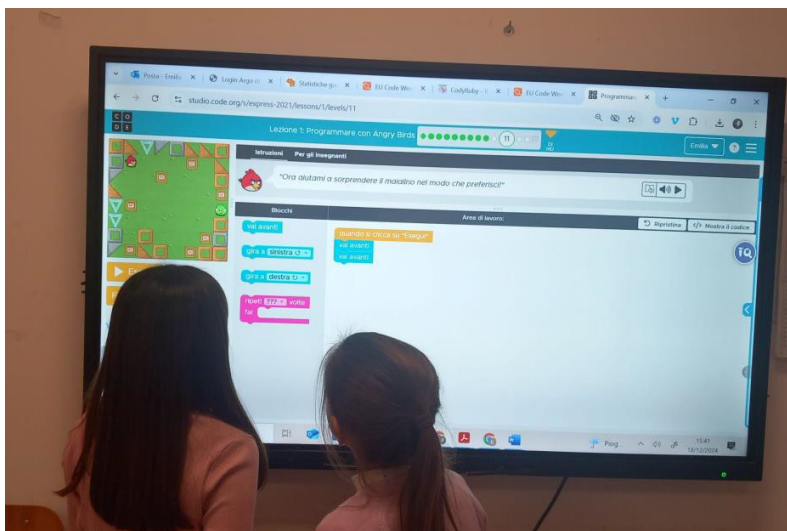


Figura 8: Attività Angry Bird prevista dal corso Fondamenti di Informatica Corso Rapido Code.org.

Gli alunni di quinta, inoltre, hanno realizzato il progetto JamSession nell'ambito dell'Ora del Codice, stimolando la creatività e la collaborazione. Questa attività ha favorito il lavoro di gruppo, la presa di decisioni in autonomia e ha consentito di sperimentare la programmazione con un obiettivo finale concreto e motivante (Fig. 9). L'insegnamento non si è limitato all'uso di strumenti, ma ha favorito una vera e propria alfabetizzazione informatica: non imparare a usare i computer, ma imparare a pensare con il linguaggio dei computer, come sottolinea la visione didattica di Bogliolo.

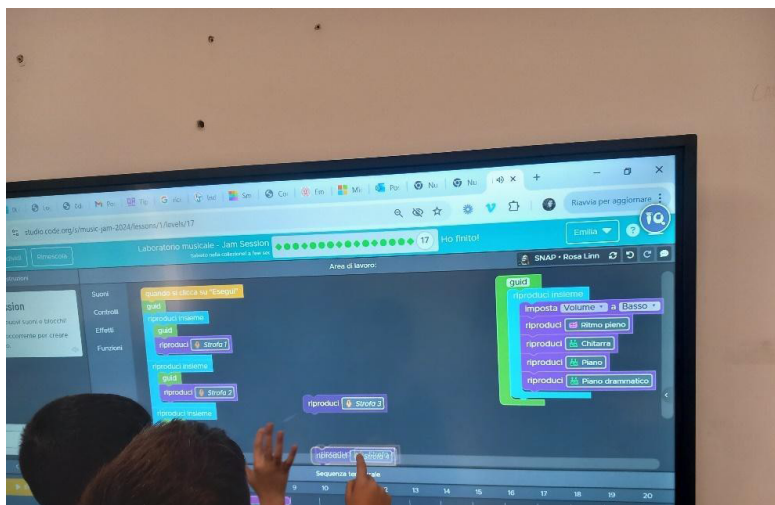


Figura 9: Realizzazione del progetto JamSession previsto nell'Ora del Codice.

4. Creatività computazionale e comunicazione attraverso i codici

4.1 Codici comunicativi con i pentamini

Uno degli aspetti più originali e apprezzati del progetto è stato l'incontro tra il coding e l'espressione creativa. Gli alunni, infatti, sono stati guidati nell'esplorazione di diverse forme di comunicazione attraverso il codice: dalle istruzioni scritte per guidare un compagno in un labirinto, fino alla creazione di veri e propri alfabeti simbolici. Un'attività particolarmente significativa ha coinvolto l'uso dei pentamini – figure geometriche composte da cinque quadrati uniti tra loro in modi diversi. Questi strumenti si presentano in 12 forme distinte a ciascuna delle quali sono state attribuite delle lettere dell'alfabeto (Fig. 10).

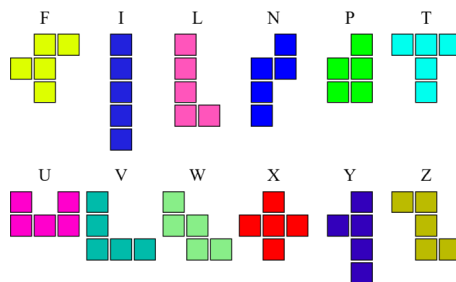


Figura 10: Pentamini: figure geometriche composte da cinque quadrati uniti tra loro in modi diversi.

Alcune di queste forme hanno ulteriori proprietà combinatorie poiché possono ruotare o presentarsi in forma simmetrica: L, N, P, F e Y si possono orientare in 8 modi: 4 tramite rotazione ed altri 4 aggiungendo una simmetria; Z si può orientare in 4 modi: 2 tramite rotazione ed altri 2 aggiungendo una simmetria; T, V, U e W si possono orientare in 4 modi tramite rotazione; I si può orientare in 2 modi tramite rotazione; X si può orientare in un solo modo. I bambini durante il laboratorio hanno sperimentato le varie combinazioni e ricercato figure geometriche risultanti dall'unione di più pentamini, allo scopo di trovare di trovare alcune proprietà, oltre a quella della equiestensione, come la isoperimetria (Fig. 11).

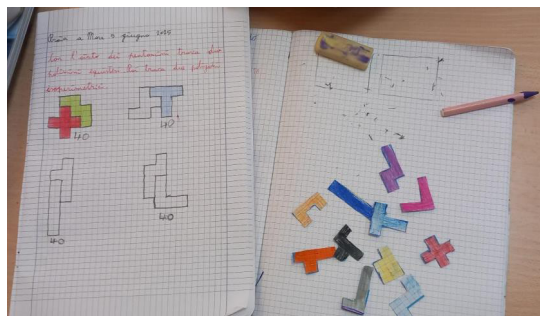


Figura 11: Utilizzo dei pentamini per trovare le proprietà di equiestensione e isoperimetria dei poligoni risultanti.

Un'ulteriore variazione sull'utilizzo di questi strumenti è stata la proposta di utilizzarli come codici comunicativi, secondo un criterio suggerito dai bambini stessi, consistente nell'attribuire una sillaba alle diverse combinazioni rotatorie e simmetriche di un pentamino, laddove possibile, come nell'esempio (Fig. 12):

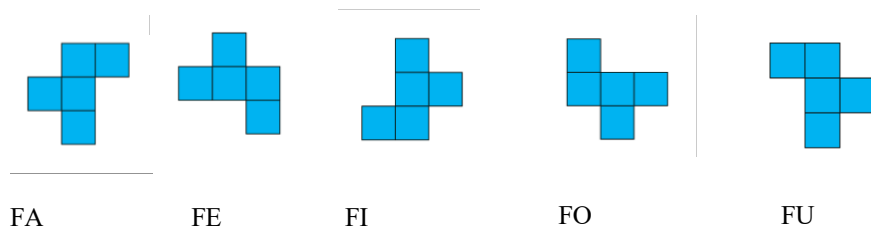


Figura 12: Associazione rotazione spaziale- sillabe.

I bambini, organizzati in squadre, ispirandosi a queste forme hanno “inventato” nuove parole e linguaggi visivi, associando a ogni figura un suono, una lettera o un significato. Ad esempio, la combinazione della rotazione a 180° del pentamino F, corrispondente alla sillaba FI, con la rotazione a 270° del pentamino L, sillaba LO, produce la parola bisillaba piana FILO (fig. 13):

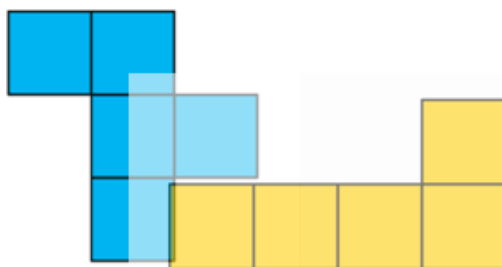


Figura 13: Combinazione di due pentamini per produrre una parola bisillaba piana.

Ciascuna squadra si è sfidata proponendo delle forme con la parola da indovinare. Questa esperienza ha unito logica e creatività, stimolando la riflessione sul valore simbolico del codice e sul modo in cui il pensiero computazionale può tradursi in linguaggio, arte e immaginazione. Attraverso questa attività è emerso chiaramente come il coding possa diventare un linguaggio espressivo, capace di veicolare idee e significati e non solo istruzioni operative.

4.2 Coding e storytelling

Il coding unplugged si è rivelato un'efficace modalità per lavorare sullo storytelling, offrendo agli alunni l'opportunità di ideare e sviluppare trame e personaggi in modo creativo e strutturato. Attraverso semplici comandi e sequenze logiche, gli studenti hanno potuto costruire narrazioni complesse, trasformando il linguaggio del coding in un vero e proprio strumento narrativo. Un valore aggiunto dell'esperienza è stato rappresentato dalla collaborazione in verticale con

la scuola dell'infanzia di Aieta, uno dei plessi dell'Istituto Comprensivo di Praia a Mare e parte del Movimento delle Piccole Scuole – INDIRE. Alcune attività di storytelling sono state realizzate proprio in sinergia con questo ordine di scuola, grazie a incontri in videoconferenza, rafforzando l'applicazione concreta del curriculum verticale. Questa modalità ha permesso agli alunni della primaria di sentirsi parte attiva di un progetto educativo condiviso, sviluppando responsabilità, empatia e efficacia comunicativa, specialmente nel rapporto con i bambini più piccoli. Gli studenti hanno prodotto diversi racconti originali, tra cui: "Il coniglietto Flip" e "L'ape Lella". Particolarmente significativa è stata la partecipazione alla creazione dei Diginauti, un progetto interattivo e multimediale disponibile al seguente link: <https://sites.google.com/view/infanzia-aieta-2024-2025/home-page?authuser=0>

4.3 Impatto sul Multilinguismo

L'integrazione del multilinguismo è stata perseguita attraverso l'uso consapevole dell'**inglese come lingua veicolare** in diverse fasi del corso. Sebbene non si trattasse di un corso di lingua specifico, l'obiettivo era familiarizzare gli alunni con il lessico tecnico-scientifico in inglese e promuovere un approccio naturale all'uso della lingua in contesti pratici:

- Sono stati introdotti e utilizzati termini specifici del coding in inglese (es. "code", "bug", "loop", "sequence", "command", "forward", "backward").
- Le istruzioni e le brevi spiegazioni sono state spesso presentate anche in inglese, incoraggiando la comprensione e l'utilizzo di frasi semplici.
- Durante lo storytelling, sono stati inseriti elementi narrativi o comandi in inglese, stimolando la creatività bilingue.
- L'interazione con la piattaforma code.org, se usata in inglese, ha contribuito all'esposizione linguistica.

L'impatto si è manifestato in una maggiore familiarità e minore reticenza nell'accostarsi a termini in inglese legati all'ambito STEM e digitale, ponendo le basi per un futuro apprendimento più approfondito.

5. Risultati e prospettive

Il progetto ha riscosso grande entusiasmo tra gli studenti e i docenti. Gli alunni hanno acquisito familiarità con concetti informatici complessi in modo naturale e coinvolgente, sviluppando una nuova consapevolezza delle potenzialità del linguaggio computazionale come strumento per capire e trasformare il mondo. Tra i principali risultati osservati:

- Maggiore autonomia e capacità di pianificazione nei compiti.
- Sviluppo di un atteggiamento positivo verso le sfide.
- Crescita nelle abilità comunicative e collaborative.
- Competenze disciplinari informatiche consolidate e spendibili anche in altri contesti.

In conclusione, il percorso di coding dell'IC Praia a Mare ha rappresentato un esempio virtuoso di come l'informatica possa diventare veicolo di cittadinanza, espressività e pensiero critico, nel rispetto delle Indicazioni Nazionali e con una visione pedagogica innovativa, centrata sull'alunno e sul valore formativo della creatività digitale.

Bibliografia

- [1] Bogliolo, Alessandro. A scuola con CodyRoby. Il coding come gioco di ruolo. Giunti Scuola. 2020
- [2] Bogliolo, Alessandro. Coding in your classroom. Now!. Giunti Scuola. 2016
- [3] World Health Organization. (1997). Life skills education for children and adolescents in schools. Geneva: WHO.
- [4] Di Nuovo, Santo. Competenze trasversali e life skills: si possono insegnare a scuola?. Psicologia e scuola, n. 10, 2022
- [5] Antonietti, A., & Valenti, C. (2022). Imparare a pensare per analogie: una skill per l'apprendimento e la conoscenza. In Skill training. Interventi psicologici per allenare la mente (pp. 35-61). Franco Angeli Editore, Milano.
- [6] Pecini, Chiara, Ruffini, Costanza. I fondamenti cognitivi delle soft skills. Psicologia e scuola, n. 10, 2022
- [7] Diamond, A. (2013). Executive functions. Annual review of psychology, 64(1), 135-168.
- [8] Verlinden, M., Veenstra, R., Ghassabian, A., Jansen, P. W., Hofman, A., Jaddoe, V. W., ... & Tiemeier, H. (2014). Executive functioning and non-verbal intelligence as predictors of bullying in early elementary school. Journal of abnormal child psychology, 42(6), 953-966.
- [9] Szcześniewska, P., Hanć, T., Bryl, E., Dutkiewicz, A., Borkowska, A. R., Paszyńska, E., ... & Dmitrzak-Węglarz, M. (2021). Do hot executive functions relate to BMI and body composition in school age children?. Brain Sciences, 11(6), 780.
- [10] Giorgetti, Marisa, Pizzingrilli, Paola, Antonietti, Alessandro. Creatività: come promuoverla a scuola? Psicologia e scuola, n. 6, 2009

**RACCONTI DI ESPERIENZE SUL CAMPO E
RELAZIONI SCIENTIFICHE**

PER LA SECONDARIA DI SECONDO GRADO

Machine Learning Unplugged e Python: un'esplorazione pratica

Francesco Picca¹

¹ I.I.S.S. "Augusto Righi" - Taranto

f.picca@righitaranto.edu.it

Abstract

Il presente contributo illustra l'esperienza condotta all'I.I.S.S. Augusto Righi di Taranto, nell'ambito del percorso PNRR Linea di Intervento A - Realizzazione di percorsi didattici, formativi e di orientamento per studentesse e studenti, Percorsi di orientamento e formazione per il potenziamento delle competenze STEM, digitali e di innovazione, 1224-ATT-827-E-9 "Analisi dei Dati e Machine Learning". Al fine di facilitare la comprensione dei principi fondamentali del Machine Learning, l'attività ha previsto una fase "Unplugged" con la "Macchina a Bicchieri", seguita dalla successiva implementazione dello stesso algoritmo attraverso il linguaggio di programmazione Python.

1. Introduzione

Nel corso dell'A.S. 2024-25, la classe 4CI ad indirizzo informatica dell'IISS Augusto Righi di Taranto ha partecipato a un webinar interattivo e coinvolgente, condotto dai professori Massimo Ferri (Università di Bologna) e Alessandro Bogliolo (Università di Urbino). Durante il webinar, gli studenti hanno avuto l'opportunità di esplorare concetti chiave dell'intelligenza artificiale e delle sue applicazioni pratiche, approfondendo tematiche di grande rilevanza per il loro percorso formativo. Tale webinar [1] ha introdotto i concetti fondamentali del Machine Learning attraverso la presentazione e la discussione della "Macchina a Bicchieri": un artefatto didattico che trae ispirazione da MENACE [2] (Matchbox Educable Noughts And Crosses Engine), una "macchina" ideata nel 1961 dal pioniere dell'informatica Donald Michie, un'IA realizzata con 304 scatole da fiammiferi e bottoncini colorati. La MAB [3] (macchina a bicchieri) utilizza invece 21 bicchieri per simulare un semplice gioco di una macchina contro un umano. Da questa esperienza è nata poi l'idea di implementare in Python una simulazione della MAB, per esplorare ulteriormente i principi dell'apprendimento automatico.

2. Machine Learning Unplugged

Per la MAB sono stati ideati tre giochi specifici, chiamati Gioco 1, 2 e 3. Tutti si svolgono su un esagono con vertici A, B, C, D, E, F (Figura 1), secondo le regole:

1. I giocatori, Ada (pedine rosse) e Bruno (rappresenta la macchina MAB con le pedine blu), mettono alternativamente le pedine sui vertici vuoti.

2. Ada gioca sempre per prima, obbligatoriamente mettendo una pedina rossa su A.
3. Ada conosce la configurazione vincente segreta che vuole raggiungere, mentre Bruno NON la conosce.
4. Ada vince la partita se riesce a completare la sua configurazione vincente, mentre Bruno vince se glielo impedisce.
5. Alla fine di ogni partita, le pedine vengono rimosse.
6. Una “sessione” consiste in un numero deciso di partite (k) dello stesso gioco, ed è vinta da chi vince più partite.
7. Le configurazioni vincenti per Ada (pedine rosse) sono diverse per ogni gioco:
 - Gioco 1: Tre pedine in posizioni non adiacenti; l'unica configurazione vincente è ACE.
 - Gioco 2: Tre pedine in posizioni adiacenti; le configurazioni vincenti sono ABC, FAB, EFA.
 - Gioco 3: Due pedine in posizioni adiacenti e la terza non adiacente alle altre; le configurazioni vincenti sono ABD, ABE, FAC, FAD, ACD, ADE.

Funzionamento della MAB

Per preparare la MAB, si applicano 21 etichette (scaricabili da un file PDF) a 21 bicchieri. (le Figure 2 - 5 mostrano la preparazione). Ogni etichetta riporta una configurazione di gioco e la lista delle lettere (B, C, D, E, F) da inserire inizialmente nel bicchiere corrispondente.

Nel momento in cui sceglie, Bruno (la MAB) identifica il bicchiere che corrisponde alla configurazione attuale sull'esagono e pesca a caso una delle lettere contenute nel bicchiere. Bruno mette la sua pedina blu sul vertice indicato dalla lettera estratta.

	B C D E F		B E F		B C F
	D E F		D E F		B D F
	C E F		C E F		C D F
	C D F		B E F		C D E
	C D E		B C F		B D E
	B D E		B C E		B C E
	B D F		B C D		B C D

Figura 1: Configurazioni possibili della “Macchina a Bicchieri”

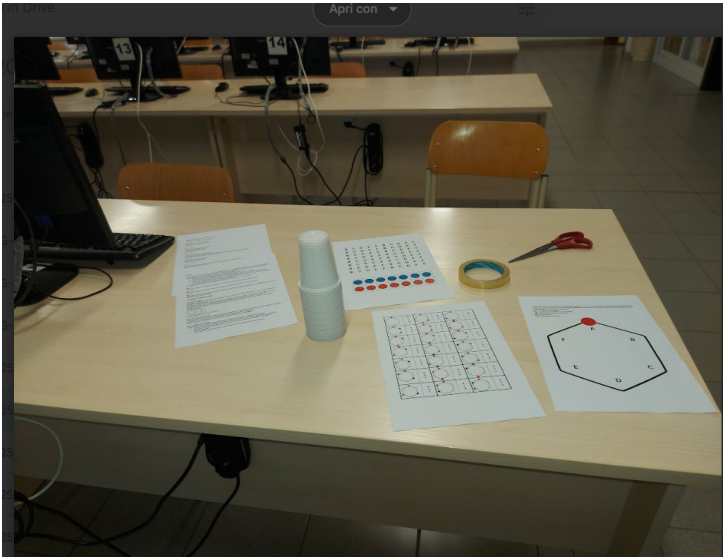


Figura 2: Materiali occorrenti per la “Macchina a Bicchieri”

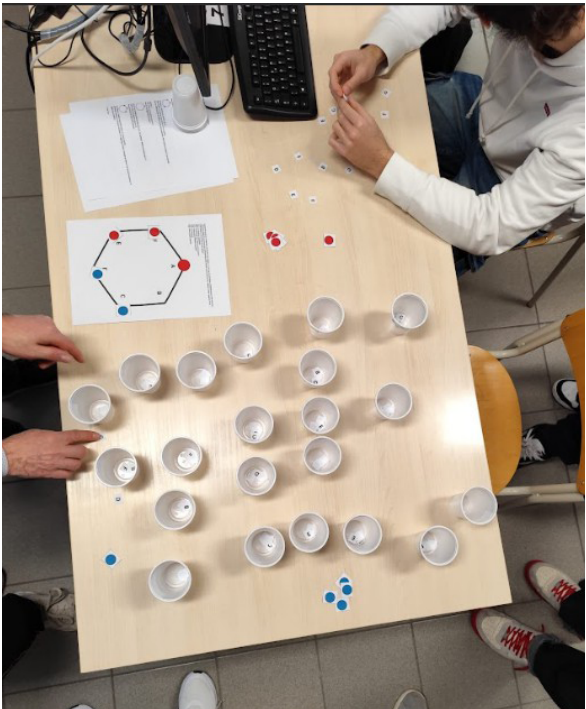


Figura 3: Costruzione della “Macchina a Bicchieri”



Figura 4: “Macchina a Bicchieri” operativa



Figura 5: Ulteriori dettagli ed inizio del Gioco

Addestramento della MAB

Per l'attività didattica presentata, si è scelto il Gioco 2: tre pedine in posizioni adiacenti; le configurazioni vincenti sono ABC, FAB, EFA.

L'addestramento avviene partita per partita, secondo le specifiche:

1. Se Bruno (la MAB) vince la partita, le lettere estratte dai bicchieri vengono rimesse nei rispettivi bicchieri e ne viene aggiunta un'altra uguale (nell'apprendimento per rinforzo corrisponde al premio).
2. Se Bruno perde la partita (cioè Ada vince), le lettere estratte dai bicchieri NON vengono rimesse; vengono cioè tolte definitivamente (nell'apprendimento per rinforzo corrisponde alla punizione).
3. Se un bicchiere si svuota completamente, le lettere indicate sulla sua etichetta vengono reinserite per riposizionarlo nella configurazione iniziale.

3. Implementazione in Python della Macchina a Bicchieri

Il programma Python sviluppato [4] simula il funzionamento della MAB (Figura 6). In particolare, è stato implementato, così come è accaduto con le simulazioni reali, il Gioco 2. Ogni stato del gioco è rappresentato da una struttura dati che corrisponde ai “bicchierini” della macchina, ciascuno contenente un insieme di possibili mosse (analoghe alle biglie). Dopo ogni partita, se la macchina vince, le mosse che ha scelto vengono rafforzate (aumentando la loro probabilità di essere selezionate in futuro); se perde, le mosse vengono penalizzate o rimosse. Questo meccanismo riproduce il principio del rinforzo positivo/negativo.

Il programma consente di osservare il comportamento della macchina nel tempo: inizialmente le scelte sono casuali, ma partita dopo partita la MAB tende a preferire le mosse che hanno portato alla vittoria, fino a sviluppare una strategia vincente. Questa simulazione mostra come anche un algoritmo semplice, privo di intelligenza vera e propria, possa apprendere in modo efficace solo attraverso l'interazione con l'ambiente. In un contesto didattico, questo progetto permette di esplorare concretamente concetti chiave dell'intelligenza artificiale, dell'algoritmica e della teoria dell'apprendimento. È anche un'occasione per riflettere su cosa significhi davvero “imparare” e su come, in certi casi, basti solo selezionare e adattarsi, senza capire.

In particolare, è utile osservare l'evoluzione del comportamento della macchina nel tempo dopo un numero sufficienti di partite, e valutare così quanto rapidamente sviluppa una strategia efficace. (Figure 7-9).

Prendendo in considerazione le versioni successive, il programma python potrebbe includere alcuni parametri, modificabili per personalizzare l'esperimento e osservare come cambiano le dinamiche di apprendimento della MAB:

- **Regole di rinforzo:** si possono modificare le quantità di biglie aggiunte in caso di vittoria e rimosse in caso di sconfitta, influenzando la velocità di apprendimento.
- **Strategia dell'avversario:** invece di un avversario casuale, si può definire un avversario strategico, per testare la capacità di adattamento della MAB.

```
def main():
    """Gioca n partite e conta quante volte vince Player 1 e Player 2."""
    player1_wins = 0
    player2_wins = 0
    num_partite = 50
    player1_wins_list=[]
    player2_wins_list=[]
    for partita in range(1, num_partite + 1):
        winner = play_game()

        if winner == 1:
            player1_wins += 1
        else:
            player2_wins += 1

        # Salviamo i progressi per il grafico
        player1_wins_list.append(player1_wins)
        player2_wins_list.append(player2_wins)

        if partita % 100 == 0:
            print(f"partita {partita} giocate... (Player 1: {player1_wins}, Player 2: {player2_wins})")

    print(f"\nEsito delle {num_partite} partite:")
    print(f"Player 1 ha vinto {player1_wins} partite.")
    print(f"Player 2 ha vinto {player2_wins} partite.")

    if player1_wins > player2_wins:
        print("Player 1 ha vinto più partite!")
    elif player2_wins > player1_wins:
        print("Player 2 ha vinto più partite!")
    else:
        print("È un pareggio!")

    # Creazione del grafico
    plt.figure(figsize=(10, 6))
    plt.plot(range(1, num_partite + 1), player1_wins_list, label="Player 1 Ada umano", color="red")
    plt.plot(range(1, num_partite + 1), player2_wins_list, label="Player 2 Bruno la MAB", color="blue")
    plt.xlabel('Numero di partite')
    plt.ylabel('Numero di vittorie')
    plt.title(f'Avanzamento delle vittorie nelle {num_partite} partite')
    plt.legend()
    plt.grid(True)
    plt.show()

if __name__ == "__main__":
    main()
```

Figura 6: Estratto del codice Python, metodo main()

Esito delle 50 partite:
 Player 1 ha vinto 14 partite.
 Player 2 ha vinto 36 partite.
 Player 2 ha vinto più partite!

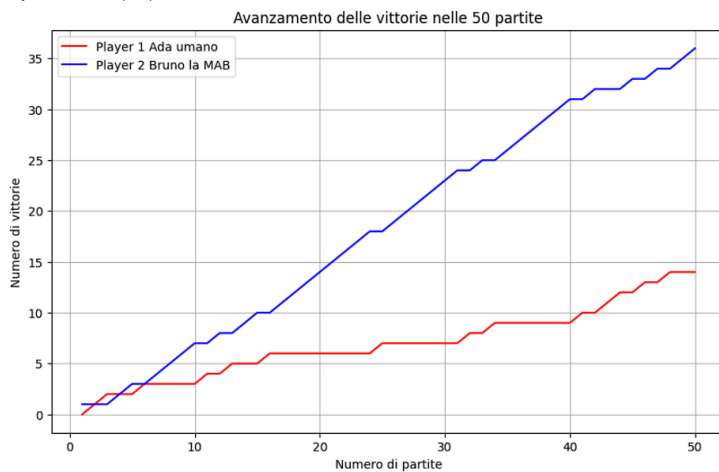


Figura 7: Simulazione di 50 partite - dopo una fase iniziale, la MAB vince il 72% delle partite

Esito delle 1000 partite:
Player 1 ha vinto 153 partite.
Player 2 ha vinto 847 partite.
Player 2 ha vinto più partite!

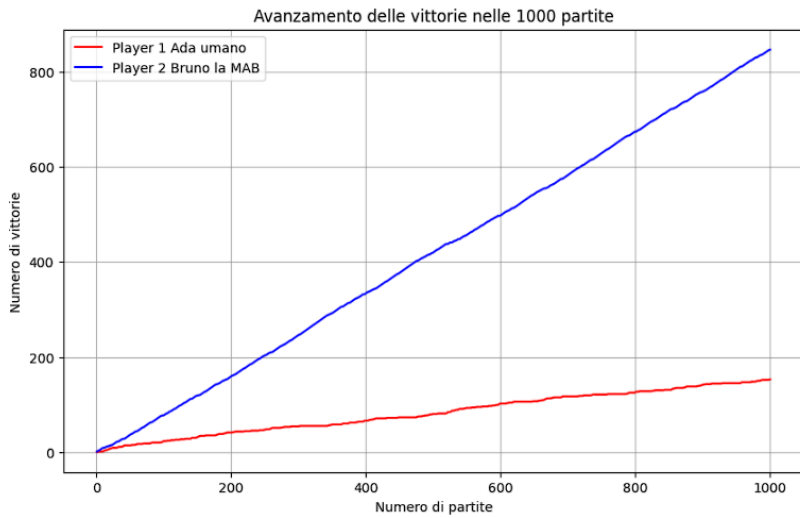


Figura 8: Simulazione di 1000 partite, la MAB vince l'84% delle partite

Esito delle 100000 partite:
Player 1 ha vinto 13401 partite.
Player 2 ha vinto 86599 partite.
Player 2 ha vinto più partite!

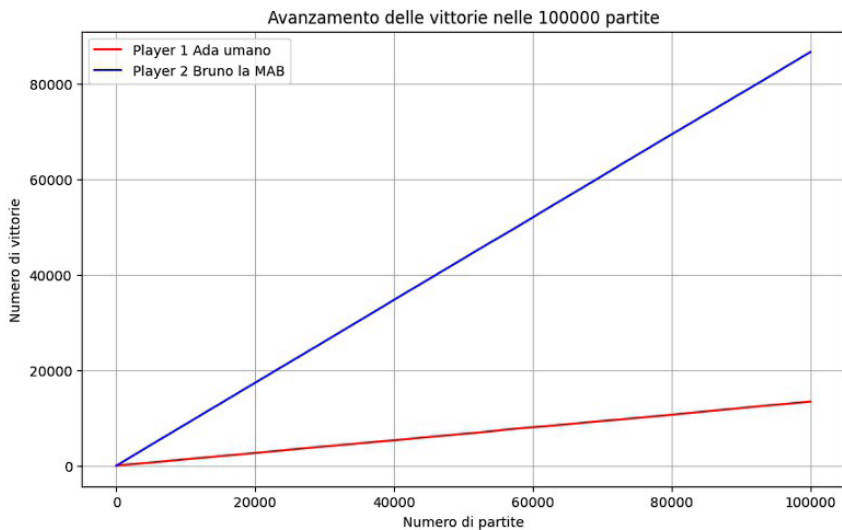


Figura 9: Simulazione di 100000 partite, la MAB vince l'86% delle partite

4. Conclusioni

La MAB: una Macchina Intelligente che non sapeva di esserlo

A prima vista, la Macchina a Bicchieri può sembrare solo un curioso esperimento didattico, una costruzione artigianale fatta di bicchierini, biglie e compartimenti. Eppure, dietro la sua semplicità apparente, si nasconde una dinamica sorprendente: la capacità di apprendere, migliorare e vincere senza alcuna consapevolezza.

Non ha un cervello, né sensori, né linguaggio. Non comprende le regole del gioco, e non ha alcuna nozione di sé. Eppure, attraverso il meccanismo del rinforzo, la MAB riesce ad affinare le sue scelte e ad assumere comportamenti vincenti. Questo ci obbliga a ripensare il concetto stesso di intelligenza: è davvero necessaria la coscienza per apprendere? O può bastare un semplice processo di selezione e adattamento?

L'esperimento con la MAB, se non se ne comprende il funzionamento interno, può sembrare abbastanza impressionante. I 21 bicchierini della MAB non creano un modello mentale del gioco, non cercano regole o strategie, non sanno di vincere o perdere, e non "sanno" nemmeno di giocare. Eppure, finiscono per vincere.

Questo paradosso apparente ci mostra qualcosa di fondamentale dell'apprendimento: non sempre serve una comprensione cosciente o una rappresentazione simbolica per migliorare il comportamento. La MAB funziona semplicemente premiando le mosse che hanno portato al successo e scoraggiando quelle che hanno portato alla sconfitta. In termini tecnici, si tratta di un meccanismo di apprendimento per rinforzo, in cui le configurazioni che ottengono risultati positivi tendono a essere selezionate più spesso.

Da un punto di vista didattico, questo ci permette di osservare in modo tangibile il principio darwiniano di selezione applicato non alla biologia, ma all'informazione e al comportamento. Le mosse "vincenti" sopravvivono, quelle "perdenti" vengono eliminate. È un esempio potente di come l'intelligenza artificiale, anche nella sua forma più elementare, possa imitare certi aspetti del comportamento intelligente senza comprendere nulla.

L'attività proposta ha suscitato grande entusiasmo negli studenti, in particolare durante la fase di costruzione fisica della Macchina a Bicchieri (MAB) e nella sua sperimentazione diretta. La manipolazione concreta dei materiali ha favorito un approccio attivo e partecipato, stimolando curiosità e spirito collaborativo. Parallelamente, gli studenti sono stati guidati nella comprensione e nella scrittura del codice Python che simula il comportamento della MAB, avviandoli a un primo livello di astrazione e riflessione sul funzionamento algoritmico del modello.

Questa doppia esperienza – fisica e digitale – ha permesso di attivare processi cognitivi e metacognitivi complementari: la MAB fisica ha facilitato la comprensione intuitiva dei concetti di base, mentre la MAB digitale ha offerto un'opportunità per formalizzare e rappresentare il comportamento dell'algoritmo in chiave computazionale.

La MAB ci invita a riflettere su cosa significhi davvero "imparare" o "capire". Se un sistema può avere successo senza sapere cosa sta facendo, allora forse l'intelligenza, come la interpretiamo comunemente, non è una condizione necessaria per adattarsi o migliorare. La macchina a bicchieri, pur essendo composta solo da oggetti inanimati e regole meccaniche, riesce a mostrare un tipo primitivo di apprendimento che anticipa molte idee moderne sull'IA.

Osservando la MAB, capiamo che l'efficacia non dipende dalla complessità strutturale, ma dalla capacità di adattarsi all'ambiente e migliorarsi nel tempo. In un'epoca in cui le macchine intelligenti stanno diventando parte integrante della nostra società, comprendere come anche

le forme più semplici di apprendimento possano produrre risultati complessi ci aiuta a riflettere su cosa significhi davvero “essere intelligenti” per le macchine, e forse anche per noi.

Bibliografia

- [1] Webinar di Massimo Ferri/Alessandro Bogliolo: <https://www.youtube.com/live/sBOPFAgxbsA>
- [2] MENACE: the pile of matchboxes which can learn (https://www.youtube.com/watch?v=R9c-_neaxeU)
- [3] La “Macchina a Bicchieri”: <https://maddmaths.simai.eu/divulgazione/ai-bicchierini/>
- [4] Riferimento Programma Python (esecuzione con scelte casuali)

Insegnare informatica nell'era dell'IA: pratiche, sfide e prospettive

Paolo Ciancarini², Marcello Missiroli^{1,2}, Giancarlo Succi²

¹ IIS Corni Liceo e Tecnico, Modena

m.missiroli@istitutocorni.it

² Università di Bologna, DISI

paolo.ciancarini@unibo.it, marcello.missiroli@unibo.it, giancarlo.succi@unibo.it

Abstract

L'intelligenza artificiale (IA) generativa è ormai uno strumento di uso quotidiano nel mondo del lavoro, incluso quello informatico. Ignorarne o vietarne l'utilizzo nella didattica rischia di creare un divario tra la scuola e la realtà professionale. Questo articolo esamina l'impiego dell'IA nel contesto lavorativo e in due situazioni educative di diverso grado, evidenziandone analogie e differenze. Si conclude proponendo alcune linee guida per una didattica dell'informatica che integri l'IA come risorsa formativa, evitando che diventi una pericolosa scorciatoia.

1. Introduzione

L'intelligenza artificiale generativa è già da tempo parte integrante del flusso di lavoro di un informatico [8]. Strumenti come GitHub Copilot, Cursor o ChatGPT sono usati per generare codice, documentare, creare prototipi, eseguire test automatici e molto altro. Secondo vari report industriali [11], una fetta crescente di sviluppatori integra quotidianamente questi strumenti per velocizzare compiti ripetitivi ed ottenere soluzioni rapide a problemi specifici.

Tuttavia, l'adozione diffusa non equivale a una sostituzione del programmatore umano. Al contrario, proprio l'utilizzo concreto di questi strumenti rivela quanto siano ancora centrali le competenze di sviluppo tradizionali. Per questo motivo, formare gli studenti a usare questi strumenti non significa solo insegnare cosa chiedere all'IA, ma soprattutto quando e perché intervenire in prima persona. È necessario abituarli a leggere criticamente l'output dell'IA, a riconoscere segnali di incoerenza, e a sviluppare la capacità di "rientrare nel codice" in modo efficace. L'idea che basti saper scrivere prompt è illusoria: serve una competenza tecnica solida per uscire dal pit of death e completare progetti reali.

2. Stato dell'arte della IA nella didattica dell'informatica

L'integrazione dell'intelligenza artificiale (IA), in particolare dei chatbot generativi come ChatGPT, nella didattica della programmazione sta trasformando in profondità il modo in cui gli studenti apprendono concetti computazionali e affrontano problemi algoritmici.

Una prima conferma è fornita da Albadarin *et al.* [1], tramite una revisione sistematica della letteratura riguardante l'uso di ChatGPT in ambito educativo. Lo studio evidenzia come ChatGPT funga da assistente virtuale, offrendo supporto su richiesta, assistenza nella scrittura e apprendimento personalizzato per gli studenti, oltre ad aiutare gli insegnanti nella pianificazione delle lezioni e nella generazione di contenuti. Becker *et al.* [2], pur confermando gli aspetti positivi dell'uso dell'intelligenza artificiale, osservano che essa potrebbe ostacolare lo sviluppo di competenze più profonde nella risoluzione dei problemi e generare un'"illusione di competenza". Quest'ultimo aspetto è confermato da Groothuisen *et al.* [4]. Olga Petrovska *et al.* [9] analizzano le modalità di integrazione dell'IA nei percorsi formativi di sviluppo software, dimostrando come tali strumenti possano essere usati per stimolare competenze critiche e tecniche nei discenti, incluse capacità di codifica, argomentazione e consapevolezza etica. Li *et al.* [6] propongono un quadro sistematico per l'integrazione dei Large Language Models nella formazione in ingegneria del software, sottolineando l'urgenza di aggiornare i programmi educativi per rispondere alle trasformazioni che l'AI sta apportando alla pratica professionale, soprattutto nei processi di sviluppo e qualità del software. Permane una carenza di materiali didattici operativi per l'integrazione dell'IA, con l'eccezione del lavoro di Porter [10] che propone tecniche - solo operative - di prompt engineering, scomposizione in sottoproblemi affrontabili dall'IA, interpretazione dei risultati e debugging "AI-first".

3. L'IA nel mondo del lavoro

L'uso dell'IA è ampiamente diffuso e incentivato nel mondo del lavoro, integrandosi in tutte le fasi dello sviluppo software: dalla generazione del codice alla documentazione, passando per testing e pratiche DevOps. Questo emerge sia da fonti aneddotiche – blog di programmatori e annunci su LinkedIn – sia da studi accademici [5, 3]. Gli utilizzi principali includono:

- Assistenza nella scrittura del codice, riducendo attività ripetitive e aumentando l'efficienza.
- Supporto in debugging, revisione e documentazione, migliorando la qualità del software.
- Potenziamento delle competenze, specie per sviluppatori junior o in contesti tecnologici poco noti.

Restano tuttavia criticità su affidabilità, trasparenza ed etica.

Un fenomeno emblematico è il pit of death (fig. 1), termine coniato da Steve Sewell [12]: inizialmente l'IA fornisce risultati rapidamente, ma al crescere della complessità – requisiti avanzati, comportamenti specifici o bug – l'output diventa incoerente o inutilizzabile. L'intervento umano diventa allora necessario per correggere, riscrivere e ripristinare il codice.

Questo ciclo è ricorrente: allo stato attuale, l'IA è utile per prototipi e compiti standard, ma nella complessità reale dello sviluppo, le competenze umane di progettazione, debugging e refactoring tornano centrali. Con l'evolvere dell'applicazione, gli interventi umani diventano sempre più frequenti, segno che le LLM non sostituiscono i programmatori, ma ne trasformano il ruolo in chiave più strategica.

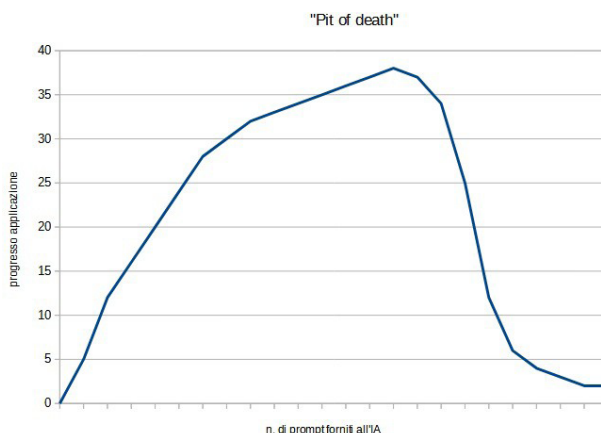


Figura 1: Pit of Death

4. Nostra esperienza: scuola superiore

L'IIS "Fermo Corni" di Modena, istituto tecnico a forte indirizzo tecnologico, comprende cinque sezioni di Liceo delle Scienze Applicate con potenziamento informatico e tre corsi di indirizzo "Informatica". Nonostante la vocazione tecnologica, l'uso dell'IA non è formalmente integrato nella didattica, salvo un impiego marginale nella preparazione di esercizi. È vietato durante le verifiche, anche se alcuni studenti tentano di aggirare il divieto. Spesso accettano acriticamente soluzioni errate o complesse proposte dall'IA, costringendo i docenti a rilevare questi usi impropri – ad esempio tramite costrutti mai affrontati in classe – e ad applicare sanzioni. Questa situazione ha progressivamente svuotato di significato il lavoro a casa, rendendo difficile valutare le competenze individuali.

Aggiungiamo un caso particolare, relativo a una particolare forma di Alternanza Scuola-Lavoro ideata nel 2017: la "Developer's Week" [7]. Si tratta di un progetto didattico che simula un contesto aziendale, in cui piccoli team affrontano in una settimana la realizzazione di una soluzione digitale, partendo da un bisogno reale, con tempi e consegne scanditi come in un vero ambiente di lavoro.

La diffusione massiva di strumenti di intelligenza artificiale generativa ha impattato il progetto significativamente. L'uso dell'IA non poteva essere vietato né ignorato: la natura stessa del progetto, che mira a responsabilizzare gli studenti e stimolare l'autonomia, mal si conciliava con un controllo rigido o con il blocco di strumenti ormai di uso comune. Abbiamo quindi scelto di consentire l'uso dell'IA, senza alcun tipo di formazione da parte nostra, chiedendo in cambio trasparenza. Ogni gruppo partecipante è stato invitato a documentare in una sezione specifica della relazione finale l'utilizzo degli strumenti di IA, indicando sia i principali prompt utilizzati sia una riflessione sul valore aggiunto (o sugli eventuali limiti) riscontrati. Questa scelta si è rivelata utile su più livelli: ha permesso di analizzare concretamente l'impatto dell'IA sul processo creativo e produttivo, ha favorito un uso consapevole degli strumenti digitali e ha offerto agli studenti l'occasione per riflettere criticamente sul proprio modo di lavorare. Nei

paragrafi seguenti presentiamo alcune evidenze raccolte, mettendo a confronto modalità d’uso, vantaggi percepiti, rischi emersi e spunti per le future edizioni del progetto.

Di particolare interesse notare l’evoluzione dell’utilità percepita da parte degli studenti nel biennio 24-25. Al termine del progetto, ad ogni studente è stata posta la domanda “Quanto è stato utile chatGPT?”, con una risposta fornita da una scala Likert a 5 livelli. Come si desume dalla tabella, il giudizio medio è totalmente mutato in pochi mesi, passando da poco o marginalmente utile ad estremamente utile. Dai commenti testuali si desume inoltre che l’IA è stata usata nel 2025 in modo più completo rispetto al passato: in fase iniziale per orientarsi sul progetto o ottenere spunti, ma anche per chiedere snippet di codice, debugging e verifica.

Tuttavia, gli insegnanti interessati al progetto non hanno rilevato cambiamenti di rilievo né nel prodotto finale, né nella qualità media degli artefatti (codice, grafici), né nei tempi di consegna. Questo suggerisce che molti studenti utilizzano ancora l’intelligenza artificiale in modo superficiale, poco critico e spesso inefficace.

Valutazione	AS 2023-24	AS 2024-25
1 (basso)	15.8%	0.0%
2	31.6%	5.0%
3	31.6%	20.0%
4	21.1%	30.0%
5 (alto)	0.0%	45.0%

Tabella 1: Utilità percepita dell’IA - superiori

5. Nostra esperienza: progetti universitari

Gli autori insegnano da diversi anni il corso di Ingegneria del Software, tenuto al terzo anno di Scienze dell’Informazione presso l’Università di Bologna. Una parte consistente della valutazione consiste nella realizzazione di un progetto complesso, sviluppato in team piuttosto numerosi, con ampia libertà nella scelta di strumenti e modalità organizzative, incluso ovviamente l’uso dell’IA; anche in questo caso, non sono previsti momenti formativi sull’argomento. Al termine dei progetti, gli studenti (mediamente 80 per coorte) devono compilare un questionario che, negli ultimi due anni, include anche domande sull’utilità dell’IA nello svolgimento del progetto.

Valutazione	2023–24 (%)	2024–25 (%)
1 (basso)	8.0	1.9
2	12.0	11.1
3	30.0	42.6
4	34.0	31.5
5 (alto)	16.0	13.0

Tabella 2: Utilità percepita dell’IA - università

Anche qui si nota una evoluzione, che prevede una valutazione medio-alta dell'utilità e in generale aumento dell'arco dei 12 mesi. Tuttavia, la distribuzione si concentra maggiormente sul voto 3 (utilità moderata). Si registra un calo marcato delle valutazioni più basse ("1"), segno che sempre meno studenti considerano l'IA del tutto inutile. Parallelamente, si rileva una lieve diminuzione delle valutazioni massime ("5"), probabilmente perché l'entusiasmo iniziale si è attenuato, lasciando spazio a una valutazione più realistica, che tiene conto anche dei limiti dello strumento. Nei commenti testuali forniti dagli studenti a fine progetto non si riscontra alcun riferimento diretto o implicito all'uso dell'intelligenza artificiale, nonostante i dati quantitativi (domanda 36) mostrino che una buona parte degli studenti ne ha fatto uso. Questo divario tra uso e riflessione suggerisce che l'IA, pur essendo presente nel processo, non viene ancora percepita come parte integrante del progetto. L'assenza di consapevolezza critica o di valutazioni sull'uso dell'IA conferma l'urgenza di introdurre strumenti e attività che aiutino gli studenti a integrare in modo più riflessivo e produttivo questi strumenti nella loro pratica progettuale.

Sul piano dei risultati, i docenti hanno notato la tendenza a un completamento più rapido del progetto, ma non hanno rilevato grandi differenze nella qualità del codice prodotto (correttezza, manutenibilità, ecc.).

6. Conclusioni e proposte

Quanto emerso ci fa ritenere urgente e necessario un percorso di alfabetizzazione all'IA, in particolare in ambienti collaborativi o soggetti a regolamentazione, e l'importanza di una migliore integrazione con gli strumenti di sviluppo già esistenti. È quindi necessario sviluppare una didattica mirata all'uso consapevole e competente di questi strumenti. L'intelligenza artificiale è ormai una realtà consolidata, impossibile da rimuovere dal contesto attuale. È invece possibile – e auspicabile – costruire percorsi formatici che aiutino gli studenti a integrare l'IA come risorsa utile, senza delegare ad essa la responsabilità del pensiero o della soluzione dei problemi. Alcune possibili direttive didattiche potrebbero essere:

- **Educazione al Prompting:** laboratori su come scrivere richieste efficaci e verificabili all'IA.
- **Analisi critica dell'output:** attività in cui gli studenti devono individuare e correggere errori presenti nelle risposte dell'IA.
- **Co-creazione controllata:** esercizi in cui studenti e IA collaborano alla stesura di codice o documentazione, con una riflessione finale sull'apporto effettivo dell'IA.
- **Riflessione obbligatoria sull'uso dell'IA:** ogni progetto dovrebbe includere una sezione che descriva se, come e quanto l'IA è stata utilizzata, con esempi di prompt e valutazioni critiche.
- **Avvio di sperimentazioni in rete:** progetti didattici guidati, documentati e condivisi tra scuole e università, al fine di costruire una base comune di pratiche didattiche efficaci e replicabili.

L'obiettivo non è solo "insegnare a usare l'IA", ma **formare cittadini digitali in grado di guidare il proprio pensiero**, riconoscere i limiti degli strumenti e assumersi la responsabilità delle proprie scelte anche quando passano attraverso un algoritmo.

Appendice: Questionari

Link ai moduli delle domande

Developers' Week (scuola superiore): <https://docs.google.com/forms/d/e/1FAIpQLSfzCfm2sEg0K9nQNOT2vNRkDseAAQ52FCjtgE-gnTh4rxOG9g/viewform?usp=header>

Capstone Project (università) https://docs.google.com/forms/d/e/1FAIpQLSfKuVvADK_eAtygdXrctXBQxiWAQyOED_RhjdZ5DQs0Tfa4SA/viewform?usp=dialog

Link alle risposte (Dropbox)

<https://www.dropbox.com/scl/fo/c5agffkcqhysqokz0iavs/ABTYs4V-MQwqqMOIhLacVXw?rlkey=zw6z7x4k748mgn2n68mij71be&st=2wo0n58f&dl=0>

Bibliografia

- [1] Yazid Albadarin, Mohammed Saqr, Nicolas Pope, and Markku Tukiainen. A systematic literature review of empirical research on ChatGPT in education. *Discover Education*, 3(1):60, 2024.
- [2] Brett A Becker, Michelle Craig, Paul Denny, Hieke Keuning, Natalie Kiesler, Juho Leinonen, Andrew Luxton-Reilly, Lauri Malmi, James Prather, and Keith Quille. Generative AI in introductory programming. In *ACM Computer Science Curricula*, pages 438–439. 2023.
- [3] I. Damyanov, N. Tsankov, and I. Nedyalkov. Applications of generative artificial intelligence in the software industry. *TEM Journal*, 2024.
- [4] Suzanne Groothuisen, Jan van der Meij, Robert Harmsen, and Ton de Jong. Ai chatbots in programming education: Students' use in a scientific computing course and consequences for learning. *Computers and Education: Artificial Intelligence*, 7:100290, 2024.
- [5] V. Jackson, G. Vaz Pereira, and R. Prikladnicki. Exploring GenAI in software development: Insights from a case study in a large brazilian company. 2025.
- [6] Y. Li, J. Keung, and X. Ma. Integrating generative ai in software engineering education: Practical strategies. In *2024 International Symposium on Educational Technology (ISET)*, pages 49–53. IEEE, 2024.
- [7] Marcello Missiroli, P Ciancarini, D Russo, and Paolo Torricelli. Developers' week: alternanza scuola lavoro rovesciata. *Mondo Digitale*, 18(81), 2019.
- [8] Stack Overflow. Stack overflow developer survey 2024: Artificial intelligence. <https://survey.stackoverflow.co/2024/ai>, 2024. Accessed: 2025-05-23.
- [9] Olga Petrovska, Lee Clift, Faron Moller, and Rebecca Pearsall. Incorporating generative ai into software development education. In *Proceedings of the 8th Conference on Computing Education Practice (CEP '24)*, pages 37–40. Association for Computing Machinery, 2024.
- [10] Leo Porter and Daniel Zingaro. *Learn AI-Assisted Python Programming: With Github Copilot and ChatGPT*. Simon and Schuster, 2024.
- [11] Rudolf Ramler, Michael Moser, Lukas Fischer, Markus Nissl, and René Heinzl. Industrial experience report on ai-assisted coding in professional software development. In *Proceedings of the 1st International Workshop on Large Language Models for Code*, pages 1–7, 2024.
- [12] Steve Sewell. Why ai is making software dev skills more valuable, not less. YouTube video, 2024. Pubblicato ca. 7 mesi fa, su Builder.io / YouTube :contentReference[oaicite:1]index=1

Hackathon a scuola: un'esperienza didattica per lo sviluppo delle competenze non cognitive

Laura Castellana¹, Giuseppe Mastrandrea²

¹DS – ITT Panetti Pitagora

dirigente@panettipitagora.edu.it

²ITT Panetti Pitagora

giuseppe.mastrandrea@panettipitagora.edu.it

Abstract

L'articolo presenta un'esperienza didattica basata su un hackathon, progettata per studenti del quarto anno di un istituto tecnico con indirizzo informatico, volta a colmare il divario tra apprendimento a scuola e mondo reale, con un focus sullo sviluppo di competenze non cognitive (NCS). Descrive la progettazione interdisciplinare dell'evento, la collaborazione con partner corporativi e realtà associative del territorio, e le attività svolte dagli studenti per affrontare la sfida sul tema "Innovare la scuola". Si analizza come l'esperienza abbia mirato a potenziare competenze tecniche specifiche dell'informatica, abilità multidisciplinari e un ampio spettro di NCS cruciali per il successo professionale, quali problem solving, teamwork e comunicazione efficace. Infine, si discutono le implicazioni per la valutazione e si propongono direzioni future per un assessment strutturato delle competenze sviluppate.

1. Introduzione

Il settore informatico richiede professionisti con solide competenze tecniche e un insieme di competenze non cognitive (NCS), come pensiero critico e adattabilità, fondamentali per il capitale umano [1]. Tuttavia, la didattica tradizionale fatica a integrarle con l'insegnamento disciplinare [2]. Gli hackathon emergono come una promettente strategia pedagogica esperienziale per affrontare questa sfida [3]. Questo articolo descrive e analizza un'iniziativa di hackathon (a tema "Innovare la scuola") implementata in un istituto tecnico italiano di Bari, indirizzata a studenti di informatica. L'obiettivo è duplice: illustrare come tale formato possa favorire lo sviluppo integrato di competenze informatiche, multidisciplinari e trasversali, e discutere il suo contributo alla formazione del capitale umano degli studenti.

2. Inquadramento teorico e Contesto del Progetto

2.1 L'Importanza Crescente delle Competenze Non Cognitive (NCS)

La letteratura sottolinea il ruolo cruciale delle NCS nello sviluppo professionale e nella competitività sul mercato del lavoro [1, 4]. Alcuni studi [3] evidenziano carenze nella loro formazione sistematica, nonostante siano essenziali per affrontare ambienti dinamici e complessi [5]. Parallelamente il concetto di capitale umano si è ampliato, includendo queste “higher-order skills” come contributi importanti di innovazione nello sviluppo sociale, culturale ed economico delle nazioni [7].

2.2 L'Hackathon come approccio pedagogico innovativo

La parola “hackathon” deriva dall'unione di due parole: “hacker” e “marathon”; può essere definito come un evento intensivo e collaborativo, spesso della durata di un giorno o un weekend, durante il quale team multidisciplinari si riuniscono per progettare, sviluppare e presentare soluzioni innovative (tipicamente prototipi software o hardware) a sfide specifiche o problemi predefiniti. Originariamente radicati nella cultura degli sviluppatori software (la prima evidenza di hackathon risale al 1999 con l'obiettivo di rendere più sicuro il sistema operativo OpenBSD), gli hackathon si sono evoluti in un fenomeno più ampio, adottato in contesti educativi, aziendali e sociali come strumento per stimolare la creatività, l'apprendimento esperienziale, la prototipazione rapida con un occhio sull'innovazione aperta. Essi, definiti da Briscoe & Mulligan [3] come eventi di “innovazione digitale collaborativa”, offrono un ambiente di apprendimento immersivo e intensivo, e simulano sfide progettuali reali, richiedendo ai partecipanti, suddivisi in gruppi, di applicare conoscenze teoriche in contesti pratici e con vincoli di tempo, promuovendo così l'apprendimento esperienziale [8].

3. Descrizione dell'esperienza didattica

L'esperienza ha coinvolto le classi quarte dell'indirizzo “Informatica e Telecomunicazioni” dell'ITT Panetti Pitagora di Bari, con un hackathon sul tema “Innovare la scuola”, pensato per stimolare soluzioni creative a problemi del contesto scolastico.

- **Fase preparatoria e attori coinvolti.** Nei due mesi precedenti, il Consiglio di Classe, con il referente di progetto e la dirigenza, ha curato la progettazione. Sono state attivate partnership con realtà del territorio (PugliaTechs, GDG Bari, Global Shapers, PyBari) che hanno contribuito alla definizione del tema, fornito mentor e partecipato alla valutazione. Agli studenti sono stati offerti brevi sessioni su metodologie agili e design thinking in ottica di consegna dei prodotti finali.
- **Svolgimento dell'hackathon.** Gli studenti, organizzati in team, hanno lavorato per sviluppare un prototipo di soluzione (es. applicazione web/mobile, piattaforma) in risposta alla sfida. Hanno avuto accesso a strumenti hardware e software professionali, simulando un ambiente lavorativo. Mentor interni (docenti) ed esterni (professionisti) hanno fornito supporto non direttivo, stimolando l'autonomia e il problem solving.
- **Prodotti finali e valutazione.** Ogni team ha realizzato un prototipo funzionante, un video pitch e un deck di slide, valutati da una commissione mista sulla base di criteri come innovazione, fattibilità, presentazione e design.

4. Analisi dei prototipi e delle competenze sviluppate

4.1 Prototipi sviluppati

I progetti presentati spaziano dall'orientamento scolastico al supporto per DSA, dalla gestione ambientale al miglioramento della vita scolastica quotidiana, proponendo soluzioni concrete alle sfide vissute dagli studenti.

- **Alpha Dev.** Un registro elettronico intelligente con premi per motivare gli studenti e facilitare la gestione delle presenze da parte dei docenti.
- **Breakeasy.** Una piattaforma digitale per semplificare e ottimizzare le ordinazioni al bar scolastico, riducendo caos e tempi d'attesa.
- **EcoSphere.** Sistema integrato con AI e "bidoni intelligenti" per promuovere la raccolta differenziata e l'educazione ambientale nelle scuole tramite competizioni e gamification.
- **OpenBook.** Una piattaforma per la condivisione e il prestito digitale di libri scolastici, facilitando l'accesso ai testi e riducendo sprechi.
- **semplificaMente.** Un portale di supporto per studenti con DSA che offre lettura vocale, mappe concettuali automatiche ed esercizi assistiti.
- **UniSearch.** Un sito con quiz e assistente AI per aiutare gli studenti delle superiori a scegliere consapevolmente l'università più adatta a loro.
- **NextStep.** Una web app che guida i ragazzi di terza media nella scelta della scuola superiore in base a passioni e attitudini personali.

4.2 Competenze specifiche dell'informatica

Gli studenti hanno applicato e approfondito conoscenze relative alla progettazione software, sviluppo di prototipi (con i linguaggi per il web studiati nei percorsi curricolari), e potenzialmente gestione dati e UI/UX design. L'introduzione a metodologie come Agile e Design Thinking ha fornito strumenti concettuali e operativi tipici dell'industria tech [3]. Qui di seguito vengono dettagliati i principali ambiti di competenza e le tecnologie affrontate:

- **Sviluppo software e programmazione.** Gli studenti hanno realizzato prototipi di applicazioni web, usando stack basati su HTML5, CSS3, JavaScript, chiamate asincrone ad API e uso di librerie front-end per il Machine Learning.
- **Protocolli di comunicazione.** Per la realizzazione di prototipi che prevedevano l'interazione tra client e server, gli studenti hanno dovuto comprendere i principi base della trasmissione dati, configurazione di API JSON e gestione delle chiamate HTTP.
- **User Interface (UI) e User Experience (UX) Design.** È stata posta particolare attenzione alla progettazione dell'interfaccia utente tramite strumenti di prototyping (es. Figma).
- **Strumenti per lo sviluppo collaborativo.** Gli studenti hanno utilizzato Git come strumento di versioning e pubblicato i propri prototipi web su piattaforme free come GitHub Pages o Altvista.

4.3 Competenze trasversali ed NCS

Questa è stata l'area di maggiore impatto atteso.

- **Teamwork e Collaborazione.** Lavorare in team eterogenei e auto-organizzati su un compito complesso con scadenze strette ha potenziato la collaborazione, la negoziazione e la gestione di dinamiche di gruppo [2].

- **Problem solving e pensiero critico.** Affrontare una sfida aperta come “Innovare la scuola” ha stimolato l’analisi critica dei problemi, la generazione di soluzioni creative e il pensiero laterale [8].
- **Comunicazione efficace.** Oltre alle presentazioni formali, la comunicazione interna ai team e con i mentor è stata fondamentale.
- **Adattabilità e Gestione dello Stress.** La natura intensiva dell’hackathon ha esposto gli studenti a contesti variabili e alla pressione del tempo, favorendo l’adattabilità e la resilienza [3].
- **Autonomia e Leadership.** La formazione spontanea dei gruppi e il supporto non direttivo dei mentor hanno incoraggiato l’iniziativa e l’emergere di ruoli di leadership. Queste NCS sono riconosciute come essenziali per l’impiegabilità e la progressione di carriera [2] e costituiscono una parte significativa del capitale umano valorizzato dal mercato [1].

5. Valutazione dell’impatto e prospettive future

L’impatto dell’hackathon è stato valutato attraverso un questionario di gradimento somministrato ai 26 studenti partecipanti.

5.1 Riscontri dei partecipanti

L’esperienza è stata accolta con entusiasmo unanime: il 66,7% dei partecipanti l’ha definita “Molto positiva” e il restante 33,3% “Positiva”. Tuttavia, sebbene l’apprezzamento generale sia elevato, l’analisi approfondita delle risposte rivela aspetti critici utili alla valutazione dell’impatto formativo. La quasi totalità degli studenti (86,7%) si è sentita solo “abbastanza” preparata, e nessuno ha dichiarato di sentirsi completamente pronto, suggerendo che l’hackathon, pur motivante, ha posto sfide superiori alle aspettative iniziali.



Figura 1: Valutazione complessiva dell’esperienza

Le competenze maggiormente sviluppate sono state il “Lavoro di squadra” (15 menzioni), la “Creatività e innovazione” e la “Comunicazione” (10 ciascuna), seguite da “Problem solving” (8), “Gestione del tempo” (6) e “Sviluppo tecnico” (4). Questi dati indicano un impatto rilevante sul piano delle competenze trasversali (NCS), con effetti anche su aspetti disciplinari, soprattutto nei team che hanno fatto uso di tecnologie avanzate. Dai commenti aperti emergono riferimenti specifici all'apprendimento di strumenti pratici (es. Flask, AI, strumenti di prototipazione), ma anche all'importanza del confronto, della collaborazione e della capacità di adattarsi a vincoli stringenti.

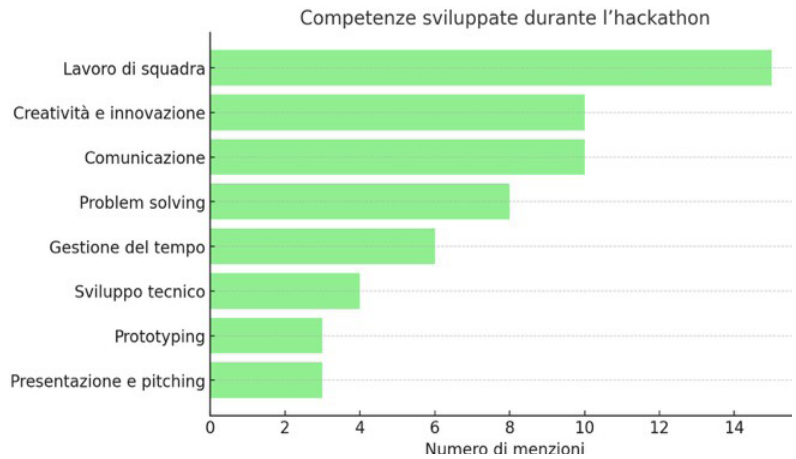


Figura 2: Competenze sviluppate durante l'hackathon

Tra le difficoltà riscontrate, la “Gestione del tempo” è risultata predominante (13 menzioni), riflettendo sia la struttura intensiva dell'evento sia una necessità latente di sviluppo di strategie organizzative. Alcuni studenti hanno inoltre segnalato “problemi tecnici o mancanza di competenze specifiche” (3) e difficoltà legate alla “chiarezza delle richieste” (2), suggerendo che la complessità delle consegne e l'approccio progettuale richiedono una preparazione più mirata.

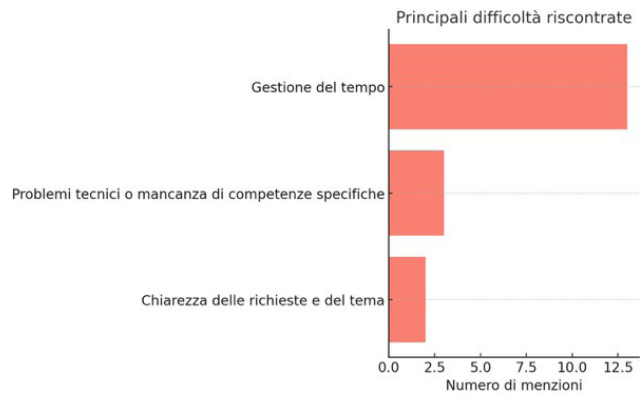


Figura 3: Principali difficoltà riscontrate

Il supporto dei coach ha ricevuto una valutazione media di 3,4 su 5: sufficiente ma **migliorabile**, secondo alcune osservazioni che chiedevano “**tutor più informati**” e un **accompagnamento più tecnico**. Questo dato segnala un’area critica su cui investire per garantire un’esperienza più efficace, soprattutto per i team con minore autonomia iniziale. Tra gli aspetti più apprezzati figurano lo “Sviluppo del prototipo” (7) e ancora una volta il “Lavoro di squadra” (6), confermando l’efficacia del formato esperienziale e collaborativo. Le proposte di miglioramento convergono sulla richiesta di “più tempo a disposizione”, “formazione preliminare” più solida e “maggiore specificità nelle richieste”.

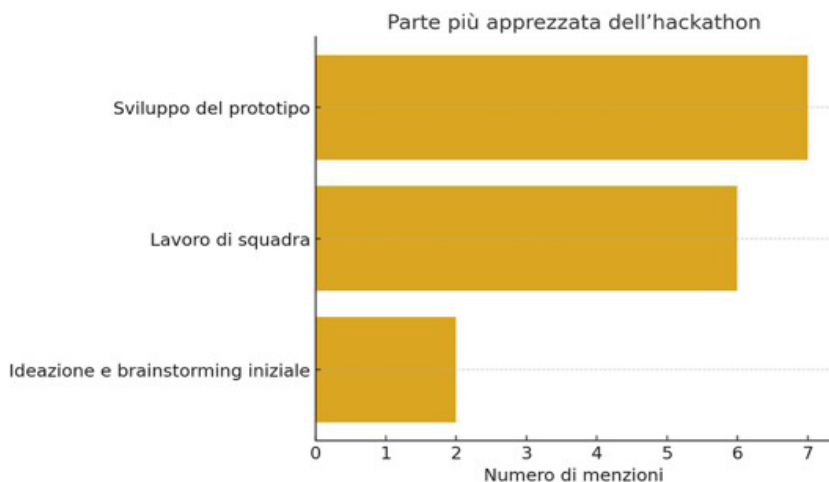


Figura 4: Parte più apprezzata dell'hackathon

5.2 Limiti della valutazione attuale e prospettive future

I dati raccolti tramite il questionario di gradimento non costituiscono una misurazione strutturata e oggettiva delle competenze effettivamente sviluppate. La letteratura enfatizza la necessità di adottare approcci di valutazione più rigorosi per misurare l’impatto di interventi formativi sulle NCS [9, 1]. In risposta a questa esigenza e ai feedback degli studenti (che suggeriscono aree di potenziamento nella preparazione), si propone l’introduzione di un sistema di valutazione più articolato per le future edizioni dell’hackathon. L’approccio valutativo potrebbe includere:

- **Rubriche Osservative.** Utilizzate da docenti e mentor per valutare le soft skill durante le fasi di svolgimento dell’hackathon.
- **Self-Reflection Guidate.** Questionari pre e post evento per stimolare la metacognizione degli studenti sul proprio percorso di apprendimento e sullo sviluppo percepito delle competenze.

Bibliografia

- [1] National Bureau of Economic Research (NBER). Skills, Human Capital, and the Future of Work, 2024
- [2] Groeneveld, M., van der Klink, M., Bücker, J., & Segers, M. Non-cognitive abilities of exceptional software engineers: a Delphi study, 2020
- [3] Briscoe, G., & Mulligan, C. Digital Innovation: The Hackathon Phenomenon, 2014
- [4] Avanesian, G. Do Non-Cognitive Skills Produce Heterogeneous Returns Across Different Wage Levels Amongst Youth Entering the Workforce? A Quantile Mixed Model Approach, 2025
- [5] Sultanova, G. et al., Exploring the influence of non-cognitive skills on academic achievement in STEM education: the case of Kazakhstan, 2024
- [6] Steven, P., Weale, M., Education and Economic growth, 2004
- [7] Bowen, H. (a cura di), Investment in Learning: the individual and social value of American higher education, 2018, Routledge, Londra
- [8] Gorlatova, M. et al., Project-based Learning within a Large-scale Interdisciplinary Research Effort, ITiCSE 2013
- [9] Lipnevich A. et al., Assessing Non-Cognitive Constructs in Education: A Review of Traditional and Innovative Approaches, Oxford Handbooks Online, 2013

Il modello tra problema e soluzione: un percorso tra competizione e consapevolezza

Fabrizio Marinelli^{1*}, Gionata Massi^{2**}

¹ Dipartimento di Ingegneria dell'Informazione Università Politecnica delle Marche Ancona, Italia

fabrizio.marinelli@staff.univpm.it

² Istituto di Istruzione Superiore Savoia Benincasa Ancona, Italia

gionata.massi@savoiaabenincasa.it

Abstract

Il corso presentato in questo articolo mira a integrare le conoscenze multidisciplinari degli studenti, rafforzare i saperi nelle discipline scientifiche e favorire un orientamento consapevole per le scelte post-diploma. I suoi obiettivi formativi includono lo sviluppo di competenze di *problem solving* e modellazione quantitativa, potenziando la sinergia tra informatica e matematica e la motivazione verso le materie STEM¹. Il corso è proposto con una metodologia didattica innovativa che integra: (a) un approccio *problem-based*, (b) elementi di *gamification*, (c) l'attivazione dei processi *metacognitivi* e (d) la *segmentazione* della lezione. Lo sviluppo procede per problemi, in funzione degli schemi cognitivi già formati. Sono stati affrontati alcuni problemi decisionali tipici della programmazione matematica: zaino, rettangolo isoperimetrico di area massima, mix ottimo di produzione e dieta di costo minimo. Dopo una breve introduzione di un problema nuovo o già noto, gli studenti si sfidano nella ricerca di una soluzione al problema dato, riflettono sul loro processo cognitivo, sul quale “scommettono”, e solo dopo aver preso visione delle soluzioni e dei metodi sviluppati, ottengono una breve spiegazione teorica. La spiegazione, che cristallizza i contenuti e i metodi di ricerca, risulta più efficace rispetto alla lezione frontale. L'impatto sul coinvolgimento degli studenti è stato elevato, con significativi risultati nell'attivazione dei meccanismi di *problem solving* e modellazione. Sulla base dei feedback forniti dai questionari di gradimento, il percorso didattico ha pienamente realizzato gli obiettivi prefissati. Il metodo didattico, infine, appare facilmente estendibile ad argomenti quali la programmazione, le basi di dati e l'ingegneria del software.

1. Introduzione

L'esperienza didattica descritta è proposta dall'Istituto di Istruzione Superiore “Savoia Benincasa” di Ancona nell'ambito del DM 65/2023², finalizzato al *potenziamento delle compe-*

* Ha concepito e sviluppato il corso e tutto il materiale didattico. Ha svolto il ruolo di formatore esperto del progetto.

** Ha svolto il ruolo di tutor.

1 STEM è un acronimo che sta per *Science, Technology, Engineering and Mathematics*, ovvero Scienze, Tecnologia, Ingegneria e Matematica.

2 Decreto del Ministro dell'Istruzione e del Merito, 12 aprile 2023, n. 65.

tenze STEM, digitali e di innovazione³. La proposta, frutto della collaborazione tra i dipartimenti di Informatica, Matematica e Scienze Giuridiche, Economiche e Geografiche, si è concretizzata grazie al supporto dell'Università Politecnica delle Marche. Il corso è stato offerto alla classe quinta dell'Istituto Tecnico Economico articolazione *Sistemi Informativi Aziendali*, composta da 16 studenti, essendo l'unica classe a includere l'Informatica nel curriculum. La scelta metodologica, condizionata dai requisiti del DM 65/2023 [5], ha fatto tesoro delle esperienze didattiche, in ambito universitario, del primo autore di questo articolo, ispirate alle principali metodologie per l'insegnamento e l'apprendimento delle discipline STEM nel ciclo terziario [2].

2. Metodologia didattica

La proposta didattica, articolata in fasi diverse in un'ottica di lezione segmentata [4], integra e rielabora vari approcci attivi, fondendoli in un'unica metodologia: si ispira al *Problem-Based Learning* (PBL) [3], ma con una fase risolutiva più estesa e la presenza di risposte oggettivamente verificabili; adotta la *gamification* [1], arricchita da un meccanismo di scommessa che funge da incentivo; infine, stimola la *metacognizione* [8], sia implicitamente, tramite esercizi di modellazione e un meccanismo di scommesse, sia esplicitamente durante la discussione. Questo impianto si traduce in un *serious game*, in cui ogni partecipante ha una dotazione iniziale di 1000 *doblioni* e guadagna doblioni rispondendo correttamente a domande individuali. I partecipanti collaborano in squadra alla risoluzione di problemi e, insieme al proprio gruppo, possono scommettere da 10 a 200 doblioni sulla qualità della propria soluzione, con la possibilità di incrementare o ridurre la propria dotazione in base all'esito. Per l'esperienza descritta, sono state formate quattro squadre di quattro studenti ciascuna, con composizione casuale e invariata per tutta la durata del corso. La struttura del corso è basata sullo studio dei problemi descritti nel par. 3. Ogni problema è affrontato attraverso una sequenza di attività che si sviluppano in modo progressivo: si parte con la presentazione del contesto, seguita da domande individuali su casi semplici o concetti di base. Dopo un primo aggiornamento della classifica e un momento di confronto, viene introdotta una soluzione algoritmica naïve, a cui seguono esempi reali e più complessi. Le squadre si cimentano poi in una sfida su problemi di maggiore difficoltà, seguita da un nuovo aggiornamento della classifica e da una discussione finale. L'attività si conclude con la spiegazione del modello o dell'algoritmo risolutivo.

Le *domande individuali* hanno un tempo di risposta di pochi minuti, in base alla difficoltà, e sono gestite tramite uno strumento online (smartphone o computer). Le *sfide di gruppo* prevedono un tempo di qualche decina di minuti e consistono nella risoluzione di un problema decisionale presentato in modo accattivante e realistico. A differenza dei problemi classici del PBL, quelli proposti ammettono spesso una risposta numerica di cui è possibile verificare l'esattezza ma non l'*ottimalità* e quindi il livello di competitività. Gli studenti utilizzano fogli elettronici di calcolo o carta e penna, e la raccolta dei risultati avviene manualmente oppure tramite strumenti online. Contestualmente alla consegna, ogni gruppo scommette sulla qualità della propria soluzione. L'introduzione della "scommessa" stimola una riflessione metacognitiva

3 Progetto scolastico "Citizen scientists of the future" M4C1I3.1-2023-1143-P-31174, C.U.P. H34D23002330006 finanziato dal Piano nazionale di ripresa e resilienza, Missione 4 – Istruzione e ricerca – Componente 1 – Potenziamento dell'offerta dei servizi di istruzione: dagli asili nido alle università – Investimento 3.1 "Nuove competenze e nuovi linguaggi", finanziato dall'Unione europea – Next Generation EU – "Azioni di potenziamento delle competenze STEM e multilinguistiche".

sul ragionamento che ha condotto alla soluzione, e attiva un “bisogno di verifica” dell’ipotesi formulata. L’induzione di questo desiderio di conoscenza è affine al *Necessity Learning Design* (NLD), ma in questo contesto la *necessità* non origina dall’impossibilità di fornire una soluzione, bensì dalla difficoltà di certificare la correttezza – e spesso l’ottimalità – della soluzione stessa. Al termine della fase di attribuzione dei punteggi, durante la quale il formatore mostra pubblicamente le soluzioni e ne valuta le performance, uno studente per gruppo illustra le strategie adottate rendendo esplicito il processo cognitivo. Infine, il formatore spiega la teoria, innescando la conoscenza formale quando gli studenti ne dovrebbero aver percepito la necessità.

3. Problemi e sfide

Questa sezione descrive i problemi proposti nel corso, evidenziando i contenuti e le competenze disciplinari sviluppate.

3.1 *Problema dello Zaino* [7]

Il corso ha introdotto il Problema dello Zaino attraverso un’attività progressiva: inizialmente, una sfida individuale ha invitato gli studenti a selezionare, tra quattro oggetti con peso e valore assegnati, quelli da inserire in un bagaglio a mano per massimizzare il valore senza superare un limite di peso. In seguito, è stata analizzata l’enumerazione esaustiva delle possibili combinazioni. Il problema è stato poi contestualizzato con esempi reali, come il caricamento di container. Nella sfida di gruppo, gli studenti hanno affrontato l’ottimizzazione del carico di un furgone con 50 oggetti. La discussione conclusiva si è concentrata sulle strategie adottate dai vari gruppi e sull’importanza della verifica dell’ottimalità delle soluzioni proposte.

3.2 *Problema del rettangolo isoperimetrico di area massima*

Successivamente, allo scopo di mettere a confronto l’approccio procedurale, centrato sulla descrizione dei passi che portano al calcolo di una soluzione, con quello dichiarativo, focalizzato invece sulla descrizione delle caratteristiche della soluzione stessa, è stato proposto il problema di determinare il rettangolo di area massima, dato il perimetro. Le domande individuali hanno guidato gli studenti nell’esplorazione delle possibili soluzioni che, successivamente, sono state confrontate con quella ottenuta tramite un primo esempio di modello di programmazione matematica. È stato quindi introdotto il linguaggio AMPL per la programmazione matematica e, nella seconda sfida di gruppo, gli studenti hanno formalizzato un problema decisionale, specificando variabili decisionali, funzione obiettivo e vincoli.

3.3 *Ancora sul Problema dello Zaino*

Si è poi ritornati al Problema dello Zaino per un approfondimento metacognitivo sulle strategie risolutive. Sono state proposte sfide individuali sulla determinazione del numero di sottoinsiemi di un insieme dato. La discussione ha portato alla comprensione dell’impraticabilità della ricerca esaustiva nella soluzione di questo e altri problemi reali analoghi, evidenziata dal numero esponenziale di soluzioni ammissibili (250 nel caso proposto). Dal problema reale, quindi, sono state astratte le entità del modello di programmazione matematica (variabili, funzione obiettivo, vincoli), richiamando concetti già noti come variabili booleane, array e codifiche binarie.

3.4 Programmi lineari nel piano: un esempio di mix ottimo di produzione [6]

Dai problemi combinatori si è passati alla programmazione lineare, introdotta attraverso una sfida di gruppo su un problema di mix ottimo di produzione in due variabili. Lo sviluppo della conoscenza è proseguito con domande individuali mirate all'identificazione delle variabili decisionali, dei vincoli e della funzione obiettivo, nonché alla ricerca della soluzione ottima. La discussione delle strategie adottate ha messo in luce principi riconducibili al metodo del simplesso e al test di ammissibilità. Attraverso un processo di elicitazione guidata, è stato costruito il modello matematico completo e illustrato il metodo del simplesso in forma grafica. La successiva sfida di gruppo ha previsto l'applicazione del metodo grafico del simplesso a un'istanza di programma lineare, richiamando conoscenze pregresse su rette, derivate, gradienti e curve di livello.

3.5 Programmazione Lineare: il problema della dieta economica [6]

Infine, è stata proposta una sfida di gruppo su un problema di programmazione lineare in più variabili, nello specifico il problema della dieta economica. Anche per questo problema, tramite domande individuali, sono state identificate le variabili di decisione, i vincoli e la funzione obiettivo. È seguita un'illustrazione sui concetti di spazi a dimensioni maggiori di tre, fornendo le basi per la risoluzione di problemi di programmazione lineare più complessi.

4. Risultati

Gli studenti hanno mostrato un coinvolgimento e una motivazione mediamente superiori rispetto alle lezioni tradizionali: 15 su 16 hanno partecipato ad almeno 7 ore del corso, con un'unica assenza prolungata dovuta a motivi di salute. La maggior parte ha manifestato un rinnovato interesse per matematica e informatica, percepite in una nuova luce. I questionari anonimi confermano questa tendenza: 10 studenti su 15 hanno ritenuto il formatore *capace di suscitare interesse e coinvolgimento*, 9 hanno giudicato il corso *utile per i contenuti*, 8 per le *abilità/capacità sviluppate*, e 7 per l'*interesse complessivo*.

Le competenze di *problem solving* e modellazione sono emerse chiaramente nelle sfide di gruppo e nelle domande individuali, dove gli studenti hanno saputo analizzare i problemi, individuare le entità rilevanti e tentare una formalizzazione. Il meccanismo della scommessa ha favorito la riflessione critica sulla validità delle strategie adottate, pur non essendo al centro delle discussioni nei gruppi che invece erano principalmente rivolte alla soluzione dei problemi.

La spiegazione formale è risultata più efficace dopo il confronto attivo con i problemi, favorendo una comprensione più profonda del ruolo complementare di matematica e informatica, in particolare nella distinzione tra linguaggi dichiarativi (come il linguaggio matematico o SQL) e procedurali. Il lavoro di gruppo ha coinvolto quasi tutti: con l'unica eccezione di uno studente meno motivato, anche gli studenti solitamente meno partecipi hanno contribuito attivamente alle sfide.

5. Conclusioni

L'esperienza ha evidenziato l'efficacia dell'approccio didattico proposto nel promuovere coinvolgimento, motivazione e sviluppo di competenze di *problem solving* e modellazione, soprattutto nei contesti in cui matematica e informatica si intrecciano. L'integrazione di elementi

ludici e la stimolazione metacognitiva hanno favorito una comprensione più profonda dei concetti formali. Il modello si è dimostrato coerente con gli obiettivi formativi del DM 65/2023, sostenendo la costruzione di competenze operative attraverso un approccio attivo. La *gamification* ha rafforzato motivazione e collaborazione, mentre il meccanismo della scommessa ha stimolato riflessione critica e autonomia nella validazione delle soluzioni. La teoria, introdotta dopo il confronto empirico, è risultata più significativa, e l'alternanza tra fasi pratiche e momenti di ascolto ha contribuito all'efficacia complessiva. Tuttavia, un limite rilevato riguarda la durata del corso: le 10 ore disponibili si sono rivelate strette; 15 ore avrebbero consentito un consolidamento più approfondito delle competenze.

Il modello didattico è facilmente adattabile ad altri temi di informatica o matematica e, secondo gli autori, si presta in modo particolare al Liceo Scientifico (indirizzo Scienze Applicate), mentre per altri ordini o annualità, i contenuti andrebbero modulati in base alle capacità di astrazione della specifica fascia d'età e alle conoscenze pregresse. In particolare, si ritiene naturale estendere questa metodologia ad altre aree dell'informatica. Ad esempio, nella programmazione dei calcolatori, si potrebbe definire un percorso che richieda l'uso di strutture di controllo o di strutture dati a complessità crescente, mentre nella progettazione di basi di dati e nell'ingegneria del software, si potrebbero proporre problemi con pattern strutturali comuni, come le cardinalità delle associazioni presenti o i *design pattern*.

In prospettiva, sarebbe utile sviluppare una versione online del sistema di punteggio come *proof of concept* per una piattaforma didattica, e proporre/realizzare una seconda edizione del corso, accompagnata da un monitoraggio sistematico dell'efficacia e dell'acquisizione delle competenze che possa supportare con dati più robusti le osservazioni qualitative finora raccolte.

Bibliografia

- [1] Patrick Buckley e Elaine Doyle. «Gamification and student motivation». In: *Interactive learning environments* 24.6 (2016), pp. 1162–1175.
- [2] Richard M. Felder e Rebecca Brent. *Teaching and Learning STEM: A Practical Guide*. Hoboken, New Jersey: John Wiley & Sons, 2016.
- [3] Cindy E Hmelo-Silver. «Problem-based learning: What and how do students learn?» In: *Educational psychology review* 16.3 (2004), pp. 235–266.
- [4] Dany Maknouz. *La lezione segmentata: Ritmata, varia, integrata*. Bologna: Zanichelli, 2021. ISBN: 978-8808720184.
- [5] Ministero dell'Istruzione e del Merito. *Linee guida per le discipline STEM*. Documento ufficiale del Ministero dell'Istruzione e del Merito. 2023. URL: <https://www.mim.gov.it/documents/20182/0/Linee+guida+STEM.pdf/2aa0b11f-7609-66ac-3fd8-2c6a03c80f77> (visitato il giorno 31/05/2025).
- [6] Carlo Vercellis. *Ottimizzazione. Teoria, metodi, applicazioni*. Milano: McGraw-Hill Education, 2008. ISBN: 9788838664427.
- [7] Wikipedia contributors. *Problema dello zaino*. Wikipedia, l'enciclopedia libera. 2025. URL: https://it.wikipedia.org/wiki/Problema_dello_zaino (visitato il giorno 04/06/2025).
- [8] Aman Yadav, Haeseong Hong e Zhenyu Liu. «Computational Thinking and Metacognition». In: *The Cambridge Handbook of Computing Education Research*. A cura di Sara Fincher e Joshua Tenenbergs. Cambridge University Press, 2022, pp. 253–276. DOI: 10.1017/9781108781496. 012.

Codice che “pensa”. Modelli ed algoritmi di intelligenza artificiale con Python per la didattica STEAM

Flavia Giannoli¹

¹ Liceo scientifico A. Volta, MIM

flavia.giannoli@gmail.com

Abstract

L'articolo presenta un approccio innovativo per introdurre modelli ed algoritmi di intelligenza artificiale nella didattica scolastica attraverso Python. Si propone un percorso graduale che parte dall'esplorazione dei dati (dataset Iris) per arrivare alla classificazione supervisionata e al clustering non supervisionato. La metodologia privilegia l'approccio pratico su quello teorico, utilizzando strumenti come Google Colab e librerie come Scikit-learn, Pandas, Matplotlib. Questo approccio introduttivo diventa trasversale alle discipline, coinvolgendo studenti su competenze diverse e sviluppando pensiero critico. L'esperienza ha dimostrato come Python faciliti l'apprendimento grazie alla sua sintassi semplice; inoltre, i progetti basati sull'elaborazione di dati reale tratti dai diversi dataset anche istituzionali accessibili in rete hanno trasformato l'aula in laboratorio attivo, contribuendo a preparare cittadini più consapevoli.

1. Introduzione

L'Informatica nelle nostre scuole sta cambiando pelle. Non più solo Scratch, fogli Excel o presentazioni PowerPoint e Canva, ma c'è un campo tutto da esplorare al centro della rivoluzione digitale. E in questo panorama, l'approccio ai modelli ed algoritmi di intelligenza artificiale brilla e attira curiosità e domande. Dai filtri che trasformano i propri selfie in ritratti rinascimentali o punk con facilità, agli algoritmi usati nei reparti di oncologia per analizzare mammografie, all'app che consiglia cosa guardare la sera, gli esempi-stimolo portati in classe non mancano mai di suscitare reazioni: “Quindi è così che Netflix sa cosa mi piace?”. Esattamente. “Ma quindi Netflix mi suggerisce solo film che confermano quello che già mi piace?” “Ma davvero i computer imparando da soli?”. Così nascono discussioni spontanee sulle Echo Chambers e sulle responsabilità degli algoritmi. Lavorare su algoritmi di intelligenza artificiale implica il coinvolgimento di interessi diversificati negli studenti: chi eccelle in matematica mette in pratica in modo autonomo e critico le formule; chi ama le scienze sociali si appassiona nell'analisi dei bias negli algoritmi; chi è portato per il pensiero critico si diverte a interrogare i risultati. È come avere a disposizione una palestra dove allenare contemporaneamente diversi “muscoli” mentali. È stato scelto Python come importante alleato in questa avventura didattica. Quando viene mostrato agli studenti come bastino pochissime righe di codice per classificare un'immagine, perfino i più scettici alzano la mano per provare. Ma, se la barriera all'ingresso è bassa, la volta della stanza a cui si accede sarà altissima! L'importante è iniziare.

Ciò che segue è una proposta di percorso per classi terze di Liceo scientifico, un quaderno di appunti di un'insegnante convinta che comprendere gli algoritmi che sempre più plasmano il nostro mondo sia importante quanto conoscere i meccanismi della fotosintesi o i sonetti di Dante e Petrarca. Il percorso è offerto su base volontaria.

2. Perché utilizzare Python

L'obiettivo è introdurre Python come strumento didattico per sviluppare il pensiero computazionale, aiutando a sviluppare logica e Problem solving, e per promuovere la creatività digitale e avvicinare gli studenti alle competenze richieste dal mondo moderno e professionale scientifico, soprattutto negli ambiti come la Data science, il Machine learning e l'Intelligenza artificiale. Python è uno dei linguaggi di programmazione di alto livello più diffusi ed è considerato il linguaggio ideale per chi inizia a programmare perché è semplice da imparare, ha una sintassi intuitiva ed è diffusamente adottato dalle principali università e aziende innovative.

La sintassi di Python è molto semplice e leggibile per gli alunni. Niente graffe, niente punti e virgola, solo l'indentazione che rispecchia la struttura logica. Non più pagine e pagine di codice: per scrivere un classificatore di immagini bastano una decina di righe. Inoltre, scrivere "if numero > 10" è quasi come leggere l'italiano. Python è come un buon amico: non ti fa sentire inadeguato quando commetti errori. Così gli studenti si possono concentrare sul problema piuttosto che sul combattimento con la sintassi.

La vera magia arriva quando si introducono le librerie: Python è un ecosistema ricco di librerie (es. Scikit-learn, Pandas, Matplotlib, Tensorflow, Numpy). Dire "Ragazzi, oggi vi presento Scikit-learn", è quasi come presentare una rockstar. E in effetti un po' lo è. È impagabile vedere gli occhi che si illuminano quando gli allievi capiscono che con `model.fit(X, y)` stanno addestrando un modello matematico complesso.

L'utilizzo delle librerie in questa esperienza è stato black-box, ma gli studenti hanno appreso i rudimenti per la modifica del codice. In futuro sarà possibile analizzare qualsiasi tipo di dato. Per esempio, divertirsi con quelli delle recensioni online del ristorante vicino alla scuola: Pandas per gestire i dati, Matplotlib per creare i grafici e con Scikit-learn sarà possibile fare previsioni sul gradimento dei nuovi piatti.

Gli strumenti educativi di Google Colab (potente strumento gratuito offerto da Google), che consente agli utenti di scrivere ed eseguire codice Python direttamente nel browser, come anche Jupyter Notebook sono altri notevoli punti di forza. È come usare un quaderno, ma "vivo": poter inserire spiegazioni, codice, output e grafici nello stesso documento trasforma il modo in cui i ragazzi studiano e collaborano. Python si dimostra essere anche un ottimo strumento per l'inclusione digitale perché abbassa le barriere tecniche e offre pari opportunità.

La cosa più gratificante è vedere gli studenti lavorare in gruppo, discutendo animatamente se il loro modello di regressione dovrebbe usare più o meno features; non stanno solo imparando Python, ma stanno facendo un po' di Data Science. Stanno imparando a pensare criticamente, a testare ipotesi, a interpretare risultati. Ed è questo il punto focale.

3. Un possibile percorso didattico

Obiettivi Formativi:

- Comprendere i concetti base del machine learning e della programmazione;
- Stimolare il pensiero computazionale;
- Applicare Python a problemi pratici e multidisciplinari;
- Eseguire semplici applicazioni digitali.

Valutazione:

- Partecipazione attiva;
- Progetto di gruppo;
- Quiz interattivi con Kahoot (valutazione formativa) e Exam.net (valutazione sommativa).

Attività proposte:

Cominciare con la teoria astratta del codice delle reti neurali per insegnare il machine learning (ML) significherebbe perdere metà della classe dopo neanche dieci minuti. È più utile introdurre i principi di funzionamento brevemente ed in modo accattivante, per esempio mostrando filmati ed animazioni del funzionamento delle reti neurali, e poi partire subito dai dati e dalle storie che essi raccontano. Il codice verrà dopo, quasi naturalmente.

È fondamentale introdurre i concetti del ML in modo graduale, partendo dagli algoritmi:

1. Dati e rappresentazione;
2. Modelli supervisionati: classificazione e regressione;
3. Modelli non supervisionati: clustering.

Un classico dataset utilizzabile nell'ambito dell'apprendimento automatico come esempio di classificazione statistica è il Dataset Iris, un dataset multivariato introdotto da Ronald Fisher nel 1936. Esso consiste in 150 istanze di Iris misurate da Edgar Anderson e classificate secondo tre specie (Iris setosa, Iris virginica e Iris versicolor), tenendo conto di quattro variabili: la lunghezza e la larghezza del sepal e quelle del petalo.

3.1 Fase 1. Esplorazione dei dati

“Prof, ma questi fiori Iris sono tutti uguali!” è stato il commento quando ho mostrato le prime immagini, ma è bastato caricare il dataset “Iris” e visualizzare i primi grafici per avviare discussioni animate (Fig. 1). Gli occhi si sono illuminati quando si sono visti i puntini colorati raggrupparsi in modo decisamente distinto. “È come se i fiori si mettessero in fila da soli!”, disse qualcuno. Esattamente quello che volevo sentire.

Questa prima attività ha aiutato a comprendere i dataset tabulari, visualizzare le relazioni tra variabili, introdurre il concetto di “classe” come categoria da predire.

```

from sklearn.datasets import load_iris
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Caricamento e visualizzazione dei dati
iris = load_iris()
df = pd.DataFrame(iris.data, columns=iris.feature_names)
df['species'] = iris.target

# Grafico a dispersione
sns.pairplot(df, hue="species")
plt.show()

```

Figura 1: Pseudocodice di analisi del dataset “Iris”

3.2 Fase 2. Apprendimento supervisionato: classificazione

Il passo successivo è stato naturale. Alla classe è stato domandato: “Se noi vedessimo un nuovo fiore, come faremmo a capire di che specie è?” Un'alunna molto pragmatica ha risposto: “Guarderei quanto è simile agli altri che già conosco.” Ottimo! Ecco, quindi, l'introduzione del classificatore con K-Nearest Neighbors (KNN) (Fig 2):

```

from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score

# Divisione tra training e test
X_train, X_test, y_train, y_test = train_test_split(
    iris.data, iris.target, test_size=0.3, random_state=42)

# Creazione e addestramento del modello
model = KNeighborsClassifier(n_neighbors=3)
model.fit(X_train, y_train)

# Predizione e valutazione
y_pred = model.predict(X_test)
print("Accuratezza:", accuracy_score(y_test, y_pred))

```

Figura 2: Pseudocodice di classificazione con KNN

Quando il programma ha ottenuto un'accuratezza del 97%, c'è stato un applauso spontaneo in aula. “Ma come fa a essere così preciso?”. Ottima domanda, che ci ha portato a discutere di

divisione del dataset in training set e test set, di addestramento del modello, di over fitting e di valutazione delle prestazioni al variare della proporzione dei dati di training e di test.

3.3 Apprendimento non supervisionato: clustering

“E se non sapessimo già quali sono le specie?” Con questa domanda sono stati introdotti i concetti di modelli esplorativi e di clustering. Si è deciso quindi di sfidare il computer: “Vediamo se riesce a scoprire da solo i gruppi”, cioè di confrontare cluster trovati automaticamente con le specie reali. Quando è apparso il grafico i gruppi trovati dal computer sono risultati quasi identici alle specie reali. “È come se avesse indovinato!” esclamò qualcuno. Il momento perfetto per parlare di come i pattern nei dati possono essere scoperti automaticamente dagli algoritmi.

3.4 Approccio laboratoriale e inclusivo

Molti casi reali potrebbero essere qui proposti: la previsione dei voti finali (regressione), o l'analisi della distribuzione degli studenti per indirizzo scolastico o per regione, lo studio della dispersione scolastica, il confronto tra numero di diplomati e risultati INVALSI (a partire da Open data del MIM), oppure l'analisi della mobilità e dei trasporti pubblici (dati dei Comuni) ed anche previsioni di fenomeni complessi (dati ARPA, Protezione Civile, Sanità, Dataset Open).

Dietro ogni algoritmo da proporre in classe c'è una storia, una domanda, un problema reale da risolvere che necessita di accurata analisi dei dati, formulazione di ipotesi, interpretazione dei risultati e discussione in gruppo per le decisioni e sugli errori.

```
from sklearn.cluster import KMeans

# Modello K-means
kmeans = KMeans(n_clusters=3, random_state=42)
kmeans.fit(iris.data)

# Aggiunta dei cluster al dataframe
df['cluster'] = kmeans.labels_

# Visualizzazione
sns.scatterplot(data=df, x="petal length (cm)", y="petal width (cm)",
                hue="cluster", palette="deep")
plt.title("Cluster trovati da KMeans")
plt.show()
```

Figura 3: Pseudocodice di Clusterizzazione

Questa consapevolezza vale più di mille righe di codice: si chiama competenza. L'uso di ambienti condivisi in Google Colab inoltre può facilitare l'inclusione e la collaborazione tra studenti con competenze eterogenee: gli studenti più avanzati spesso supportano quelli in difficoltà, spiegando i concetti con parole loro.

4. Sfide e opportunità nella scuola

Tra registri e verbali da compilare, riunioni e programmazioni didattiche, chiedere ai docenti di diventare anche esperti di intelligenza artificiale è pura utopia. Spesso si aggiunge il problema dei computer presenti nella scuola, spesso obsoleti. Ci viene però incontro la semplicità d'uso di Python ed anche il fatto che Google Colab funziona anche sullo smartphone e permette a tutti di addestrare e fare inferenza con i modelli di algoritmi. Inoltre, con Kaggle si possono facilmente trovare dataset su tutto, dal cambiamento climatico alle canzoni di Sanremo. Molti pensano che si debbano creare materie nuove, ma questo approccio algoritmico può infiltrarsi ovunque: in matematica per dare senso alle statistiche, in scienze per analizzare dati ambientali, in italiano per studiare l'evoluzione del linguaggio, in arte per classificare le immagini a seconda dei diversi stili pittorici. La vera sfida non è tecnica, è di visione: creare un progetto sui modelli predittivi dell'inquinamento locale riguarda la preparazione attiva di futuri cittadini.

5. Riflessioni conclusive

Ripensando al percorso fatto proponendo Python per esplorare gli algoritmi di intelligenza artificiale, vedo ragazzi che discutono animatamente se il loro modello predittivo sulla qualità dell'aria funziona male per colpa dei dati incompleti o per un errore nel pre-processing. Non è solo tecnica: è pensiero critico in azione. C'è chi in matematica fa un po' di fatica ed ha imparato a scomporre un problema complesso in passaggi più piccoli perché questo gli viene naturale quando programma. Altri, solitamente silenziosi, che si sono trovati a guidare un dibattito acceso sui bias di genere nei sistemi di riconoscimento vocale. Python è stata una scommessa vincente: la curva di apprendimento iniziale non ha scoraggiato nessuno: "Prof, ma questo è più facile dell'inglese". Le librerie Scikit-learn ci hanno permesso di concentrarci sulle domande importanti invece che perderci nella sintassi. Il progetto sui dati della centralina meteo locale ha trasformato l'aula in un laboratorio vivo, con telefoni che per una volta non servivano per i social ma per visualizzare grafici e previsioni. E quando i ragazzi del gruppo "recommender system" hanno presentato alla classe un sistema che suggeriva libri basandosi sui gusti di ciascuno, perfino i più disinteressati hanno cominciato a fare domande. Non sono sicura che tutti diventeranno data scientist, ma ho la certezza che guarderanno agli algoritmi che ormai governano occultamente le nostre vite con occhi più consapevoli. Questa è la lezione più importante.

Bibliografia

- [1] Barberis, Giuliana e Dari, Emil. "Informatica bit a bit." Tramontana (2025).
- [2] Blog Boolean Careers: <https://boolean.careers/blog/10-motivi-per-cui-dovresti-imparare-a-programmare-in-python>
- [3] Matthes, Eric. "Python. Introduzione pratica alla programmazione ad oggetti." Ed. LSWR (2024)

Tre concetti chiave per un corso sul Machine Learning alle scuole superiori

Nicola Dalla Pozza¹, Emanuele Scapin¹

¹ ITT G.Chilesotti, Thiene, Vicenza, Italia

ndallapozza@chilesotti.it, escapin@chilesotti.it

Abstract

Negli ultimi due anni è stato proposto ai nostri studenti dell'Istituto Tecnico Tecnologico un corso introduttivo sull'Intelligenza Artificiale (IA). Questo contributo descrive i contenuti chiave attorno ai quali è stato progettato: 1) le reti neurali sono un modello computazionale in grado di approssimare qualsiasi funzione matematica; 2) l'addestramento delle reti necessita di grandi moli di dati, evidenziando l'impegno necessario per ottenere esempi di qualità; e 3) il paradigma di programmazione delle reti neurali è ben diverso dalla programmazione algoritmica "tradizionale" cui sono abituati gli studenti.

1. Introduzione

L'ubiquità delle tecnologie basate sull'Intelligenza Artificiale (IA) sta rapidamente ridefinendo il panorama tecnologico e sociale. In questo contesto, l'apprendimento automatico (machine learning) emerge come una competenza computazionale fondamentale, destinata a integrare e, in parte, trasformare l'attuale didattica della programmazione. Riconoscendo l'importanza di questa transizione, il nostro Istituto Tecnico Tecnologico ha introdotto negli ultimi due anni un corso propedeutico di IA. Tale iniziativa mira a fornire agli studenti una solida base concettuale e operativa in questo campo, stimolando al contempo un interesse critico che possa guidarli nelle future scelte professionali e accademiche.

Per facilitare l'integrazione di contenuti innovativi e tecnicamente complessi, tipicamente assenti dai curricula tradizionali, con l'esigenza di offrire esperienze didattiche accessibili agli studenti del triennio superiore, il corso è stato strutturato attorno a tre concetti fondamentali delle reti neurali (si veda la sezione "Metodologie"). La didattica si avvale di attività laboratoriali di gruppo, impiegando Google Colaboratory per lo sviluppo di codice e kit Arduino Explore IoT per applicazioni pratiche, sfruttando la strumentazione scolastica disponibile.

2. Contesto di riferimento

Negli ultimi anni, l'integrazione dell'Intelligenza Artificiale (IA) nell'educazione scolastica è stata oggetto di numerosi studi, evidenziando l'urgenza di introdurre tali concetti fin dalla scuola primaria [1, 2]. Un aspetto cruciale in questo processo è la formazione degli insegnanti,

i quali devono affrontare le sfide poste dall'IA e rinnovare le proprie metodologie didattiche in un contesto in rapida evoluzione. Linee guida internazionali, come quelle emanate dalla Commissione Europea sull'uso dei dati e dell'IA nell'educazione, offrono riferimenti essenziali per armonizzare le pratiche educative con gli standard globali [3, 4]. L'integrazione efficace del Machine Learning (ML) nei curricula scolastici è possibile tramite attività pratiche e coinvolgenti [5], anche in assenza di pregresse competenze avanzate. Il coinvolgimento diretto degli studenti nella costruzione e nel test di sistemi ML rafforza la motivazione e il pensiero computazionale [6], mentre l'ambiente collaborativo stimola l'apprendimento tra pari [7].

Un recente studio di Tenberga e Daniela [8] si concentra sulle competenze di alfabetizzazione all'IA richieste ai docenti. La ricerca, basata su questionari e analisi statistica (PCA), rivela che tali competenze costituiscono una categoria autonoma ma strettamente interconnessa alla competenza digitale generale. I risultati sottolineano l'urgenza di sviluppare programmi di formazione professionale specifici per supportare un'efficace adozione dell'IA nelle pratiche educative. In un'ottica complementare all'introduzione dell'IA, la robotica ha acquisito crescente rilevanza nell'ultimo decennio come strumento didattico. Questo ha creato un terreno fertile per l'esplorazione di applicazioni pratiche, specialmente attraverso piattaforme come Arduino, che si integrano perfettamente con i concetti di intelligenza artificiale sviluppati nel nostro corso. Sebbene i primi studi si siano concentrati sull'istruzione universitaria [9], ricerche più recenti si focalizzano sull'educazione K-12 [10]. Diversi lavori [11, 12] evidenziano come la robotica in classe stia evolvendo da semplice supporto tecnologico a strumento didattico multidisciplinare, capace di stimolare motivazione, creatività e inclusività. Inoltre, la robotica contribuisce allo sviluppo del pensiero computazionale e all'interesse verso le discipline STEM [13]; essa alimenta anche effetti positivi su creatività e problem solving [14]. L'efficacia delle attività robotiche, tuttavia, dipende dall'età degli studenti, dalla durata del programma e dalla qualità della progettazione didattica.

L'utilizzo del kit Arduino Explore IoT ha permesso di sfruttare i vantaggi e le opportunità che questo offre agli studenti delle scuole secondarie [15, 16]. Infatti, uno dei metodi più promettenti per la formazione nell'ambito delle discipline STEM è il metodo dei progetti [17], che consente agli studenti di analizzare attivamente un problema reale, applicare le conoscenze acquisite nella pratica e sviluppare capacità di cooperazione, comunicazione e pensiero critico. L'implementazione di progetti con Arduino contribuisce quindi alla combinazione di diverse attività – cognitive, creative, pratiche e di ricerca – promuovendo al contempo lo sviluppo di capacità di pensiero, motivazione e senso di responsabilità. In questo contesto, il nostro corso si inserisce come esempio pratico di tale integrazione.

3. Metodologia

Il corso, della durata complessiva di 10 ore distribuite su quattro pomeriggi nel primo quadrimestre, è offerto su base volontaria a un massimo di 24 studenti del biennio finale, con priorità ai diplomandi. La selezione di questo target di studenti è motivata dalla necessità di una preesistente familiarità con la programmazione tradizionale, considerata propedeutica alla piena comprensione delle tecnologie trattate. Le sessioni si svolgono in laboratorio e alternano lezioni frontali, supportate da slide condivise con gli studenti¹, a esperienze pratiche che richiedono l'applicazione di competenze di programmazione.

¹ La presentazione introduttiva è reperibile al link https://gitlab.com/nicola.dallapozza/three-ai-key-concepts/-/raw/main/3_AI_key_Concepts_-_Intro.pdf?inline=false

Il corso è stato concepito con una sezione introduttiva dedicata alla terminologia chiave, differenziando esplicitamente concetti quali Intelligenza Artificiale (Artificial Intelligence), apprendimento automatico (Machine Learning) e reti neurali (Neural Networks). Successivamente, viene introdotto il linguaggio di programmazione Python. Sebbene Python non sia parte del curriculum standard della nostra istituzione, che privilegia linguaggi fortemente tipizzati come C# e Java, la sua inclusione è giustificata dal suo ampio utilizzo nella comunità del Machine Learning. Successivamente, vengono esplorati i paradigmi dell'apprendimento supervisionato e con rinforzo attraverso esperienze laboratoriali. La trattazione si concentra su tre concetti chiave:

1. Le reti neurali come approssimatori universali: Si evidenzia la capacità delle reti neurali di approssimare con precisione arbitraria qualsiasi funzione [18, 19]. A differenza della programmazione algoritmica tradizionale, che si basa su sequenze di istruzioni, selezioni e cicli, le reti neurali raggiungono tale universalità attraverso una struttura di unità fondamentali di calcolo (neuroni artificiali) organizzata in livelli, collegati tra loro per scambiarsi dati. Vengono presentati esempi concreti che dimostrano questa capacità, come la mappatura dall'insieme dei pixel di un'immagine alla descrizione del suo contenuto semantico, collegando così due contesti apparentemente eterogenei.
2. Addestramento delle reti e necessità di grandi volumi di dati: Una volta definita l'architettura di una rete neurale, la sua "calibrazione" comporta l'ottimizzazione dei parametri interni per raggiungere una configurazione con buone prestazioni. L'addestramento è presentato come un processo iterativo di perfezionamento graduale di tali parametri. Si enfatizza la necessità di disporre di un'ingente quantità di dati di alta qualità, ovvero accurati, rappresentativi e mancanti di errori per il problema affrontato, e si discute la complessità e l'onere associati alla loro raccolta e pre-elaborazione. Questo concetto viene direttamente sperimentato dagli studenti durante le sessioni pratiche di laboratorio.
3. La distinzione dalla programmazione tradizionale: La metodologia di sviluppo basata su reti neurali si articola principalmente in tre fasi: raccolta e preparazione dei dati, addestramento (training) del modello utilizzando tali dati, e inferenza (inference), ovvero l'applicazione della rete addestrata a nuovi input. Questo paradigma di programmazione si discosta notevolmente dall'approccio algoritmico a cui gli studenti sono tipicamente abituati.

Le sessioni laboratoriali dedicate all'apprendimento supervisionato hanno incluso due esperienze principali: la "Classificazione di cifre scritte a mano" e il "Riconoscimento di gestualità tracciate in aria". La prima, ampiamente usata nella trattazione del Machine Learning, mira a classificare immagini di cifre manoscritte, identificando il numero corrispondente. Questo compito si rivela arduo per la programmazione tradizionale, che richiederebbe la definizione algoritmica di innumerevoli pattern (ad esempio, le diverse varianti di un cerchio per lo zero). Al contrario, una semplice rete neurale consente di ottenere facilmente prestazioni di classificazione significativamente superiori, dimostrando l'efficacia di questo approccio. In questa esperienza, gli studenti hanno lavorato su un notebook Colaboratory contenente codice Python², rielaborato da [20]. È importante sottolineare che non sono state utilizzate librerie esterne: la definizione e l'addestramento della rete sono stati implementati direttamente in Python,

2 Il notebook Colaboratory è scaricabile dal link https://gitlab.com/nicola.dallapozza/three-ai-key-concepts/-/raw/main/Handwritten_digits_classification.ipynb?inline=false. Dopo averlo aperto con l'App Google Colab (<https://colab.research.google.com/>) e aver eseguito le celle delle sezioni "Handwritten digit classification", "Codice" ed "Importazione codice", gli studenti completano le celle della sezione "Creazione della rete neurale" (già completate nel link a disposizione) ed eseguono le celle successive, effettuando così training e test della rete.

permettendo agli studenti interessati di analizzare a fondo il codice, completando poi poche righe essenziali per la costruzione, l'addestramento e il testing della rete.

La seconda esperienza è stata introdotta con l'obiettivo di applicare la tecnologia precedente in un diverso contesto: il riconoscimento di movimenti eseguiti in aria, come ad esempio simboli. A tale scopo, abbiamo impiegato i kit "Arduino Explore IoT", che include un Arduino MKR Wi-Fi 1010 montato su una scheda dotata di accelerometro. Il kit rileva l'accelerazione sui tre assi durante l'esecuzione del gesto e trasmette la sequenza di misurazioni tramite monitor seriale, fungendo da sorgente dati per l'input della rete neurale.

Abbiamo fornito agli studenti il codice Arduino preimpostato per l'acquisizione dei dati e un notebook Colaboratory contenente il codice per la definizione e l'addestramento della rete. Agli studenti è stato richiesto di selezionare le cifre o i simboli da classificare, raccogliere circa cinquanta sequenze di misurazioni per ciascun simbolo tramite il monitor seriale, definire autonomamente la rete neurale all'interno del notebook Colaboratory, addestrare e testare il modello risultante. Questa attività ha messo in evidenza come la fase di raccolta e preparazione dei dati possa spesso richiedere un impegno maggiore e rivelarsi più complessa rispetto all'effettivo processo di addestramento delle reti neurali.

4. Risultati

Al termine del corso, è stato somministrato un questionario di gradimento che utilizzava una scala di Likert da 0 (valutazione molto negativa) a 10 (valutazione molto positiva). I risultati indicano un elevato interesse e partecipazione alle lezioni (valutazione media di 9.0/10) e una elevata auto-valutazione della comprensione dei fondamenti teorici da parte degli studenti (valutazione media di 8.2/10). La difficoltà percepita della parte laboratoriale è risultata moderata (valutazione media di 3.8/10, dove 0/10 indica nessuna difficoltà e 10/10 difficoltà elevata). In media, gli studenti hanno espresso un giudizio positivo sull'adeguatezza del tempo dedicato (8.6/10) e sull'utilità del corso sulle future scelte professionali ed universitarie (7.9/10).

L'introduzione a Python non ha rappresentato un ostacolo significativo durante le sessioni laboratoriali. Ciò è probabilmente attribuibile sia alla quantità ridotta di codice che gli studenti dovevano produrre, sia alla loro pregressa esperienza con altri linguaggi di programmazione.

È degno di nota il fatto che un gruppo di studenti ha sviluppato autonomamente, nell'ambito di un progetto curricolare, una web application e un'applicazione mobile Android con architettura client-server per il riconoscimento di cifre disegnate. Questo mostra la loro capacità di adattare e integrare le tecnologie apprese in un contesto applicativo più ampio, utilizzando librerie di machine learning basate su PHP non introdotte nel corso.

5. Discussione

La sfida principale in fase di progettazione è stata ideare esperienze laboratoriali per gli studenti che fossero stimolanti e accessibili, capaci di bilanciare l'introduzione di argomenti tecnici e innovativi, non previsti dal curriculum standard, senza eccedere in complessità. Per le future edizioni del corso, si sta valutando l'opportunità di estenderne la durata e di integrare esperienze laboratoriali di apprendimento non supervisionato, attualmente non incluse. Si considera inoltre la possibilità di introdurre concetti di apprendimento automatico all'interno

delle discipline curriculari esistenti. Quest'ultima transizione implicherebbe una rimodulazione dei contenuti didattici attuali per far spazio a queste nuove tematiche. L'adozione di Python come linguaggio di programmazione curriculare, in sostituzione di altri linguaggi attualmente in uso, potrebbe facilitare significativamente tale integrazione. A posteriori, durante la stesura di questo articolo, è stato individuato un lavoro [21] che descrive un'esperienza simile alla nostra sul Riconoscimento di gestualità tracciate in aria.

6. Conclusione

Il corso è stato accolto con entusiasmo dagli studenti, che hanno mostrato un notevole interesse e una partecipazione attiva sia nella parte teorica sia in quella laboratoriale. Di conseguenza, il corso verrà riproposto negli anni successivi ed è attualmente in fase di valutazione per un possibile inserimento nel curriculum scolastico come modulo di approfondimento sulle tecnologie emergenti. Attraverso la collaborazione con docenti di diverse discipline, miriamo a sviluppare un quadro formativo più ampio dedicato alle reti neurali e agli strumenti di intelligenza artificiale. L'approccio laboratoriale, elemento distintivo della pedagogia del nostro istituto, è stato particolarmente apprezzato dagli studenti. Le attività future potranno includere l'esplorazione della possibilità di trasferire reti neurali addestrate su Colab direttamente su dispositivi Arduino, nonché la progettazione di esercitazioni pratiche dedicate all'apprendimento non supervisionato, in modo da integrare e ampliare il curriculum attualmente incentrato sull'apprendimento supervisionato. L'applicazione mobile Android e la web application, prodotte da un gruppo di studenti a seguito del corso in un progetto curriculare, saranno impiegate nelle attività di orientamento e pubblicata sul sito ufficiale dell'istituto.

Bibliografia

- [1] James Finnie-Ansley, Paul Denny, Brett A. Becker, Andrew Luxton-Reilly, and James Prather. The robots are coming: Exploring the implications of openai codex on introductory programming. In *Proceedings of the 24th Australasian Computing Education Conference, ACE '22*, page 10–19, New York, NY, USA, 2022. Association for Computing Machinery.
- [2] Anne Ottenbreit-Leftwich, Krista Glazewski, Minji Jeon, Katie Jantaraweragul, Cindy Hmelo-Silver, Adam Scribner, Seung Lee, Bradford Mott, and James Lester. Lessons learned for ai education with elementary students and teachers. *International Journal of Artificial Intelligence in Education*, 33, 09 2022.
- [3] Brett A. Becker, Michelle Craig, Paul Denny, Hieke Keuning, Natalie Kiesler, Juho Leinonen, Andrew Luxton-Reilly, Lauri Malmi, James Prather, and Keith Quille. *Generative ai in introductory programming*, 2023.
- [4] Paul Denny, James Prather, Brett A. Becker, James Finnie-Ansley, Arto Hellas, Juho Leinonen, Andrew Luxton-Reilly, Brent N. Reeves, Eddie Antonio Santos, and Sami Sarsa. *Computing education in the era of generative ai*, 2023.
- [5] Radu Marinescu-Istodor and Ilkka Jormanainen. Machine learning for high school students. In *Proceedings of the 19th Koli Calling International Conference on Computing Education*

- Research, Koli Calling '19, New York, NY, USA, 2019. Association for Computing Machinery.
- [6] Shashank Srikant and Varun Aggarwal. Introducing data science to school kids. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education, SIGCSE '17*, page 561–566, New York, NY, USA, 2017. Association for Computing Machinery.
- [7] Richard M. Ryan and Edward L. Deci. Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *The American psychologist*, 55 1:68–78, 2000.
- [8] Ieva Tenberga and Linda Daniela. Artificial intelligence literacy competencies for teachers through self- assessment tools. *Sustainability*, 2024.
- [9] Briana Lowe Wellman, James Davis, and Monica Anderson. Alice and robotics in introductory cs courses. In *The Fifth Richard Tapia Celebration of Diversity in Computing Conference: Intellect, Initiatives, Insight, and Innovations, TAPIA '09*, page 98–102, New York, NY, USA, 2009. Association for Computing Machinery.
- [10] B.S. Barker. *Robots in K-12 Education: A New Technology for Learning*. Premier reference source. Information Science Reference, 2012.
- [11] Mohammad Ehsanul Karim, Séverin Lemaignan, and Francesco Mondada. A review: Can robots reshape k-12 stem education? In *2015 IEEE International Workshop on Advanced Robotics and its Social Impacts (ARSO)*, pages 1–8, 2015.
- [12] Nicholas Alexander Bascou and Muhsin Menekse. Robotics in k-12 formal and informal learning environments: A review of literature. In *2016 ASEE Annual Conference & Exposition*, 2016.
- [13] Yanjun Zhang, Ronghua Luo, Yijin Zhu, and Yuan Yin. Educational robots improve K-12 students' computational thinking and stem attitudes: Systematic review. *Journal of Educational Computing Research*, 59(7):1450–1481, 2021.
- [14] Yanjun Zhang and Yijin Zhu. Effects of educational robotics on the creativity and problem-solving skills of K-12 students: A meta-analysis. *Educational Studies*, 50(6):1539–1557, 2024.
- [15] Ugur Sari, Hu`seyin Mirac, Pektas, O`mer Faruk Sen, and Harun C, elik. Algorithmic thinking development through physical computing activities with Arduino in stem education. *Education and information technologies*, 27(5):6669–6689, 2022.
- [16] José-Antonio Marín-Marín, Pedro Antonio García-Tudela, and Pablo Duo-Terrón. Computational thinking and programming with arduino in education: A systematic review for secondary education. *Heliyon*, 10(8):e29177, 2024.
- [17] Oleksii Voronkin and Sergiy Lushchin. Project method in stem education using arduino software and hardware platform. *Physical and Mathematical Education*, 38(4):24–30, Sep. 2023.
- [18] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.
- [19] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems (MCSS)*, 2(4):303–314, December 1989.
- [20] M.A. Nielsen. *Neural Networks and Deep Learning*. Determination Press, 2015.
- [21] Hongyu Zhang, Lichang Chen, Yunhao Zhang, Renjie Hu, Chunjuan He, Yaqing Tan, and Jiajin Zhang. A wearable real-time character recognition system based on edge computing-enabled deep learning for air- writing. *Journal of Sensors*, 2022(1):8507706, 2022.

NextPyter: piattaforma collaborativa e open-source per la didattica dell'informatica

Linda Burchiellaro¹, Francesco Faenza², Claudia Canali¹

¹ *University of Modena and Reggio Emilia, Department of Physics, Informatics and Mathematics, Modena, Italy
273342@studenti.unimore.it*

² *University of Modena and Reggio Emilia, Department of Engineering "Enzo Ferrari", Modena, Italy
francesco.faenza@unimore.it, claudia.canali@unimore.it*

Abstract

La crescente diffusione di approcci didattici collaborativi e interdisciplinari pone nuove sfide nell'insegnamento dell'informatica, in particolare per quanto riguarda l'adozione di strumenti digitali accessibili e gestibili in autonomia. In questo contesto, presentiamo NextPyter, una piattaforma open-source che integra Jupyter Notebook con Nextcloud per supportare attività didattiche basate su coding, analisi dei dati e lavoro cooperativo. Pensata per rispondere alle esigenze della scuola contemporanea, offre un ambiente intuitivo e flessibile che consente la condivisione dei materiali e il lavoro congiunto tra studenti e docenti. Dopo aver analizzato i limiti delle soluzioni esistenti, il contributo presenta l'approccio adottato da NextPyter e ne illustra possibili sviluppi futuri, inclusa una fase di sperimentazione in aula.

1. Introduzione

La promozione della collaborazione online rappresenta una sfida cruciale e al tempo stesso un'opportunità significativa nell'ambito dell'insegnamento dell'informatica. In un contesto educativo sempre più orientato al lavoro cooperativo e all'uso di strumenti digitali, emerge la necessità di soluzioni che facilitino attività condivise in ambienti strutturati, accessibili e sostenibili.

Numerosi problemi sono stati identificati nel caso specifico delle attività didattiche in ambito informatico. Il principale di questi riguarda la necessità di strumenti e soluzioni che non richiedano competenze tecniche avanzate e che agevolino l'adozione da parte di docenti e studenti. Gli ambienti di sviluppo tradizionali, infatti, risultano spesso complessi da configurare e gestire, specialmente per coloro che non possiedono una solida familiarità con la programmazione [8].

Ulteriore criticità riguarda la gestione dei file e la condivisione del codice, soprattutto in progetti che coinvolgono gruppi di lavoro. Nel caso delle materie informatiche all'interno dei corsi curricolari, la possibilità di integrare codice e materiale didattico all'interno di un unico ambiente semplificherebbe notevolmente il processo di apprendimento, consentendo agli studenti di concentrarsi sui concetti fondamentali piuttosto che sulle complessità tecniche degli strumenti [3]. Tali esigenze, combinate con il sempre maggiore utilizzo di linguaggi di

programmazione caratterizzati da una barriera iniziale di apprendimento più bassa, come ad esempio Python, favoriscono l'adozione di strumenti come i Notebook Jupyter [9] per l'organizzazione delle lezioni di informatica. In risposta a tali sfide, NextPyter¹ si propone come una soluzione che integra la potenza analitica dei Jupyter Notebook con la semplicità operativa di gestione documentale di Nextcloud². NextPyter fornisce un'interfaccia user-friendly che abilita sia docenti che studenti senza competenze tecniche avanzate al suo utilizzo completo, rispondendo così alle esigenze di una scuola sempre più orientata all'interdisciplinarietà e alla didattica laboratoriale. L'idea di Nextpyter [5] nasce dall'osservazione di due tendenze convergenti: da un lato, l'adozione crescente di Python come linguaggio introduttivo alla programmazione nelle scuole [7, 12]; dall'altro, la diffusione dei Jupyter Notebook come ambienti educativi capaci di integrare testo, codice ed elementi visivi in un'unica interfaccia interattiva, perfettamente in linea con i principi della didattica attiva e costruttivista. La progettazione dello strumento è stata inoltre ispirata dalle pratiche consolidate di condivisione dei materiali tramite piattaforme digitali, come Moodle, comunemente adottate in ambito scolastico, e da strumenti di archiviazione online come Google Drive o OneDrive. In un panorama in cui l'infrastruttura digitale deve essere sostenibile, accessibile e gestibile in autonomia, la scelta di integrare l'ambiente Jupyter con un sistema open-source di archiviazione e condivisione come NextCloud risponde a tali esigenze. Infatti, NextPyter è rilasciata free e open-source in tutte le sue componenti su Gitlab³. Lo scopo è da un lato quello di favorirne l'adozione e dall'altro quello di creare una comunità di utilizzatori e sviluppatori che si occupino di rispondere alle esigenze in continua trasformazione di studenti e insegnanti.

2. Motivazione e background

Tra le piattaforme cloud per l'esecuzione di codice online, strumenti come Google Colaboratory [2], Binder [10], Observable e Deepnote sono sempre più diffusi in diverse attività interdisciplinari. Tuttavia, tali soluzioni presentano alcune limitazioni significative soprattutto per la loro adozione in ambito scolastico. Piattaforme come Kaggle [11], originariamente progettate per comunità di data scientist, risultano inadeguate per l'ambiente scolastico a causa di restrizioni nelle risorse disponibili, limitazioni nelle funzionalità collaborative e scarsa possibilità di personalizzazione. Sebbene Google Colaboratory [2] risponda più da vicino alle esigenze didattiche, le difficoltà nella collaborazione sincrona, nel controllo delle versioni e la sua natura proprietaria ne compromettono l'adozione a lungo termine nelle istituzioni scolastiche.

Soluzioni commerciali quali Deepnote⁴ e Observable⁵ offrono strumenti validi ma impongono limitazioni dovute ai costi, alle risorse limitate per gli utenti gratuiti e a una curva di apprendimento non sempre agevole per docenti e studenti. Inoltre, la mancanza di opzioni open-source o di installazioni locali rappresenta un elemento critico in termini di autonomia, privacy e sostenibilità delle infrastrutture digitali scolastiche.

La piattaforma open-source Binder [10], basata su JupyterHub, consente l'esecuzione di Notebook ospitati su repository Git direttamente nel browser; tuttavia, la necessità di competenze tecniche specifiche ne limita l'utilizzo in contesti didattici generalisti. Altre piattaforme

1 <https://gitlab.com/nextpyter/nextpyter-aio>

2 <https://nextcloud.com/>

3 <https://gitlab.com/nextpyter>

4 <https://deepnote.com/>

5 <https://deepnote.com/>

avanzate, quali Agave [4] e NaaVRE [13], progettate per la ricerca scientifica e l'integrazione con ambienti di calcolo remoti, risultano invece eccessivamente complesse per un uso quotidiano nell'ambito scolastico. Esperienze significative, come quella dello European Spallation Source [6], hanno mostrato il potenziale dell'integrazione tra Jupyter e sistemi di archiviazione open-source come Nextcloud. La progettazione di Nextpyter prevede l'integrazione di Jupyter Notebook con Nextcloud, pensata specificamente per l'ambiente educativo.

3. Nextpyter

NextPyter utilizza Nextcloud come piattaforma di base per la gestione dei file, sfruttandone la natura agnostica e la semplicità d'uso che permettono a studenti e docenti di caricare, conservare e condividere materiali didattici di ogni tipo, come presentazioni, esercizi, dati, codici e documenti testuali. Grazie all'integrazione nativa dei Jupyter Notebook all'interno della struttura di cartelle di Nextcloud, i Notebook stessi beneficiano automaticamente di tutte le funzionalità offerte dalla piattaforma, come il controllo granulare degli accessi, la condivisione sicura e la sincronizzazione multiplatforma. In questo modo, si mantiene un flusso di lavoro didattico intuitivo, integrato e facilmente accessibile, che non richiede competenze tecniche avanzate da parte di docenti e studenti. Per garantire un ambiente di lavoro sicuro, stabile e scalabile, NextPyter esegue i Notebook all'interno di container Docker⁶ isolati. Questa scelta tecnica permette di separare ogni sessione dall'ambiente principale e da quelle degli altri utenti, proteggendo l'infrastruttura scolastica da possibili errori o conflitti, anche quando gli studenti hanno competenze tecniche molto diverse tra loro. Inoltre, Docker consente di preparare ambienti personalizzati, preconfigurati con librerie e strumenti specifici permettendo l'accesso a numerosi linguaggi di programmazione, come Python, R, Rust, C, e altri.

Un aspetto fondamentale per favorire la collaborazione in aula è la gestione condivisa dei dati tra gli utenti. A tal fine, Nextpyter maschera gli aspetti implementativi, nello specifico quando uno studente o docente accede a un Notebook Jupyter situato in un percorso specifico nella struttura delle cartelle NextCloud, quel percorso diventa il punto di partenza per il notebook stesso. Questo meccanismo consente agli utenti di interagire con il container tramite un iframe nella pagina mantenendo l'accesso agli stessi documenti presenti su NextCloud, creando un'esperienza diretta, trasparente e intuitiva.

3.1 Esempi di utilizzo in aula

Uno degli scenari didattici più efficaci per l'utilizzo di NextPyter è rappresentato dalle attività di laboratorio collaborativo. Ad esempio, gli studenti, organizzati in piccoli gruppi, possono lavorare contemporaneamente su uno stesso Notebook Jupyter, condiviso tramite Nextcloud. Questo ambiente consente di unire la scrittura del codice, la documentazione testuale e la visualizzazione dei risultati in un unico spazio digitale. I docenti possono assegnare compiti di programmazione, analisi di dati o simulazioni interattive, osservando in tempo reale il contributo di ciascun gruppo. La possibilità di accedere al Notebook da qualsiasi dispositivo connesso alla rete scolastica (o anche da casa) favorisce una didattica partecipativa, inclusiva e flessibile.

Un ulteriore vantaggio è dato dalla possibilità di effettuare correzioni in tempo reale. Il docente può intervenire direttamente sul Notebook condiviso, come mostrato in Figura 1, suggerendo modifiche, evidenziando errori o inserendo commenti esplicativi mentre gli studenti

⁶ <https://www.docker.com/>

stanno lavorando. Questo approccio rende il processo di feedback più immediato ed efficace rispetto alla consegna differita dei compiti.

Un altro importante beneficio derivante dall'uso di NextPyter riguarda la possibilità di tracciare e archiviare l'intera cronologia delle modifiche apportate ai Notebook. Grazie alla sincronizzazione e al controllo delle versioni integrati in Nextcloud, sia gli insegnanti sia gli studenti possono consultare le diverse versioni di un progetto, analizzando l'evoluzione del lavoro svolto nel tempo.

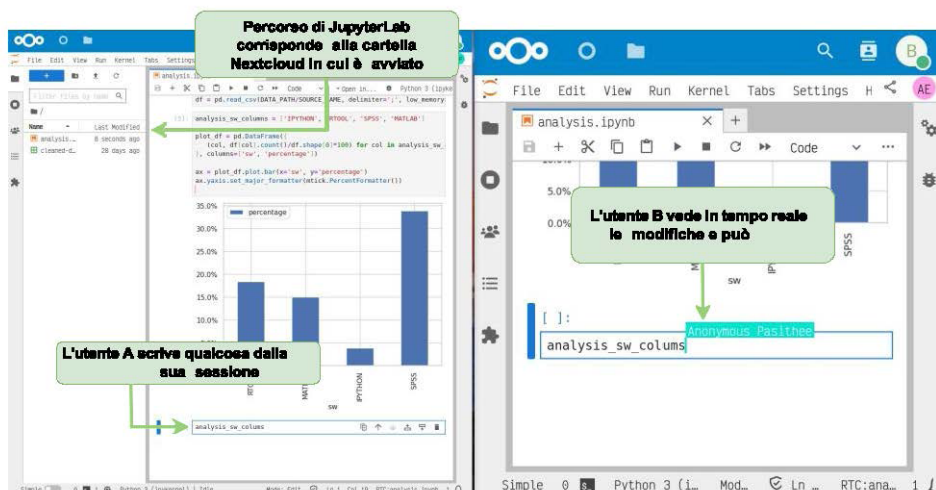


Figura 1: Esempio di collaborazione su NextPyter

4. Conclusioni e sviluppi futuri

NextPyter, grazie alla sua integrazione con Nextcloud e all'uso di container Docker isolati, garantisce un ambiente sicuro e stabile durante attività didattiche. Le risorse di calcolo possono essere allocate dinamicamente in base al numero di studenti, e la gestione dei permessi consente al docente di organizzare le attività in modo ordinato e controllato. Questo lo rende uno strumento ideale per supportare l'insegnamento dell'informatica sia in contesti specialistici, sia in progetti multidisciplinari, dove il coding si intreccia con altri contesti disciplinari.

Per valutarne l'impatto reale in contesti scolastici, le prossime fasi del progetto NextPyter prevedono una fase strutturata di sperimentazione in aula, in collaborazione con docenti di informatica e di discipline affini. L'obiettivo è osservare direttamente l'impatto dello strumento sulla didattica laboratoriale e sull'apprendimento degli studenti. A tal fine, sarà avviata una raccolta sistematica di feedback attraverso diversi strumenti [1]: questionari a risposta chiusa per rilevare percezioni generali e domande aperte per approfondire opinioni ed esperienze rivolte ai docenti. Questo approccio misto consentirà di ottenere una visione ricca e articolata sull'efficacia della piattaforma, sull'usabilità percepita e sugli ostacoli eventualmente incontrati.

Bibliografia

- [1] Yousuf Ali Alharbi and Steve Drew. Technology acceptance model: a literature review from 1986 to 2013. In PACIS 2014 Proceedings, page Paper 145, 2014.
- [2] Ekaba Bisong et al. Building machine learning and deep learning models on Google cloud platform. Springer, 2019.
- [3] KL Cristiano and DA Triana. Google classroom as a tool-mediated for learning. In Journal of Physics: Conference Series, volume 1161, page 012020. IOP Publishing, 2019.
- [4] Rion Dooley, Steven R Brandt, and John Fonner. The agave platform: An open, science-as-a-service platform for digital science. In Proceedings of the Practice and Experience on Advanced Research Computing: Seamless Creativity, pages 1–8. 2018.
- [5] Francesco Faenza, Claudia Canali, Lisa Fregni, and Emiliano Maccaferri. Nextpyter: Open-source research collaborative platform. In Practice and Experience in Advanced Research Computing 2024: Human Powered Computing, pages 1–9. 2024.
- [6] Leandro Fernández, Riccard Andersson, Hakan Hagenrud, Timo Korhonen, Emanuele Laface, Bla Zupanc, et al. Jupyterhub at the ess. an interactive python computing environment for scientists and engineers. In Proceedings of IPAC2016, Busan, Korea, 2016.
- [7] Linda Grandell, Mia Peltomaki, Ralph-Johan Back, and Tapio Salakoski. Why complicate things?: introducing programming in high school using python. In ACM International Conference Proceeding Series, volume 165, pages 71–80, 2006.
- [8] Riko Kelter, Matthias Kramer, and Torsten Brinda. Teachers’ perspectives on learning and programming environments for secondary education. In Empowering Learners for Life in the Digital Age: IFIP TC 3 Open Conference on Computers in Education, OCCE 2018, Linz, Austria, June 24–28, 2018, Revised Selected Papers, pages 47–55. Springer, 2019.
- [9] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica Hamrick, Jason Grout, Sylvain Corlay, et al. Jupyter notebooks—a publishing format for reproducible computational workflows. In Positioning and power in academic publishing: Players, agents and agendas, pages 87–90. IOS press, 2016.
- [10] Benjamin Ragan-Kelley, Carol Willing, F Akici, D Lippa, D Niederhut, and M Pacer. Binder 2.0-reproducible, interactive, sharable environments for science at scale. In Proceedings of the 17th python in science conference, pages 113–120. F. Akici, D. Lippa, D. Niederhut, and M. Pacer, eds., 2018.
- [11] April Yi Wang, Dakuo Wang, Jaimie Drozdal, Xuye Liu, Soya Park, Steve Oney, and Christopher Brooks. What makes a well-documented notebook? a case study of data scientists’ documentation practices in Kaggle. In Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems, pages 1–7, 2021.
- [12] Michael Weigend. Programming languages for secondary schools, python. In Encyclopedia of education and information technologies, pages 1370–1378. Springer, 2020.
- [13] Zhiming Zhao, Spiros Koulouzis, Riccardo Bianchi, Siamak Farshidi, Zeshun Shi, Ruyue Xin, Yuandou Wang, Na Li, Yifang Shi, Joris Timmermans, et al. Notebook-as-a-vre (naavre): From private notebooks to a collaborative cloud virtual research environment. Software: Practice and Experience, 52(9):1947–1966, 2022.

Dalla scuola allo Spazio: attività di coding con Python e Sense HAT per apprendere l'Informatica

Giuseppe Fiamingo¹

¹ Istituto d'Istruzione Superiore di Tropea

giuseppe.fiamingo@scuola.istruzione.it

Abstract

L'engagement degli studenti nelle discipline STEM (Scienza, Tecnologia, Ingegneria e Matematica), e in particolare nell'informatica, rappresenta una sfida cruciale per l'educazione contemporanea. In un panorama globale in rapida evoluzione tecnologica, lo sviluppo del pensiero computazionale e le competenze pratiche di coding non sono più una specializzazione, ma delle abilità trasversali indispensabili, competenze fondamentali per la cittadinanza attiva e per le future carriere. Tuttavia, la percezione dell'Informatica come disciplina astratta o puramente teorica, priva di connessioni dirette con il mondo reale, può talvolta limitarne l'attrattiva per gli studenti e limitarne lo studio. In questo contesto, il presente contributo descrive un'esperienza didattica innovativa realizzata con la classe IVB dell'I.I.S. Tropea, composta da 15 studenti di età compresa tra i 17 e i 18 anni, nell'A.S. 2022/2023, nell'ambito della Mission Space Lab promosso dall'Agenzia Spaziale Europea (ESA) e della Raspberry Pi Foundation attraverso la piattaforma ESERO (European Space Education Resource Office). L'iniziativa ha offerto agli studenti l'opportunità unica di condurre attività scientifiche nello spazio attraverso il coding, mirate allo studio dell'ambiente per la salvaguardia del nostro pianeta. Lavorando in team, gli studenti hanno sviluppato un codice Python che è stato successivamente eseguito su un computer Raspberry Pi a bordo della Stazione Spaziale Internazionale (ISS). La prima fase dell'esperimento consisteva nell'inviare un messaggio di saluto agli astronauti sul display LED dell'AstroPi ricevuto, e nelle fasi successive, è stato programmato un algoritmo per acquisire immagini durante il passaggio della ISS con l'obiettivo di misurare la luminanza delle acque terrestri, con finalità di analisi ambientale. L'algoritmo realizzato ha permesso di raccogliere dati scientifici, successivamente analizzati e riportati in grafici. L'attività, certificata da Samantha Cristoforetti, ha stimolato il pensiero computazionale, l'approccio scientifico e l'interdisciplinarietà. Ha inoltre dimostrato le profonde connessioni dell'informatica con la fisica, le scienze della Terra e l'astronomia, offrendo un modello replicabile per introdurre l'informatica come disciplina viva, utile e motivante nella scuola secondaria di secondo grado in linea con le Indicazioni Nazionali del CINI.

1. Introduzione

Nel contesto attuale, dove la tecnologia è ovunque ma l'informatica è ancora poco compresa come scienza, proporre agli studenti esperienze autentiche e interdisciplinari è essenziale per dare significato allo studio e all'importanza del coding. L'esperienza Astro Pi nasce da questa esigenza: unire la programmazione alla scienza e all'osservazione del nostro pianeta, sfruttando la

Stazione Spaziale Internazionale (ISS) come laboratorio orbitale. Nel corso dell'A.S. 2022/2023, la classe IV B del Tecnico Turistico dell'IIS Tropea, ha realizzato un innovativo progetto didattico che ha coinvolto studentesse e studenti della scuola secondaria di secondo grado nella partecipazione alla missione Astro Pi, promossa dall'Agenzia Spaziale Europea (ESA) e dalla Raspberry Pi Foundation. L'attività ha previsto la scrittura di un codice Python, che è stato successivamente eseguito sui computer AstroPi a bordo della ISS, e supervisionato dagli astronauti a bordo della Stazione Spaziale Internazionale (ISS), culminando con il conferimento di un attestato di merito firmato dall'astronauta Samantha Cristoforetti (Figura 1), a testimonianza dell'autenticità e del successo dell'impresa. Questa iniziativa ha trasformato un semplice esercizio di programmazione in una vera e propria attività di ricerca, promuovendo l'acquisizione di competenze tecniche come il coding e la programmazione, nonché abilità trasversali quali il problem solving, la collaborazione e la creatività. Sebbene la nostra scuola abbia avuto il privilegio di ricevere un dispositivo Astro Pi fisico, è importante sottolineare che l'ESA rende l'esperienza del coding accessibile ad ogni studente mettendo a disposizione un emulatore gratuito online (disponibile sul sito Raspberry Pi - <https://trinket.io/sense-hat> - Figura 2). Gli studenti hanno potuto scrivere e testare brevi algoritmi per visualizzare immagini, colori o testi sulla matrice LED dell'Astro Pi, come ad esempio animali, simboli, volti sorridenti o tristi. Il coding si è così rivelato uno strumento formativo divertente, stimolante e inclusivo, con la possibilità di personalizzare le attività proposte e di visualizzare immediatamente i risultati del proprio lavoro (Figura 3).



Figura 1: Attestato di merito

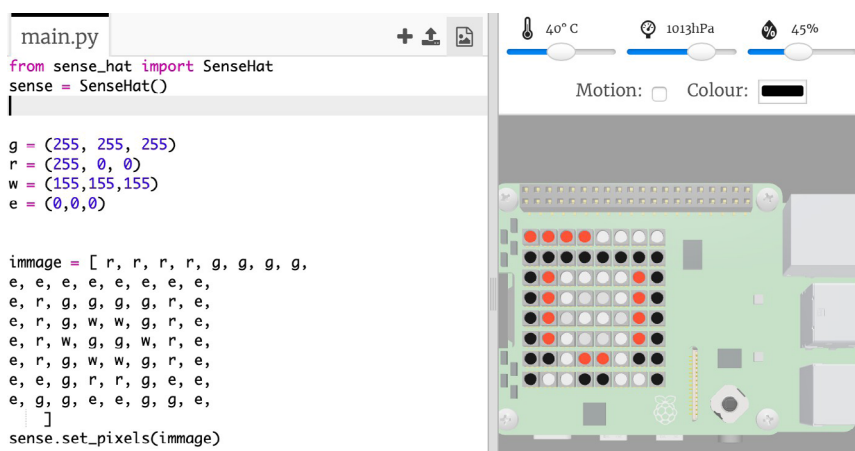


Figura 2: Esempio di display LED con output dell'esperimento

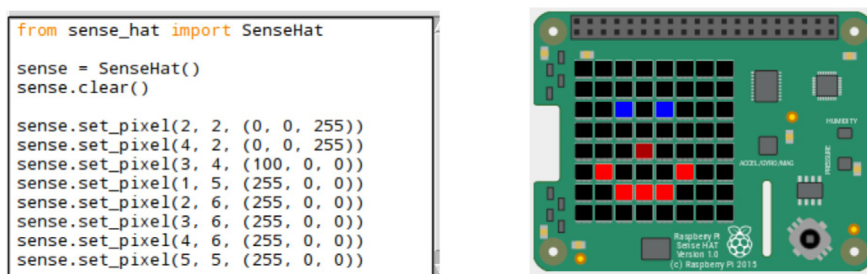


Figura 3: Componenti del Sense HAT Link video: <https://www.youtube.com/watch?v=4jAwcYMYHag>

2. Il progetto didattico, Python, il coding e il pensiero computazionale

Il progetto è stato articolato in diverse fasi progressive, calibrate per il livello di apprendimento degli studenti. La fase preliminare, ha visto la presentazione del programma Mission Space Lab, e le attività di ricerca dell'ISS. Introduzione al Raspberry Pi e al Sense HAT, con esempi pratici delle funzionalità dei sensori. Richiamo e approfondimento delle basi del linguaggio Python, con la scrittura di semplici programmi per visualizzare immagini, colori e testi sulla matrice LED dell'Astro Pi. In una fase successiva, gli studenti, organizzati in piccoli gruppi, hanno partecipato a sessioni di brainstorming per identificare possibili esperimenti scientifici da condurre.

È stata selezionata l'analisi della luminanza dell'acqua del mare in diversi momenti durante il passaggio della ISS. La scelta è stata motivata dall'importanza della luminanza come parametro in applicazioni scientifiche e ambientali, quali lo studio del clima, il monitoraggio dell'inquinamento delle acque e l'osservazione di eventi naturali (es. fioriture algali).

2.1 Che cos'è un Astro Pi?

Astro Pi è un computer basato su Raspberry Pi, progettato per essere utilizzato a bordo della ISS. La versione più recente, Astro Pi MK II, integra:

1. Raspberry Pi 4.
2. Sense HAT.
3. Fotocamera HQ.
4. Acceleratore Coral USB per machine learning.

Il computer Astro Pi, consente di realizzare esperimenti avanzati, dalla fotografia della Terra al riconoscimento di immagini. Il nostro esperimento ha utilizzato la versione VIS, dedicata all'osservazione terrestre in luce visibile. (Figure 6-9)

2.2 Fase di Sviluppo del Codice Python (Figura 3)

Ogni gruppo ha sviluppato il proprio script Python. Successivamente, un gruppo più avanzato si è dedicato a:

1. Inizializzare il Sense HAT e i suoi sensori.
2. Acquisire i dati di luminanza ad intervalli regolari durante il periodo di esecuzione (tipicamente 30 secondi o 60 secondi, come da specifiche del bando Mission Space Lab).
3. Registrare i dati raccolti (luminanza, ora di acquisizione, coordinate geografiche della ISS) in un formato strutturato (es. file CSV) per la successiva analisi.
4. Visualizzare messaggi personalizzati o stati di avanzamento sul display a LED 8x8 del Sense HAT, dimostrando la capacità di controllare l'hardware da codice.
5. Supporto e tutoraggio verso i compagni nelle varie fasi

Durante questa fase, è emersa una sfida cruciale legata alla gestione del ciclo di acquisizione dati. Nello specifico, il codice prevedeva un'iterazione fissa di 10 cicli per l'acquisizione delle immagini e il salvataggio dei dati (e delle immagini nei directory "shot" e "processed") nei dizionari "data" e "data_raw". Questa scelta, sebbene corretta sintatticamente, ha rappresentato un errore concettuale in relazione all'obiettivo di un monitoraggio continuo o prolungato della luminanza. L'intenzione originaria era probabilmente quella di acquisire dati per l'intera durata dell'esperimento (tipicamente 30 o 60 secondi), ma la limitazione a sole 10 iterazioni ha di fatto compresso drasticamente la finestra di osservazione, fornendo un campione molto limitato e potenzialmente non rappresentativo dei fenomeni in corso. L'esperienza dell'errore ha reso tangibile il concetto di debugging e di pensiero critico applicato alla propria produzione. Hanno imparato che l'errore non è un fallimento, ma un'opportunità di apprendimento, di affinamento della logica e di miglioramento continuo. Questo processo ha rafforzato notevolmente le loro competenze di problem-solving e la loro autonomia nel ragionamento computazionale, elementi chiave che trascendono la semplice sintassi di un linguaggio.

2.3 Fase di Sottomissione e Monitoraggio dell'Esperimento Spaziale

Una volta completati e validati, gli script Python sono stati sottomessi alla Raspberry Pi Foundation/ESA attraverso la piattaforma dedicata del programma Mission Space Lab. Gli studenti hanno avuto l'opportunità di seguire lo stato del loro esperimento e ricevere notifiche sull'avvenuta esecuzione a bordo della ISS.

2.4 Fase di Raccolta e Analisi dei Dati

Al termine dell'esecuzione in orbita, i dati grezzi raccolti dagli Astro Pi (inclusi quelli del nostro esperimento) sono stati resi disponibili per il download. Gli studenti hanno utilizzato strumenti come fogli di calcolo (es. Microsoft Excel, Google Sheets) e script Python aggiuntivi (utilizzando librerie come pandas per la manipolazione dati e matplotlib per la visualizzazione) per elaborare e analizzare i dati di luminanza. L'analisi ha incluso la creazione di grafici temporali per visualizzare le variazioni della luminanza

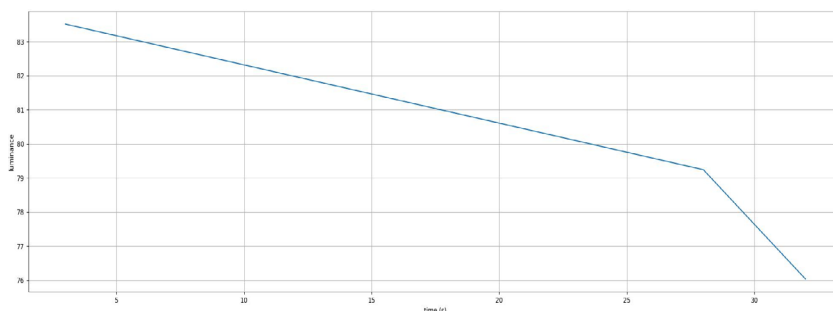


Figura 4: Analisi della luminance chart

3. Il cuore tecnologico del progetto: il Sense HAT

Il Sense HAT (Figura 2), è una scheda elettronica che si collega al Raspberry Pi. Questo HAT (“Hardware Attached on Top”) si posiziona sopra la scheda madre e, grazie ai suoi sensori, permette di monitorare l’ambiente e visualizzare dati in modo interattivo. Le sue componenti principali, ciascuna con un ruolo didattico specifico nell’apprendimento dell’informatica, includono:

1. Una matrice LED 8x8 RGB per visualizzare scritte, immagini e animazioni.
2. Sensori di temperatura e umidità, utili per introdurre il concetto di acquisizione e gestione di dati ambientali;
3. Un barometro per la pressione atmosferica;
4. Un sensore di movimento (accelerometro, giroscopio, magnetometro).

4. Python e programmazione

Python è un linguaggio di programmazione, uno strumento altamente didattico, leggibile e potente, che consente di introdurre:

- Strutture sequenziali e condizionali;
- Variabili, cicli e funzioni;
- Utilizzo di librerie esterne (come sense_hat, OpenCV e NumPy);
- Manipolazione diretta dei LED tramite array RGB;
- Elaborazione automatica di immagini e visualizzazione di grafici.

Accanto alla visualizzazione di messaggi sulla matrice LED della ISS, gli studenti hanno realizzato un algoritmo più complesso (Figura4) per acquisire e analizzare immagini in tempo reale. Il programma, basato sull'uso delle librerie OpenCV per la manipolazione delle immagini e NumPy per la gestione dei dati, Matplotlib per la creazione dei grafici, eseguiva i seguenti passaggi:

- Acquisizione di immagini da una videocamera PiCamera;
- Elaborazione dei pixel per calcolare la luminanza secondo la formula percettiva (ponderazione RGB);
- Scarto dei dati anomali (troppo chiari o rumorosi);
- Calcolo della media luminosa per ogni scatto e sua rappresentazione grafica nel tempo;
- Esportazione dei dati in formato JSON.

Questo esercizio ha permesso agli studenti di comprendere come un'immagine sia in realtà un array tridimensionale di dati numerici, su cui è possibile applicare operazioni di filtro e analisi. Il grafico risultante, salvato come immagine e correlato da un file .data, ha mostrato chiaramente l'evoluzione della luminanza in funzione del tempo.

Esempio di un semplice algoritmo elaborato dalla classe che permetteva la scrittura di un messaggio personalizzato sulla matrice LED, visibile direttamente dagli astronauti a bordo della ISS.

```
from sense_hat import SenseHat
from time import sleep
sense = SenseHat() sense.set_rotation(270)
sense.show_message ("Ciao astronauti, siamo i ragazzi della
classe IV B del Tecnico Turistico di Tropea! Missione terra: Proteg-
giamo il pianeta!")

https://trinket.io/sense-hat
```

5. Conclusioni e prospettive future

Dopo aver condotto l'esperimento e analizzato i dati raccolti, gli esperti dell'ESA/Raspberry Pi Foundation hanno condotto un controllo rigoroso del codice inviato per verificarne la sicurezza, la conformità e la funzionalità. Superata questa verifica, l'Astro Pi a bordo della ISS ha autonomamente eseguito il nostro codice, acquisendo i dati di luminanza come richiesto. Successivamente, i dati grezzi sono stati resi disponibili per il download e la successiva analisi e interpretazione da parte dei ragazzi a terra, che hanno poi elaborato i relativi grafici e visualizzato le immagini. Dalla lettura dei relativi grafici, sono emerse diverse conclusioni significative. In primo luogo, si è confermato che la luminanza dell'acqua è un parametro utile per indagare fenomeni climatici, inquinamento marino, e il monitoraggio di eventi naturali come le correnti oceaniche e può fornire informazione utili per verificare il corretto funzionamento di sensori e strumenti scientifici. Il grafico ha mostrato un calo progressivo della luminanza con un picco a 28 secondi, suggerendo possibili correlazioni con copertura nuvolosa, oli, agenti chimici o alghe tossiche. Dal punto di vista metodologico, abbiamo imparato a utilizzare librerie come **OpenCV** e **NumPy** per l'elaborazione di immagini e calcoli scientifici. Infine, ci siamo resi conto che il minimo errore può influenzare significativamente i risultati ottenuti. L'esperienza Astro Pi ha rappresentato un esempio concreto di didattica attiva e laboratoriale dell'Informatica, in

linea con le più recenti proposte curriculari nazionali. Dal coding base in ambiente simulato per le scuole medie, fino alla programmazione reale in Python con hardware fisico per le scuole secondarie. Gli alunni, grazie a questa esperienza, hanno potuto:

- Comprendere il funzionamento di algoritmi e dispositivi;
- Riflettere sul ruolo dell'applicazione dell'informatica nella vita reale e nello spazio;
- Sviluppare competenze digitali e computazionali;
- Allenare il pensiero logico e la capacità di progettazione.

```
import cv2
import numpy as np
from picamera import PiCamera
import time
import matplotlib.pyplot as plt
import json
import random
import os

WIDTH = 480
HEIGHT = 360
TOT_PIXELS = WIDTH*HEIGHT
def process_image(in_file_path: str, out_file_path: str, data,
data_raw, start_time: int):
    image = cv2.imread(in_file_path)
    image = np.array(image, dtype=int)
    height, width, channels = image.shape
    values = []
    for i in range(height):
        for j in range(width):
            blue = image[i][j][0]
            green = image[i][j][1]
            red = image[i][j][2]
            if (abs(red - green) > 40 or abs(min(red, green) - blue)
< 10) or (blue < red or blue < green) or (green > 100 and blue > 100
and red > 100):
                image[i][j] = [0, 0, 0]
            else:
                values.append((0.2126*red + 0.7152*green +
0.0722*blue))
    lumCount = len(values)
    if lumCount < 0:
        median = 0;
    elif lumCount < (TOT_PIXELS * 10 / 100):
        median = -1;
    else:
        summed = 0
        for lum in values:
            summed += lum
        median = summed / lumCount
    print(median)
    write_time = int(time.time()) - start_time;
    data_raw[write_time] = median;
    if median > 0:
        data[write_time] = median;
    cv2.imwrite(out_file_path, image)
    return median
picamera = PiCamera()

picamera.resolution = (WIDTH, HEIGHT)
picamera.contrast = 10
data = {}
data_raw = {}

if not os.path.isdir('shot'):
    os.mkdir('shot');

if not os.path.isdir('processed'):
    os.mkdir('processed');

start_time = int(time.time())
for x in range(10):
    process_image_time = int(time.time())
    image_name = str(process_image_time) + '.jpg';
    picamera.capture('shot/' + image_name)
    brightness = process_image('shot/' + image_name, 'processed/' +
image_name, data, data_raw, start_time)

    optimalWaitTime = 1 - int(time.time() - process_image_time);
    if optimalWaitTime < 0:
        optimalWaitTime = 1

    time.sleep(optimalWaitTime)

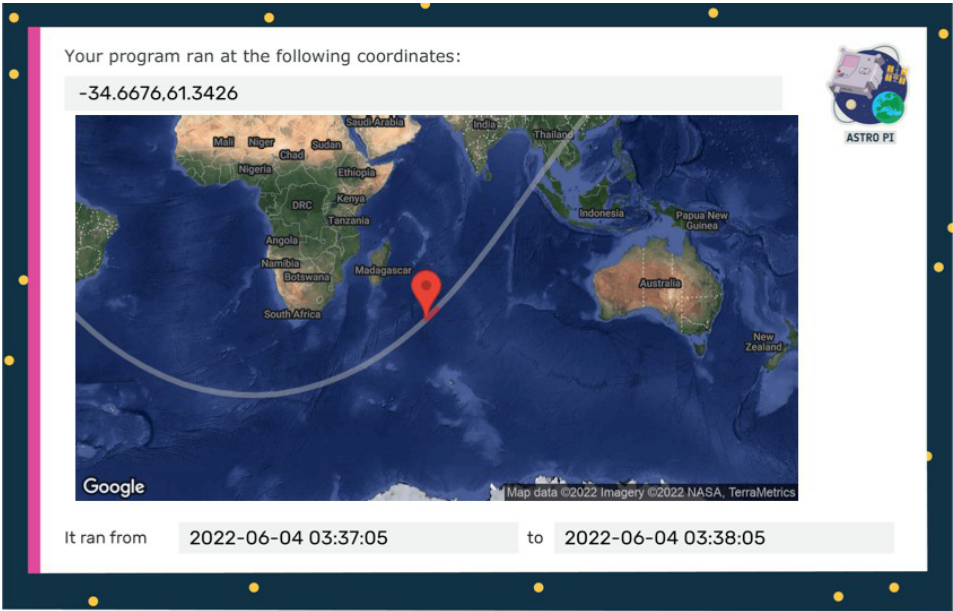
names = list(data.keys())
values = list(data.values())
fig, ax = plt.subplots(figsize=(25, 7))
ax.plot(names, values)

ax.set(xlabel='time (s)', ylabel='luminance')
ax.grid()

jsonDumpFile = open('luminance.data', 'w')
jsonDumpFile.write("startTime: " + str(start_time) + "\n")
jsonDumpFile.write("stopTime: " + str(int(time.time())) + "\n")
jsonDumpFile.write("json clean data: " + json.dumps(data) + "\n")
jsonDumpFile.write("json complete data: " + json.dumps(data_raw))
jsonDumpFile.close()

fig.savefig("fig.jpg")
```

Figura 5: Algoritmo Python utilizzato per eseguire l'esperimento



ASTRO-PI.ORG

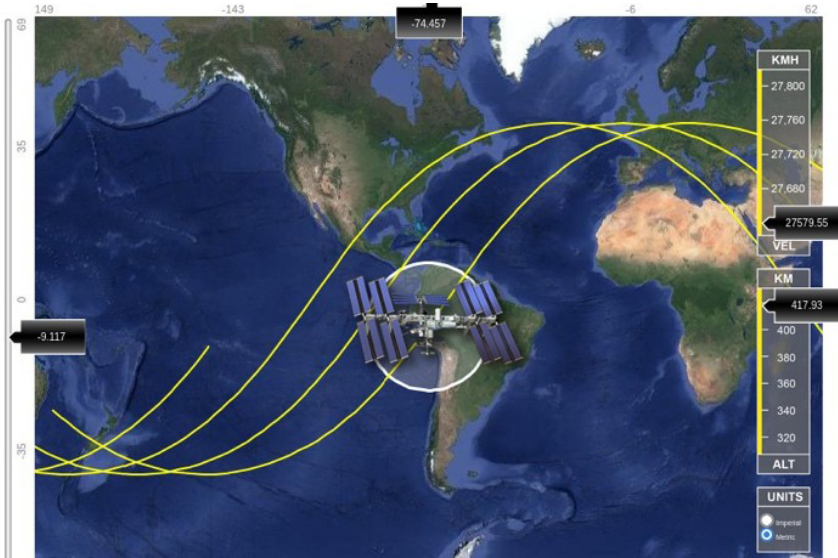


Figura 6: Passaggio della stazione orbitante



Figura 7: Articolo di giornale



Figura 8: Samantha Cristoforetti con AstroPi

Educare all'Informatica con SteamCity

Manon Ballester¹, Mauro D'Angelo², Sébastien Nedjar³, Maria Angela Pellegrino^{2,4,5}

¹ *Les Emulsionneurs, Aix-en-Provence, France*

manon.ballester@lesemulsionneurs.fr

² *Perlatecnica, Maddaloni, Caserta, Italy*

mauro.dangelo@perlatecnica.it

³ *Laboratoire d'Aix-périmentation et de Bidouille, Aix-en-Provence, France*

sebastien.nedjar@labaixbidouille.com

⁴ *Università degli Studi di Salerno, Italy*

mapellegrino@unisa.it

⁵ *Laboratorio Nazionale del CINI - Gruppo di Lavoro "Informatica & Scuola"*

Abstract

Il progetto *SteamCity* propone un modello innovativo per l'insegnamento dell'informatica nella scuola secondaria di secondo grado, integrando attività hands-on e apprendimento per indagine. Attraverso protocolli didattici interdisciplinari centrati sulle sfide urbane, come l'inquinamento o la qualità dell'aria, il progetto coinvolge gli studenti in esperimenti concreti che richiedono la progettazione di oggetti intelligenti, la raccolta, rappresentazione, e elaborazione di dati, la comprensione dei meccanismi alla base degli strumenti di intelligenza artificiale per abilitarne l'uso consapevole. Questo contributo presenta l'impianto pedagogico di *SteamCity*, con particolare attenzione all'informatica come leva per la comprensione critica dei sistemi urbani. Vengono discussi i primi risultati emersi da laboratori con insegnanti e sperimentazioni in classe, mettendo in luce l'efficacia dell'approccio per stimolare motivazione, pensiero computazionale e consapevolezza civica.

1. Introduzione

Negli ultimi anni, la didattica dell'informatica nella scuola secondaria ha assunto un ruolo centrale nell'educazione scientifica, con l'obiettivo di formare cittadini capaci di comprendere e intervenire criticamente nei processi digitali che regolano la società contemporanea, rendendoli scienziati capaci di osservare fenomeni esterni e interpretarli correttamente [2]. Nel contestualizzare l'apprendimento nella vita quotidiana, il progetto *SteamCity*¹ propone un modello di educazione scientifica integrata che unisce l'informatica a contesti urbani concreti. Utilizzando protocolli di indagine (a.k.a. inquiry-based learning) [1, 6, 8] strutturati secondo il metodo scientifico, il progetto mira a trasformare la città in un laboratorio vivente per esplorare i concetti

1 *SteamCity*: <https://www.steamcity.eu>

fondamentali dell'informatica e delle scienze computazionali, promuovendo allo stesso tempo creatività, partecipazione e responsabilità civica. SteamCity, progetto Erasmus+, finanziato per il triennio 2022-2025, ha prodotto 25 protocolli validati in 75 esperimenti coinvolgendo attivamente docenti e studenti della scuola superiore di secondo grado. Focalizzandoci sui protocolli inerenti all'apprendimento dell'informatica, i 13 individuati sono liberamente accessibili² in italiano, riportando descrizione dell'attività e obiettivi di apprendimento, materiale di supporto, linee guida per abilitare la conduzione in aula dell'attività da parte di docenti interessati, e livello di difficoltà su una scala da 1 a 5 con 5 difficoltà massima. Seppur tutti i protocolli partono con una fase di familiarizzazione per il raggiungimento degli obiettivi individuati, i docenti dovrebbero optare per l'adeguata difficoltà in base alla natura dell'istituto e le competenze pregresse dei partecipanti.

2. Protocolli focalizzati sull'Apprendimento dell'Informatica

La Tabella 1 abilita il confronto e l'accesso agevolato alle risorse per ogni protocollo. Di seguito vengono categorizzati per obiettivi di apprendimento:

- **Pensiero computazionale e Programmazione.** In questa categoria rientrano attività che favoriscono la comprensione delle basi della programmazione, dell'elaborazione algoritmica e del problem solving computazionale in contesti realistici. *Smart Object Safari* [10] consente ai ragazzi di padroneggiare con la sensoristica per la progettazione e programmazione di oggetti intelligenti rispondendo a problemi reali. Ad esempio, i partecipanti possono essere invitati a rendere il parco della loro città intelligente chiamandoli a individuare l'oggetto da rendere interattivo, ideare il comportamento automatico che vogliono realizzare raccontandone la modalità di interazione, individuare la sensoristica adeguata al progetto proposto, programmare del comportamento e prototipazione dell'oggetto. In questo caso il pensiero computazionale e la programmazione sono messi al servizio della valorizzazione delle città urbane. In modo analogo, *Avventura di Bot Buddy* propone la progettazione di chatbot che garantiscano l'inclusività delle città rendendo accessibili servizi urbani a cittadini e visitatori con disabilità, stimolando la pianificazione di interazioni condizionate e la strutturazione del dialogo, offrendo un'esperienza trasversale che unisce informatica, linguistica e progettazione dell'interazione. Nella direzione di avere città sempre più autonome, *Roobopoli* [3] guida gli studenti nella progettazione e simulazione della mobilità autonoma attraverso la programmazione di robot, promuovendo l'acquisizione di abilità logiche, controllo dei sensori [7] e riflessione su implicazioni ambientali [4] nell'ambito di hackathon educativi [9]. Ciò permette ai partecipanti di affrontare sfide reali, scomporle in sottoproblemi, e risolverle sfruttando un approccio collaborativo.
- **Alfabetizzazione Intelligenza Artificiale (IA) e Machine Learning (ML).** Questa categoria include attività per introdurre gli studenti ai concetti fondamentali dell'IA e dell'apprendimento automatico. Coerentemente con le pratiche esplorate in ambito educativo [5], queste attività si incentrano sull'apprendimento di strumenti di classificazione, coprendo dati eterogenei per natura e formato. Ad esempio, *Processi di apprendimento bio-ispirati* affronta l'adattamento nei sistemi intelligenti attraverso simulazioni unplugged di apprendimento per rinforzo, rendendo accessibili le logiche decisionali dei sistemi adattivi. *Esploratore del*

² Protocolli SteamCity per l'apprendimento dell'informatica: <https://www.notion.so/steamcityproject/ITADINFO-20e229db4fb681dd90d3c677f877dcb>

canto degli uccelli invita gli studenti a raccogliere, etichettare e classificare suoni della natura, affrontando i concetti di addestramento, bias, precisione e valutazione dei modelli. Anche *Raccolta differenziata* propone una sperimentazione attiva sul riconoscimento automatico dei rifiuti, in cui i ragazzi addestrano un sistema di classificazione visiva per il riciclo, imparando a valutare le prestazioni e i limiti dei modelli supervisionati. In *La segnaletica stradale non ambigua* gli studenti progettano segnaletica stradale non ambigua testando e addestrando modelli di classificazione. In *Alberi VS Macchine* vengono padroneggiati alberi decisionali per distinguere esemplari di classi disgiunte. In tutti i casi, l'apprendimento non si limita a usare un classificatore come una black box, ma acquisire le competenze necessarie per addestrare i modelli, comprenderne i meccanismi alla base, e valutarne le prestazioni.

- **Acquisizione e rappresentazione dei dati.** Le attività incluse in questa categoria sviluppano la competenza nell'acquisizione, analisi e interpretazione di dati ambientali. *Monitoraggio qualità dell'aria esterna e interna* permettono agli studenti di esplorare l'inquinamento urbano attraverso sensori di particolato e CO₂, incoraggiando il confronto tra condizioni diverse e la comprensione della variabilità e del rumore nei dati raccolti. In *Muri sussurranti*, *Decibel Detective*, *Luce VS Zzzz* e *Sound Squad* la rilevazione sonora viene utilizzata per valutare l'efficacia di materiali fonoassorbenti e riflettere sull'impatto del rumore sulla salute mentale mentre in *Dai muri caldi alle città fresche* i partecipanti usano sensoristica ambientale per monitorare le emissioni di gas serra e aumentare la consapevolezza sui cambiamenti climatici. In tutte queste attività, i dati vengono utilizzati non solo per descrivere fenomeni, ma anche per prendere decisioni informate su problemi reali, scalando la piramide DIKW che guida dai dati alla conoscenza [11].

Protocollo	Obiettivi	Strumento	Link	Diff.
Pensiero Computazionale e Programmazione				
Smart Object Safari	Progettazione di sistemi interattivi	Sensoristica, strumento unplugged basato sul gioco del memory, Micro:bit, MakeCode	https://shorturl.at/zhuUK	1/5
Avventura di Bot Buddy	Rendere le città accessibili mediante	Strumenti di geo-localizzazione, editor a blocchi, strumenti di riconoscimento vocale	https://shorturl.at/S0G9b	3/5
Roobopoli	Progettazione, implementazione e prototipazione di veicoli a guida autonoma	Robotica educativa	https://shorturl.at/Qkn3b	5/5

Alfabetizzazione Intelligenza Artificiale (IA) e Machine Learning (ML)				
La segnaletica stradale non ambigua	Progettazione di segnaletica stradale via addestramento e valutazione di reti neurali	Classificatore (addestrato e da addestrare), dataset di segnali stradali	https://shorturl.at/9tuhS	1/5
Raccolta differenziata	Apprendimento di ML supervisionato per classificare immagini e valutare modelli	Interfaccia per costruzione di dataset, addestramento di modelli di ML e metriche di errore	https://shorturl.at/EfFTG	1/5
Alberi VS Automobili	Classificazione basata su alberi decisionali	Alberi logici, griglie di classificazione	https://shorturl.at/d11Gt	3/5
Processi di apprendimento bio-ispirati	Apprendimento automatico e basato su rinforzo	Simulazione (anche con strumenti unplugged), apprendimento basato sul gioco	https://shorturl.at/1SnBm	3/5
Esploratore del canto degli uccelli	Apprendimento supervisionato (etichettatura dati, classificazione, valutazione modelli)	Smartphone, classificatori, dati geo-referenziati, infografiche	https://shorturl.at/dR4q1	3/5

Alfabetizzazione Intelligenza Artificiale (IA) e Machine Learning (ML)				
Monitoraggio della qualità dell'aria esterna	Calibrazione sensori, acquisizione dati real-time, analisi inquinamento ambientale	Microcontrollori con sensori del biossido di azoto, GPS, e fogli di calcolo per manipolazione dati	https://shorturl.at/lvjXa	1/5
Monitoraggio della qualità dell'aria interna	Acquisizione e interpretazione dati, acquisizione dati ambientali, logica condizionale	Microcontrollori con sensori del biossido di azoto, GPS, e fogli di calcolo per manipolazione dati	https://shorturl.at/nycp9	1/5
Dai muri caldi alle città fresche	Raccolta dati, confronto e classificazione, visualizzazione	Sensori Temperatura/Umidità, termo-camere o termometri infrarossi, materiali isolanti, mappe urbane, Arduino	https://shorturl.at/Lr8Uh	3/5

Decibel Detective, Luce VS Zzz, Sound-Squad	Acquisizione dati, programmazione sensori ambientali, analisi temporale, visualizzazione dati	Micro:bit, sensori luce, suono, temperatura, umidità, Make-Code	https://shorturl.at/Xc8tm https://shorturl.at/FN13N https://shorturl.at/mQ7mU	3/5
Muri sussurranti	Programmazione microcontrollori per acquisizione e analisi dati audio	STM32 IoT Node, microfono, display LCD, buzzer, materiali isolanti, fogli di calcolo	https://shorturl.at/G8hU5	3/5

Tabella 1: Protocolli di SteamCity per l’apprendimento dell’informatica nell’ambito delle smart city, ordinati per difficoltà (abbreviato come diff.).

3. Sperimentazione sul campo

Durante il 2024–2025 sono stati realizzati laboratori formativi in Italia, Belgio e Francia, coinvolgendo sia studenti delle scuole di secondo ordine e grado che docenti di informatica, scienze e materie umanistiche che hanno sperimentato direttamente i protocolli. I dati raccolti derivano dall’osservazione dei partecipanti, questionari post-attività e focus group. Dalle sperimentazioni condotte emergono risultati incoraggianti sull’efficacia dell’approccio *SteamCity* per l’insegnamento dell’informatica nella scuola secondaria. In particolare, si osserva un aumento significativo della motivazione da parte degli studenti, attribuibile alla rilevanza concreta delle attività proposte e alla possibilità di affrontare questioni urbane reali attraverso strumenti tecnologici accessibili. L’uso di dispositivi fisici, microcontrollori e interfacce di programmazione visuale consente anche agli studenti con minore familiarità tecnica di partecipare attivamente, promuovendo un coinvolgimento inclusivo e trasversale. Tenendo in considerazione la difficoltà del protocollo, i risultati raccolti mediante sondaggi dimostrano che queste iniziative sono adatte ai diversi indirizzi di scuole secondarie di secondo grado.

Le attività stimolano lo sviluppo del pensiero computazionale e delle competenze di base nell’ambito dell’intelligenza artificiale, in particolare grazie all’utilizzo di modelli supervisionati, classificatori, chatbot e simulazioni logiche. Questi strumenti, lungi dall’essere affrontati in modo astratto, vengono inseriti in contesti significativi, come la gestione del traffico, la qualità dell’aria o il riconoscimento di segnali visivi, trasformando la programmazione in un’attività orientata alla risoluzione di problemi reali.

Un altro risultato rilevante riguarda la crescita della consapevolezza critica nei confronti della tecnologia e di informatica come scienza. Gli studenti dimostrano di sviluppare una maggiore

sensibilità verso temi etici come la privacy, i bias algoritmici e l'affidabilità dei dati, e sono stimolati a riflettere su come i sistemi intelligenti influenzino la vita urbana e le decisioni collettive. In questo senso, l'informatica viene presentata come una lente per comprendere e discutere la complessità della società contemporanea, collocandola correttamente come scienza capace di osservare e modellare fenomeni, estrarre dati e elaborare informazioni, per prendere decisioni informate. Dal punto di vista dei docenti e in base ai risultati dei sondaggi acquisiti, l'approccio è stato accolto positivamente per la sua flessibilità, la chiarezza delle strutture didattiche e la possibilità di adattare facilmente i protocolli a diversi livelli scolastici. Gli insegnanti apprezzano l'equilibrio tra rigore scientifico, creatività progettuale e interdisciplinarietà, e riconoscono nel modello *SteamCity* uno strumento valido per innovare la didattica e rendere più significative le esperienze di apprendimento. Diverse testimonianze indicano inoltre l'intenzione concreta di integrare le attività nei curricula, a dimostrazione del potenziale trasformativo del progetto.

Fondi

Questo lavoro è stato supportato dal progetto *SteamCity*, finanziato dal programma Erasmus+ dell'Unione Europea, nell'ambito dell'accordo di sovvenzione n. 2022-1-FR01-KA220-SCH-000087922. La presente comunicazione riflette esclusivamente il punto di vista degli autori e la Commissione non è responsabile dell'uso che potrà essere fatto delle informazioni in essa contenute.

Bibliografia

- [1] Yurdagul Bogar. Literature review on inquiry-based learning in science education. *Uluslararası Bilim ve Eğitim Dergisi*, 1(2):91–118, 2019.
- [2] Peter J Denning. Computing is a natural science. *Communications of the ACM*, 50(7):13–18, 2007.
- [3] Mauro D'Angelo and Maria Angela Pellegrino. Roobopoli: a project to learn robotics by a constructionism- based approach. In *Methodologies and Intelligent Systems for Technology Enhanced Learning*, 10th International Conference. Workshops: Volume 2, pages 249–257. Springer, 2021.
- [4] Rosella Gennari, Alessandra Melonio, and Mauro D'Angelo. Engaging learners in the collaborative design of sustainable smart cities. In *Proceedings of the 1st Sustainable, Secure, and Smart Collaboration Workshop in conjunction with CHITALY*, volume 3574 of *CEUR Workshop Proceedings*, pages 9–19. 2023.
- [5] Rosella Gennari, Alessandra Melonio, Maria Angela Pellegrino, and Mauro D'Angelo. How to playfully teach AI to young learners: a systematic literature review. In *Proceedings of the 15th CHItaly*, pages 1:1–1:9. ACM, 2023.
- [6] Zuzana Jesková, Stanislav Lukáč, Martina Hancová, Lubomír Sňajder, Ján Gunis, Brigita Balogová, and Marián Kires. Efficacy of inquiry-based learning in mathematics, physics and informatics in relation to the development of students' inquiry skills. *Journal of Baltic Science Education*, 15(5):559–574, 2016.
- [7] Alessandra Melonio, Maria Angela Pellegrino, and Mauro D'Angelo. Exploiting collaboration

- to engage high-school learners in experiencing sensors and actuators. In MIS4TEL, volume 1171, pages 67–79. Springer, 2024.
- [8] Nikolina Nikolova and Eliza Stefanova. Inquiry-based science education in secondary school informatics – challenges and rewards. In *Information Technology and Open Source: Applications for Education, Innovation, and Sustainability*, pages 17–34. Springer Berlin Heidelberg, 2014.
- [9] Jari Porras, Antti Knutas, Jouni Ikonen, Ari Happonen, Jayden Khakurel, and Antti Herala. Code camps and hackathons in education-literature review and lessons learned. *Hawaii International Conference on System Sciences (HICSS)*, 2019.
- [10] Eftychia Roumelioti, Maria Angela Pellegrino, Mehdi Rizvi, Mauro D’Angelo, and Rosella Gennari. Smart- thing design by children at a distance: How to engage them and make them learn. *International Journal Child Computer Interaction*, 33:100482, 2022.
- [11] Jennifer Rowley. The wisdom hierarchy: representations of the DIKW hierarchy. *Journal of information science*, 33(2):163–180, 2007.

Prototipo di un Gioco Unplugged per insegnare l'IA Generativa: Comprendere Costi, Bias e Sostenibilità

Leonardo Martino¹, Gennaro Iaccarino¹, Maddalena Braccesi¹

¹ Direzione Istruzione e Formazione Italiana via del Ronco 2, 39100 Bolzano, Italy

{leonardo.martino, gennaro.iaccarino, maddalena.braccesi}@provincia.bz.it

Abstract

In questo articolo proponiamo un prototipo di gioco “unplugged” concepito per facilitare la comprensione dei principi fondamentali alla base dell’addestramento dei modelli di Intelligenza Artificiale generativa. Al contempo, il gioco intende stimolare una riflessione critica sui temi dell’educazione civica digitale. L’implementazione del prototipo è pianificata per la seconda metà dell’anno scolastico 2025-2026, con una fase di sperimentazione pilota rivolta alle classi del primo biennio della scuola secondaria di secondo grado nella Provincia Autonoma di Bolzano.

1. Introduzione

Negli ultimi anni, l’Intelligenza Artificiale (IA) generativa ha conosciuto uno sviluppo senza precedenti, influenzando profondamente ambiti quali l’educazione, l’arte, l’informazione e il lavoro. La crescente diffusione di tali strumenti rende necessario promuovere nei giovani una comprensione critica dei meccanismi necessari al loro funzionamento, sin dalle fasi iniziali del percorso scolastico di secondo grado. Il CINI, nella sua proposta del 2017 [4], indica la consapevolezza digitale come competenza chiave per l’utilizzo professionale della tecnologia. In tale prospettiva, i giochi unplugged – attività didattiche che non prevedono l’uso di strumenti digitali – si configurano come una metodologia efficace per introdurre i fondamenti dell’IA generativa in modo accessibile e inclusivo [5, 8, 9, 13]. Attraverso la mediazione ludica, è possibile facilitare l’apprendimento di concetti complessi, stimolando al contempo competenze cognitive e metacognitive [3]. L’introduzione di principi di educazione civica digitale inoltre consente di superare un’impostazione esclusivamente tecnica, promuovendo l’interdisciplinarietà e coinvolgendo diverse aree del sapere. Come già evidenziato in [2, 7] le cosiddette attività socialmente rilevanti favoriscono il processo di apprendimento e stimolano il pensiero critico.

2. Il Gioco

All’inizio del gioco vengono distribuiti a ciascun giocatore, in maniera casuale, una *plancia*, una *carta Modello* e un *foglio addestramento* (Figura 1). L’obiettivo del gioco consiste nell’addestrare la propria IA in coerenza con il modello ricevuto, al fine di generare una risposta corretta e pertinente ad una richiesta dell’utente.

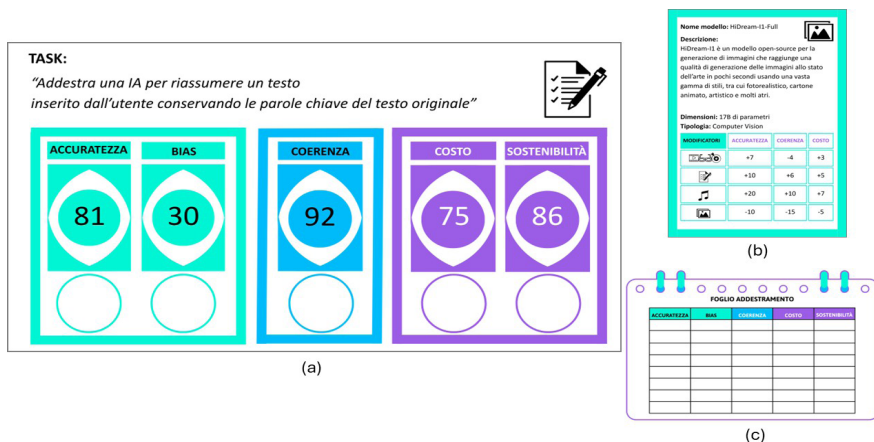


Figura 1: Setup iniziale per ogni giocatore. (a) Plancia da gioco contenente descrizione del task, icona corrispondente alla tipologia e parametri di addestramento, ognuno accompagnato dal relativo valore obiettivo iniziale. In basso sono collocati dei cerchi in cui il giocatore scrive gli obiettivi aggiornati, determinati dai modelli di AI ricevuti (carte Modello) e dagli eventi che si verificano durante la partita (carte Evento). (b) Carta Modello: aggiorna gli obiettivi della plancia come indicato dai modificatori. (c) Foglio addestramento: in ogni colonna vanno sommati i punteggi indicati nelle carte Dati, fino al raggiungimento degli obiettivi della plancia

Ogni *plancia* fornisce una descrizione del compito (*task*) da eseguire specificandone la tipologia, secondo la classificazione proposta in [14], nonché i parametri di addestramento (Sezione 2.2) e i relativi obiettivi da raggiungere, espressi da un valore numerico.

La fase di addestramento richiede l'utilizzo di *carte Dati*, le quali rappresentano input eterogenei per tipologia, accuratezza, presenza di bias, coerenza con il task, costo e sostenibilità (Figura 3). Questi cinque parametri costituiscono i criteri principali che ogni giocatore è chiamato a bilanciare nel corso del gioco, al fine di costruire un prompt efficace, sostenibile e il più possibile libero da bias e allucinazioni.

Il processo di addestramento può essere influenzato da eventi esterni – *carte Evento* – che introducono elementi di perturbazione in grado di modificare i valori obiettivo della plancia o attribuire un nuovo modello all'IA. Tali eventi possono avere effetti localizzati, agendo su una singola IA, oppure avere un impatto globale, coinvolgendo simultaneamente tutti i giocatori (Sezione 2.3).

Al proprio turno, ogni giocatore lancia il dado (Figura 2a) e prende la carta corrispondente al risultato. La probabilità associata a ciascuna tipologia di carta non è uniforme; questa distribuzione probabilistica è stata progettata per favorire una prevalenza dei dati *ordinari* rispetto agli eventi *straordinari*, rispecchiando in modo semplificato la realtà dell'addestramento di modelli di IA, in cui eventi eccezionali, seppur meno frequenti, possono avere impatti significativi. Se esce una *carta Dati*, il giocatore riporta i punteggi sul proprio *folio addestramento* e li somma ai precedenti, al fine di raggiungere i valori obiettivo indicati sulla plancia. Se esce una *carta Evento*, il giocatore deve aggiornare i valori obiettivo sulla plancia ed eventualmente cambiare modello. Nel corso della partita, ciascun giocatore aggiorna progressivamente sia il proprio

foglio addestramento sia gli obiettivi della plancia, in funzione delle caratteristiche del modello iniziale e degli avvenimenti che si verificano durante il gioco.

La partita si conclude quando un giocatore, al termine del proprio turno, dichiara di aver raggiunto gli obiettivi della plancia. A quel punto, tutti i partecipanti procedono al calcolo dello scarto assoluto, per ciascun parametro, tra il punteggio ottenuto e il corrispondente obiettivo della plancia. Vince il giocatore la cui somma degli scarti assoluti risulta più vicina allo zero.

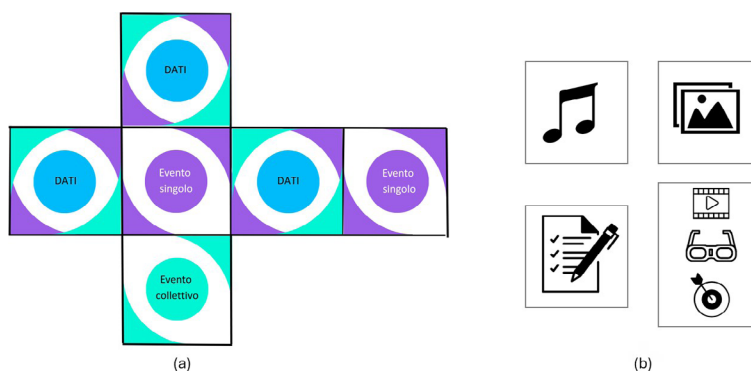


Figura 2: (a) Dado da gioco: ogni lancio prevede tre possibili esiti con probabilità differente: dati (1/2), evento singolo (1/3), evento collettivo (1/6). I tre esiti indicano la carta che il giocatore dovrà pescare. (b) Icone rappresentative delle tipologie di modello

2.1 Carta Modello

La funzione didattica della carta Modello è quella di presentare in forma ludica diversi modelli di IA, introducendo le loro caratteristiche, espresse con un opportuno linguaggio tecnico. I modelli nel campo dell'IA sono sistemi matematici addestrati per svolgere compiti specifici (task), quali il riconoscimento di immagini, la comprensione del linguaggio naturale o la generazione automatica di testi. Tuttavia, non esiste un modello universale in grado di eccellere in tutte le tipologie di task: alcuni risultano ottimizzati per l'elaborazione di dati visivi, altri per quella del linguaggio naturale, altri ancora sono specializzati in compiti di previsione numerica. La carta Modello fornisce agli studenti una sintesi visiva e concreta dei principali concetti legati ai modelli di IA. La Figura 2(b) mostra le icone rappresentative delle quattro tipologie di modello scelte:

1. *Multimodal*: integra simultaneamente diverse modalità informative per fornire risposte coerenti e contestualizzate. È alla base di sistemi avanzati in grado di combinare linguaggio, visione e audio in modo sinergico.
2. *Natural Language Processing (NLP)*: analizza e genera linguaggio naturale, scritto o parlato. Si concentra esclusivamente su dati testuali, apprendendo struttura, significato e contesto linguistico.
3. *Audio*: elabora segnali acustici, parlato e musica. Identifica, classifica o genera contenuti audio, includendo funzioni di trascrizione e sintesi vocale.

4. *Computer Vision*: consente l'interpretazione automatica di immagini e video, attraverso il riconoscimento di oggetti, forme, colori e movimenti, e l'estrazione di informazioni visive complesse.

La carta, quindi, riporta il nome di un modello reale, la tipologia di task per cui risulta più adatto, una breve descrizione dei suoi parametri costitutivi [14], nonché i modificatori che il giocatore dovrà utilizzare per adattare gli obiettivi di accuratezza, coerenza e costo della propria plancia. La quantificazione numerica di tali caratteristiche, tradotte nei parametri di gioco, è ottenuta tramite un confronto prestazionale tra modelli differenti [1, 6, 10, 11], con l'obiettivo di offrire una semplificazione verosimile della realtà. I modificatori influenzano il raggiungimento dei tre obiettivi facilitando, in caso di valori negativi, o peggiorando, nel caso di valori positivi, il raggiungimento del punteggio da ottenere per soddisfare i requisiti del task. Il giocatore che riceve una carta Modello, all'inizio della partita o a causa di una carta Evento, deve aggiornare i propri valori obiettivo secondo i valori dei modificatori (Figura 1(a)). La Figura 4(a) mostra una carta Modello con i valori obiettivo da modificare nella plancia di gioco.

2.2 Carta Dati

I dati rappresentano un elemento cruciale nell'addestramento delle IA. Essi forniscono la base da cui il modello apprende, influenzando direttamente la sua accuratezza, affidabilità e coerenza rispetto al task. Dati fuorvianti o distorti possono introdurre bias significativi, con implicazioni etiche e sociali. La qualità e la rappresentatività del dataset sono quindi determinanti per le prestazioni del modello di IA. In questo contesto, le carte Dati rappresentano i dataset utilizzati per l'addestramento dei modelli di IA assegnati ai giocatori. Al centro di ciascuna carta è indicata, tramite un'icona, la tipologia di task per cui il dato risulta particolarmente efficace. Se questa tipologia coincide con quella riportata sulla plancia del giocatore, il valore di coerenza sarà elevato (Figura 3); in caso contrario, il valore sarà molto basso o addirittura negativo. In entrambi i casi, il giocatore annota sul proprio foglio addestramento il valore del parametro coerenza corrispondente alla tipologia del task riportato sulla propria plancia. Ai quattro vertici della carta sono riportati i punteggi per i restanti parametri da riportare sul foglio addestramento: accuratezza, bias, costo e sostenibilità. Il valore dei punteggi di questi quattro parametri è spesso interconnesso tra loro. Ad esempio, dataset gratuiti o a basso costo possono essere associati a livelli bassi di accuratezza; viceversa, dataset ad alto costo possono comportare una sostenibilità ambientale ridotta, a causa dell'elevato impiego di risorse computazionali. Di seguito, una breve descrizione dei parametri considerati:

- *Costo*: indica il costo computazionale ed economico per l'acquisizione del dato. In una IA bisogna, infatti, considerare i costi per acquisire le informazioni, per gli strumenti tecnologici e per le risorse necessarie a rendere i dati utilizzabili.
- *Coerenza*: indica la pertinenza tra il dato e la tipologia di task. Molto spesso nell'addestramento di un modello bisogna procedere ad una seconda fase, chiamata *fine tuning*, che permette di migliorarne le risposte ad uno specifico task.
- *Sostenibilità*: indica l'impatto ambientale dell'acquisizione di un dato in termini consumo energetico. L'analisi di grandi quantità di dati implica un notevole consumo energetico che deve essere gestito da chi progetta l'addestramento dell'IA.
- *Accuratezza*: indica il grado di correttezza e precisione del dato. La raccolta di elementi per l'addestramento implica inevitabilmente anche l'acquisizione di dati incompleti, sintetici,

non corretti o poco precisi. Un buon addestramento deve ottimizzare la qualità dei dati raccolti.

- **Bias:** indica la presenza di elementi derivanti da stereotipi impliciti o rappresentazioni parziali della realtà. Minimizzare i bias è fondamentale per non compromettere l'equità e l'affidabilità etica delle decisioni prodotte dal modello.

La raccolta di carte Dati contribuisce al raggiungimento degli obiettivi indicati nella plancia di gioco.

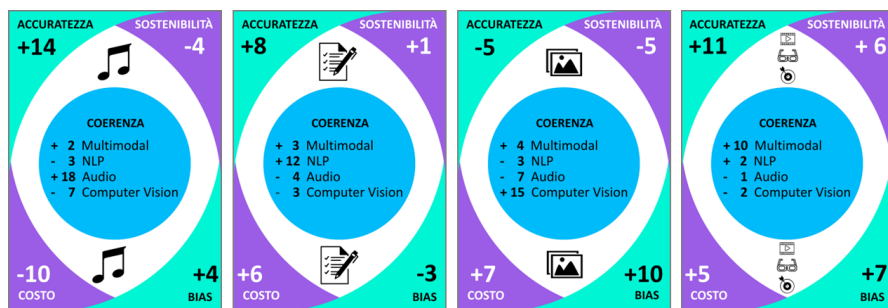


Figura 3: Esempio di carte Dati. L'icona rappresenta la tipologia di task a cui si riferisce. Il valore del parametro coerenza varia in base dalla tipologia di task riportato sulla plancia del singolo giocatore

2.3 Carta Evento

All'interno del gioco, le *carte Evento* rappresentano stimoli esterni che simulano l'impatto di dinamiche del mondo reale, quali notizie, trend emergenti, modifiche normative o fenomeni sociali, sul processo di addestramento di un'IA. Da un punto di vista didattico, tali carte evidenziano come le fonti di dati e i contesti socio-culturali possano influenzare in modo significativo sia la fase di addestramento che il comportamento risultante del modello. In termini pratici, impongono ai giocatori o di modificare uno o più valori obiettivo della plancia, o di sostituire il modello in uso con uno differente. Le carte evento agiscono come meccanismo di perturbazione controllata, stimolando il pensiero critico, la capacità di problem solving e la comprensione delle dinamiche adattive dell'IA, rispetto al cambiamento dei parametri iniziali (Figura 1).

- **Evento singolo.** Questa carta prevede le modifiche del valore di alcuni degli obiettivi della plancia, esclusivamente per il giocatore che ha preso la carta. Al termine delle modifiche, il turno passa al giocatore successivo. Tra le carte evento singolo è previsto anche il cambio del modello di IA. In tal caso il giocatore dovrà prendere dal mazzo *carta Modello* un nuovo modello di IA e aggiornare gli obiettivi della propria plancia di gioco. In Figura 4(b) un esempio di carta evento singolo.
- **Evento collettivo.** Questa carta prevede le modifiche del valore di alcuni degli obiettivi della plancia, per tutti i giocatori. Un evento collettivo ha carattere globale e influenza dunque tutte le IA in gioco. In Figura 4(c) un esempio di carta evento collettivo. Al termine del turno di gioco, dopo che tutti i giocatori hanno modificato i valori della propria plancia, a seconda delle indicazioni della carta, si procede a una fase di riequilibrio globale. In questa

fase, ogni giocatore seleziona, tra le carte Dati in suo possesso, quella che ritiene meno utile all'addestramento della propria IA e la colloca coperta al centro del tavolo. Tutte le carte così raccolte vengono mescolate dal giocatore che ha lanciato il dado e poi redistribuite casualmente tra i partecipanti. Questo meccanismo simula la circolazione dei dati nel sistema, sottolineando l'imprevedibilità e la complessità nell'approvvigionamento delle fonti per l'IA. Tutti i giocatori inseriscono i nuovi punteggi nel proprio *foglio addestramento* (Figura 1(c)).

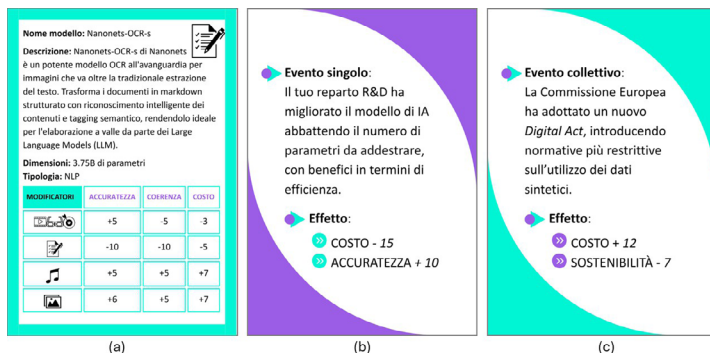


Figura 4: Prototipo di carte da gioco. (a) Carta Modello - In questo esempio prendiamo in considerazione il modello Nanonets-OCR-S [12]. (b) Carta Evento singolo. (c) Carta Evento collettivo

3. Conclusioni

L'attività proposta, sviluppata in forma di prototipo, si pone l'obiettivo di introdurre gli studenti e le studentesse del primo biennio della scuola secondaria di secondo grado alle principali dinamiche che caratterizzano il processo di addestramento di un sistema di IA generativa. Parallelamente, intende stimolare una riflessione critica su tematiche di forte attualità nell'ambito dell'educazione civica digitale, quali la sostenibilità ambientale connessa al consumo energetico dei modelli di IA, la presenza di bias e le implicazioni sociali derivanti dal loro impiego. In questa prospettiva, il gioco non si configura unicamente come strumento didattico, ma anche come ambiente formativo orientato alla promozione della consapevolezza e della cittadinanza digitale attiva [4]. Il gioco permette inoltre di evidenziare come i modelli generativi di IA operino su base statistica, attribuendo significato ai dati in modo radicalmente diverso rispetto al pensiero umano. Emerge con chiarezza l'importanza del modello e della qualità dei dati, a fronte del contenuto semantico delle informazioni trattate. Questo aspetto è particolarmente utile per sviluppare negli studenti una prospettiva critica sul rapporto tra tecnologia, conoscenza e interpretazione.

In un prossimo futuro sarà possibile implementare il gioco e verificarne l'efficacia nelle classi del primo biennio di tutti i percorsi scolastici della Provincia Autonoma di Bolzano.

Bibliografia

- [1] Artificial Analysis. Independent analysis of AI. <https://artificialanalysis.ai/>, 2025.
- [2] Anna Brancaccio, Luis Corral, Ilenia Fronza, and Gennaro Iaccarino. Building smart apps for smart cities: un esempio di sinergia tra PCTO ed educazione civica, concluso ai tempi di COVID-19. DIDAMATICA 2020 - Atti del convegno, pages 380–389, 2020.
- [3] Peng Chen, Dong Yang, Ahmed Hosny Saleh Metwally, Jari Lavonen, and Xiao Wang. Fostering computational thinking through unplugged activities: A systematic literature review and meta-analysis. *International Journal of STEM Education*, 10(1):47, 2023.
- [4] CINI. Proposta di Indicazioni Nazionali per l'insegnamento dell'Informatica nella Scuola. <https://www.consortio-cini.it>, 2017.
- [5] Yun Dai. Integrating unplugged and plugged activities for holistic ai education: An embodied constructionist pedagogical approach. *Education and Information Technologies*, pages 1–24, 2024.
- [6] Colin White et al. Livebench: A challenging, contamination-free LLM benchmark. In *Proceedings of the Thirteenth International Conference on Learning Representations*, 2025.
- [7] Gennaro Iaccarino, Sara Tosi, Ilenia Fronza, and Luis Corral. Laboratorio di #bioinformatica: un PCTO innovativo in periodo pandemico e post pandemico. DIDAMATICA 2022 - Atti del convegno, pages 296–305, 2022.
- [8] Lukas Lehner and Martina Landman. Unplugged decision tree learning—a learning activity for machine learning education in k-12. In *International Conference on Creative Mathematical Sciences Communication*, pages 50–65. Springer, 2024.
- [9] Hansol Lim, Wookhee Min, Jessica Vandenberg, Veronica Cateté, Judith Uchidiuno, and Bradford Mott. Supporting student engagement in k-12 ai education with a card game construction toolkit. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 2*, pages 1718–1719, 2024.
- [10] LLM-Stats. LLM leaderboard. <https://llm-stats.com/>, 2025.
- LMarena. Leaderboard overview. <https://lmarena.ai/leaderboard>, 2025.
- [11] Souvik Mandal, Ashish Talewar, Paras Ahuja, and Prathamesh Juvatkar. Nanonets-ocr-s: A model for transforming documents into structured markdown with intelligent content recognition and semantic tagging, 2025.
- [12] Yukyeong Song, Xiaoyi Tian, Nandika Regatti, Gloria Ashiya Katuka, Kristy Elizabeth Boyer, and Maya Israel. Artificial intelligence unplugged: Designing unplugged activities for a conversational ai summer camp. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*, pages 1272–1278, 2024.
- [13] Hugging Face Team. Hugging face - The AI community building the future. <https://huggingface.co>, 2025.

CriptoLab... alla scoperta della crittografia. Dalla teoria degli algoritmi al coding crittografico

Francesco Paolo Caforio¹

¹ IISS Majorana, Martina Franca, TA, Italia

francescopaolo.caforio@majorana.net

Abstract

CriptoLab – Alla scoperta della crittografia è stato un laboratorio extracurricolare di 30 ore rivolto ad un gruppo di studenti dell'IISS Majorana di Martina Franca (TA) e realizzato nel corso dell'anno scolastico 2024/25 nell'ambito del progetto *SySTEM – Let's bridge the gap*. Il percorso è stato strutturato in moduli teorico-pratici, iniziando dall'analisi dei cifrari storici e giungendo agli algoritmi di cifratura moderni. Attraverso progressive esercitazioni in Python e l'adozione di metodologie didattiche attive, gli studenti hanno potenziato competenze logico-computazionali e acquisito consapevolezza sui meccanismi di protezione dei dati. L'esperienza ha inoltre valorizzato l'approccio scientifico all'informatica, favorendo l'integrazione tra teoria, sperimentazione e riflessione critica. Le attività finali hanno rivelato una crescente autonomia nella progettazione di algoritmi crittografici e un rinnovato entusiasmo verso le discipline STEM, suggerendo l'efficacia e la scalabilità del seguente percorso in contesti didattici differenti.

1. Introduzione

L'informatica rappresenta oggi una disciplina cardine per interpretare e agire nella complessità del mondo contemporaneo che costituisce un sistema teorico e applicativo, dotato di linguaggi propri, strutture formali e una forte interconnessione con la logica matematica. Promuoverla come disciplina scientifica significa evidenziarne i fondamenti epistemologici e stimolare il pensiero computazionale. In tale prospettiva, le esperienze didattiche sul campo rivestono un ruolo cruciale. Esse coniugano dimensione teorica e operatività, stimolano competenze trasversali e avvicinano gli studenti alle discipline STEM in maniera concreta, motivante e orientativa. Il decreto del Ministro dell'Istruzione e del Merito del 12 aprile 2023 n. 65 si inserisce pienamente in questa visione strategica, in quanto riconducibile alla linea di investimento 3.1 “Nuove competenze e nuovi linguaggi” della Missione 4 – Componente 1 del Piano Nazionale di Ripresa e Resilienza (PNRR), finalizzata a integrare nei curricula scolastici attività, metodologie e contenuti innovativi per lo sviluppo delle competenze STEM, digitali e di innovazione [3]. La crittografia, in particolare, rappresenta un ambito significativo in cui convergono modelli teorici, applicazioni pratiche e implicazioni sociali [5]. Introdurla nel contesto scolastico significa offrire agli studenti una chiave di lettura concreta per comprendere come funziona la protezione delle informazioni nei sistemi digitali, quali sono i rischi legati

alla sicurezza dei dati, e come la matematica e la programmazione possano contribuire alla costruzione di ambienti digitali più sicuri e affidabili.

Il presente contributo documenta un'esperienza realizzata presso l'IISS Majorana di Martina Franca, svolta nel laboratorio *CriptoLab* – *Alla scoperta della crittografia*, finanziato con risorse PNRR – DM 65/2023 e attivato nell'ambito del progetto *SySTEM* – *Let's bridge the gap*. L'iniziativa è stata concepita con l'obiettivo di promuovere un approccio attivo all'informatica, attraverso compiti operativi, collaborazione tra pari, riflessione critica e pratiche di codifica autentica.

2. *CriptoLab*: specifiche formative e ambito applicativo

Il laboratorio *CriptoLab*, condotto dal docente Francesco Paolo Caforio, ha coinvolto un gruppo di circa 30 studenti frequentanti l'ultimo triennio dell'Istituto Tecnico – Settore Tecnologico, indirizzo “Informatica e Telecomunicazioni” – Articolazione “Informatica”, e del Liceo Scientifico – opzione Scienze Applicate. Il percorso extracurricolare si è sviluppato attraverso una sequenza di attività didattiche progressive, che hanno guidato gli studenti in un viaggio tra conoscenze storiche, concetti informatici e strumenti matematici, fino all'analisi e alla realizzazione di algoritmi crittografici. La durata complessiva del laboratorio è stata di 30 ore, suddivise in 10 incontri pomeridiani da 3 ore ciascuno.

I contenuti affrontati sono sintetizzati nella Tabella 1, che evidenzia l'articolazione tematica, metodologica e la scansione temporale delle attività proposte.

Modulo	Contenuti trattati	Attività e metodologie	Durata
<i>Crittografia</i>	Fondamenti di crittografia. Crittografia simmetrica e asimmetrica. Applicazioni reali	Lezione frontale-dialogata con esempi reali, cooperative learning	3h
<i>Aritmetica modulare</i>	Conversioni tra sistemi numerici. Proprietà dell'aritmetica modulare. Operazioni modulo n. Fondamenti matematici per la cifratura	Esercitazione guidata e individuale, attività di gruppo su problemi logico-matematici	6h
<i>Cifrari classici</i>	Studio di cifrari storici: Cesare (monoalfabetico), Vigenère (polialfabetico) e Vernam (one-time pad). Analisi della sicurezza e tecniche base di crittoanalisi	Analisi di casi, codifica e decodifica manuale, costruzione collaborativa di infografiche	6h
<i>Crittografia moderna</i>	Introduzione agli algoritmi a blocchi (DES, AES) e a chiave pubblica (RSA)	Discussione guidata, simulazioni, cooperative learning su scenari di cifratura. Attività laboratoriale, codifica assistita, esercitazione individuale e di gruppo	6h

<i>Firma digitale</i>	Funzioni hash: proprietà. Firma digitale: autenticazione, integrità e non ripudio. Applicazioni: firme elettroniche, verifica file, blockchain	Lezione partecipata, esercizi su scenari reali, discussione guidata	6h
<i>Monitoraggio</i>	Valutazione dei risultati di apprendimento. Autovalutazione. Discussione finale sulle competenze acquisite e sugli strumenti utilizzati	Discussione guidata, restituzione e validazione dei prodotti in uscita, monitoraggio finale e riflessione metacognitiva	3h

Tabella 1: Elenco dei moduli del laboratorio CriptoLab

Il laboratorio ha favorito un dialogo significativo tra informatica e matematica, affrontando contenuti fondamentali per la comprensione della crittografia in chiave computazionale.

Dal punto di vista metodologico, ogni modulo ha previsto: lezioni dialogate introduttive supportate da schede, infografiche strutturate e discussioni guidate; esercitazioni manuali sui cifrari per stimolare il ragionamento logico e la comprensione dei meccanismi; attività individuali e di gruppo, anche su casi pratici o simulazioni di attacchi; laboratori digitali interattivi su *Google Colab*, dove gli studenti hanno esplorato i meccanismi della crittografia codificando in Python diversi algoritmi crittografici. Per favorire la replicabilità del laboratorio in altri contesti, una selezione dei principali materiali didattici, delle fonti bibliografiche, delle risorse di approfondimento e dei codici sviluppati è stata pubblicata in un repository dedicato [2].

3. Struttura e funzionamento di algoritmi crittografici

Il laboratorio ha innanzitutto previsto un approfondimento sui cifrari classici, con l'obiettivo di introdurre i concetti fondamentali della crittografia simmetrica e delle operazioni bitwise.

Per il *cifrario di Cesare* è stato analizzato il meccanismo di sostituzione monoalfabetica, in cui ciascun carattere P del testo in chiaro è stato traslato di un numero fisso K di posizioni lungo l'alfabeto, secondo la formula $C = (P + K) \bmod n$, dove n rappresenta la dimensione dell'alfabeto. Di seguito è stata proposta un'attività di crittoanalisi volta a evidenziare i limiti dello stesso cifrario.

Nel caso del *cifrario di Vernam*, ogni carattere del messaggio in chiaro è stato convertito in codice ASCII e successivamente trasformato in una rappresentazione binaria a 8 bit. È stata poi generata una chiave binaria casuale della stessa lunghezza del messaggio, sulla quale è stata applicata l'operazione XOR per ottenere il testo cifrato. La stessa operazione è stata utilizzata in fase di decifrazione, dimostrando sperimentalmente la proprietà di reversibilità dello XOR.

L'approfondimento del *Data Encryption Standard* ha poi rappresentato il nucleo centrale del percorso laboratoriale. Dopo un inquadramento storico e concettuale dell'algoritmo [4], gli studenti ne hanno esaminato le componenti strutturali attraverso una versione semplificata, ricostruendone passo dopo passo il funzionamento logico con il supporto dello schema visivo riportato in Figura 1. Il percorso si è concentrato sulle seguenti fasi:

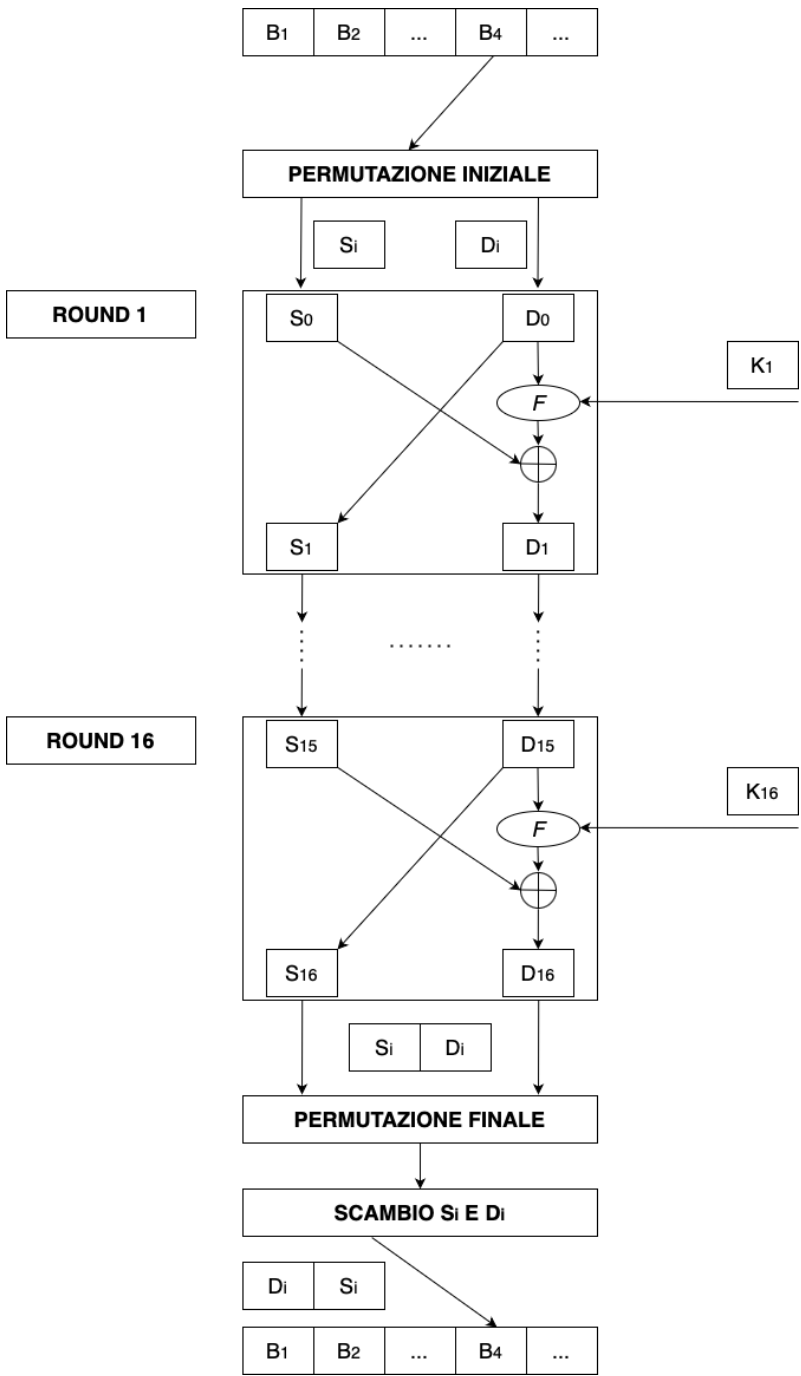


Figura 1: Struttura semplificata dell'algoritmo DES

1. **permutazione iniziale** (IP) sul blocco di 64 bit in ingresso;
2. **suddivisione del blocco** in due semiblocchi da 32 bit, L_i ed R_i ;
3. **ripetizione di 16 round**, in cui:
 - si calcola $R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$
 - si imposta $L_i = R_{i-1}$
 - all'interno della funzione di Feistel F , si effettuano:
 - l'espansione del semiblocco destro da 32 a 48 bit;
 - l'operazione di XOR con la sottochiave K_i ;
 - la sostituzione tramite *S-box*: i 48 bit ottenuti vengono suddivisi in 8 blocchi da 6 bit ciascuno. Ogni blocco viene trasformato in un blocco da 4 bit mediante una funzione di sostituzione (*S-box*);
4. **permutazione finale** (FP) al termine dei 16 round.

Per ragioni didattiche, si è scelto di semplificare il processo di generazione delle 16 sottochiavi; gli studenti, infatti, hanno lavorato con un insieme di sottochiavi predefinite, in modo da concentrarsi sulla struttura e sul funzionamento dell'algoritmo DES.

Il laboratorio ha anche proposto un'introduzione applicativa alle *firme digitali*. Gli studenti hanno analizzato le caratteristiche delle *funzioni hash*, osservandone le principali proprietà, tra cui la sensibilità alle modifiche minime del messaggio e la non reversibilità. Attraverso questa attività è stato possibile comprendere in modo operativo come la *firma digitale* consenta di garantire integrità, autenticità e non ripudio delle informazioni trasmesse.

Le attività hanno previsto un'articolazione tra esercitazioni tradizionali e momenti di codifica. Gli studenti hanno eseguito tracciamenti su carta dei passaggi per comprendere a fondo la logica trasformativa degli algoritmi, implementato in modo modulare le diverse componenti degli algoritmi crittografici in *Python* e utilizzato un notebook *Google Colab*, arricchito da spiegazioni, output intermedi e sezioni dedicate alla sperimentazione guidata, con possibilità di modifica e test attivo del codice.

4. Studio strutturato e implementazione semplificata di algoritmi crittografici

Per consolidare la comprensione degli algoritmi crittografici e renderne tangibili i meccanismi interni, il laboratorio *CriptoLab* ha previsto attività implementative in linguaggio *Python* realizzate in ambiente *Google Colab*. Le attività si sono articolate come segue:

- per il *cifrario di Cesare*, è stata realizzata una funzione per la cifratura basata sulla traslazione dei caratteri secondo una chiave intera, seguita dalla funzione inversa di decifratura; gli studenti hanno inoltre implementato un attacco a forza bruta mediante ciclo iterativo, testando tutte le possibili chiavi;
- relativamente al *cifrario di Vernam*, è stata costruita una funzione di conversione carattere per carattere in codice ASCII binario, accompagnata da una procedura per la generazione di una chiave casuale e da due funzioni per la cifratura e la decifratura mediante operazione XOR;
- l'implementazione del DES ha rappresentato il nucleo centrale del percorso, in quanto la struttura dell'algoritmo [4] offre un modello chiaro e sistematico per introdurre i concetti fondamentali della crittografia moderna, quali la cifratura a blocchi, l'uso delle chiavi simmetriche e le operazioni di permutazione e sostituzione;

- per la *firma digitale*, gli studenti hanno realizzato funzioni per la generazione del *digest* del messaggio originale mediante funzione hash, per la concatenazione tra messaggio e *digest* volta a simulare la trasmissione firmata, e di verifica per il confronto tra *digest* ricevuto e ricalcolato.

Questo approccio ha rafforzato le competenze computazionali e crittografiche degli studenti, promuovendo l'autonomia nella progettazione e codifica di algoritmi. Nel repository [1] sono archiviate le più importanti implementazioni realizzate durante il laboratorio, corredate da codice eseguibile, commenti esplicativi, esempi di output e sezioni dedicate alla sperimentazione autonoma. L'uso di *Python* ha abbassato la soglia di accesso grazie alla sua sintassi semplice e leggibile, mentre l'integrazione con *Google Colab* ha facilitato l'apprendimento collaborativo e l'interazione continua tra studenti e docente. Il passaggio dalla teoria all'implementazione, quindi, ha contribuito a consolidare in modo significativo le conoscenze, a rafforzare il pensiero computazionale e a stimolare l'interesse verso l'informatica come disciplina scientifica, creativa e critica.

5. Valutazione dell'efficacia e impatto del laboratorio formativo

Il laboratorio *CriptoLab* ha dimostrato come la crittografia possa costituire un potente motore didattico, in grado di integrare informatica e matematica, attivare competenze logico-analitiche, promuovere il pensiero computazionale e orientare concretamente gli studenti verso percorsi formativi e professionali in ambito STEM.

L'efficacia del percorso è stata monitorata attraverso una combinazione di strumenti valutativi. Durante lo svolgimento delle attività sono state condotte *osservazioni sul campo* finalizzate a rilevare livelli di partecipazione, modalità di interazione nei gruppi e capacità di risoluzione autonoma dei problemi.

Sono state previste *consegne* individuali e di gruppo, valutate attraverso rubriche strutturate che prendevano in esame la qualità dei prodotti in uscita (correttezza, funzionalità, aderenza agli obiettivi), il processo di lavoro (cooperazione, gestione del tempo, strategie adottate) e la dimensione metacognitiva (capacità di autovalutazione, riflessione sugli errori, consapevolezza).

L'esperienza ha contribuito positivamente al percorso formativo, accompagnandosi a un riscontro generale di miglioramento negli apprendimenti delle discipline informatiche al termine dell'anno scolastico. In particolare, sono emersi progressi in alcune aree chiave: maggiore sicurezza nella traduzione di algoritmi teorici in codice; crescente autonomia nella progettazione di semplici soluzioni algoritmiche; migliorata capacità di cooperare nella risoluzione di problemi tecnici. L'iniziativa ha pertanto rappresentato non solo un esempio di didattica STEM integrata, ma anche un'occasione efficace per promuovere un apprendimento orientato alla comprensione profonda e al trasferimento delle competenze in contesti nuovi. Esperienze di questo tipo possono contribuire a rendere l'insegnamento più coinvolgente e significativo, sostenendo lo sviluppo di competenze trasversali – come il pensiero logico, la curiosità, la capacità di analisi e la creatività – utili per affrontare i cambiamenti e le sfide professionali del presente e del futuro.

Bibliografia

- [1] Francesco Paolo Caforio. Criptolab: implementazioni in Python di algoritmi di crittografia. <https://github.com/francescocaforio/CriptoLab>, 2025. Accessed: 2025-06-01.
- [2] Francesco Paolo Caforio. Criptolab: materiali didattici, risorse e fonti. <https://github.com/francescocaforio/CriptoLab/wiki>, 2025. Accessed: 2025-06-01.
- [3] Ministero dell'Istruzione e del Merito. Nuove competenze e nuovi linguaggi (dm 65/2023). <https://pnrr.istruzione.it/competenze/nuove-competenze-e-nuovi-linguaggi/>, 2023. Accessed: 2025-06-01.
- [4] Behrouz A Forouzan. Introduction to Cryptography & Network Security. McGraw-Hill, Inc., 2007.
- [5] Michael Lodi, Marco Sbaraglia, Simone Martini, et al. Programmare per imparare la crittografia al liceo matematico. *Rendiconto del Seminario Matematico*, 80(2):49–81, 2022.

Didattica del prompting: “SalvIAmo il pianeta”

Luca Basteris¹, Maria Cristina Daperno², Alessandra Vassallo³, Valerio Milano⁴

¹ Liceo Classico e Scientifico Statale “Silvio Pellico – Giuseppe Peano”

luca.basteris@liceocuneo.it

² Liceo Classico e Scientifico Statale “Silvio Pellico – Giuseppe Peano”

cristina.daperno@liceocuneo.it

³ I.I.S. Arimondi Eula - Savigliano – Racconigi

vassallo.alessandra@arimondienla.edu.it

⁴ Liceo Scientifico G. B. Bodoni

valerio.milano@liceobodoni.edu.it

Abstract

In questo articolo viene presentata un'attività progettata da alcuni componenti delle Equipe Formative Territoriali e la sua sperimentazione didattica nell'ambito del progetto IA-LAB, un percorso formativo basato sulle recenti “AI competency framework” elaborati dall'UNESCO che si pone come obiettivo quello di integrare le competenze legate all'intelligenza artificiale nella pratica didattica. In particolare, l'attività proposta pone particolare attenzione alle implicazioni legate alla scrittura del prompt e a quanto generato dall'Intelligenza Artificiale. Attraverso la simulazione di alcuni contesti, gli studenti collaborano per progettare e testare prompt strategici, sviluppando competenze di problem solving e riflettendo sull'impatto umano delle loro soluzioni.

1. Introduzione

In questo articolo viene presentata la sperimentazione didattica di un'attività progettata da alcuni componenti delle Equipe Formative Territoriali nell'ambito del progetto IA-LAB, un percorso formativo basato sulle recenti “AI competency framework” elaborati dall'UNESCO che si è posta l'obiettivo quello di integrare le competenze legate all'intelligenza artificiale nella pratica didattica. Hanno lavorato alla progettazione dell'attività i professori: Massimo Berardi EFT Lombardia, Lara Corvatta EFT Marche, Roberto Ghelli EFT Toscana, Raffaella Giacobbi EFT Veneto, Marta Sanz Manzanedo EFT Toscana, Giovanni Silvestro EFT Toscana e Sandra Troia EFT Puglia. Le classi in cui è stata sperimentata l'attività sono:

- classe 2°E del Liceo Classico- Scientifico “Pellico-Peano” di Cuneo: formata da 24 studenti del Liceo Scientifico Opzione Scienze Applicate e il docente che ha condotto la sperimentazione è la prof.ssa Maria Cristina Daperno.
- classe 1°D AFM dell'Istituto Superiore Arimondi-Eula di Savigliano: formata da 20 studenti e i docenti che hanno condotto la sperimentazione sono il prof. Luca Basteris e la prof.ssa Alessandra Vassallo.

- classe 1°E del Liceo Scientifico G. B. Bodoni di Saluzzo: formata da 24 studenti classe del Liceo Scientifico Opzione Scienze Applicate e il docente che ha condotto la sperimentazione è il prof. Valerio Milano.
- classe 1°E dell'Istituto Superiore Arimondi-Eula di Savigliano: formata da 22 studenti classe del Liceo Scientifico Opzione Scienze Applicate e il docente che ha condotto la sperimentazione è la prof.ssa Alessandra Vassallo.

L'attività sviluppa la competenza "Mentalità incentrata sull'uomo" dell'UNESCO AI Competency Framework, favorendo un approccio etico e responsabile all'IA. Attraverso la simulazione di una crisi, gli studenti collaborano per progettare e testare prompt strategici, sviluppando competenze di problem solving e riflettendo sull'impatto umano delle loro soluzioni. La fase di debriefing li aiuta a comprendere le caratteristiche di prompt efficaci, stimolando creatività e consapevolezza sull'uso etico dell'IA. Lavorando in gruppo, imparano a confrontarsi e a considerare valori umani fondamentali, promuovendo una visione collaborativa e riflessiva sull'innovazione tecnologica al servizio della società.

L'attività prevede l'utilizzo di una piattaforma di AI. Tra quelle proposte è stata scelta Perplexity in quanto rappresenta uno strumento gratuito, non richiede accesso e soprattutto è la più utilizzata in ambito scolastico e accademico in quanto fornisce, senza doverne fare richiesta, i riferimenti utilizzati per la generazione del testo. In questi mesi in cui è terminata la sperimentazione didattica anche altre piattaforme di intelligenza artificiale hanno implementato direttamente questo servizio.



Figura 1: Logo del progetto

2. Attività didattica

L'attività didattica prevede un'attività in laboratorio di informatica della durata di 2 ore totali ed è suddivisa in 5 fasi:

- **Fase 1:** Icebreaker.
- **Fase 2:** Presentazione.
- **Fase 3:** Attività laboratoriale.
- **Fase 4:** Debriefing.
- **Fase 5:** Condivisione e feedback tra pari.

2.1 Fase 1: Icebreaker

Nella prima fase viene presentata un'attività interattiva per stimolare la creatività degli studenti chiedendo loro di immedesimarsi in un personaggio di fantascienza e di dialogare con un chatbot per affrontare un problema, (es. nel suo modulo spaziale sta per terminare l'ossigeno) Gli studenti annotano su di una bacheca digitale le interazioni con l'intelligenza artificiale, trascrivendo o facendo copia e incolla del testo ottenuto oppure, se il chatbot lo consente, riportando il link alla conversazione. Al termine il docente invita gli studenti a riflettere su quali quesiti hanno generato le soluzioni più efficaci.

2.2 Fase 2: Presentazione

Nella seconda fase si entra nel vivo dell'attività spiegando agli studenti il contesto narrativo: “Siete in un futuro prossimo. Il vostro mondo sta affrontando una crisi idrica senza precedenti: le risorse d'acqua dolce sono sempre più scarse e la sopravvivenza della comunità dipende da decisioni rapide e strategiche. Come possiamo preservare questa risorsa fondamentale? Le soluzioni che individuerete saranno cruciali! Facciamoci aiutare dall'intelligenza artificiale”. In questa fase è necessario dare alcune indicazioni di metodo ai ragazzi, in quanto l'IA può offrire idee, spunti e simulazioni, ma sarà la loro capacità di fare domande mirate e interpretare le risposte a guidare il processo decisionale. NON bisogna seguire ciecamente l'IA, ma utilizzarla come strumento per stimolare la creatività e risolvere problemi complessi.

Sarà necessario ricordare che quando si dialoga con l'Intelligenza artificiale, è fondamentale che il prompt sia chiaro, dettagliato e funzionale all'obiettivo: più la domanda è ben formulata, più la risposta sarà efficace per la risoluzione del problema.

Ecco alcuni esempi di prompt:

- Esempio 1 – Richiesta generica: “Spiega le cause principali della scarsità d'acqua nel mondo.”
- Esempio 2 – Richiesta specifica con dettagli: “Suggerisci tre soluzioni sostenibili per risparmiare acqua in una città con 50.000 abitanti. Ogni soluzione deve includere vantaggi e potenziali difficoltà.”
- Esempio 3 – Creatività e contesto narrativo: “Immagina di essere un esperto in gestione delle risorse idriche e devi aiutare una città in crisi idrica. Crea una strategia a lungo termine per risolvere il problema.”

2.3 Fase 3: Attività laboratoriale

In questa terza fase gli studenti iniziano a formulare le domande e a ipotizzare le azioni necessarie per risolvere il problema. Lavorano in piccoli gruppi per formulare prompt funzionali in modo da ottenere risposte in linea con le loro aspettative. L'obiettivo è quello di far capire loro che, per ottenere delle risposte adeguate all'AI è necessario l'intervento dell'uomo che deve elaborare prompt mirati e chiari. Basandosi sulle risposte fornite dall'assistente cibernetico e collaborando tra loro, gli studenti pianificano interventi funzionali alla risoluzione del problema. Gli studenti vengono quindi suddivisi in “squadre di esperti” incaricate di trovare soluzioni alla crisi idrica. All'interno di ciascuna squadra vengono definiti dei ruoli specifici per ciascun componente, come ad esempio: scrittore di prompt (formula i prompt da inviare al chatbot), analista (interpreta e sintetizza le risposte fornite dal chatbot), responsabile delle soluzioni (propone come applicare le risposte ricevute al contesto narrativo), presentatore (espone alle altre squadre le soluzioni individuate). Il docente a questo punto invita a interagire con il chatbot,

formulando prompt funzionali per individuare informazioni e soluzioni legate alla crisi idrica e successivamente a riflettere sulle risposte più rilevanti e discuterle come applicarle al problema proposto. Agli studenti viene richiesto di condividere il link alla conversazione con la chatbot, preparando una breve relazione scritta, in cui indicare i prompt utilizzati, le risposte ricevute, le soluzioni ideate e le soluzioni adottate, ovvero gli interventi funzionali alla risoluzione del problema, presentando le azioni da intraprendere a breve, medio, lungo termine, individuate grazie al supporto dell'IA.

2.4 Fase 4: Debriefing

In questa fase gli studenti rivedono le loro domande e strategie, con il supporto del docente e vengono incoraggiati a riflettere sull'efficacia e sulla coerenza delle risposte ottenute dalla chatbot in relazione ai prompt utilizzati, apportando eventuali miglioramenti. Ad esempio, gli studenti sono invitati a rispondere alle seguenti domande:

1. I prompt hanno guidato l'IA verso output mirati o troppo generici? Come si può affinare il focus delle richieste?
2. Gli output sono risultati chiari, accurati e utilizzabili? Ci sono state informazioni errate, incomplete o difficili da comprendere?
3. Quanto sono stati affidabili gli output? Sono state incluse fonti, dati o riferimenti chiari?
4. I risultati ottenuti rispondono pienamente alla domanda posta nel prompt? Ci sono aspetti non considerati o risultati imprevisti?
5. Quali competenze abbiamo messo in gioco durante l'attività? Quali dovremmo sviluppare ulteriormente?
6. Se potessimo ripetere questa attività, cosa faremmo diversamente e perché?

2.5 Fase 5: Condivisione e feedback tra pari

In quest'ultimo momento le soluzioni di ciascun gruppo vengono inserite all'interno di una bacheca digitale e la classe vota la soluzione migliore abbinato al prompt più efficace. Sarà il presentatore di ciascuna squadra a esporre il lavoro svolto ovvero: le domande formulate (prompt), le risposte della chatbot e le soluzioni ideate per affrontare la crisi idrica. La classe, guidata dal docente, discute e riflette sui vari approcci utilizzati e su come l'IA possa supportare il processo decisionale, esaminando domande, risposte e soluzioni.

3. Sperimentazione classe 2°E del Liceo Classico-Scientifico “Pellico-Peano”

Una delle classi in cui è stata sperimentata l'attività è la classe 2°E del Liceo Classico-Scientifico “Pellico-Peano” di Cuneo, classe di Liceo scientifico opzione Scienze applicate formata da 24 studenti, con prevalenza di ragazzi. Si tratta di una classe mediamente vivace e interessata, non sempre precisa o attenta nell'esecuzione soprattutto di compiti ripetitivi, ma solitamente molto interessata a sperimentare attività nuove o non convenzionali. Il progetto è stato presentato come una opportunità di venire a contatto con tematiche legate all'intelligenza artificiale: obiettivamente è un argomento che gli studenti apprezzano o a cui sono interessati, anche se difficilmente ammettono di fare uso di strumenti di intelligenza artificiale generativa a livello scolastico. Gli alunni avevano già una minima formazione sull'intelligenza artificiale,

grazie ad un altro progetto a cui hanno partecipato nelle settimane precedenti, leggermente più mirato alla sperimentazione anche informatica di strumenti di questo tipo.

La prima attività proposta come icebreaker ha richiesto agli studenti, individualmente, di provare a risolvere la seguente questione: “sei un astronauta in missione nello spazio e nel tuo modulo spaziale c’è un problema con il sistema dell’ossigeno, che sta per terminare. Come potresti risolvere il problema?”. La richiesta ed il vincolo, in questo primo approccio, erano di farsi aiutare da un sistema di intelligenza artificiale generativa, con il vincolo di poter fare esclusivamente una domanda. Al termine dell’attività gli studenti hanno dovuto incollare su un padlet le loro domande e risposte ottenute, per permettere alcune osservazioni successive. Gli studenti sono stati lasciati liberi di decidere se utilizzare Chat-GPT o Perplexity o Magic-School ed hanno scelto in maggioranza Chat-GPT, alcuni Perplexity. In questa prima fase gli studenti, mediamente, non hanno percepito come grossa limitazione quella di non poter affinare il prompt e, pertanto, molti studenti si sono limitati ad incollare la domanda posta dalla docente nei motori.

Come si vede nell’immagine allegata (Figura 2) solo alcuni studenti hanno riformulato il problema, con alcune indicazioni utili.

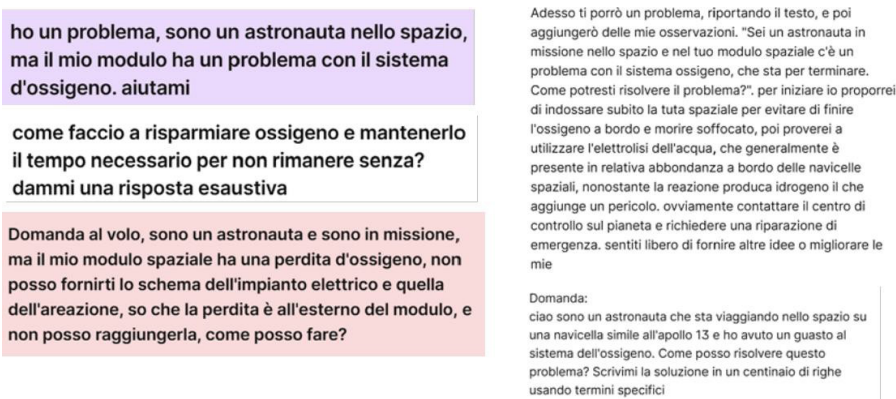


Figura 2: Esempi di prompt studenti parzialmente originali

Le risposte ottenute sono state abbastanza simili: sono stati dati consigli su come risparmiare il più possibile l’ossigeno, sul mettersi in comunicazione col centro a terra, in alcuni casi è stato consigliato di produrre aria con sistemi di elettrolisi.

Dopo questo primo step la docente ha proposto una breve analisi dei risultati e alcune “regole” per generare prompt efficaci: questa fase è stata la meno partecipata e seguita dagli studenti, nonostante un tentativo di renderlo più interessante, proponendo anche tre regole per generare prompt efficaci generate da Chat-GPT.

Nonostante queste difficoltà, gli alunni hanno, in generale, compreso l’importanza di creare prompt efficaci ed hanno anche iniziato a comprendere l’utilità di affinare le richieste via via.

La fase principale dell’attività proposta è stata l’attività a gruppi; il problema è stato proposto con questa formulazione: “Siete in un futuro prossimo. Il vostro mondo sta affrontando una crisi idrica senza precedenti: le risorse d’acqua dolce sono sempre più scarse e la sopravvivenza della comunità dipende da decisioni rapide e strategiche. Come possiamo preservare questa risorsa fondamentale? Le soluzioni che individuerete saranno cruciali!”

In questa fase dell'attività la sollecitazione è stata di ricorrere all'aiuto dell'intelligenza artificiale, affinando le richieste. Dato il poco tempo disponibile, circa 40 minuti, alcuni gruppi si sono limitati a proporre come prompt il testo assegnato e poi aggiungere qualche richiesta, ma altri gruppi hanno ipotizzato le dimensioni di una città come quella in cui si trova la nostra scuola ed hanno anche fatto generare il testo di un ipotetico discorso pronunciato dal Sindaco per sollecitare i cittadini a partecipare attivamente alle attività messe in atto per salvaguardare l'ambiente.

Ciascun gruppo doveva, poi, preparare una presentazione con la risposta più valida ottenuta o con un mix di richieste e risposte e presentarlo al resto della classe, in un tempo molto limitato di 3 minuti, come tipico degli hackathon. Per rendere più accattivante l'attività, al termine ciascun gruppo doveva votare la soluzione che riteneva migliore, ad esclusione della propria.

Durante le presentazioni è emerso da un lato che gli studenti hanno compreso la necessità o opportunità di affinare i prompt, dall'altro che si sono concentrati sulle proposte fatte da Chat-GPT e Perplexity per risparmiare acqua: si è creata come effetto "secondario" una ricaduta di educazione civica su possibili azioni che riducano il consumo di acqua.

Dai commenti raccolti in forma anonima al termine dell'attività è emerso che tutti gli studenti sono stati molto interessati e stupiti; hanno compreso, almeno in parte, che per poter utilizzare bene l'intelligenza artificiale si deve essere consapevoli il più possibile di come funzioni, come presentare le richieste e come affinarle. Dai commenti è emersa una certa consapevolezza che l'intelligenza artificiale non possa sostituirsi all'essere umano nel risolvere il problema ma possa fornirgli molti spunti utili. Molti hanno sottolineato come abbiano ottenuto in pochissimi secondi una raccolta di possibili soluzioni, alcune delle quali molto comuni o semplici ma che non sempre vengono in mente. Indubbiamente gli studenti hanno riflettuto di più anche sulle tematiche attuali del risparmio idrico con la necessità di dover porre domande fatte bene all'intelligenza artificiale che se fosse stata proposta una analoga attività di educazione civica o di scienze.

4. Sperimentazione 1°D AFM dell'Istituto Superiore Arimondi-Eula di Savigliano

Una delle classi in cui è stata sperimentata l'attività descritta è la classe 1°D AFM dell'Istituto Superiore Arimondi-Eula di Savigliano: si tratta di una classe per Tecnici di Amministrazione, Finanza e Marketing formata da 20 studenti con un numero equilibrato di ragazzi e ragazze. L'attività è stata presentata nel mese di marzo 2025 e la classe nei primi mesi di lezione aveva già evidenziato alcuni problemi disciplinari e di attenzione. Per questo motivo l'attività in accordo con i docenti di classe non è stata presentata nella forma molto narrativa descritta al punto 2, ma si sono inserite parti più operative.

Le riflessioni, indicazioni e osservazioni fatte dal docente sono state integrate anche dal supporto di alcuni video a corredo in quanto fin da subito si è osservato come la classe fosse maggiormente "catturata" dalle immagini audio-video. Si è spiegato cosa si intendeva per prompting e sono state date alcune regole di base per scrivere un buon prompt.

In particolare, è stata aggiunta una fase di prompting per la generazione delle immagini. Rimanendo sempre sul tema del pianeta Terra è stato chiesto agli studenti di generare un'immagine del pianeta Terra tra 100 anni utilizzando la piattaforma di Ai Artguru. Tale piattaforma ha dato dei problemi: anche se garantisce la generazione di 5 immagini gratuite, proponendo

l'attività da un laboratorio con stesso indirizzo IP ha bloccato l'accesso alla maggior parte dei PC rendendo necessario procedere l'attività con gli smartphone personali. Agli studenti sono stati fatti degli esempi di Ai differenti e prompt più o meno efficaci.

Ad esempio, un prompt mediocre: “Genera un’immagine del pianeta Terra tra 100 anni, visto dalla Luna e illuminato dal Sole. Rendi visibile lo stato italiano e rappresenta un traffico di satelliti e astronavi spaziali” che ha dato risultati in Figura 3 e un prompt migliore “Genera un’immagine del pianeta Terra tra 100 anni, visto dalla Luna, illuminato dal Sole. L’immagine deve comunicare speranza nel futuro, dove la tecnologia permetterà all’uomo di vivere meglio. Lo stile deve essere del futuro. I colori devono essere caldi, anche se nello spazio. Rendi visibile lo stato italiano e rappresenta un traffico di satelliti e astronavi spaziali, la nuova mobilità del futuro” che ha dato i risultati in Figura 4.

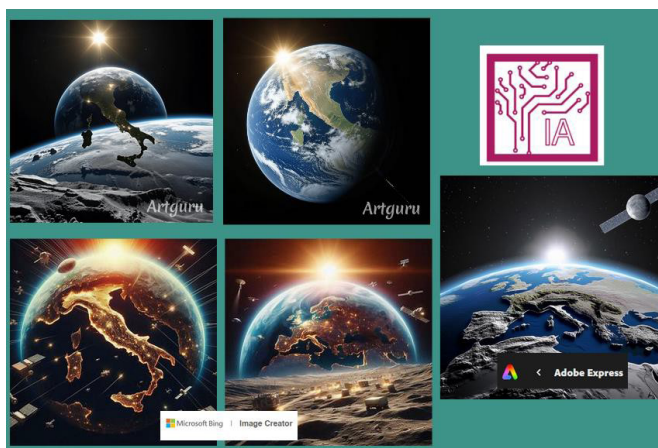


Figura 3: Prompt esempio del docente: Terra tra 100 anni con prompt mediocre

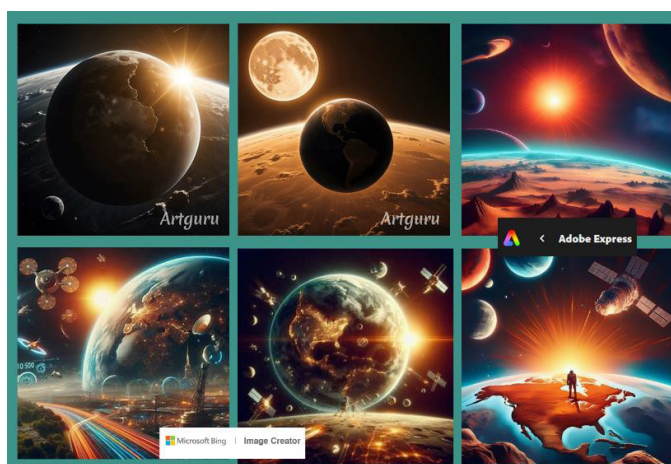


Figura 4: Prompt esempio del docente: Terra tra 100 anni con prompt più dettagliato

Lavorando più sulle immagini che non sulle parole gli studenti sono stati aiutati ad avere immediatamente un feedback sulla scrittura del proprio prompt. I lavori prodotti sono andati da prompt che hanno riscritto pedestremente quello con le indicazioni del docente, fino a prompt più complessi che chiedevano anche di produrre un'immagine che "comunicasse" delle emozioni. Si riportano alcuni prompt elementari che hanno generato immagini semplici come ad esempio: "Genera un'immagine della terra tra 100 anni con colori freddi e che provoca tristezza e angoscia" e "La pianeta fra 100 anni" le cui immagini sono riportate in Figura 5. Mentre alcuni prompt più elaborati come: "Genera un immagine della terra tra 100 anni, ormai invasa da zombie e alieni e delle astronavi nel cielo che indicano gli esseri umani che scappano verso un altro pianeta. Quest'immagine deve provocare angoscia e paura a chi la guarda." e "il pianeta terra fra 100 anni. con colori freddi. con delle persone. con mari e terre. con città e abitazioni. con macchina volanti e astronavi. persone intorno al pianeta" le cui immagini sono riportate in Figura 6.



Figura 5: Prompt di studenti semplici



Figura 6: Prompt di studenti elaborati

Questa parte inserita nella sperimentazione è stata sicuramente più efficace rispetto alle fasi successive. Lavorare sulle “parole” è stato più difficile, le risposte legate al problema “acqua” sono state difficili da valutare da parte degli studenti, anche perché nella maggior parte dei casi molti simili e generali.

5. Sperimentazione 1°E del Liceo Scientifico G. B. Bodoni di Saluzzo

La classe del Liceo Classico e Scientifico “G.B. Bodoni” di Saluzzo in cui si è svolta la sperimentazione è la 1D dell’opzione Scienze Applicate del Liceo Scientifico. La classe era composta da 24 alunni, con una leggera prevalenza della componente maschile. La classe è piuttosto variegata riguardo l’interesse e la partecipazione, con alcuni alunni particolarmente entusiasti e interessati e altri poco attivi e partecipi. Il progetto è stato un’opportunità per conoscere meglio uno strumento che gli studenti utilizzano in modo poco consapevole (anche se diversi allievi hanno dichiarato di aver solo sentito nominare l’IA ma di non averla mai utilizzata). Non avendo mai svolto alcuna formazione al riguardo, la classe ha, comunque, apprezzato questa occasione.

Nella prima attività è stato richiesto agli studenti di provare a risolvere, individualmente ma con il supporto di un sistema di Intelligenza Artificiale generativa, la seguente questione: “Sei un astronauta in missione nello spazio e nel tuo modulo spaziale c’è un problema con il sistema dell’ossigeno, che sta per terminare. Come potresti risolvere il problema?”. Il vincolo è stato quello di poter fare una sola domanda. Al termine dell’attività gli studenti hanno incollato su un padlet opportunamente predisposto dal docente i loro prompt e le risposte ottenute. Agli studenti è stato richiesto di utilizzare Perplexity.

In questa prima fase praticamente tutti gli studenti hanno presentato come prompt il quesito posto dal docente, con minime variazioni. Per esempio: *“Sono un astronauta, adesso la mia navicella sta perdendo ossigeno, COSA DEVO FARE? Scrivimelo in un elenco chiaro, facilmente leggibile e con un linguaggio semplice e soprattutto funzionale.”* oppure *“Se fossi un astronauta in una navicella che sta perdendo ossigeno e dovessi ritornare sulla terra prima che l’ossigeno finisca, cosa mi consiglieresti di fare?”*

Conseguenza di richieste sostanzialmente identiche sono state risposte molto simili che prevedono l’uso di tute spaziali o bombole di riserva per sfruttare tutto l’ossigeno potenzialmente disponibile o il ricorso all’elettrolisi per produrre ossigeno dall’acqua presente sull’astronave.

Al termine di questa prima fase, è stata fatta una breve analisi dei risultati e sono state fornite alcune buone prassi per generare prompt efficaci: questa fase è stata quella meno coinvolgente per gli allievi che, pur comprendendo il problema, non hanno potuto fare a meno di commentare tra loro i prompt forniti e le risposte ricevute.

Gli alunni hanno, comunque, generalmente compreso l’importanza di creare prompt efficaci e, avendo avuto la possibilità di inserire un solo prompt, l’utilità di utilizzare prompt multipli, migliorando le richieste procedendo per affinamento successivi.

La seconda dell’attività proposta è stata, invece, svolta dopo aver diviso la classe in sei gruppi da quattro studenti.

È stato richiesto alla classe di sviluppare, con il supporto dell’IA, soluzioni al seguente problema: *“Siete in un futuro prossimo. Il vostro mondo sta affrontando una crisi idrica senza precedenti: le risorse d’acqua dolce sono sempre più scarse e la sopravvivenza della comunità dipende da decisioni rapide e strategiche. Come possiamo preservare questa risorsa fondamentale? Le soluzioni che individuerete saranno cruciali!”*

In questa fase non c'erano vincoli alle richieste e gli allievi sono stati, anzi sollecitati ad affinare le richieste. Il tempo disponibile era limitato a circa 40 minuti. Ogni gruppo, alla fine, ha nominato un rappresentante che ha presentato all'intera classe, in circa 3 minuti, la proposta sviluppata. Per rendere più l'attività più stimolante, al termine ciascun gruppo doveva votare la soluzione che riteneva migliore, ad esclusione della propria. Durante le presentazioni è emerso che gli studenti, dopo l'esperienza della prima fase, hanno compreso la necessità di procedere operando per affinamenti successivi dei prompt, ma, tranne il caso di un gruppo, non hanno contestualizzato e ben definito la situazione, ottenendo, in generale, risposte abbastanza generiche riguardo la protezione, il risparmio e il riciclo delle risorse idriche.

Nella discussione finale tutti gli studenti hanno manifestato apprezzamento per le attività svolte, che hanno consentito loro di acquisire maggiore consapevolezza riguardo l'utilizzo e il funzionamento degli strumenti di Intelligenza Artificiale. Si segnala che, nel corso dei successivi colloqui con le famiglie, diversi genitori hanno riferito della soddisfazione dei figli manifestando, a loro volta, apprezzamento per la sperimentazione svolta, nella consapevolezza dell'importanza di sviluppare competenze che permettano di utilizzare correttamente e consapevolmente gli strumenti di IA, incoraggiando il docente a proseguire con l'approfondimento su tali argomenti.

6. Sperimentazione 1°D dell'Istituto Superiore Arimondi- Eula di Savigliano

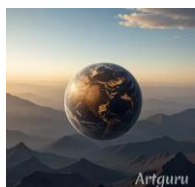
L'attività è stata proposta nel mese di marzo alla classe 1D del Liceo Scientifico – opzione Scienze Applicate, composta da 26 studenti (12 maschi e 14 femmine). Si tratta di un gruppo molto vivace, sia dal punto di vista didattico che disciplinare, ma al contempo curioso, propositivo e particolarmente reattivo quando vengono proposte tematiche di loro interesse.

Data la predisposizione del gruppo classe al “saper fare”, si è scelto un approccio didattico che bilanciassero la componente narrativa con attività operative e concrete. L'intento era quello di introdurre i concetti base del prompting, ovvero la formulazione di richieste efficaci per strumenti di intelligenza artificiale generativa. Per catturare l'attenzione degli studenti e coinvolgerli attivamente, la docente ha fatto uso di supporti audiovisivi e ha fornito esempi pratici.

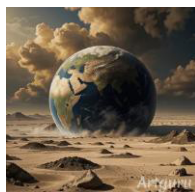
Utilizzando il generatore di immagini AI Artguru, la classe è stata chiamata a immaginare e visualizzare il futuro del nostro pianeta attraverso la scrittura di un prompt. Nonostante non conoscessero in precedenza la piattaforma, gli studenti si sono rapidamente orientati e hanno superato autonomamente il limite tecnico del laboratorio (relativo all'uso del medesimo indirizzo IP) grazie all'uso degli smartphone personali.



Genera un'immagine della terra
fra cento anni resa inabitabile dal
surriscaldamento globale



genera un immagine del pianeta
tra 100 anni. Ci saranno ancora
gli uomini? le intelligenze
artificiali comanderanno il
mondo e avranno pieni diritti?



genera la terra tra 100 anni distrutta
dalle guerre, tutto deserto senza
niente



crea un'immagine della terra tra 100
anni con tante industrie e palazzi, tutta
moderna ma con un cielo pulito e
molti spazi verdi

Figura 7: Immagini e Prompt degli studenti

I risultati ottenuti – immagini generate da prompt talvolta fantasiosi, talvolta ingenui – hanno rivelato un elemento comune: la preoccupazione per il futuro del pianeta, un dato che, considerata l'età (14-15 anni), meriterebbe un approfondimento educativo e psicologico.

Per stimolare il pensiero critico e l'uso strategico dell'IA, è stato proposto un secondo scenario: “Sei un astronauta in missione nello spazio e nel tuo modulo spaziale c'è un problema con il sistema dell'ossigeno che sta per terminare.”

Gli studenti sono stati invitati a formulare domande e strategie risolutive utilizzando Perplexity, una piattaforma di AI generativa da loro mai utilizzata prima. Successivamente, hanno potuto confrontare i risultati con quelli ottenuti usando altre IA (ChatGPT, Magic School), osservando similitudini e differenze nei contenuti prodotti.

sono un astronauta in missione nello spazio. sto andando verso marte, con due miei compagni (un uomo e una donna, entrambi trentenni, come me). siamo in viaggio da un'ora e c'è un problema con il sistema dell'ossigeno, in più non abbiamo altre bombole. La bombola è quasi vuota e manca l'aria. non riusciamo a parlare e non riusciamo a contattare la centrale della nasa, abbiamo perso il collegamento. Mi aspetto un aiuto prezioso per riuscire a risolvere il problema,

Sono un'astronauta in missione nello spazio e nel mio modulo spaziale c'è un problema con il sistema dell'ossigeno, che sta per terminare. Come posso ripararlo definitivamente? L'ha danneggiato un urto con un meteorite, non ho più collegamenti con la terra ma ho comunque molti attrezzi a disposizione. Cerca le informazioni sul sito della nasa o su altri siti affidabili.

Figura 8: Esempi di Prompt degli studenti

Le risposte fornite dall'IA, sebbene pertinenti e dettagliate, sono risultate piuttosto lunghe. Questo ha comportato una certa perdita di attenzione da parte degli studenti, che tendevano a non portare a termine la lettura dei testi generati. La difficoltà di mantenere la concentrazione su risposte articolate ha fatto emergere l'importanza di educare alla lettura integrale dei contenuti e allo sviluppo della capacità critica, essenziale non solo nell'interazione con l'IA, ma più in generale nella formazione personale e scolastica. L'attività ha evidenziato l'utilità e l'interesse che gli strumenti di IA possono suscitare anche in contesti scolastici. Tuttavia, si conferma fondamentale accompagnare l'uso delle tecnologie con un percorso formativo che insegni a non fermarsi alla superficie delle risposte, ma ad analizzarle in profondità. Solo così gli studenti potranno imparare a valutare autonomamente le soluzioni proposte, sviluppando competenze trasversali e capacità decisionali fondamentali per il loro futuro.

7. Conclusioni

La valutazione complessiva della sperimentazione del progetto è indubbiamente positiva, per una serie di ragioni e di riscontri, pur prestandosi ad alcuni aggiustamenti e migliorie, visto che si è trattato di una prima sperimentazione. Gli aspetti sicuramente positivi sono stati di intercettare una tematica, quale è oggi l'intelligenza artificiale, che interessa molto gli alunni, sui quali in genere c'è poca trattazione a scuola, rischiando di delegare la formazione e l'uso dell'IA ad un uso solitario da parte degli studenti. Il fatto di aver proposto attività pratiche ed operative ha contribuito a mantenere viva l'attenzione anche di studenti che tendono a distrarsi spesso. È nata una buona sinergia anche tra problema “simulato” e analoga tematica di attualità ed educazione civica, visto l'esempio della scarsità di acqua utilizzato nella parte più lunga dell'attività.

Dalla sperimentazione che ha fatto uso di immagini è emersa la conferma che gli studenti di oggi sono più recettivi sulle immagini che sulle parole e che, pertanto, occorre investire

sulla didattica anche in quest’ambito. Sicuramente sarebbe opportuno inserire la parte dedicata alla generazione di immagini nell’attività e riservarle un posto non marginale. Si è evidenziato invece il limite della problematica proposta da far risolvere all’AI. Se il problema è troppo complesso, come nel caso del problema idrico, le risposte dell’AI sono generiche e gli studenti devono essere accompagnati nell’interpretare che non sono risposte attendibili.

Il progetto si presta sicuramente ad essere riproposto anche ad altre classi, non solo del biennio della scuola secondaria di secondo grado. Si può proporre a studenti più giovani, magari puntando più sulla generazione di immagini che di testi anche lunghi, ed a studenti del triennio, insistendo su prompt efficaci e consegne leggermente più complesse. Occorre trovare un modo ancora più accattivante per far sperimentare come i prompt possano essere efficaci o non efficaci, forse facendo eseguire più prove e provando a far emergere dagli studenti che cosa ha funzionato e che cosa è stato poco efficace.

Storie Sonore di Donne nella Scienza: Narrazione, Musica, Coding e Intelligenza Artificiale

Ersilia Pagano¹

¹ IIS Nitti Portici

ersilia.pagano@gmail.com

Abstract

Il progetto interdisciplinare “*Storie Sonore di Donne nella Scienza*” valorizza il contributo femminile in ambito scientifico e unisce educazione civica, musica, informatica e scienze. Gli studenti hanno sviluppato una web app interattiva con Pictoblox e l'estensione ChatGPT, dando voce a cinque scienziate (Rita Levi Montalcini, Ipazia, Clementina Sasso, Anna Grassellino, Filomena Nitti) attraverso narrazioni multimediali dinamiche e personalizzate.

L'attività si configura come un'esperienza autentica di didattica dell'informatica nel primo biennio, attraverso l'utilizzo di coding creativo, intelligenza artificiale e progettazione interattiva. Il percorso ha promosso la partecipazione attiva delle studentesse, favorendo l'inclusione e la valorizzazione della diversità nei contesti STEM, e sviluppando soft skill come creatività, pensiero critico e collaborazione. Grazie all'integrazione tra musica, narrazione e tecnologie computazionali, il progetto ha stimolato lo sviluppo di competenze digitali, progettuali e di cittadinanza, contribuendo a formare una visione critica e consapevole dell'uso della tecnologia.

1. Introduzione

Il progetto nasce con l'obiettivo di **stimolare la riflessione sulla parità di genere**, mettendo in luce il ruolo significativo, e troppo spesso invisibile, delle donne nel campo della scienza, e di promuovere **l'apprendimento interdisciplinare attraverso l'uso di tecnologie digitali**.

L'attività ha coinvolto **due classi del biennio della scuola secondaria di secondo grado, un Istituto Tecnico** (1B Turistico Quadriennale e 2B Informatica Quadriennale), impegnate in un percorso cooperativo che ha previsto l'adozione della metodologia del tutoraggio tra pari, con il supporto della classe 3B Informatica Quadriennale.

2. Obiettivi formativi e approccio interdisciplinare

L'attività ha perseguito obiettivi trasversali:

- sviluppo del pensiero computazionale e creativo;
- promozione della cittadinanza attiva e digitale;
- valorizzazione delle competenze musicali, narrative e tecniche;
- conoscenza del contributo femminile nella scienza.

Le discipline coinvolte sono:

- **Educazione civica:** cittadinanza digitale, diritti delle donne, parità di genere.
- **Italiano:** scrittura creativa e sceneggiature.
- **Informatica (e Tecnologia musicale):** programmazione a blocchi, IA, ambienti di sviluppo visuale, produzione e montaggio audio.
- **Storia (e Filosofia):** approfondimenti biografici e riflessioni etico-scientifiche.
- **Scienze (Chimica/Fisica):** studio delle scoperte scientifiche delle protagoniste, analisi del metodo scientifico, riflessioni sui contributi femminili alla ricerca.

3. Aspetti disciplinari dell'informatica

L'attività informatica ha previsto:

- progettazione e programmazione in Pictoblox;
- uso dell'estensione ChatGPT per interazioni dinamiche;
- creazione di sprite intelligenti;
- integrazione audio-narrazione-codice.

4. Metodologia didattica e fasi

La didattica si è basata su **co-progettazione**, **didattica laboratoriale** e **peer learning**, promuovendo la collaborazione e la valorizzazione dei talenti individuali. Gli studenti sono stati **suddivisi in gruppi eterogenei di 4–5 membri, mischiando le due classi** coinvolte, al fine di favorire l'interscambio di competenze, il tutoraggio tra pari e una più ampia prospettiva interdisciplinare. Le fasi operative sono state:

1. **Brainstorming** e scelta della scienziata da narrare.
2. **Scrittura creativa** e costruzione dei dialoghi.
3. **Produzione sonora** con Band on the Cloud.
4. **Programmazione** interattiva in Pictoblox.
5. **Test** del progetto finale.
6. **Presentazione** pubblica e condivisione digitale.

Per facilitare la realizzazione del progetto, ho organizzato le attività in laboratorio in compresenza, pianificando ore comuni tra le classi coinvolte. I contenuti necessari per avviare lo sviluppo (uso di Pictoblox, introduzione all'IA e alla narrazione interattiva, elementi di progettazione audio) sono stati forniti in forma laboratoriale, con attività introduttive guidate e schede operative condivise, adatte a un pubblico eterogeneo. Gli studenti non avevano bisogno di prerequisiti formali: è stato sufficiente un livello base di dimestichezza con ambienti digitali. La classe terza ha potuto svolgere un ruolo di tutoraggio grazie alla maggiore familiarità con l'ambiente di sviluppo e alle competenze informatiche già consolidate nel percorso curricolare.

5. Strumenti e ambienti

- **Pictoblox + estensione ChatGPT:** ambiente di **programmazione a blocchi**, utilizzato per realizzare la struttura **narrativa interattiva** del progetto e integrare risposte dinamiche basate sull'IA.
- **Band on the Cloud** (piattaforma INDIRE): per l'audio design.
- **Google Sites, Padlet, Classroom:** per collaborazione e pubblicazione.
- **Canva:** per le presentazioni finali.

6. Valutazione e risultati

La valutazione è stata **multidimensionale**, considerando competenze:

- digitali e narrative,
- collaborative e trasversali,
- informatiche, in linea con gli obiettivi previsti per il biennio della secondaria di secondo grado.

Strumenti di valutazione utilizzati:

- **Griglia delle competenze** (digitali, narrative, informatiche);
- **Portfolio digitale degli studenti** (con riflessioni e documentazione del processo);
- **Prova autentica:** presentazione pubblica del progetto finale interattivo.

Nella **componente informatica**, si è tenuto conto di indicatori coerenti con il documento CINI, tra cui:

- progettazione e realizzazione di algoritmi narrativi interattivi (O-S-A-2, O-S-P-6),
- uso consapevole e creativo della programmazione a blocchi con IA (O-S-R-1, T-S-6, T-S-16),
- integrazione di contenuti digitali multimediali (O-S-D-2, O-S-R-2),
- consapevolezza degli impatti etici e sociali dell'IA (T-S-14).

Livelli raggiunti dagli studenti:

- Livello base: 6 studenti – comprensione iniziale degli strumenti, partecipazione guidata.
- Livello intermedio: 9 studenti – autonomia nella creazione di contenuti sonori e narrativi con elementi programmati.
- Livello avanzato: 7 studenti – padronanza completa di progettazione narrativa, audio e programmazione interattiva con IA.

7. Contributo ai temi del convegno ITADINFO

Il progetto si inserisce pienamente nei temi del convegno ITADINFO, offrendo **un'esperienza didattica centrata sull'insegnamento dell'Informatica come disciplina scientifica**, con particolare attenzione a:

- **Sviluppo di curricula sperimentali** nel primo biennio della scuola secondaria di secondo grado, integrando programmazione visuale, AI generativa e progettazione interattiva.

- **Metodologie didattiche innovative**, basate su **coding narrativo, didattica per progetti e intelligenza artificiale**, che permettono di costruire conoscenze attraverso il fare e il pensare computazionale.
- **Partecipazione femminile**: il progetto è stato realizzato da un gruppo misto, con una **forte presenza di studentesse**, valorizzando figure di scienziate e promuovendo l'inclusione di genere nel contesto STEM.
- **Formazione delle competenze digitali e trasversali**: il percorso ha sviluppato competenze chiave come il pensiero computazionale, la collaborazione, la creatività digitale e la riflessione etica sul ruolo della tecnologia.

L'interazione tra storytelling, musica e tecnologie computazionali si configura come un esempio efficace di **didattica trasversale e inclusiva**, capace di **avvicinare all'Informatica anche studenti con attitudini non tecniche**, offrendo uno **spazio di espressione personale e di cittadinanza digitale attiva**.

8. Collegamenti ai Traguardi e Obiettivi della Proposta CINI – Biennio Scuola Secondaria di Secondo Grado

Il progetto si inserisce in piena coerenza con la *Proposta di Indicazioni Nazionali per l'insegnamento dell'Informatica nella Scuola* del CINI, contribuendo al raggiungimento di numerosi **traguardi di competenza e obiettivi specifici**, distribuiti nei diversi ambiti disciplinari.

Traguardi per lo sviluppo delle competenze raggiunti:

- **T-S-1 / T-S-2 / T-S-3 / T-S-4**. Gli studenti hanno appreso la necessità di definire algoritmi chiari e generali per simulare conversazioni e comportamenti interattivi delle scienziate attraverso sprite programmati.
- **T-S-6 / T-S-12**. Il progetto ha previsto la modellazione di un sistema interattivo (la web app narrativa) che simula un'interazione personalizzata tra utente e contenuti educativi, incentrata sull'esperienza dell'utente e sulla fruibilità del prodotto finito.
- **T-S-14 / T-S-13**. Ampio spazio è stato dato alla riflessione critica sull'uso dell'intelligenza artificiale, all'impatto sociale dell'informatica e all'etica della rappresentazione e divulgazione scientifica, specie in ottica di genere.
- **T-S-15 / T-S-16**. Gli studenti hanno saputo combinare strumenti e tecnologie digitali (Pictoblox con estensione ChatGPT, Band on the Cloud, Canva, Padlet, Google Sites) per sviluppare un prodotto multimediale, personalizzato e creativo.

Obiettivi specifici per ambito:

- **Ambito: Algoritmi O-S-A-2**. Gli studenti hanno valutato e implementato diverse logiche di interazione tra sprite e utente, simulando percorsi di conversazione differenziati.
 - **O-S-A-4**. La progettazione degli script ha tenuto conto delle possibilità e dei limiti dell'ambiente di sviluppo Pictoblox e dell'estensione ChatGPT.
- **Ambito: Programmazione**.
 - **O-S-P-1 / O-S-P-2 / O-S-P-3 / O-S-P-4 / O-S-P-5 / O-S-P-6**. Durante lo sviluppo, gli studenti hanno analizzato il ruolo funzionale dei diversi blocchi di codice, predetto il

comportamento dei programmi e strutturato funzioni modulari, ad esempio per le risposte dinamiche delle scienziate.

- **Ambito: Dati e Informazione.**

- **O-S-D-2.** Le scelte relative alla strutturazione delle informazioni (testi, audio, metadati) nelle diverse sezioni del progetto riflettono consapevolezza nella gestione efficace dei dati in un contesto narrativo e interattivo.

- **Ambito: Creatività Digitale**

- **O-S-R-1.** Il progetto ha valorizzato l'espressività computazionale: ogni gruppo ha realizzato una narrazione audiovisiva e sonora originale, basata su script personalizzati.
- **O-S-R-2.** L'integrazione tra l'ambiente di sviluppo (Pictoblox), strumenti per la produzione audio (Band on the Cloud) e piattaforme di pubblicazione (Google Sites, Canva, Padlet) ha permesso la combinazione funzionale di servizi digitali per il raggiungimento degli obiettivi espressivi e comunicativi.

- **Ambito: Consapevolezza Digitale**

- **O-S-N-2 / O-S-N-3.** L'attenzione all'utente finale, al pubblico di riferimento e alla possibilità di riutilizzo e diffusione dei contenuti ha orientato le scelte progettuali e comunicative.
- **O-S-N-4.** Discussioni e riflessioni etiche in aula legate all'uso dell'IA e alla rappresentazione biografica delle scienziate hanno stimolato negli studenti una maggiore consapevolezza delle responsabilità sociali connesse all'utilizzo della tecnologia.

9. Conclusioni

Il progetto ha dimostrato il valore del coding creativo e dell'IA generativa nella narrazione educativa, promuovendo la partecipazione attiva e consapevole degli studenti. L'integrazione tra musica, storia, informatica e cittadinanza ha favorito l'acquisizione di competenze trasversali e la valorizzazione delle figure femminili nella scienza.

Il modello può essere replicato in altri contesti scolastici, anche in collaborazione tra gradi scolastici differenti e può essere facilmente esteso, ad esempio includendo nuove scienziate, anche di rilievo informatico, come Ada Lovelace, così da arricchire il percorso con ulteriori riferimenti storici e scientifici e potenziare il legame con i fondamenti dell'informatica.

Bibliografia

- [1] Pictoblox AI. <https://pictoblox.ai>
- [2] INDIRE – Band on the Cloud. <https://bandonthecloud.indire.it/musicacollaborativa/>
- [3] Proposta CINI. Indicazioni Nazionali per l'Informatica. <http://consorzio-cini.it/gdl-informatica-scuola>
- [4] Canva. <https://www.canva.com>
- [5] Padlet. <https://padlet.com>

Insegnare Informatica con progetti autentici: il monitoraggio delle acque del fiume Aterno

Luca Forlizzi^{2,3}, Maria Lizzi¹, Giovanna Melideo^{2,3}, Giovanna Patrizio¹, Alessia Verticchio¹

¹ Istituto di Istruzione Superiore “Amedeo D’Aosta” L’Aquila

{marializzi, giovannapatrizio, alessiaverticchio}@iisdaostalaquila.it

² Dipartimento di Ingegneria e scienze dell’informazione e matematica, Università degli Studi dell’Aquila

{luca.forlizzi, giovanna.melideo}@univaq.it

³ Laboratorio Nazionale del CINI - Gruppo di Lavoro “Informatica & Scuola”

Abstract

L’articolo descrive il progetto interdisciplinare “Monitoraggio delle acque del fiume Aterno”, svolto nel 2025 presso l’IIS “A. D’Aosta” dell’Aquila, in collaborazione con l’Università degli Studi dell’Aquila e il Club Alpino Italiano, con l’obiettivo di potenziare le competenze tecnico-scientifiche e trasversali degli studenti, promuovendo la consapevolezza ambientale. Gli studenti hanno condotto campionamenti e analisi microbiologiche, per poi realizzare un’applicazione per la gestione e la visualizzazione dei dati, oltre a un sito web1 che documenta il progetto e consente la consultazione di una mappa interattiva per il monitoraggio delle acque. L’approccio didattico basato sul Project-Based Learning, supportato da seminari specialistici, ha coinvolto gli studenti in un progetto reale, favorendo un apprendimento significativo e rendendo l’esperienza coinvolgente.

1. Introduzione

L’insegnamento delle discipline tecniche e scientifiche nelle scuole si limita spesso all’applicazione di conoscenze teoriche a casi di studio didattici, che non sempre riflettono la complessità del mondo reale né le sfide connesse allo sviluppo di sistemi software complessi. Per superare queste limitazioni, l’IIS “D’Aosta” dell’Aquila, tra gennaio e maggio 2025, ha realizzato il progetto “Monitoraggio delle acque del fiume Aterno”, in collaborazione con l’Università degli Studi dell’Aquila (UnivAQ) e il Club Alpino Italiano - sezione dell’Aquila (CAI). L’iniziativa ha mirato a rafforzare conoscenze teoriche e competenze tecnico-scientifiche e trasversali (problem solving, pensiero computazionale, lavoro in team, comunicazione anche in inglese), in linea con gli obiettivi di apprendimento delineati in [1, 3], e a promuovere al contempo la cittadinanza attiva e la consapevolezza ambientale. Collegando discipline scientifiche con l’educazione civica, il progetto è coerente con gli obiettivi dell’Agenda 2030 (Goal 6) e il Digital Education Action Plan della Commissione Europea (2020).

L’esperienza ha coinvolto le classi 4°A e 5°C dell’Istituto Tecnico Tecnologico - Indirizzo “Informatica e Telecomunicazioni” (ITT) e la 4°A del Liceo Scientifico - Opzione Scienze

Applicate (LSA). Gli studenti del Liceo hanno effettuato campionamenti delle acque, in collaborazione con il CAI, e poi condotto analisi microbiologiche dei campioni per valutare l'impatto delle attività antropiche sugli ecosistemi fluviali. A supporto delle attività di analisi e comunicazione dei risultati, è nata una sinergia interdisciplinare con gli studenti dell'ITT che hanno progettato e realizzato un'applicazione in C++ per la gestione e visualizzazione degli esiti delle analisi microbiologiche e un sito web dinamico basato su dati reali. Il sito¹ consente la consultazione di una mappa interattiva per il monitoraggio delle acque e integra funzionalità di autenticazione, inserimento dati, visualizzazione e analisi statistica dei batteri individuati.

Oltre agli aspetti funzionali, gli studenti hanno creato il logo ufficiale e progettato la grafica del sito, realizzando uno strumento digitale completo per la sensibilizzazione ambientale e la valorizzazione del territorio. Infine, hanno presentato i risultati, anche in inglese, agli enti coinvolti e ai visitatori Erasmus, potenziando il valore formativo dell'esperienza. Integrando competenze scientifiche e informatiche, hanno partecipato attivamente al progetto, favorendo una pratica di "computational action", espressione che sintetizza l'idea di utilizzare l'informatica per creare artefatti digitali che abbiano un impatto concreto su individui, comunità e società [8]. Come suggerito da Tissenbaum et al., promuovere la "computational action" insieme al "computational thinking" può rendere l'apprendimento dell'informatica più inclusivo, motivante ed efficace per gli studenti.

Parte integrante del progetto è stata la partecipazione degli studenti a seminari specialistici tenuti da ricercatori universitari, che hanno rappresentato importanti momenti di approfondimento e contestualizzazione delle conoscenze curriculari su argomenti fondamentali come ciclo di vita del software, modellazione UML, e algoritmi, strutture dati e complessità computazionale. L'esperienza formativa si è basata sull'approccio del Project-Based Learning (PjBL) integrato dai seminari volti a fornire scaffolding teorico al momento opportuno. Il PjBL è ampiamente adottato nell'insegnamento dell'informatica per la sua efficacia nel promuovere la motivazione, le competenze pratiche e teoriche degli studenti, soprattutto quando sostenuto da un adeguato scaffolding docente. Questo approccio non solo facilita l'apprendimento di contenuti tecnico-scientifici, ma sviluppa anche competenze trasversali fondamentali per il successo scolastico e professionale (pensiero critico, problem solving, collaborazione), risultando particolarmente efficace nel garantire un apprendimento significativo e duraturo [2, 4, 5, 6, 7]. In contesti interdisciplinari, favorisce l'apprendimento esperienziale e l'integrazione delle conoscenze [4, 6]. L'approccio basato sul PjBL, integrato dai seminari specialistici, il confronto con un problema reale come il monitoraggio della qualità delle acque di un fiume urbano, e l'interdisciplinarietà, hanno caratterizzato il progetto, favorendo un apprendimento significativo e rendendo l'esperienza più coinvolgente.

2. Obiettivi del progetto

L'esperienza ha perseguito due obiettivi generali fondamentali:

- *Potenziamento delle conoscenze teoriche e delle competenze tecnico-scientifiche e trasversali:*
 - lavorare in team e comunicare efficacemente;
 - riconoscere e integrare le esigenze dell'utente come elemento centrale nel processo di sviluppo software;
 - presentare pubblicamente i risultati, anche in lingua inglese;

¹ Url: <http://109.239.246.86/monitoraggioacque/index.php>

- riflettere sulle capacità espressive degli strumenti informatici e sull'impatto dei prodotti digitali su individui, comunità e società.
- *Sviluppo di competenze civiche in ambito ambientale*, con particolare attenzione alla tutela delle risorse idriche.

Gli obiettivi specifici che hanno costituito tappe fondamentali anche per il consolidamento delle competenze generali hanno riguardato:

1. l'approfondimento delle conoscenze sull'ingegneria del software, applicate a un caso concreto;
2. l'adozione di un approccio critico all'ottimizzazione del software;
3. la gestione di un progetto complesso, dalla fase di analisi dei requisiti fino alla realizzazione di prodotti funzionanti.

3. Organizzazione del progetto

Il progetto ha seguito due linee parallele e in costante dialogo: i seminari specialistici e la realizzazione concreta del progetto, avvenuta in diverse fasi operative²:

1. definizione dei requisiti (gennaio 2025);
2. progettazione e sviluppo dell'applicazione in C++ e del sito web (febbraio-maggio 2025);
3. progettazione e realizzazione del logo (aprile 2025);
4. presentazione del progetto finale, anche in lingua inglese (maggio 2025).

L'approccio integrato rispecchia quanto evidenziato dalla letteratura scientifica, che sottolinea come il PjBL favorisca l'apprendimento attraverso la realizzazione di progetti concreti, supportati da un adeguato scaffolding. In particolare, l'inserimento di seminari specialistici si allinea con le raccomandazioni per un bilanciamento efficace tra istruzione frontale e apprendimento indagativo, facilitando la comprensione di concetti complessi e migliorando le competenze tecnico-scientifiche [2].

3.1 Seminari specialistici

I seminari hanno rappresentato momenti chiave di approfondimento, integrando conoscenze teoriche e pratiche per supportare sia la progettazione sia l'implementazione del progetto. Hanno affrontato due temi poco approfonditi nei programmi tradizionali:

1. *Ciclo di vita del software e modellazione UML*: per introdurre i principi fondamentali della modellazione dei sistemi software, essenziali per rappresentare in modo chiaro e condiviso requisiti, struttura e comportamento del sistema;
2. *Algoritmi, strutture dati e complessità computazionale*: per stimolare un approccio critico all'analisi e all'ottimizzazione del codice.

Alternando teoria, casi di studio e attività pratiche in sottogruppi, i seminari, erogati nei momenti strategici, hanno consentito agli studenti di rivedere criticamente le scelte progettuali e di sviluppo. In particolare, la parte più interessante per gli studenti è stata quella su "Ciclo di vita del software e modellazione UML", che ha permesso loro di acquisire consapevolezza sull'importanza di applicare le buone pratiche di progettazione spesso percepite come astratte.

2 Per esigenze di sintesi, la trattazione si limita agli aspetti informatici.

3.2 Realizzazione del progetto

L'attività progettuale ha riprodotto il modello cliente-fornitore tipico del contesto lavorativo, assegnando ruoli specifici e gestendo tutte le fasi del ciclo di vita del software. Una classe ha assunto il ruolo di "cliente", definendo requisiti e aspettative, le altre due quello di "fornitori", responsabili della progettazione, sviluppo e consegna delle soluzioni digitali. Tale modalità ha favorito l'acquisizione di competenze pratiche nella gestione delle dinamiche di progetto, nella comunicazione "interprofessionale" e nella collaborazione tra team con ruoli complementari, rispecchiando la prassi lavorativa. Si è proceduto alla suddivisione dei compiti per classe, secondo le rispettive competenze e ruoli assegnati:

- *4°A LSA*: esecuzione di analisi batteriologiche e interpretazione dei dati. La classe ha assunto il ruolo di cliente, definendo le esigenze di progetto;
- *4°A ITT*: realizzazione della parte statica del sito web e sviluppo dell'applicazione in C++
La classe ha assunto il ruolo di fornitore;
- *5°C ITT*: sviluppo delle pagine dinamiche per l'autenticazione, l'inserimento, la visualizzazione e la gestione dei dati, inclusa la mappa interattiva, anch'essa in qualità di fornitore.

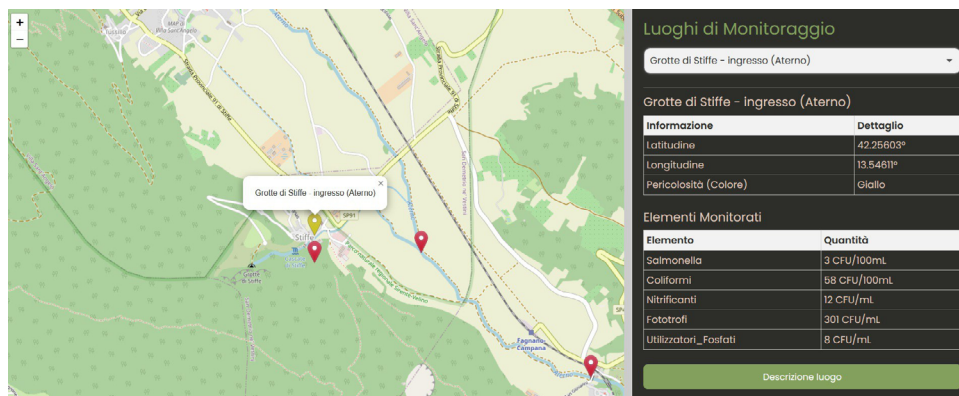


Figura 1: Visualizzazione dei dati microbiologici relativi ad un punto di campionamento.

I clienti e i fornitori si sono incontrati in tre momenti chiave: all'inizio per definire le specifiche di progetto, a metà percorso per verificare la rispondenza delle implementazioni alle esigenze espresse, e al termine per effettuare i test finali e procedere ai necessari raffinamenti. L'individuazione di un referente per classe ha facilitato le comunicazioni tra le parti, soprattutto nella fase cruciale dell'analisi dei requisiti. All'interno di ogni classe, sono stati costituiti sottogruppi con compiti specifici, organizzati in base al livello di competenza e, ove possibile, alle preferenze degli studenti. Le attività si sono svolte nelle ore laboratoriali delle discipline coinvolte, con il supporto dei docenti, e in orario extracurricolare in maniera autonoma. I sottogruppi della stessa classe si sono confrontati settimanalmente per la revisione dei progressi, la pianificazione delle attività successive e l'eventuale ridefinizione delle priorità. Con cadenza bisettimanale, si sono svolti incontri interclasse per discutere, integrare e armonizzare i diversi contributi. Un punto di forza del progetto è stata la partecipazione attiva dei fornitori ad attività pratiche dei clienti, quali la raccolta e l'analisi dei campioni d'acqua, al fine di comprendere meglio le necessità e i vincoli operativi, migliorando così l'efficacia delle soluzioni sviluppate.

4. Architettura del sistema e tecnologie web

L'applicazione per la gestione e visualizzazione dei dati è stata sviluppata in C++, linguaggio chiave per apprendere la programmazione e le tecniche di ottimizzazione del codice. Questa attività ha consentito di sviluppare una solida padronanza del linguaggio mediante l'implementazione di operazioni CRUD su file tramite interfaccia testuale. Per ogni campione d'acqua si registrano il punto di prelievo con coordinate GPS e i valori dei batteri rilevati (coliformi, salmonella, fototrofi, utilizzatori di fosfati, nitrificanti), con controlli di coerenza dei dati. La ricerca può avvenire per luogo, batterio o concentrazione, mentre la visualizzazione include istogrammi orizzontali che rappresentano la numerosità dei batteri per ogni punto di prelievo. I dati raccolti sono stati inoltre archiviati nel database del sito web, strutturato su una classica architettura three-tier: frontend, backend e database. Il frontend, realizzato con HTML, CSS e JavaScript, offre un'interfaccia intuitiva e responsive. Il backend in PHP gestisce la logica applicativa e la comunicazione con un database MySQL, scelto per affidabilità e scalabilità. La Figura 1 mostra un esempio di visualizzazione dei dati di un punto di campionamento tramite l'interfaccia web. Il percorso formativo, dallo sviluppo di applicazioni in C++ alle soluzioni web, è particolarmente efficace nel fornire agli studenti una formazione completa e versatile, coprendo sia gli aspetti tecnici e performanti del C++, sia le tecnologie web più diffuse e accessibili. In questo modo, gli studenti saranno preparati ad affrontare sfide diversificate e ad adattarsi a molteplici contesti professionali.

5. Conclusioni e sviluppi futuri

Il buon esito del progetto, evidenziato dall'osservazione sistematica dei processi di apprendimento (partecipazione, impegno, collaborazione, contributo personale) e dai risultati ottenuti durante le attività, conferma l'efficacia di un approccio integrato, che unisce il PjBL a solide basi teoriche, per realizzare progetti reali e interdisciplinari. Rappresenta, pertanto, un modello replicabile capace di coniugare apprendimento e impegno civico. L'esperienza ha promosso la collaborazione tra scuola, università, enti esterni e comunità locale, promuovendo una sensibilizzazione attiva alla tutela delle risorse idriche. Il progetto apre la strada a uno studio più approfondito delle acque locali con campionamenti estesi e stagionali. I dati raccolti potrebbero favorire lo sviluppo di modelli predittivi semplificati, a scopo didattico, per avvicinare gli studenti alla statistica e all'analisi dei dati, oltre a consentire la valutazione dell'inquinamento in relazione a clima e attività antropiche, e a suggerire pratiche di gestione sostenibile. Per supportare queste attività, si prevede di integrare con nuove funzionalità l'applicativo in C++ e il sito web, introducendo inoltre un sistema di versioning per una gestione collaborativa e controllata del codice e della documentazione.

Contributo degli autori al progetto

Il coordinamento delle attività è stato affidato, per l'area informatica, alle Prof.sse M. Lizzi e A. Verticchio, mentre per le scienze biologiche alla Prof.ssa G. Patrizio. In qualità di referenti scientifici, L. Forlizzi e G. Melideo (UnivAQ) hanno supervisionato i seminari specialistici, tenendone personalmente alcuni. L'organizzazione dei seminari è supportata dal "Progetto Nazionale Informatica del Piano Lauree Scientifiche".

Ringraziamenti

Gli autori ringraziano tutte le persone e gli enti che hanno contribuito con competenza e passione alla realizzazione del progetto, con un particolare riconoscimento ai Proff. M. Badia, R. Di Gregorio, G. Giuliani, F. Massimi, C. Moscardi, C. Pagliariccio e M. Panzanaro per il loro costante supporto. Esprimono inoltre gratitudine alle Dott.sse M. Pellegrini e B. Farda (UnivAQ) per il prezioso contributo scientifico nelle analisi microbiologiche dei campioni, ai relatori dei seminari Dott. R. Capuano, L. Di Donato, R. Rubei e C. Di Sipio (UnivAQ) per gli approfondimenti specialistici, e al CAI, alla Croce Rossa Italiana, al Gruppo Speleologico Aquilano ETS, alla società Grotte di Stiffe e al Comune di San Demetrio per l'assistenza sul campo. Infine, ringraziano la Dirigente Scolastica Prof.ssa M. C. Marola per il sostegno e la fiducia dimostrati.

Bibliografia

- [1] L. Forlizzi, M. Lodi, V. Lonati, C. Mirolo, M. Monga, A. Montresor, A. Morpurgo, and E. Nardelli. A core informatics curriculum for Italian compulsory education. In Sergei N. Pozdniakov and Valentina Dagiené, editors, *Informatics in Schools. Fundamentals of Computer Science and Software Engineering*. LNCS, volume 11169 of ISSEP 2018, pages 141–153, Cham, 2018. Springer International Publishing.
- [2] D. Kokotsaki, V. Menzies, and A. Wiggins. Project-based learning: A review of the literature. *Improving Schools*, 19(3):267–277, 2016.
- [3] E. Nardelli, L. Forlizzi, M. Lodi, V. Lonati, C. Mirolo, M. Monga, A. Montresor, and A. Morpurgo. Proposal for a national informatics curriculum in the Italian school. Technical report, CINI, 2017.
- [4] D. Nguyen. Project-based learning (PjBL) as an experiential pedagogical methodology in interdisciplinary education: A review of the literature. *International Journal of Education in Mathematics, Science and Technology*, 13:1016–1039, 06 2025.
- [5] R. Pucher and M. Lehner. Project based learning in computer science - a review of more than 500 projects. *Procedia - Social and Behavioral Sciences*, 29:1561–1566, 12 2011.
- [6] Sadaqat Ur Rehman. Trends and challenges of project-based learning in computer science and engineering education. In *Proceedings of the 15th International Conference on Education Technology and Computers, ICETC '23*, page 397–403, New York, NY, USA, 2024. Association for Computing Machinery.
- [7] A. Saad and S. Zainudin. A review of teaching and learning approach in implementing project-based learning (PBL) with computational thinking (CT). *Interactive Learning Environments*, 32(10):7622–7646, 2024.
- [8] Mike Tissenbaum, Josh Sheldon, and Hal Abelson. From computational thinking to computational action. *Commun. ACM*, 62(3):34–36, February 2019.

Generazione digitale, ma non consapevole: giovani e IA fra percezioni e pratiche

Laura Cesaro¹, Giovanni Doderò²

¹ Università degli Studi di Padova

laura.cesaro@phd.unipd.it

² Liceo statale scientifico e classico "Martin Luther King", Genova - EFT Liguria

giovanni.dodero@gmail.com

Abstract

L'Intelligenza Artificiale (IA) è parte integrante della quotidianità giovanile, ma rimane assente dai curricula scolastici italiani. Questa indagine presenta due esperienze educative in Veneto e Liguria con studenti di scuole superiori per indagare rappresentazioni, usi e percezioni dell'IA. Attraverso sondaggi digitali e metodologie esperienziali, la ricerca ha coinvolto otto classi di istituti tecnici e licei. I risultati evidenziano una semplificazione concettuale: l'IA viene identificata prevalentemente con ChatGPT, utilizzata in modo diffuso ma con scarsa consapevolezza dei processi tecnologici. Emerge una dimensione emotiva complessa e crescente attenzione verso le implicazioni sociali e ambientali. Lo studio sottolinea l'urgenza di sviluppare competenze critiche per una cittadinanza digitale consapevole, proponendo un ripensamento del ruolo educativo della scuola nell'era dell'IA.

1. Introduzione

L'Intelligenza Artificiale (IA) è diventata una presenza costante e trasversale nella vita quotidiana, soprattutto per le nuove generazioni. I giovani la utilizzano attraverso motori di ricerca, assistenti vocali, social network, piattaforme video, chatbot generativi e strumenti creativi. Nonostante la sua crescente diffusione e pur essendo presente in diversi riferimenti istituzionali [8-12], l'IA rimane tuttavia in larga parte assente dai curricula scolastici ufficiali italiani, e il suo utilizzo avviene per lo più senza mediazione didattica o consapevolezza critica. La mancanza di una sistematizzazione educativa pone interrogativi urgenti: come si formano le rappresentazioni giovanili dell'IA? Quali usi ne fanno gli studenti? E soprattutto, quali competenze servono per trasformare l'interazione quotidiana in una cittadinanza digitale consapevole?

Per esplorare questi aspetti, sono state condotte due esperienze educative distinte ma complementari, nelle regioni Veneto e Liguria. Entrambe avevano come obiettivo l'ascolto degli studenti e la comprensione delle loro conoscenze, percezioni e aspettative nei confronti dell'IA, attraverso attività laboratoriali, strumenti interattivi e sondaggi digitali.

Nel contesto veneto, l'attività si è svolta con quattro classi del triennio di istituti tecnici, coinvolte in percorsi di PCTO in presenza. A partire da sondaggi svolti attraverso piattaforme

digitali, sono stati indagati quattro aspetti chiave: quali strumenti vengono utilizzati per la ricerca di informazioni, quali parole vengono associate all'IA, quali emozioni essa evoca e quanto se ne conosce la storia.

In Liguria, l'investigazione è stata condotta in quattro classi terze e quarte liceo, attraverso un'attività esperienziale ispirata alla metodologia del “barometro delle opinioni” o “grafico umano”, finalizzata a esplorare la consapevolezza e l'utilizzo di tre concetti e strumenti chiave della cultura digitale: bit e byte, dispositivi digitali e memoria; Internet; Intelligenza artificiale.

Pur essendo state condotte con approcci differenti, le due esperienze hanno fatto emergere convergenze significative.

2. Rappresentazioni spontanee dell'IA

In entrambe le sperimentazioni, è stato chiesto agli studenti di associare parole al concetto di Intelligenza Artificiale. In Veneto, le parole sono state raccolte attraverso un sondaggio digitale in tempo reale: gli studenti, utilizzando il proprio cellulare, potevano scrivere fino a 3 parole in appositi campi; in Liguria, tramite un brainstorming iniziale rispondendo con una parola, poi trascritta alla lavagna, alla domanda: “Cos'è per te l'Intelligenza Artificiale?”. L'obiettivo era raccogliere termini e riferimenti che emergono in assenza di definizioni guidate, per cogliere quali conoscenze, idee, bias o falsi miti i ragazzi associano all'IA.

Le parole raccolte offrono un primo quadro del modo in cui l'intelligenza artificiale viene percepita e collocata culturalmente, segnalando connessioni con la tecnologia, la quotidianità, la scuola, il futuro o altri contesti significativi per il loro vissuto.



Figura 1: Parole associate all'IA

In entrambi i contesti, la parola “ChatGPT” è emersa come dominante, al punto da rappresentare per molti l'intera idea di IA. Il suo nome viene spesso utilizzato come etichetta generica, segno di una semplificazione concettuale e della mancanza di familiarità con altri strumenti o applicazioni basate su IA.

In Veneto, accanto a ChatGPT, compaiono frequentemente termini come “robot”, “velocità”, “informazioni”, “utile”, “aiuto”, che delineano un immaginario prevalentemente pragmatico e funzionale: l'IA è vista come tecnologia che accelera, supporta e semplifica, utile

soprattutto in ambito scolastico. Altre parole, come “innovazione”, “futuro” e “intelligenza”, indicano una collocazione simbolica dell’IA in una dimensione trasformativa, proiettata verso il cambiamento.

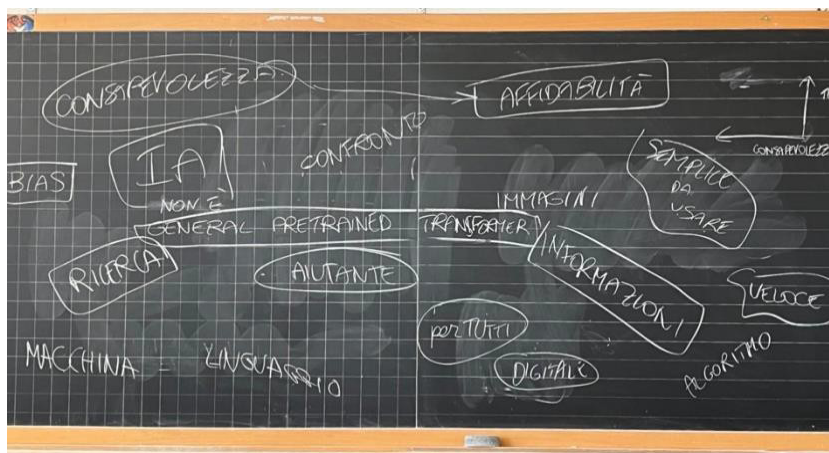


Figura 2: Brainstorming

In Liguria sono emersi anche aspetti legati all’impatto ambientale notevole prodotto dalle interrogazioni all’AI: ChatGPT stessa, chiedendoglielo, ha spiegato quanto è il suo impatto in termini di consumo energetico, idrico e produzione di CO₂. Si evince che i data center, per l’addestramento e l’esecuzione dei modelli IA consumano grandi quantità di energia elettrica e di risorse idriche, avendo un impatto ambientale non completamente trascurabile.

Tutti gli studenti hanno riconosciuto i vantaggi funzionali dell’IA: velocità, risparmio di tempo, semplificazione e supporto nelle attività scolastiche. In Liguria, grazie all’approccio laboratoriale, sono emersi anche utilità più articolate come l’automazione di compiti ripetitivi e la capacità di processare un numero elevatissimo di dati assieme, ma anche svantaggi strutturali come le possibili perdite di posti di lavoro umano e l’assoggettamento dell’uomo alla macchina.

3. Consapevolezza nell’uso dell’IA

Per comprendere davvero quanto gli studenti siano consapevoli nell’uso delle tecnologie digitali, non basta chiedere loro cosa sanno: è necessario osservare come si rapportano concretamente a concetti e pratiche digitali fondamentali, mettendo in relazione ciò che dicono di conoscere con il modo in cui effettivamente li utilizzano.

Per questo motivo, in Liguria è stata proposta un’attività esperienziale basata su un “grafico umano”, ispirato al barometro delle opinioni. Gli studenti si sono posizionati fisicamente lungo due assi: sulle ascisse “da quanto tempo lo uso” e sulle ordinate “quanto lo conosco” o “quanto lo uso”, riferiti a tre concetti chiave della cultura digitale: dispositivi digitali e memoria (bit/byte), Internet, e Intelligenza Artificiale. Si sono poste domande specifiche e si è data ai ragazzi la possibilità di posizionarsi nello spazio e, dopo momenti di riflessione e confronto, di spostarsi in modo da rendere il grafico coerente e non dipendente dalla sensazione individuale.

L'attività ha reso visibile quindi attraverso le loro disposizioni la discrepanza tra uso e conoscenza, facendo emergere bias diffusi e conoscenze implicite.

Per quanto riguarda i dispositivi digitali, gli studenti riconoscono di utilizzarli da molti anni, ma dimostrano una conoscenza limitata dell'informatica di base: solo circa il 50% sa cosa siano i bit e i byte, che sono le unità fondamentali dell'informazione. Rispetto a Internet, la rete viene comunemente identificata con Google, e concepita non come infrastruttura, ma come semplice strumento di ricerca, senza consapevolezza del trattamento dei dati o della loro localizzazione.

Quanto all'intelligenza artificiale, gli studenti affermano di utilizzarla chi da alcuni mesi chi due anni, ma sono consapevoli di avere una comprensione ancora superficiale. Solo attraverso la riflessione collettiva emerge la consapevolezza di interagire con sistemi IA già da tempo, ad esempio tramite algoritmi di raccomandazione o filtri automatici. È evidente anche in questa indagine che negli studenti ci sia il preconcetto che Intelligenza Artificiale sia un sistema Generative Pre-trained Transformer (GPT) basato su Large Language Model (LLM) con interazione via chat, questo coerentemente con quanto sta avvenendo nella società.

A complemento di questo quadro, in Veneto è stato chiesto agli studenti in che periodo collocassero la nascita dell'IA, scegliendo fra quattro date e quale strumento utilizzassero per cercare informazioni.

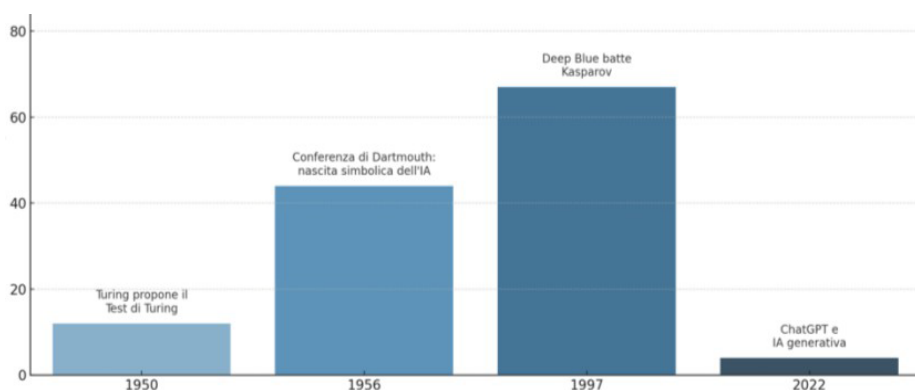


Figura 3: Collocazione della nascita dell'IA

Per quanto riguarda la prima domanda, la maggior parte degli studenti riconosce che l'IA ha origini precedenti a ChatGPT, ma tende a collocarla in modo impreciso. È probabile che le risposte corrette siano state date dagli studenti con un background scolastico in informatica.

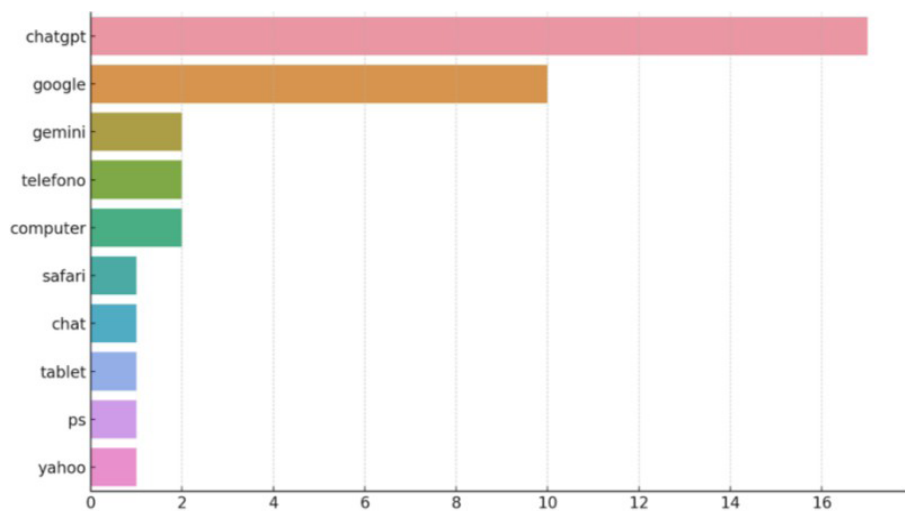


Figura 4: *Quale strumento usi quando devi cercare qualcosa? - Risposte*

Per quanto riguarda invece la ricerca di informazioni, la maggior parte ha indicato ChatGPT in misura superiore a Google, a conferma di un trend già osservato nella letteratura recente: l'abbandono della logica di navigazione per link in favore di interazioni conversazionali e personalizzate, mediate da modelli linguistici di grandi dimensioni. I giovani adottano forme di information sensibility, basate su pratiche intuitive ma spesso prive di criteri sistematici di verifica [2,4]. Questo uso prevalente dei chatbot generativi solleva interrogativi educativi urgenti: in che modo la scuola può sviluppare competenze critiche che permettano agli studenti di orientarsi in modo consapevole nel nuovo ecosistema informativo? È necessario insegnare ai giovani la differenza tra informazioni e conoscenze: la prima è un mero dato, la seconda comporta connessioni, confronti e complessità. Quando non si è esperti di un tema, non si riesce a comprenderne le sfaccettature, per cui la propria opinione, basata su una rapida ricerca in rete o su pregiudizi personali, viene considerata valida, credibile e soprattutto superiore alle altre. È fondamentale sviluppare il senso critico degli studenti, aiutandoli a riconoscere che l'IA è uno strumento da interrogare e interpretare, non una fonte di verità assoluta, e i docenti svolgono un ruolo chiave nel guidare questo processo. Rispetto ad altri strumenti digitali, infatti, la facilità di interazione grazie all'uso del linguaggio naturale e l'immediatezza di una risposta almeno apparentemente coerente con la domanda, può far sottovalutare il fondamentale passaggio dell'interpretazione e della rielaborazione del testo restituito.

4. IA ed emozioni

L'intelligenza artificiale non è percepita solo come uno strumento tecnico, ma anche come un interlocutore simbolico ed emotivo, soprattutto tra i più giovani. Studi recenti mostrano come gli adolescenti utilizzino i chatbot generativi anche come spazi di dialogo e conforto, attribuendo loro capacità di ascolto ed empatia [3]. Alcune ricerche ne evidenziano l'utilità nel favorire l'accesso al supporto psicologico di base, sebbene richiedano garanzie etiche e una

supervisione umana [1]. Altre, invece, mettono in guardia rispetto al cosiddetto empathy gap: la tendenza a sovrastimare l'intelligenza emotiva delle macchine, con il rischio di fraintendimenti e dipendenza [5]. In questo contesto, chiedere agli studenti "che cosa provano" pensando all'IA non serve solo a rilevare un atteggiamento pratico, ma diventa un'occasione per cogliere la varietà di emozioni che l'IA suscita contribuendo così a restituire la complessità di un fenomeno che è al tempo stesso tecnologico, psicologico e culturale.

Le parole più citate dagli studenti veneti riflettono un ampio spettro emotivo: amore (15), paura (10), gratitudine (9), curiosità (7). Seguono, con frequenza minore ma significativa, termini come sicurezza, salvezza e utilità (5 ciascuno), e infine intelligenza, dubbio, ammirazione e comodità (4). Accanto alle risposte secche, emergono anche formulazioni più articolate, che introducono elementi di riflessione critica e uso consapevole. Non mancano infine risposte che esprimono indifferenza, ambivalenza o incertezza.

Anche in Liguria, durante il brainstorming iniziale, è emerso il ruolo delle emozioni nel rapporto con lo sviluppo tecnologico: l'attività ha permesso di svelare ansie, ma anche aspettative e fiducia, mostrando come l'IA venga vissuta anche a livello personale ed emotivo, oltre che cognitivo.

5. Conclusioni

Nonostante non sia possibile mettere le due esperienze a confronto in modo sistematico, le informazioni analizzate mostrano molti elementi comuni: l'associazione semplificata tra IA e ChatGPT, la scarsa consapevolezza dei processi tecnologici, un uso diffuso ma raramente critico, e una crescente attenzione verso le implicazioni sociali e ambientali della tecnologia. In un contesto in cui l'IA è già integrata nella quotidianità degli studenti, la scuola è chiamata a fornire strumenti per interpretarla e valutarla con consapevolezza.

Pur essendo condotta su un campione ridotto di studenti di due regioni del Nord Italia, la convergenza dei risultati fa intuire l'importanza di approfondire con studi più sistematici e con strumenti comuni la ricerca su un campione più numeroso e rappresentativo di tutta Italia. L'uso educativo dell'IA deve andare oltre l'efficienza funzionale e promuovere creatività, collaborazione e senso critico, mettendo in luce anche i rischi legati alla dipendenza, alla delega cognitiva e all'illusione dell'empatia artificiale. La tendenza a proiettare emozioni sui chatbot rende urgente educare alla distinzione tra supporto digitale e relazione umana.

Per questo è fondamentale formare i docenti e costruire alleanze tra scuola, studenti e famiglie, in modo da favorire un uso dell'IA orientato non solo all'apprendimento, ma anche al benessere e alla cittadinanza digitale attiva. Importante anche che i docenti abbiano coscienza che lo sviluppo tecnologico abbia ormai un andamento esponenziale [5,6] e questo significa che più che inseguire le ultime novità conviene dare cornici di senso e significato nel dialogo educativo che facilitino tale consapevolezza.

Bibliografia

- [1] Dray, J., Symons, D. Review of Innovative Mental Health Support for Children and Young People: Generative AI Co-design Applications and Challenges. *Curr Dev Disord Rep* 12, 13 (2024). <https://doi.org/10.1007/s40474-025-00328-z>
- [2] Hassoun, A., Beacock, I., Consolvo, S., Goldberg, B., Kelley, P. G., & Russell, D. M. (2023). Practicing Information Sensibility: How Gen Z Engages with Online Information. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. ACM. <https://doi.org/10.1145/3544548.3581328>
- [3] Huang, S., Lai, X., Ke, L., Li, Y., Wang, H., Zhao, X., Dai, X., & Wang, Y. (2024). AI Technology panic-is AI Dependence Bad for Mental Health? A Cross-Lagged Panel Model and the Mediating Roles of Motivations for AI Use Among Adolescents. *Psychology research and behavior management*, 17, 1087–1102. <https://doi.org/10.2147/PRBM.S440889>
- [4] Klarin, J., Hoff, E., Larsson, A., & Daukantaitė, D. (2024). Adolescents' use and perceived usefulness of generative AI for schoolwork: exploring their relationships with executive functioning and academic achievement. *Frontiers in artificial intelligence*, 7, 1415782. <https://doi.org/10.3389/frai.2024.1415782>
- [5] Kurian, N. (2024). 'No, Alexa, no!': designing child-safe AI and protecting children from the risks of the 'empathy gap' in large language models. *Learning, Media and Technology*, 1–14. <https://doi.org/10.1080/17439884.2024.2367052>
- [6] KURZWEIL - Progresso e tecnologia esponenziale <https://startegy.it/kurzweil-la-tecnologia-e-un-processo-esponenziale/>
- [7] Quintarelli, S., Ferrauto, C.G., Corea, F., Fossa, f., Loreggia, A., Sapienza, S., *Intelligenza Artificiale*.
- [8] Cos'è davvero, come funziona, che effetti avrà. 2020, Bollati Boringhieri
- [9] UNESCO - Miao, F., Holmes, W., Ronghuai H., Hui Z., *AI and education: guidance for policy-makers*, 2021, <https://doi.org/10.54675/PCSP7350>
- [10] UNESCO – Miao, F. Cukurova, M., *AI competency framework for teachers*, 2024, <https://doi.org/10.54675/ZJTE2084>
- [11] UNESCO – Miao, F., Shiohira, K., Lao, N., *AI competency framework for students*, 2024, <https://doi.org/10.54675/JKJB9835>
- [12] Unione Europea, Regolamento (UE) 2024/1689 che stabilisce regole armonizzate sull'intelligenza artificiale (AI Act), *Gazzetta ufficiale dell'Unione europea*, 2024.
- [13] Il "Disegno di legge sull'intelligenza artificiale", che recepisce le indicazioni dell'AI Act in Italia, è identificato con la sigla DDL S. 1146 (19ª Legislatura). Approvato dal Senato, è attualmente in commissione alla Camera dei deputati.
- [14] Vuorikari, R., Kluzer, S. e Punie, Y., *DigComp 2.2, The Digital Competence framework for citizens: with new examples of knowledge, skills and attitudes*, Ufficio delle pubblicazioni dell'Unione europea, 2022, <https://data.europa.eu/doi/10.2760/115376>

Liceo Classico Digitale: Proposta Contenuti Disciplinari e Prime Sperimentazioni

Simone Cuconato¹, Kristian Reale¹

¹ Università della Calabria, Rende (CS)

simone.cuconato@unical.it kristian.reale@unical.it

Abstract

Nel contesto dell'introduzione di nuovi indirizzi curriculari innovativi nelle scuole, le cosiddette curvature, questo lavoro si concentra su un Liceo Classico che, da due anni, ha avviato la Curvatura Digitale denominata "*Cl@ssico Digitale*". Il percorso ha previsto l'inserimento, nel programma tradizionale, di due nuove materie: "Comunicazione Digitale" e "Scienza e Tecnologie Digitali". In questo lavoro si analizza quanto svolto nel primo biennio, con l'obiettivo di riflettere sui contenuti proposti, valutare possibili modifiche e progettare gli sviluppi per il triennio successivo.

1. Introduzione

In un momento storico in cui la transizione digitale coinvolge profondamente la società, la cultura e l'economia, anche la scuola è chiamata a rinnovarsi. L'introduzione della didattica digitale integrata, con strumenti come tablet, ambienti online (es. Google Classroom) e materiali digitali, ha rappresentato un primo passo importante verso una didattica più flessibile e inclusiva, affiancando i metodi tradizionali. Fin dai primi inserimenti dell'Informatica come disciplina scolastica, specialmente nei Licei Scientifici a indirizzo Tecnologico e negli Istituti Tecnici, i contenuti si sono evoluti per tenere il passo con le trasformazioni tecnologiche. In questo percorso ha avuto un ruolo centrale il modello STEM (Scienza, Tecnologia, Ingegneria, Matematica), che ha promosso attività innovative: dal coding nella scuola primaria fino alla programmazione, alle reti, alle basi di dati e all'intelligenza artificiale (IA) nelle scuole superiori.

Il Piano Nazionale Scuola Digitale (PNSD) ha sostenuto questo rinnovamento, puntando su tecnologie, metodologie e formazione dei docenti, con l'obiettivo di sviluppare competenze digitali e cittadinanza consapevole. Anche i licei a vocazione umanistica hanno iniziato a ripensare la propria identità, ispirandosi al modello del Liceo Scientifico con opzione Scienze Applicate, per formare cittadini digitali critici e responsabili.

In questo contesto, in alcuni Licei Classici, nascono le prime curvature digitali: percorsi che rinnovano l'insegnamento umanistico integrandolo con strumenti e linguaggi digitali. L'obiettivo è collegare in modo costruttivo saperi umanistici e tecnologici, favorendo un dialogo tra filosofia, letteratura, scienza e intelligenza artificiale, in una prospettiva interdisciplinare.

Il Liceo Classico "*Gioacchino da Fiore*" di Rende (Cosenza) ha raccolto questa sfida, avviando, nell'A.S. 2023/24, il percorso sperimentale "*Cl@ssico Digitale*", che unisce la solidità della formazione classica alle opportunità offerte dalle nuove tecnologie dell'ICT. Il curriculum si basa

su un approccio multidisciplinare, realizzato in collaborazione con l'*Università della Calabria*, che coinvolge direttamente i suoi docenti. Il progetto introduce due nuove discipline, "*Comunicazione Digitale*" e "*Scienza e Tecnologie Digitali*", con un'ora settimanale ciascuna, per fornire agli studenti strumenti teorici e pratici per comprendere e partecipare alla società digitale.

Considerando che a livello ministeriale non esiste un *curricolo* scolastico ben definito per l'insegnamento dell'informatica nei Licei Classici, l'intento è quindi quello di progettare possibili contenuti disciplinari che possano contribuire alla definizione di un nuovo curriculum per i Licei Classici a Curvatura Digitale. Il lavoro presenta quanto realizzato nel primo biennio del Liceo Classico "*Gioacchino da Fiore*", con i relativi obiettivi, e avanza una prima ipotesi di sviluppo per il triennio successivo.

2. Proposta Contenuti Disciplinari

I contenuti dell'indirizzo *Classico Digitale* sono articolati attorno a due specifiche materie sperimentali: *Comunicazione Digitale* e *Scienza e Tecnologie Digitali*. Entrambe hanno l'obiettivo di fornire agli/alle studenti/esse gli strumenti informatici fondamentali per interpretare e partecipare in modo consapevole e attivo alla società digitale e dell'informazione.

L'approccio adottato è stato al tempo stesso preliminare e fondazionale. Preliminare, in quanto sono stati introdotti: *i)* gli strumenti logico-matematico-statistici necessari per comprendere il digitale da una prospettiva tecnico-scientifica; *ii)* le nozioni informatiche di base. Fondazionale, poiché sono state poste le basi storiche, teoriche ed epistemologiche del pensiero digitale e computazionale.

2.1 Comunicazione Digitale

L'insegnamento di *Comunicazione Digitale* nel primo biennio ha avuto lo scopo di fornire un'alfabetizzazione al ragionamento formale e critico-analitico alla base della comunicazione digitale. Attraverso il dialogo costante tra il sapere classico-umanistico e quello scientifico-tecnologico, il corso ha sviluppato le competenze necessarie per pensare l'era digitale con spirito critico. In particolare, il corso è stato articolato intorno a due nuclei tematici fondamentali. Il primo ha riguardato la logica classica, matematica e computazionale. L'attività didattica ha incluso una panoramica storico-epistemologica dalla sillogistica aristotelica fino alle basi formali dell'algebra della logica. Il secondo nucleo tematico si è concentrato sulle attività, i rischi e le sfide legate all'IA. Attraverso l'analisi e il commento di articoli selezionati, le studentesse e gli studenti sono stati introdotti in modo analitico all'IA, sviluppando una consapevolezza critica nei confronti di questa tecnologia. Più in dettaglio, il corso è stato articolato nei seguenti moduli:

- **Modulo I:** *Pensiero Classico e Pensiero Computazionale*. introduzione al pensiero classico; principi logici del pensiero classico; introduzione al pensiero computazionale; Test di Turing; rischi e sfide dell'intelligenza artificiale;
- **Modulo II:** *Logica Formale, Insiemi e Funzioni*. elementi di teoria degli insiemi; teoremi di Cantor; relazioni e funzioni;
- **Modulo III:** *Sintassi della logica proposizionale (L_p)*. linguaggio L_p ; regole di formazione; formule atomiche e complesse;
- **Modulo IV:** *Semantica di L_p* . elementi di teoria dei modelli; tavole di verità; verità logica; soddisfacibilità; spazio logico.

2.2 Scienza e Tecnologie Digitali

L'insegnamento di *Scienza e Tecnologie Digitali* è stato concepito come una disciplina di base dell'informatica, simile a quella presente nei Licei Scientifici con Opzione Scienze Applicate, ma con un'impostazione maggiormente orientata alla logica. Nel primo biennio sono stati introdotti i concetti fondamentali dell'informatica, tra cui i numeri binari, le operazioni logiche (AND, OR, NOT), la logica matematica (proposizioni, valori di verità, implicazioni, sillogismi, quantificatori), il concetto di linguaggio (vocabolario, sintassi, semantica), gli algoritmi con rappresentazione tramite pseudocodice e diagrammi di flusso, e i primi cenni ai linguaggi di programmazione, in particolare Python. Sono stati inoltre affrontati temi come le strutture dati (liste, grafi, alberi), il calcolo delle probabilità, e si sono svolte attività pratiche e digitali, tra cui l'uso di ChatGPT e strumenti per la creazione di mappe concettuali. Più in dettaglio, il corso è stato articolato nei seguenti moduli:

- **Modulo I: Fondamenti di Informatica e Logica.** Numeri binari, operazioni logiche (AND, OR, NOT), insiemi numerici, logica matematica (proposizioni, valori di verità, implicazioni, sillogismi, quantificatori).
- **Modulo II: Linguaggi e Algoritmi.** Concetto di linguaggio (vocabolario, sintassi, semantica), algoritmi, pseudocodice, diagrammi di flusso, introduzione a Python (variabili, input/output, conversioni, condizioni, cicli).
- **Modulo III: Calcolo delle Probabilità.** Probabilità semplici, eventi dipendenti e indipendenti, probabilità condizionata e congiunta, cenni al Teorema di Bayes, esperimenti pratici con monete e dadi.
- **Modulo IV: Strutture Dati e Strumenti Digitali.** Liste, liste doppiamente concatenate, grafi, alberi, analogie con alberi genealogici e mappe concettuali, uso di ChatGPT, Canva per mappe concettuali, cenni sul funzionamento della RAM.

2.3 Collegamento critico con le altre materie umanistiche

I contenuti proposti hanno posto particolare attenzione ai collegamenti interdisciplinari tra l'ambito scientifico-tecnologico e classico-umanistico. In particolare, il confronto con il pensiero filosofico ha giocato un ruolo fondamentale nella costruzione critica del pensiero computazionale. La riflessione sull'epistemologia digitale ha preso le mosse dai concetti classici di verità, dimostrazione, razionalità e linguaggio, affrontati attraverso il confronto tra logica aristotelica, razionalismo moderno e paradigmi contemporanei dell'intelligenza artificiale.

Queste sinergie disciplinari hanno permesso di sviluppare nei/lle studenti/esse una visione integrata della conoscenza, dove rigore formale e consapevolezza critica cooperano per formare cittadini digitali consapevoli, capaci non solo di utilizzare le tecnologie, ma di comprenderne i fondamenti teorici e le implicazioni etiche.

3. Sviluppi Futuri

Per la progettazione dei contenuti disciplinari del triennio successivo, nonché per una rivisita dei contenuti del primo biennio anche alla luce dei risultati raggiunti in questi primi due anni, si darà forte enfasi ad una progettazione multidisciplinare in accordo soprattutto con gli altri dipartimenti dell'Istituto Scolastico. Un possibile approccio consiste nello sviluppo di specifici programmi didattici volti a potenziare le competenze informatiche e a promuovere lo sviluppo

di un pensiero critico e consapevole sulla dimensione digitale, essenziale per affrontare le sfide poste dall'era dell'intelligenza artificiale.

Nell'ambito dell'insegnamento di Comunicazione Digitale, saranno valorizzate le potenzialità formative della logica, con particolare attenzione alle procedure algoritmiche dimostrative sia sintattiche (come il calcolo dei sequenti di Gentzen) sia semantiche (come i tableaux analitici).

Nell'ambito di Scienza e Tecnologie Digitali si intende prendere ispirazione dai contenuti già presenti nel Liceo Scientifico opzione Scienze Applicate, adattandoli però a un'impostazione meno tecnico-scientifica. Tra i temi trattati vi sono le reti informatiche, gli algoritmi e le strutture dati, la programmazione in Python avanzata, le basi di dati e l'intelligenza artificiale, quest'ultima affrontata con particolare attenzione a un approccio neuro-simbolico che integra Deep Learning e programmazione logica.

Ringraziamenti

Desideriamo ringraziare il Prof. Francesco Scarcello e il Prof. Roberto Beneduci dell'Università della Calabria per il loro prezioso contributo scientifico e didattico al progetto, e la Dirigente Scolastica Prof.ssa Brunella Baratta del Liceo Classico "Gioacchino da Fiore" per il costante supporto e la visione innovativa.

Aula Nova: Progettazione delle lezioni di Informatica nella scuola secondaria di secondo grado con i LLM

Alexandra Sheykina¹, Gabriele De Vito¹, Andrea De Lucia¹

¹SeSa Lab - Dipartimento di Informatica, Università degli Studi di Salerno, Fisciano, Italia

{asheykina@unisa.it, gadevito@unisa.it, adelucia@unisa.it}

Abstract

Negli ultimi anni, i processi di trasformazione digitale hanno avuto un impatto profondo sulla pratica educativa. L'integrazione delle tecnologie basate sull'intelligenza artificiale (AI), in particolare i modelli linguistici di grandi dimensioni (LLM), sta emergendo come un'opportunità concreta per supportare i docenti nella progettazione delle lezioni e nella produzione di materiali didattici. In questo studio presentiamo Aula Nova, uno strumento progettato per semplificare la progettazione didattica nell'insegnamento dell'informatica nelle scuole superiori di secondo grado. Lo strumento si propone di ridurre il carico cognitivo dei docenti, facilitare l'inclusione, velocizzare la creazione di materiali coerenti con gli standard formativi e favorire un approccio didattico personalizzato. I risultati preliminari di uno studio sperimentale condotto con 60 docenti indicano un'elevata soddisfazione e usabilità. Il lavoro rappresenta, ad oggi, uno dei primi tentativi di utilizzo sistematico di LLMs per la progettazione didattica e delle singole lezioni in ambito informatico.

1. Introduzione

L'innovazione tecnologica sta cambiando il modo di insegnare e apprendere, sia nella scuola che nell'università. In Italia, come in molti altri paesi, cresce l'attenzione verso l'adozione di tecnologie digitali in ambito educativo. Il lavoro quotidiano degli insegnanti si caratterizza per una complessità sempre maggiore di compiti e responsabilità, rendendo strategico l'impiego efficace delle tecnologie per supportare l'insegnamento.

In questo contesto, i LLMs stanno emergendo come strumenti promettenti per l'educazione, grazie alla loro capacità di generare contenuti, rispondere a domande, supportare l'apprendimento e assistere nella progettazione didattica. Numerosi studi hanno esplorato il loro impiego per la generazione automatica di quiz [1], la creazione di esercizi di programmazione e spiegazioni di codice [2], e la produzione di contenuti e attività in varie discipline [10, 12]. Inoltre, studi recenti dimostrano il potenziale dei LLMs nel supportare la creazione di attività didattiche inclusive e personalizzate, configurandosi come veri e propri partner pedagogici nella pianificazione delle lezioni [3, 4, 5, 6, 7, 8, 13, 15]. Il presente studio mira a ridurre la complessità della progettazione didattica, rendendola più accessibile, rapida, semplice, tecnologicamente avanzata e centrata sullo studente, nel contesto della scuola secondaria superiore. I docenti si trovano quotidianamente a fronteggiare molteplici sfide nella progettazione didattica: dalla definizione di competenze e obiettivi di apprendimento alla strutturazione delle Unità Didattiche

di Apprendimento (UDA), dalla creazione di singole lezioni alla preparazione di materiali educativi quali slide, verifiche e attività pratiche. A ciò si aggiunge la necessità di garantire percorsi inclusivi e personalizzati per studenti con bisogni educativi speciali, mantenendo al contempo coerenza con gli standard educativi e le proprie metodologie di insegnamento. L'obiettivo di Aula Nova non è quello di sostituire il ruolo del docente come professionista dell'educazione, ma offrire suggerimenti strutturati e personalizzabili che possano essere integrati e adattati in base alle esigenze e scelte pedagogiche individuali.

Questo studio affronta la progettazione delle lezioni di informatica attraverso un approccio sistemico che integra diversi parametri: dinamiche di classe, livelli cognitivi degli studenti, ritmi di apprendimento individuali e presenza di bisogni educativi speciali.

Aula Nova offre ai docenti un framework personalizzabile che garantisce elevata flessibilità attraverso la modulazione della complessità dei contenuti, l'integrazione di metodologie didattiche diversificate e la scalabilità temporale delle unità didattiche in funzione degli obiettivi formativi prestabiliti. Tale approccio rappresenta, a nostra conoscenza, il primo contributo scientifico dedicato alla progettazione curricolare completa per l'insegnamento dell'informatica nella scuola secondaria mediante l'uso di LLMs.

2. Descrizione Aula Nova

Aula Nova è uno strumento sviluppato per supportare la progettazione didattica nell'insegnamento dell'informatica nella scuola secondaria di secondo grado. L'obiettivo è semplificare e velocizzare la creazione di materiali didattici, nel rispetto delle direttive ministeriali e dei principi di didattica inclusiva. L'interazione con il sistema avviene attraverso input in linguaggio naturale: i docenti descrivono il contesto di insegnamento, il tipo di scuola, la disciplina e le eventuali esigenze educative speciali degli studenti. Il processo si articola in tre fasi: la progettazione disciplinare annuale, la pianificazione di lezioni simulate e la creazione dei materiali di supporto alla lezione. Nella prima fase, viene generato un piano dettagliato che include tutte le UDA previste per l'anno scolastico. Successivamente, è possibile selezionare una UDA e pianificare in dettaglio le singole lezioni. La lezione simulata comprende sia la fase pre-attiva, in cui vengono descritti contesto, obiettivi, prerequisiti, metodologie, strumenti compensativi, misure dispensativi e criteri di valutazione, sia la fase attiva, incentrata sull'interazione didattica. Lo strumento consente anche di generare materiali per la verifica degli apprendimenti, sia scritti che orali, e strumenti per l'autovalutazione degli studenti. Tutti i contenuti generati possono essere modificati prima o dopo la creazione, in base ai feedback forniti dal docente. Infine, i materiali didattici prodotti vengono esportati in diversi formati (Word e PowerPoint) e raccolti in una cartella compressa.

3. Metodologia e descrizione dei risultati preliminari

La ricerca ha coinvolto 60 docenti frequentanti un percorso di abilitazione all'insegnamento (60 CFU) per la disciplina informatica nella scuola secondaria di secondo grado. Il campione, caratterizzato da laureati in fase di specializzazione pedagogico-didattica, presentava elevata motivazione professionale e consolidate competenze digitali, evidenziando particolare propensione all'adozione di tecnologie emergenti, inclusi i sistemi basati su LLMs. La valutazione

di Aula Nova è avvenuta tramite tre questionari post-sperimentazione, incentrati su usabilità, soddisfazione e accettazione della nuova tecnologia. La generazione dei contenuti è stata ottimizzata attraverso tecniche avanzate di prompt engineering, pattern di persona, template di output e approccio RAG (Retrieval-Augmented Generation), al fine di ottenere risposte coerenti e contestualizzate. È stato scelto GPT-4o come modello linguistico di riferimento (LLM), per le sue capacità avanzate di comprensione contestuale e generazione naturale del linguaggio. Gli strumenti di valutazione sono stati sviluppati seguendo pratiche consolidate nel campo dell'ingegneria del software e della valutazione della qualità [9], e nel rispetto degli standard empirici ACM/SIGSOFT¹. Per la valutazione è stata adottata una scala Likert a cinque punti [11], in conformità alle raccomandazioni metodologiche per la ricerca qualitativa in ingegneria del software, mentre la presentazione dei risultati aderisce alle linee guida proposte da Wohlin et al. [14]. I risultati preliminari indicano un'esperienza molto positiva da parte dei docenti, con alti livelli di utilità percepita, produttività e facilità d'uso. Le valutazioni sono risultate costantemente elevate, con la maggior parte dei partecipanti che ha attribuito punteggi tra 4 e 5.

Tra gli aspetti più apprezzati vi sono la possibilità di personalizzare facilmente i materiali attraverso feedback in linguaggio naturale, la presenza di un punto di partenza

strutturato per la pianificazione e l'interfaccia semplice e pulita, che riduce il carico cognitivo. Sono emerse anche proposte di miglioramento, come l'introduzione di funzionalità per la gestione degli account e l'archiviazione dei materiali didattici specifici per istituto.

4. Conclusioni

L'integrazione dei LLMs nell'ecosistema educativo rappresenta un'opportunità strategica per l'ottimizzazione della progettazione curricolare e lo sviluppo di risorse didattiche adaptive e inclusive. L'efficacia di tale implementazione dipende tuttavia dalla formulazione di prompt strutturati e contestualizzati, capaci di rispondere alle specificità dell'ambiente di apprendimento e ai diversi profili cognitivi degli studenti. Se da un lato le potenzialità sono notevoli, dall'altro emergono rischi legati alla generazione di contenuti troppo generici o imprecisi. La questione della qualità dei contenuti generati va infatti considerata con attenzione, riconoscendo i limiti attuali dei modelli LLM, come le cosiddette "allucinazioni" o l'assenza di coerenza con i programmi scolastici. Il nostro strumento affronta queste problematiche permettendo ai docenti di integrare agevolmente feedback testuali per ottimizzare la progettazione didattica, preservando così il loro ruolo strategico di supervisione pedagogica. I risultati dello studio evidenziano la validità di Aula Nova come risorsa strategica per l'ottimizzazione dei processi didattici, garantendo una significativa riduzione dei tempi di preparazione e della complessità operativa. Tra le prospettive future, si prevede l'estensione dell'uso di GPT-4o ad altre discipline e gradi scolastici, con l'obiettivo di offrire ai docenti strumenti sempre più efficaci per costruire esperienze di apprendimento coinvolgenti e su misura.

Fondi

Questo lavoro è stato parzialmente finanziato dal progetto FAIR (PE0000013) nell'ambito del programma NRRP MUR finanziato dall'UE - NGEU.

¹ <https://github.com/acmsigsoft/EmpiricalStandards>

Bibliografia

- [1] R. Dijkstra, Z. Genc, S. Kayal, J. Kamps, et al., “Reading Comprehension Quiz Generation using Generative Pre-trained Transformers,” in *iTextbooks@ AIED*, 2022, pp. 4–17.
- [2] S. Sarsa, P. Denny, A. Hellas, and J. Leinonen, “Automatic generation of programming exercises and code explanations using large language models,” in *Proceedings of the 2022 ACM Conference on International Computing Education Research – Volume 1*, 2022, pp. 27–43.
- [3] B. Hu, L. Zheng, J. Zhu, L. Ding, Y. Wang, and X. Gu, “Teaching plan generation and evaluation with GPT-4: Unleashing the potential of LLM in instructional design,” *IEEE Transactions on Learning Technologies*, 2024.
- [4] B. Hu, J. Zhu, Y. Pei, and X. Gu, “Exploring the potential of LLM to enhance teaching plans through teaching simulation,” *npj Science of Learning*, vol. 10, no. 1, p. 7, 2025.
- [5] A. Goslen, Y. J. Kim, J. Rowe, and J. Lester, “LLM-based student plan generation for adaptive scaffolding in game-based learning environments,” *International Journal of Artificial Intelligence in Education*, pp. 1–26, 2024.
- [6] H. Fan, G. Chen, X. Wang, and Z. Peng, “LessonPlanner: Assisting novice teachers to prepare pedagogy-driven lesson plans with large language models,” in *Proc. 37th Annu. ACM Symp. User Interface Softw. Technol.*, pp. 1–20, 2024.
- [7] Y. Zheng, X. Li, Y. Huang, Q. Liang, T. Guo, M. Hou, B. Gao, M. Tian, Z. Liu, and W. Luo, “Automatic lesson plan generation via large language models with self-critique prompting,” in *Proc. Int. Conf. Artif. Intell. Educ.*, pp. 163–178, 2024.
- [8] M. Moundridou, N. Matzakos, and S. Doukakis, “Generative AI tools as educators’ assistants: Designing and implementing inquiry-based lesson plans,” *Computers and Education: Artificial Intelligence*, vol. 7, p. 100277, 2024.
- [9] N. Fenton and J. Bieman, *Software Metrics: A Rigorous and Practical Approach*. CRC Press, 2014.
- [10] C. K. Lo, “What is the impact of ChatGPT on education? A rapid review of the literature,” *Education Sciences*, vol. 13, no. 4, p. 410, 2023.
- [11] B. A. Kitchenham and S. L. Pfleeger, “Personal opinion surveys,” in **Guide to Advanced Empirical Software Engineering**, Springer, 2008, pp. 63–92.
- [12] N. Rutten, W. R. Van Joolingen e J. T. Van Der Veen, “The learning effects of computer simulations in science education,” **Computers & Education**, vol. 58, n. 1, pp. 136–153, 2012.
- [13] R. O. Davis and Y. J. Lee, “Prompt: ChatGPT, create my course, please!,” **Education Sciences**, vol. 14, no. 1, p. 24, 2023.
- [14] C. Wohlin, M. Höst, and K. Henningsson, “Empirical research methods in software engineering,” in **Empirical Methods and Studies in Software Engineering: Experiences from ESERNET**, Springer, 2003, pp. 7–23.
- [15] Geesje van den Berg and Elize du Plessis, “ChatGPT and generative AI: Possibilities for its contribution to lesson planning, critical thinking and openness in teacher education,” *Education Sciences*, vol. 13, no. 10, pp. 998, 2023.

**RACCONTI DI ESPERIENZE SUL CAMPO E
RELAZIONI SCIENTIFICHE**

PER OGNI ORDINE E GRADO

Vibe Coding: un paradigma emergente per la programmazione assistita da IA nella didattica

Andrea Canesi¹, Marco Canesi¹

¹ Nuvie APS

andrea.canesi@nuvie.it marco.canesi@nuvie.it

Abstract

Il presente articolo analizza il cosiddetto Vibe Coding, un approccio alla programmazione mediata da modelli di linguaggio di grandi dimensioni (LLM) in cui l'umano descrive a parole il comportamento desiderato e l'intelligenza artificiale genera il codice corrispondente. Già consolidatosi come prassi nell'industria del software, la sua adozione in contesti educativi apre nuove prospettive: diminuzione della barriera d'ingresso, rafforzamento del pensiero computazionale e ridefinizione del ruolo del docente. Dopo aver delineato il paradigma e il suo posizionamento rispetto ad altre forme di sviluppo automatico, il lavoro discute benefici, rischi e strategie di mitigazione, proponendo la delega guidata quale cornice pedagogica. Infine, vengono presentate piste di ricerca per valutare empiricamente l'impatto del Vibe Coding nei diversi ordini di scuola.

1. Introduzione

La diffusione di sistemi di intelligenza artificiale generativa ha trasformato le pratiche di sviluppo software. Tra i concetti più recenti spicca il Vibe Coding, termine coniato in un post su X da Andrej Karpathy [1] per descrivere la creazione di prototipi mediante conversazioni iterative con un LLM. A differenza dell'agentic coding (sviluppo interamente guidato da agenti autonomi), in cui un agente quasi autonomo pianifica, genera, testa e rifattorizza il software, il Vibe Coding valorizza l'interazione continua tra umano e macchina: l'IA produce, l'umano orienta e verifica. Il keynote di Sundar Pichai al Google I/O 2025 ha sottolineato come le funzionalità "prompt-centriche" (basate su istruzioni testuali all'IA) diventeranno presto standard negli IDE (Integrated Development Environment) professionali [2]. Portare questo paradigma tra i banchi di scuola significa quindi allineare l'educazione informatica a uno scenario professionale in rapida evoluzione.

2. Il paradigma del Vibe Coding

In una tipica sessione di Vibe Coding lo sviluppatore fornisce istruzioni in linguaggio naturale, talvolta arricchite da frammenti di codice o vincoli formali; l'LLM genera così uno scaffold (struttura di base del progetto) corredato di test e documentazione. Il ciclo si ripete finché l'output soddisfa i requisiti. La centralità del prompt (istruzione che guida l'IA) sposta l'attenzione dagli aspetti sintattici alla capacità di formulare richieste precise, inaugurando quella che alcuni autori definiscono *prompt literacy* (alfabetizzazione alla scrittura di prompt). Ne consegue una trasformazione del ruolo del programmatore: da esecutore di istruzioni a supervisore critico del codice generato.

3. Implicazioni didattiche

Uno dei vantaggi di maggiore impatto è l'abbattimento della barriera sintattica. Ad esempio, uno studente può richiedere “Crea una funzione che ordini una lista di studenti per voto” e l'LLM genera il codice base, permettendo di concentrarsi sulla logica di ordinamento piuttosto che sulla sintassi del linguaggio. In Italia, la piattaforma WeTambara, già adottata da numerosi istituti secondari, integra un tutor LLM che corregge in tempo reale gli errori, permettendo agli studenti di concentrarsi sulla logica algoritmica anziché su parentesi e punti e virgola [3]. A livello internazionale, il programma GitHub Copilot for Teachers offre completamente contestuale e rubriche di revisione automatica, riducendo i tempi di esercitazione e incrementando la motivazione [4]. Tali strumenti promuovono l'apprendimento per scoperta e il *peer programming* (programmazione tra pari), mentre la *prompt engineering* diventa una competenza chiave insieme alla capacità di decomporre problemi complessi.

3.1 Vibe coding e gli obiettivi del Decennio Digitale europeo

Il Digital Decade Policy Programme 2030 individua quattro assi – competenze, infrastrutture, imprese, servizi pubblici – con target quali «l'80 % della popolazione con competenze digitali di base» e «20 milioni di specialisti ICT» entro il 2030 [5]. Il Vibe Coding può fungere da fattore abilitante perché:

- **Accessibilità** – riducendo il carico sintattico, consente anche a studenti con DSA o background non tecnico di realizzare prototipi software;
- **Pensiero computazionale diffuso** – la necessità di scomporre problemi e validare codice rende operativa la raccomandazione europea di integrare il pensiero computazionale nelle discipline STEM.

Le dinamiche sopra descritte si riflettono anche sul senso di autoefficacia percepito dai docenti: uno studio di Corradini e Nardelli evidenzia un miglioramento significativo nelle classi che adottano strumenti IA rispetto a metodologie tradizionali [6].

4. Inserimento di Figure e Tabelle

L'adozione del Vibe Coding comporta criticità che la letteratura e la pratica scolastica stanno iniziando a mappare.

- **Allucinazioni dell'IA e codice non convalidato.** Gli LLM possono produrre frammenti sintatticamente plausibili ma logicamente errati o insicuri, arrivando a “inventare” endpoint API o a omettere la sanificazione degli input, con vulnerabilità quali SQL injection o buffer overflow [8].
- **Manutenzione e debito tecnico.** La qualità eterogenea dei blocchi generati – stili incoerenti, duplicazioni, scarsa documentazione – aumenta il technical debt (debito tecnico) e i costi di evoluzione della code-base [7].
- **Atrofia delle competenze fondamentali.** Un uso acritico dell'IA rischia di indebolire il pensiero computazionale, il debugging manuale e la creatività progettuale, favorendo una dipendenza passiva dallo strumento.
- **Gestione del contesto e prompt quality.** L'efficacia del Vibe Coding dipende dalla chiarezza del prompt e dalla pertinenza del contesto; senza una gestione dinamica, le allucinazioni aumentano con la dimensione della code-base.
- **Costi di calcolo e iterazione.** Raffinamenti successivi moltiplicano il consumo di token (unità di conteggio per i modelli LLM) e di energia, con ricadute sui bilanci scolastici e sull'impronta ecologica.
- **Resistenza culturale e scetticismo.** Parte della comunità docente teme un abbassamento della qualità del software e un affidamento eccessivo a “scorciatoie” automatizzate.
- **Valutazione dell'apprendimento.** Misurare competenze autentiche quando il codice è co-prodotto dall'IA impone di ripensare rubriche e prove pratiche, privilegiando *peer review* (revisione tra pari) e *project-based learning*.

Questi fattori evidenziano la necessità di un approccio bilanciato, come la delega guidata, nella quale l'IA funge da facilitatore ma la responsabilità finale su qualità e sicurezza resta umana.

5. La “delega guidata” come soluzione pedagogica

Il concetto di **delega guidata** è stato sistematizzato nella rassegna di Rossi et al. [9], che distingue tre livelli di maturità – Assistito, Collaborativo, Autonomo – scanditi da checkpoint umani obbligatori. La delega guidata preserva il ruolo formativo dell'errore: gli studenti mantengono la responsabilità di comprendere, validare e migliorare il codice generato, sviluppando spirito critico e competenze di debug. Questo approccio trasforma l'IA da “oracolo infallibile” a “partner di lavoro” che richiede supervisione continua. Cruciale, in tale modello, è l'impiego di **motori SAST/DAST**:

- **SAST (Static Application Security Testing)** – analisi statica del codice «a riposo» per individuare pattern pericolosi prima della distribuzione;
- **DAST (Dynamic Application Security Testing)** – scanner dinamico che rileva vulnerabilità di runtime in ambiente isolato.

Tra gli strumenti più accessibili in ambito educativo figurano OWASP ZAP [10] per l'analisi dinamica e Semgrep [11] per quella statica.

5.1 Implementazione didattica in cinque passi

1. **Definire** criteri di qualità (rubriche, coverage minimo, style guide – linee guida di stile) prima della generazione.
2. **Affidare** all’LLM la scrittura di boiler-plate (codice standard ripetitivo) e test automatici, mantenendo la logica centrale nelle mani degli studenti.
3. **Documentare** l’algoritmo prodotto (commenti di alto livello, descrizione in linguaggio naturale e, dove opportuno, diagrammi di flusso) per supportare la manutenzione futura e la comprensione dei principi computazionali.
4. **Validare** il codice in branch (rami di sviluppo) isolati attraverso SAST/DAST e peer review (revisione tra pari).
5. **Integrare** solo gli artefatti che superano i checkpoint, riducendo il rischio di technical debt (debito tecnico) e allucinazioni.

6. Piste di ricerca futura

L’adozione del Vibe Coding nella didattica solleva interrogativi che richiedono validazione empirica. La letteratura scientifica in questo ambito è ancora limitata, ma la rapida diffusione del paradigma nell’industria rende urgente una ricerca sistematica sui suoi effetti educativi. Servono studi controllati che confrontino classi tradizionali con quelle che adottano Vibe Coding che misurino l’impatto reale su apprendimento e competenze, superando impressioni aneddotiche per fondare su evidenze le scelte didattiche. Altre priorità includono la progettazione di percorsi che integrino gamification e attività unplugged, e la mappatura delle nuove competenze (prompting, refutazione delle allucinazioni) sulle dimensioni del pensiero computazionale.

Riteniamo prioritarie quattro macro-direttrici di indagine:

- **Efficacia didattica.** Confrontare classi che impiegano Vibe Coding con classi tradizionali su indicatori quali apprendimento del pensiero computazionale, motivazione e tempo di completamento dei progetti.
- **Inclusione e accessibilità.** Valutare se l’approccio riduce disparità di genere, background socio-economico o bisogni educativi speciali.
- **Qualità del software prodotto.** Analizzare in modo longitudinale il debito tecnico e le vulnerabilità di codice generato vs. scritto a mano, anche alla luce dell’uso di SAST/DAST.
- **Formazione docenti.** Sviluppare e validare moduli di upskilling che forniscano competenze critiche e operative per un impiego pedagogicamente consapevole dell’IA.

Studi misti, che combinino misure quantitative (test standardizzati, metriche di codice) e qualitative (interviste, osservazioni) offriranno un quadro più completo e faciliteranno la costruzione di best practice condivise.

7. Conclusioni

Il Vibe Coding è ormai prassi industriale: team di Google, Microsoft e piattaforme come Replit lo adottano quotidianamente, e le roadmap dei principali IDE indicano che le funzionalità “prompt-centriche” diventeranno lo standard. Introdurlo nelle scuole significa ridurre il divario di competenze tra aula e professione.

Per conseguire tale obiettivo è imprescindibile una formazione tempestiva e capillare dei docenti. Solo insegnanti adeguatamente preparati potranno guidare gli studenti attraverso i rischi (allucinazioni, debito tecnico, sicurezza) e valorizzarne i benefici (accessibilità, creatività, pensiero computazionale). La delega guidata resta la via più promettente: l'IA amplifica le capacità umane senza sostituirle, mentre monitoraggi empirici continuativi assicureranno che il ritmo del nuovo paradigma trovi un'armonia sostenibile nelle aule di informatica.

Bibliografia

- [1] A. Karpathy. On Vibe Coding, post su X, <https://x.com/karpathy/status/1746654017725935888>, 2024
- [2] S. Pichai. Discorso inaugurale Google I/O, 2025. <https://blog.google/intl/it-it/prodotti/cloud/google-io-2025-dalla-ricerca-alla-realta/>
- [3] WeTambara. Tutor IA per la scuola, <https://wetambara.it> 2025.
- GitHub Education. AI coding tools for beginner and expert coders. <https://github.com/resources/articles/ai/ai-coding-tools>, 2025
- [4] Commissione europea. Digital Decade Policy Programme 2030, Decision (EU) 2022/2481, 2022.
- [5] I. Corradini, E. Nardelli. “Insights into Teachers’ Self-efficacy in Informatics Education”, *The Educational Review*, 9(1), 2025
- [6] J. Smith, L. Garcia. “Assessing Maintainability of AI-Generated Code”, in *Proceedings of the 46th International Conference on Software Engineering (ICSE)*, 2024
- [7] K. Brown, S. Lee. “Risks of Large Language Model Code Generation: An Empirical Study”, *Empirical Software Engineering Journal*, 30(4), 2023.
- [8] L. Rossi, M. Chen, P. Verma. “A Review on Vibe Coding: Fundamentals, State-of-the-art, Challenges and Future Directions”, *Journal of Software Engineering Education*, 12(2), 2025.
- [9] OWASP Foundation. “OWASP Zed Attack Proxy (ZAP) Project”, 2023.
- r2c. Semgrep Documentation, <https://semgrep.dev/docs>, 2025

LABORATORI
PER L'INFANZIA E LA PRIMARIA

ScratchJr: un laboratorio di coding per la scuola dell'infanzia e primaria

Federica Lizzini¹

¹ Università degli Studi dell'Aquila

federica.lizzini@univaq.it

Abstract

Il laboratorio è dedicato all'esplorazione di ScratchJr, un ambiente di programmazione visuale a blocchi pensato appositamente per introdurre i bambini dai 4 agli 8 anni al mondo del coding (plugged) in modo intuitivo e divertente. Questo strumento offre un approccio ludico per sviluppare il pensiero computazionale e la creatività digitale fin dalla scuola dell'infanzia e primaria. Durante l'incontro, esploreremo attivamente le potenzialità di ScratchJr attraverso la realizzazione di un semplice gioco, toccando con mano come questo ambiente possa diventare un prezioso alleato didattico.

Il laboratorio, della durata di 90 minuti, sarà strutturato in modalità interattiva alternando spiegazioni del docente e attività pratiche, in cui i partecipanti affronteranno, individualmente o in piccoli gruppi, le attività che potranno essere poi riproposte ai propri allievi. In particolare, le attività includeranno:

- Introduzione a ScratchJr: Presentazione dell'ambiente di programmazione, delle sue caratteristiche principali e dei vantaggi per l'apprendimento.
- Esplorazione dell'interfaccia: Panoramica degli strumenti e delle diverse sezioni dell'applicazione.
- Attività pratica "Il nostro primo gioco": Realizzazione guidata di un semplice gioco interattivo utilizzando i blocchi di programmazione di ScratchJr. Questa attività permetterà di sperimentare concretamente i concetti di sequenza, ripetizione (cicli) e decisione (selezione ad una via).
- Riflessione didattica: Discussione sulle potenzialità di ScratchJr in relazione agli obiettivi di apprendimento definiti e alle possibili applicazioni in classe. Analisi dei limiti dello strumento.

Destinatari. Il laboratorio è rivolto ai docenti della scuola dell'infanzia e della scuola primaria interessati a integrare il coding e il pensiero computazionale nelle proprie attività didattiche. Non sono richieste competenze pregresse di programmazione.

Obiettivi. Al termine del laboratorio, i partecipanti saranno in grado di:

- Comprendere le potenzialità di ScratchJr come strumento didattico per l'introduzione al coding.
- Conoscere l'interfaccia e le funzionalità di base di ScratchJr.

- Sperimentare la creazione di semplici progetti interattivi.
- Identificare i collegamenti tra le attività proposte con ScratchJr e gli obiettivi di apprendimento relativi a:
 - Ambito algoritmi:
 - ◊ O-P3-A-2. Comprendere che problemi possono essere risolti mediante la loro scomposizione in parti più piccole.
 - Ambito programmazione:
 - ◊ O-P3-P-2. Ordinare correttamente la sequenza di istruzioni.
 - ◊ O-P3-P-3. Utilizzare i cicli per esprimere sinteticamente la ripetizione di una stessa azione un numero prefissato di volte.
 - ◊ O-P3-P-4. Utilizzare la selezione ad una via per prendere decisioni all'interno di programmi semplici.
 - Ambito creatività digitale:
 - ◊ O-P3-R-1. Creare contenuti digitali elementari.
- Valutare i limiti e le opportunità offerte da ScratchJr nel contesto didattico.

Strumenti. Ogni partecipante dovrà portare con sé:

- un PC portatile oppure un tablet;
- l'applicazione ScratchJr già scaricata e installata sul proprio dispositivo. L'applicazione è gratuita e scaricabile da:
 - <https://www.scratchjr.org/>
 - <https://jfo8000.github.io/ScratchJr-Desktop/>
- In alternativa è possibile utilizzare la versione web <https://codejr.org/scratchjr/index.html>

Introduzione al Machine Learning alla scuola primaria

Maria Cristina Carrisi¹, Sara Vergallo^{1,2}, Mirko Marras¹

¹ Dipartimento di Matematica e Informatica, Università degli studi di Cagliari

mariacri.carrisi@unica.it, mirko.marras@unica.it

² Università di Macerata

s.vergallo@unimc.it

Abstract

L'obiettivo di questo laboratorio è presentare agli insegnanti della scuola primaria (e secondaria di primo grado) alcune attività, già sperimentate in classe, per introdurre i fondamenti del machine learning e la sua applicazione ai sistemi di riconoscimento. Ci si concentrerà sull'apprendimento supervisionato basato su alberi decisionali.

La peculiarità delle attività proposte sta nell'esplicito raccordo tra i contenuti di Informatica e le preconoscenze matematiche necessarie per la comprensione degli argomenti. L'informatica, oltre ad essere oggetto dell'insegnamento, diventa una base di realtà che consente di vedere in azione concetti come le frazioni o più astratti quali gli insiemi e le loro operazioni o i connettivi logici. Le attività individuali e di gruppo, esclusivamente unplugged, sono tese a massimizzare il coinvolgimento degli studenti e prevederanno anche la costruzione di semplici artefatti.

Destinatari. Insegnanti di scuola primaria e della secondaria di primo grado. La proposta è tarata sulle conoscenze e competenze attese per studenti di una classe quinta primaria o prima della secondaria di primo grado.

Materiale. Alla pagina <https://github.com/tail-unica/ai-literacy-primary-ed/blob/main/README-IT.md> è possibile trovare il materiale utilizzato durante il laboratorio insieme ad altro materiale.

Bibliografia

- [1] Carrisi, M. C., Marras, M., & Vergallo, S. (2025) A Structured Unplugged Approach for Foundational AI Literacy in Primary Education In: arXiv preprint arXiv:[2505.21398] (<http://arxiv.org/abs/2505.21398>)

Salta, balla e rotola con il coding e l'AI

Luca Basteris¹, Maria Cristina Daperno¹

¹ Liceo Classico e Scientifico Statale "Silvio Pellico – Giuseppe Peano"

luca.basteris@liceocuneo.it cristina.daperno@liceocuneo.it

Abstract

In questo laboratorio di 1 ora e mezza si prevede di utilizzare la scheda programmabile Micro:bit per unire aspetti tradizionali del coding con quelli dell'intelligenza artificiale in particolare legati all'addestramento.

Il laboratorio sarà pratico. Verranno portati trenta micro:bit per permettere ai docenti di sperimentare l'attività. I docenti dovranno avere a disposizione i loro PC portatili. L'attività prevederà una piccola sezione introduttiva in cui si spiegherà cosa si intende per Machine Learning e cosa si intende per "addestramento". I docenti con materiale recuperabile nella loro borsa (fazzoletto, portapenne, borraccia, ecc..) simuleranno il loro peluche sul quale attaccheranno con nastro adesivo il Micro:bit. Nella prima fase mediante il proprio computer si accederà al sito Web micro:bit CreateAI (Figura 1), utilizzando un browser Web Chrome o Edge.

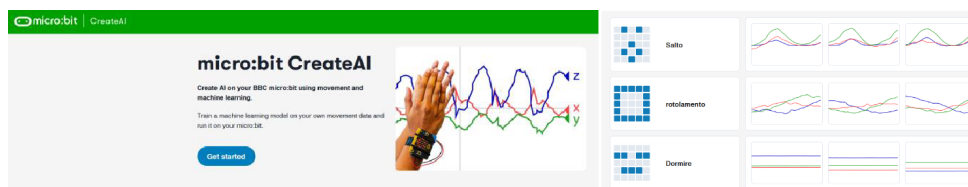


Figura 1: Interfaccia di micro:bit CreateAI

Con questa piattaforma si andrà ad addestrare il modello di Machine Learning, collegando il Micro:bit e utilizzando l'accelerometro presente al suo interno. Successivamente la piattaforma riporta direttamente a MakeCode dove possono integrarsi i vari comandi tradizionali di coding.

Destinatari. Laboratorio per scuole del primo ciclo della durata di 1,5 ore

Materiale. <https://microbit.org/projects/make-it-code-it/ai-storytelling-friend/>

LABORATORI
PER LA SECONDARIA DI PRIMO GRADO

Blockchain@School

Lorenzo Guasti¹, Maria Angela Pellegrino^{2,3}

¹INDIRE, Firenze, Italia

l.guasti.tecnologo@indire.it

²Università degli Studi di Salerno, Italia

mapellegrino@unisa.it

³ Laboratorio Nazionale del CINI - Gruppo di Lavoro "Informatica & Scuola

Abstract

La blockchain è una tecnologia sofisticata, fondata su principi avanzati di crittografia, decentralizzazione e sicurezza informatica. La sua crescente rilevanza in ambito economico, sociale e tecnologico rende sempre più urgente promuovere una cultura digitale diffusa, in cui anche le nuove generazioni – indipendentemente dall'inclinazione o interesse a diventare esperti del settore – siano messe nelle condizioni di comprenderne i concetti fondamentali. Il laboratorio "*Blockchain@School*" nasce proprio con questo obiettivo: avvicinare studenti di ogni ordine e grado al tema della blockchain attraverso un'esperienza ludica, tangibile e collaborativa, sfruttando la metafora del gioco per rendere accessibili e memorabili anche i concetti più complessi.

Descrizione del tema proposto. Durante il laboratorio, i partecipanti faranno esperienza dei concetti fondamentali della tecnologia blockchain, tra cui funzionamento del registro distribuito, meccanismo di crittografia e computazione di funzione di hash per garantire sicurezza, ruolo del consenso decentralizzato, concetti base di immutabilità e integrità dei dati. In termini di competenze, i partecipanti impareranno a simulare in modo cooperativo il funzionamento di una rete blockchain, attraverso un gioco *unplugged* da tavolo; a codificare semplici dati e calcolare manualmente valori di hash con regole condivise; verificare in modo collaborativo la validità dei blocchi e raggiungere il consenso tra nodi della rete; analizzare gli effetti di un attacco alla blockchain e comprendere le strategie di difesa del sistema. In termini di abilità trasversali, i partecipanti impareranno a lavorare in gruppo, potenziare pensiero logico attraverso la risoluzione di problemi reali in contesto gamificato, rafforzare la consapevolezza critica sull'uso delle tecnologie digitali, riflettendo sull'impatto sociale ed etico della blockchain, e stimolare curiosità verso l'innovazione tecnologica e l'apprendimento STEM attraverso esperienze accessibili e coinvolgenti.

Modalità di interazione. "*Blockchain@School*" adotta un approccio laboratoriale, in cui i partecipanti vengono coinvolti attivamente e in prima persona. Il cuore dell'esperienza è un gioco da tavolo unplugged, co-progettato da educatori, esperti di tecnologia e informatici, che simula il funzionamento di una rete blockchain. Durante il laboratorio, i partecipanti assumono il ruolo di nodi della rete, creano blocchi, calcolano hash, affrontano attacchi simulati e comprendono dinamiche come il consenso distribuito e l'immutabilità delle informazioni. L'attività

alterna momenti di gioco, sfide cooperative, e riflessioni guidate, favorendo un apprendimento coinvolgente e significativo.

Destinatari. Il laboratorio è stato progettato principalmente per la scuola secondaria di primo grado. Tuttavia, grazie alla sua struttura modulare e flessibile, può essere facilmente adattato anche per studenti della scuola secondaria di secondo grado, approfondendo le implicazioni tecniche e sociali della blockchain. In passato, lo stesso laboratorio è stato già sperimentato con successo con bambini della primaria e secondaria di primo grado, con studenti di scuola secondaria di secondo grado, e docenti durante i laboratori di DIDACTA.

Materiale (anche post-laboratorio). Per continuare ad esplorare i temi trattati anche dopo lo svolgimento del laboratorio, è disponibile una piattaforma online dedicata: <https://tecnologia.indire.it/blockchain>. Qui è possibile: consultare materiali di supporto alle fasi di gioco, scaricare le versioni cartacee del kit per simulare la blockchain in aula, accedere al modello stampabile in 3D del gioco da tavolo, così da riprodurlo autonomamente.

La crittografia? È un gioco!

Arianna Boldi¹, Sara Capecchi², Iary Davidson², Marco Viola¹

¹ *Dipartimento di Psicologia, Università degli Studi di Torino*

² *Dipartimento di Informatica, Università degli Studi di Torino*

Abstract

Il tema della crittografia permette di introdurre molti concetti di base della disciplina informatica: codifica simbolica, elaborazione dell'informazione, sicurezza dei dati. Consente inoltre collegamenti multidisciplinari (matematica, storia).

In questo laboratorio illustreremo alcune delle attività di introduzione alla crittografia svolte in circa 300 classi del primo ciclo nell'ambito del progetto Diderot di Fondazione CRT¹. Le attività toccano concetti storici, tecnici e culturali legati alla protezione delle informazioni.

Il percorso si apre con un dialogo con la classe, centrato su tre domande guida: che cos'è la crittografia, quando è nata e perché oggi è più complicata. Questo avvio ha lo scopo di attivare le conoscenze pregresse, stimolare la curiosità e mettere in evidenza la continuità storica del bisogno di protezione delle informazioni.

Alunne e alunni scoprono come la crittografia affondi le proprie radici in tempi antichissimi. Si parte dai geroglifici non standard usati nell'Antico Egitto, dalle tavolette cifrate della Mesopotamia e dalla scitila spartana, passando per esempi più noti come il codice Morse o la scrittura speculare di Leonardo da Vinci. Si affrontano inoltre le nozioni di crittoanalisi e attacco a forza bruta, evidenziando come l'evoluzione tecnologica, in particolare l'avvento dei computer, abbia reso necessarie tecniche sempre più sofisticate per garantire la sicurezza delle comunicazioni. In questo quadro si colloca anche la riflessione sull'informatica come scienza dell'elaborazione automatica dell'informazione.

Il cuore del percorso è costituito dalle attività laboratoriali come la costruzione di codici personali basati su simboli o schemi, come nel gioco Peg Code, nella creazione di sistemi inventati dagli alunni e nei giochi Secret Code e Enigma. Questi giochi, prodotti dall'azienda Quercetti in collaborazione con il Dipartimento di Informatica dell'Università di Torino, rappresentano strumenti didattici mirati che uniscono la dimensione ludica alla riflessione sui concetti.

La proposta didattica è attentamente modulata per rispondere alle necessità delle diverse fasce scolastiche, mantenendo in comune l'impostazione laboratoriale e il focus sull'apprendimento attraverso l'esperienza.

Modalità di interazione e materiale. Durante il laboratorio descriveremo la progettazione dei percorsi e faremo provare alle/ai partecipanti alcune delle attività. Il materiale verrà fornito dalla moderatrice, si richiede solo carta e penna per lo svolgimento di alcuni giochi ed esercizi.

¹ <https://www.fondazione crt.it/bandi-progetti/progetto-diderot/>.

Destinatari. primaria e secondaria di I grado.

Materiale. Link al materiale consultabile dopo il laboratorio: https://drive.google.com/drive/folders/184QXhCzZs_q5rrD6M4iLxa5EKX-jvZ__.

Approfondimento su tematiche e metodologie del percorso presentato

Alunne e alunni scoprono come la crittografia affondi le proprie radici in tempi antichissimi. Si parte dai geroglifici non standard usati nell'Antico Egitto, dalle tavolette cifrate della Mesopotamia e dalla scitila spartana, passando per esempi più noti ai ragazzi come il codice Morse o la scrittura speculare di Leonardo da Vinci.

Un elemento caratterizzante è la costante riflessione metacognitiva che accompagna le attività. Al termine di ogni laboratorio si discute insieme: quali regole hanno funzionato? Perché un codice era decifrabile o meno? Come si collega ciò che abbiamo fatto al mondo reale e alle tecnologie che usiamo ogni giorno?

La proposta didattica è attentamente modulata per rispondere alle capacità cognitive delle diverse fasce scolastiche, mantenendo in comune l'impostazione laboratoriale e il focus sull'apprendimento attraverso l'esperienza.

Scuola primaria (classi terze e quarte)

In questa fascia si privilegiano attività ludiche e manuali. Si parte da esempi vicini all'immaginario infantile, come l'alfabeto farfallino o i messaggi segreti nei giochi tra amici. Le attività prevedono:

- uso di chiodini o simboli colorati per costruire codici visivi;
- semplici cifrari come quello di Cesare, resi concreti attraverso la cifratura dei nomi degli alunni;
- esercizi per inventare simboli associati alle lettere;
- riflessioni su segreti noti ai bambini, come la ricetta della Coca-Cola, per introdurre l'idea di segreto commerciale.

Scuola primaria (quinta) e secondaria di primo grado (classi prime)

In questa fascia si alza il livello di astrazione e complessità. Si lavora su:

- costruzione di codici con regole condivise e scritte;
- riflessione sull'evoluzione storica dei cifrari e sulla relazione con lo sviluppo delle capacità umane (lettura, scrittura, calcolo);
- introduzione di cifrari più complessi come Enigma e discussione sul ruolo dei computer nella crittoanalisi;
- prime analisi sistematiche delle frequenze e simulazioni di attacchi a forza bruta.

Secondaria di primo grado (classi seconde e terze)

Le attività si concentrano su sistemi più complessi e su concetti vicini all'informatica:

- simulazione di funzionamento della macchina Enigma;
- giochi come Secret Code, che riproducono dinamiche di crittoanalisi con tentativi limitati e indizi progressivi, richiamando i meccanismi di sicurezza di PIN, carte bancomat e dispositivi digitali;
- discussioni sul rapporto tra crittografia, sicurezza e tecnologia, con attenzione ai limiti e alle vulnerabilità dei sistemi informati.

LABORATORI
PER LA SECONDARIA DI SECONDO GRADO

Dalle foglie alle radici: imparare il Debugging dalle sue componenti fondamentali

Gabriele Pozzan¹, Tullio Vardanega¹

¹ Università degli studi di Padova

gabriele.pozzan@phd.unipd.it tullio.vardanega@unipd.it

Abstract

Questo laboratorio presenta, tramite discussione e attività pratiche, una scomposizione del *Debugging* in abilità fondamentali ispirata dalla letteratura scientifica sul tema del carico cognitivo (*Cognitive Load Theory*). È indirizzato principalmente a insegnanti di Informatica a partire dalla scuola secondaria. I concetti trattati, però, sono di carattere generale e quindi sicuramente rilevanti per ogni grado di insegnamento. Il laboratorio è svolto in dialogo e non richiede dispositivi. Chi partecipa può portare esercizi di *Debugging* utilizzati in classe per discuterli alla luce dei temi del laboratorio.

La capacità di individuare e correggere errori di programmazione, chiamata *Debugging*, è una componente fondamentale del pensiero Informatico. Inoltre, è una sfida quotidiana non solo per le persone che dell'Informatica fanno la propria professione, ma anche per studentesse e studenti che imparano i rudimenti della programmazione e sviluppano il pensiero computazionale. Il *Debugging* richiede il coordinamento di diverse sottoabilità come:

1. comprendere il problema che il programma intende risolvere (i requisiti del programma);
2. comprendere come (con quale algoritmo) il programma intende risolverlo;
3. leggere e interpretare il codice eseguibile;
4. capire come il codice eseguibile si relaziona con i requisiti;
5. individuare dove il programma devia dai requisiti;
6. modificare il programma per farlo convergere verso i requisiti.

La teoria scientifica del “carico cognitivo” suggerisce che è significativamente più efficace imparare una abilità complessa come il *Debugging* a partire dall'apprendimento delle sue sottoabilità (riassunte sopra). Questo laboratorio discute un modello che scompone il *Debugging* in sottoabilità, esplicitando le loro relazioni e dipendenze.

Scansione temporale delle attività. Il laboratorio è diviso in tre segmenti successivi, della durata indicativa di 30 minuti ciascuno.

- **Introduzione:** presentazione della scomposizione in sottoabilità del *Debugging* e dei potenziali utilizzi
- **Attività pratica:** esercizi che affrontano in modo relativamente isolato specifiche sottoabilità del *Debugging*. Questi esercizi vengono svolti in modalità dialogica con il moderatore.

- **Discussione:** discussione collettiva sull'utilità potenziale della scomposizione in sottoabilità per la progettazione di percorsi educativi a medio-lungo termine. Analisi di attività di *Debugging* effettivamente utilizzate in classe dalle/dai partecipanti alla luce della scomposizione.

Materiale. <https://github.com/cornacchia/debugging-workshop-itadinfo-2025>

Laboratorio “La Natura nel Computer”

Claudio Mirolo¹

¹ *Dipartimento di Scienze Matematiche, Informatiche e Fisiche dell'Università di Udine*

claudio.mirolo@uniud.it

Abstract

Questo laboratorio, che si rivolge a insegnanti del triennio della secondaria di II grado, propone l'informatica e la programmazione come strumenti di indagine scientifica, mettendone in rilievo la valenza formativa interdisciplinare. Il contesto considerato consente, in particolare, di vedere all'opera i concetti principali della programmazione orientata agli oggetti, e nello stesso tempo introduce una prospettiva interessante sul metodo scientifico.

Il percorso proposto è stato sviluppato nell'ambito di diverse edizioni del Piano Lauree Scientifiche e ha visto coinvolte classi III, IV e V di licei e istituti tecnici. Le attività si articolano attorno allo sviluppo e alla valutazione critica di modelli per spiegare, attraverso simulazioni, due fenomeni osservati in natura: la fillostasi del girasole e la sincronizzazione di alcune specie di lucciole. Dopo aver chiarito gli obiettivi (domanda scientifica a cui si cerca di rispondere), i presupposti (meccanismi che si presumono in atto) e le semplificazioni che si intende introdurre (processo di astrazione), si procede alla messa a punto concettuale e alla codifica di modelli a partire da programmi preimpostati. L'esperienza si conclude con una discussione critica dei risultati, individuando limitazioni e possibilità di ulteriori sviluppi dei modelli.

Le idee alla base della programmazione orientata agli oggetti hanno avuto origine negli anni '60 del '900 proprio nell'ambito della simulazione scientifica (*Simula*), perciò un tale contesto si presta molto bene a svilupparne e a comprenderne i concetti. Inoltre, la creazione di immagini e animazioni in un ambiente che ne facilita la programmazione è in genere motivante. Il livello di impegno richiesto nell'affrontare la programmazione e il bilanciamento rispetto agli obiettivi interdisciplinari possono essere adattati per rispondere alle esigenze di contesti diversi.

Tematiche di informatica e interdisciplinari

- *Concetti base di programmazione orientata agli oggetti*: incapsulamento, distinzione classe vs. oggetto e concetti correlati, coesione/disaccoppiamento, riutilizzo – inoltre, laddove siano proponibili, eredità e polimorfismo.
- *Elementi di programmazione parallela*: thread, semplici modalità (non problematiche) di sincronizzazione, sistemi interattivi.
- *Approccio scientifico*: Modelli interpretativi dei fenomeni, processi di astrazione, corroborazione e confutazione di ipotesi, comportamenti emergenti, discussione critica e riconoscimento dei limiti.
- *Chiavi di lettura matematica*: Sezione aurea e interpretazione matematica del relativo ruolo nella fillostasi, modelli matematici e modelli computazionali.

Modalità di partecipazione. Nel corso del laboratorio, della durata di circa un'ora e mezza, i partecipanti saranno invitati a prendere parte attiva alla sperimentazione, e a tal fine si consiglia l'utilizzo di un dispositivo di calcolo personale in cui possa essere installato l'ambiente di programmazione *Processing* (<https://processing.org>).

Materiale. Per orientarsi sui contenuti delle attività proposte si può consultare il materiale disponibile online all'indirizzo: http://nid.dimi.uniud.it/projects/pls.html#natura_nel_computer

Cos'è e come funziona un Sistema Operativo? Scopriamolo con il ristorante Brachetti

Renzo Davoli¹

¹ *Alma Mater Studiorum, Università di Bologna DISI: Dipartimento di Informatica Scienza e Ingegneria
renzo@cs.unibo.it*

Abstract

Ci sono grandi fraintendimenti sui concetti relativi ai Sistemi Operativi e sul concetto stesso di Sistemi Operativo. Spesso vengono confusi con distribuzioni di software. C'è addirittura chi pensa che fra le caratteristiche di un sistema Operativo si debbano annoverare la posizione dei menù o l'icona del cestino presenti nell'interfaccia grafica.

Un cuoco trasformista ci conduce attraverso un gioco di teatralizzazione per comprendere cosa sia, quali siano i concetti fondamentali e come funzioni un moderno sistema operativo multi-tasking.

La lezione si può definire unplugged: non serve nessuno strumento tecnologico. Servono fantasia e voglia di partecipare a una esperienza didattica non convenzionale. Rispetto alle metodologie unplugged classiche (e.g. CS unplugged) questa esperienza differisce: perché rivolta a studenti più grandi (scuole secondarie superiori), perché vuole consentire non solo una comprensione intuitiva, ma anche creare un punto di riferimento per approfondire con lezioni più convenzionali lo studio dei Sistemi Operativi, e conseguentemente perché cerca una corrispondenza più puntuale fra i singoli personaggi, oggetti e azioni e concetti chiave della materia.

Da cinque anni “il ristorante Brachetti” costituisce la lezione introduttiva del corso di Sistemi Operativi al corso di studi in Informatica a Bologna. Oggetti, personaggi, azioni sono metafore di definizioni e concetti generali, di architettura degli elaboratori e di sistemi operativi. quali per esempio:

1. Concetti generali: programma, processo, dati,
2. Architettura: processore, modo kernel/modo user, controller di I/O, interrupt, stato del dispositivo, interrupt, risorse, memoria,
3. Sistemi Operativi: descrittore di processo, system call, protezione, scheduling di processore e di I/O, multitasking, conferma dell'interrupt,
4. ... e ovviamente la definizione stessa di sistemi operativi.

Questo metodo consente agli studenti di acquisire una idea intuitiva del funzionamento di un sistema operativo, dei suoi componenti e della interazione con l'hardware, idea che può essere poi utilizzata come canovaccio per le successive spiegazioni dettagliate e ricche di dettagli tecnologici. Spesso chiamo questo approccio play-see-do: gli studenti prima “giocano” con la teatralizzazione che consente poi di “vedere” e capire meglio la teoria ed il funzionamento dei

sistemi operativi abbinando di volta in volta i concetti a personaggi o azioni della teatralizzazione, infine per corsi altamente professionalizzanti gli studenti possono “fare”: analizzare il codice di un sistema operativo reale o scriverne uno didattico. Nascosto fra gli statement del codice troveranno sempre Brachetti che fa capolino.

La presentazione dell’esperienza comprende in 90 minuti:

1. 5 minuti per la presentazione dell’attività.
2. circa 60 minuti verranno utilizzati per il “gioco”
3. gli ultimi 25 servono per un confronto con i discenti al fine di svelare tutte le metafore presentate nel nostro ristorante.

Destinatari. Laboratorio proposto per studenti della scuola secondaria superiore. Esperienza per Liceo Scienze Applicate o lezione iniziale sui Sistemi Operativi per corso “Sistemi e Reti” Ist. Tecnico Tecnologico - Informatica.

Prerequisiti. Conoscenze di base di architettura degli elaboratori (modello Von Neumann, processore, memoria, bus, controller unità periferiche).

Materiale. <https://www.cs.unibo.it/renzo/brachetti/ristorante.pdf>

Cosa si cela dietro ai pixel: un gioco per creare un gioco

Calogero Carlino¹, Luca Forlizzini¹

¹ Dipartimento di Ingegneria e Scienze dell'Informazione e Matematica,

Università degli Studi dell'Aquila

calogero.carlino@student.univaq.it luca.forlizzini@univaq.it

Abstract

Insegnare a programmare realizzando videogiochi è un classico della didattica dell'informatica, fondato sull'interesse spontaneo che ragazze e ragazzi in età scolare spesso nutrono per tali forme di intrattenimento, parte del loro vissuto quotidiano. Tuttavia, creare videogiochi con linguaggi di programmazione general purpose in contesti scolastici, trova ostacolo nell'elevata complessità dei prodotti videoludici contemporanei, che tende da un lato a scoraggiare anche gli studenti più motivati e dall'altro a ridurre l'attrattiva di software più semplici da realizzare ma meno appaganti sotto il profilo del coinvolgimento ludico. Il laboratorio mira a superare questo problema attraverso attività di analisi, progettazione e tinkering di videogiochi concepiti per supportare la didattica dello sviluppo software a livello scolastico, offrendo al contempo un'esperienza coinvolgente e gratificante per il giocatore. Le attività si avvalgono di tecnologie specifiche come game-engine e librerie, e sono modulabili in base alle competenze pregresse dei partecipanti. La versione proposta punta ad amplificare il coinvolgimento aggiungendo un meta-livello ludico: il laboratorio si sostanzia nel prendere parte ad un videogioco multi-player nel quale i partecipanti si sfidano nel realizzare e modificare parti di un videogioco.

Destinatari. Il laboratorio è proposto per gli insegnanti della scuola secondaria di secondo grado. Non richiede competenze pregresse sulla programmazione o sullo sviluppo software.

Obiettivi di apprendimento:

1. Presentare l'informatica come disciplina abilitante per esprimere la propria creatività
2. Avvicinarsi al mondo dello sviluppo software e in particolare a programmazione e rappresentazione dell'informazione
3. Utilizzare ambienti di programmazione a fini espressivi (es. animazioni, tracce sonore, giochi, ...)¹
4. Sviluppare le competenze di problem solving e logica

Strumentazione necessaria. Smartphone, tablet/iPad o laptop personale con connessione internet per giocare in lobby² con gli altri partecipanti.

¹ Obiettivo O-S-R-1 della Proposta di Indicazioni Nazionali per l'insegnamento dell'Informatica nella Scuola <https://www.consortio-cini.it/index.php/it/component/attachments/download/745>

² Una stanza virtuale che permette ai giocatori di sfidarsi tra di loro.

Attività svolte. Il laboratorio è progettato per svolgersi in una sessione di 90 minuti in cui i partecipanti giocano collettivamente, tutti contro tutti e ognuno sul proprio dispositivo, e si sfidano a scrivere del codice, corretto o sbagliato che sia, guardando in tempo reale l'effetto che quest'ultimo ha all'interno del videogioco e accumulando più punti possibili. Due degli autori fungeranno da facilitatori.

Sentiamo l'AI attraverso Musica e Suoni

Giorgio Delzanno¹, Giovanna Guerrini¹, Guido Vallarino^{1,2}

¹ Università degli Studi di Genova, Italy

² IIT Genova

{giorgiodelzanno, giovanna.guerrini, guido.vallarino}@unige.it

Abstract

Il tema è la collaborazione tra musicisti ed intelligenza artificiale sia nell'ambito della formazione (capire concetti e potenzialità della intelligenza artificiale attraverso suoni e musica) sia nell'ambito creativo (uso dell'intelligenza artificiale nella produzione musicale), senza dimenticare aspetti etici legati alla produzione di brani musicali attraverso AI generativa come il recente caso del gruppo Velvet Sundown, popolare su Spotify. Il laboratorio nasce dalla collaborazione tra docenti del corso di laurea di informatica dell'Università di Genova e un assegnista appassionato di musica della stessa Università e dell'IIT di Genova. L'attività è in linea con il laboratorio *Ascoltiamo gli errori più comuni nella programmazione concorrente con Sonic Pi* proposto a ITADINFO 2024 [2].

Obiettivo formativo. L'obiettivo del laboratorio è quello di provare ad introdurre alcuni concetti di base dell'apprendimento automatico e dell'AI generativa attraverso suoni, segnali, strumenti e composizione musicale, con l'intento quindi di “sentire” l'AI secondo la metafora applicata alla programmazione concorrente in [1, 3].

Formato. La prima parte del laboratorio introduce i concetti di base dell'apprendimento automatico usando suoni, strumenti e segnali. Più precisamente, dopo una breve introduzione ad algoritmi di classificazione supervisionata, i partecipanti verranno coinvolti in un esperimento di allenamento di modelli in grado di riconoscere voci, strumenti musicali, segnali che possono caratterizzare un ritmo musicale. Per questa parte di laboratorio verranno usati Teachable Machine [1] e ml-machine [2]. La seconda parte del laboratorio è focalizzata sull'applicazione dell'apprendimento automatico per estrarre stem di brani musicali partendo da template predisposti tramite notebook Colab in Python e librerie comunemente usate negli AI splitter (es. spleeter). In questa parte i partecipanti proveranno ad assemblare nuovi brani partendo da una libreria di stem ragionando quindi sui formati di questo tipo di dati multimediali. La terza parte del laboratorio è dedicata all'introduzione dei concetti di base dell'AI generativa e ad esempi di generazione di brani musicali partendo da testo. I partecipanti verranno anche coinvolti in vari esempi di “Test di Turing” come spunto per discutere potenzialità, rischi e aspetti etici delle applicazioni dell'AI generativa nel campo della composizione musicale.

Prerequisiti. Nessuno.

Livello di difficoltà. Introduttivo.

Bibliografia

- [1] Giorgio Delzanno, Giovanna Guerrin, and Daniele Traversaro. Concurrency Education with Sonic Pi: “Hear” and “Play” Mistakes and Misconceptions in Multithreaded Programs. Under review, available on request. Technical report, DIBRIS, University of Genoa, July 2023.
- [2] Giorgio Delzanno, Giovanna Guerrini, and Daniele Traversaro. Ascoltiamo gli errori più comuni nella programmazione concorrente con Sonic Pi. In Atti del Convegno ITADINFO 2024, 2024.
- [3] Daniele Traversaro, Giovanna Guerrini, and Giorgio Delzanno. Sonic Pi for TBL teaching units in an introductory programming course. In Adjunct Publication of the 28th ACM Conference on User Modeling, Adaptation and Personalization, UMAP '20 Adjunct, page 143–150, New York, NY, USA, 2020. Association for Computing Machinery.

POSTER

PER LA PRIMARIA

La rappresentazione del dato “Immagine”: progettazione e sperimentazione di un percorso per il primo ciclo scolastico

Annunziata Marra¹

¹ Dipartimento di Informatica – Università di Torino

annunziata.marra@unito.it

Abstract

In questo lavoro si descrivono la progettazione, lo sviluppo e la sperimentazione di un percorso didattico pensato per studentesse e studenti frequentanti la classe quinta della scuola primaria. L’obiettivo era quello di introdurli/e alla rappresentazione digitale del tipo di dato “immagine”. La progettazione del percorso è stata, in parte, eseguita con un’insegnante curricolare per mantenere un livello adeguato ai discenti e con un’insegnante di sostegno per assicurarsi una didattica inclusiva. I feedback e i risultati ottenuti da un questionario anonimo compilato dalle studentesse e dagli studenti dopo lo svolgimento di questo percorso hanno dimostrato una notevole soddisfazione che avvalorava l’efficacia del metodo didattico intrapreso per l’insegnamento di concetti complessi nell’ambito dell’elaborazione automatica dei dati.

1. Introduzione

Nella maggior parte dei contesti scolastici si fa ancora confusione sull’insegnamento dell’informatica intesa come disciplina scientifica, combinando gli aspetti tecnici – legati alle competenze digitali – e quelli scientifici [8]. Risulta necessario che studenti/studentesse acquisiscano competenze non solo sugli aspetti generali della materia, ma anche sui principi e sulle applicazioni per modellare e interpretare le informazioni.

Come proclamato sul Manifesto di Vienna per l’Umanesimo Digitale: “L’educazione all’informatica e al suo impatto sociale devono iniziare il prima possibile. Gli studenti dovrebbero imparare a unire le competenze informatiche con la consapevolezza delle questioni etiche e sociali in gioco” [12]. In Italia il laboratorio Informatica e Scuola del Consorzio Interuniversitario Nazionale per l’Informatica (CINI) ha emanato una Proposta di Indicazioni Nazionali per l’insegnamento dell’Informatica nella Scuola [6] con lo scopo di offrire il suo contributo riguardo la formazione all’informatica come disciplina scientifica nella scuola.

Alla luce del rapporto stilato da *Informatics for all* [4] viene riconosciuta la crescente importanza della grafica computerizzata descritta come “generazione e manipolazione informatica delle immagini”. Questo articolo descrive la progettazione, realizzazione ed esperienza di un intervento di due lezioni per l’insegnamento della rappresentazione del dato *immagine* seguendo un approccio costruttivista attraverso un apprendimento attivo [1] e partecipativo.

2. La progettazione del percorso didattico

Il tema dell'intervento didattico descritto in questo lavoro è la rappresentazione delle immagini ed è rivolto a studenti/studentesse del primo ciclo scolastico.

Per la progettazione delle attività abbiamo seguito i principi del *participatory design* [2] coinvolgendo sia l'insegnante prevalente sia l'insegnante di sostegno per favorire una didattica inclusiva e una corretta gestione della classe. Sulla base del documento inerente alla Proposta di Indicazioni Nazionali per l'insegnamento dell'Informatica nella Scuola [6] gli obiettivi di apprendimento attesi rilevanti per l'intervento proposto e selezionati tra quelli relativi al termine della Scuola Primaria sono:

- Ambito dati e informazione:
 - O-P5-D-1: Utilizzare combinazioni di simboli per rappresentare informazioni familiari complesse;
 - O-P5-D-2: Utilizzare simboli per rappresentare semplici informazioni strutturate.
- Ambito consapevolezza digitale:
 - O-P3-N-1: Riconoscere l'uso dell'informatica e delle sue tecnologie nella vita comune.

Il percorso è stato esposto in due lezioni, ciascuna costituita da una parte introduttiva seguita da esercizi, simulazioni interattive e attività unplugged.

2.1 Lezione 1

Obiettivi:

- effettuare collegamenti con il concetto di informatica nell'ambito della vita quotidiana;
- comprendere cosa rappresenta un'immagine in bianco e nero a partire dal codice binario;
- svolgere un'attività di codifica/decodifica.

Nello specifico, durante la prima parte si è introdotta la definizione di informatica come scienza che studia l'elaborazione automatica dell'informazione [7] (Figura 1a), in particolare la riflessione si è concentrata su come l'informazione viene rappresentata e modellata attraverso dati e come i dati possono essere elaborati e trasformati (Figura 1b).



Figura 1: Divulgazione

Facendo collegamenti con aspetti di vita quotidiana (la passione per la fotografia, i videogiochi) è stato introdotto il concetto di immagine e, di conseguenza, codifica binaria, immagini vettoriali e bitmap, pixel.

La spiegazione non è stata totalmente frontale bensì è stato chiesto ai/alle partecipanti di interagire, facendo collegamenti con conoscenze già acquisite.

Durante la presentazione dei concetti le slide di accompagnamento sono servite per discutere due formati di rappresentazione delle immagini, evidenziando differenze, vantaggi e svantaggi (Figura 2).



Figura 2: Immagini vettoriali (a) e bitmap (b)

Per introdurre la rappresentazione delle immagini e la loro codifica all’interno di un elaboratore, in base al livello scolastico e coerentemente con obiettivi di apprendimento [6], ci si è riferiti alle immagini bitmap su cui sono state proposte le attività. Si è precisato che il calcolatore è solo lo strumento che rappresenta le informazioni mediante cifre, simboli, numeri, invitando gli/le studenti/studentesse a pensare cosa risiede dietro le immagini visibili sullo schermo.

È stata così introdotta la codifica binaria concentrandosi sulle immagini in bianco e nero.

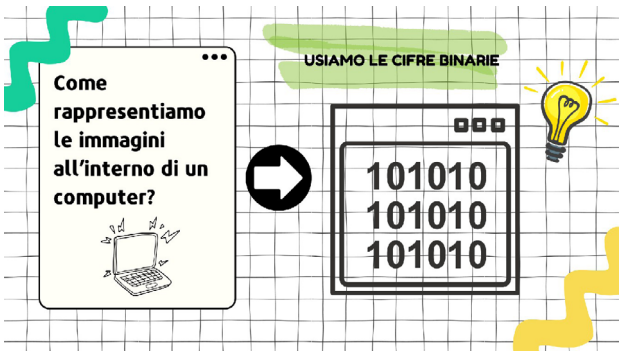


Figura 3: Cifre binarie per rappresentare le immagini

Dopo una prima spiegazione, sono state fornite istruzioni agli/alle studenti/studentesse e, utilizzando la LIM, si è svolto un primo esercizio di codifica binaria seguito da due attività di CS Unplugged [5].

Attività. La prima attività è stata fatta con l'obiettivo di rappresentare i pixel sullo schermo, utilizzando fogli con griglie prestampate e cifre binarie al loro interno, definendo una convenzione sul colore da assegnare a ciascuna cifra per poi svelare un'immagine.

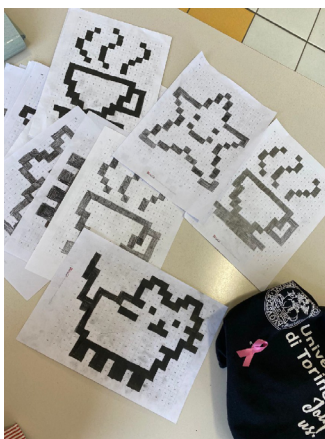


Figura 4: Risultato della prima attività

Nella seconda attività è stata svolta una codifica/decodifica. In questo caso la griglia prestampata sui fogli forniti era vuota. Con una dinamica di gioco [10] è stato proposto di lavorare in squadra così che un/una partecipante potesse creare un codice binario riempiendo i quadratini e il/la compagno/a potesse poi decodificarlo comunicando il risultato atteso.

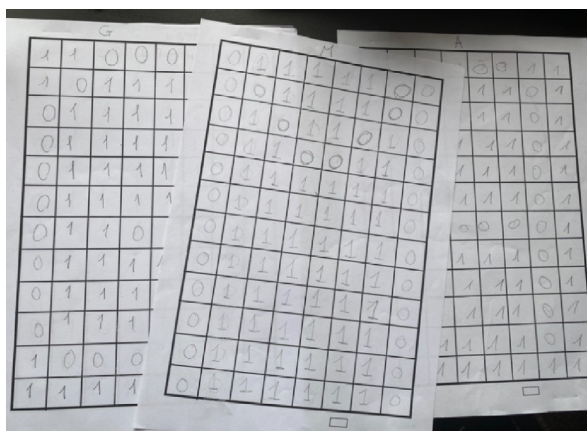


Figura 5: Risultato della seconda attività

Dato l'uso frequente delle parole “scacchiera, quadratino” è stato definito il concetto di Pixel, scomponendo la parola alla lavagna ed evidenziando la presenza delle parole “Picture Element”.

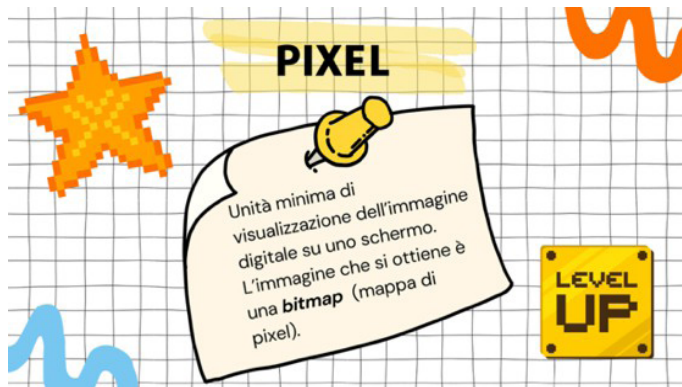


Figura 6: Definizione di Pixel

2.2 Lezione 2

Obiettivi:

- avere consapevolezza sui colori che influiscono sulla resa delle immagini a colori;
- comprendere il concetto di risoluzione.

Durante questa lezione è stato approfondito l'argomento della rappresentazione delle immagini a colori quindi la codifica del dato colore, presentando il modello colorimetrico RGB (Red, Green, Blue) dal momento che ciascun pixel è colorato sulla base di tre colori primari sommati in diverse quantità.

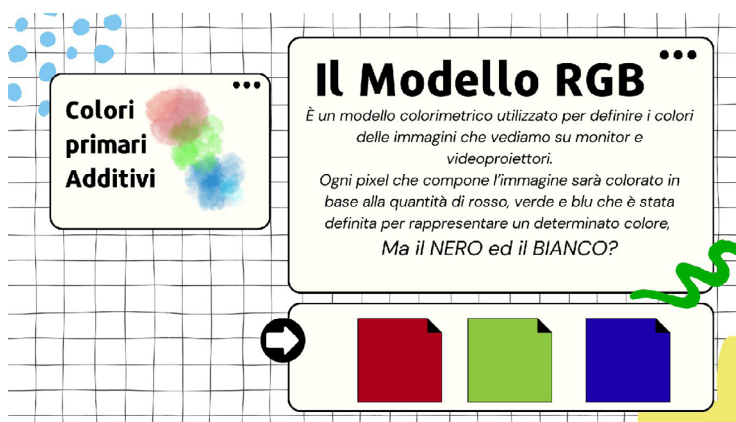


Figura 7: Presentazione del modello RGB

Per applicare quanto appreso è stata proposta una simulazione interattiva [9] in cui veniva mostrato come il colore risultante dipendesse dai singoli contributi dei tre colori.

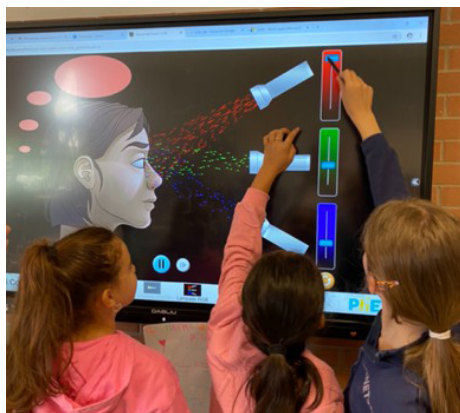


Figura 8: Simulazione interattiva “Visione dei Colori”

Si è poi passati al concetto di risoluzione delle immagini. In questo caso si è preferito esporre l'argomento a partire da esempi (Figura 9) e da un'attività selezionata dal sito InformaticaSenzaPC [3], facendo sperimentare ai/alle partecipanti la necessità di aumentare la definizione, per poi giungere alla definizione formale del concetto.



Figura 9: Confronto tra due immagini con numero e dimensione di pixel differenti

Attività. Per lo svolgimento dell'attività proposta è stato consegnato un foglio contenente tre griglie e, avendo già acquisito le competenze sulla codifica binaria, il codice è stato fornito alla lavagna.

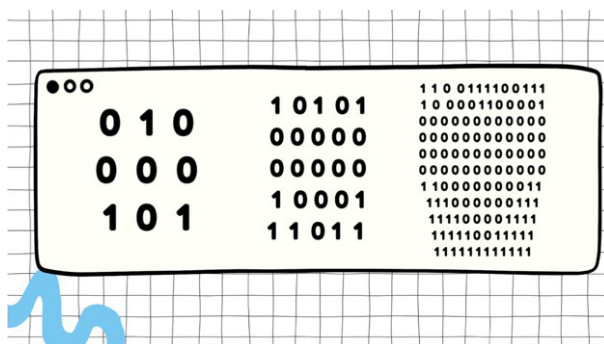


Figura 10: Attività sulla risoluzione delle immagini

L'obiettivo era quello di comprendere il concetto di risoluzione tramite la rappresentazione di un cuore a vari livelli di dettaglio (Figura 11).

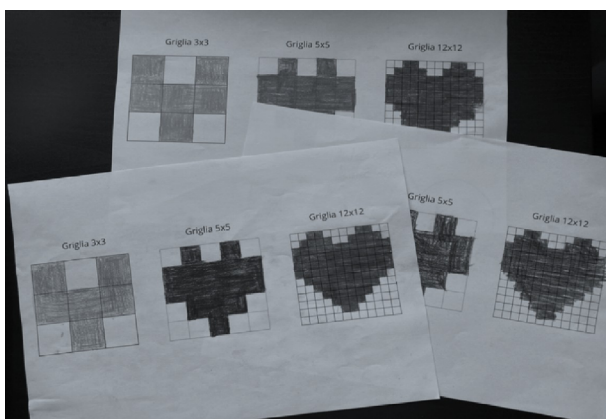


Figura 11: Risultati dell'attività sulla risoluzione

Le tre rappresentazioni sono state create con lo scopo di dimostrare come la quantità e la dimensione dei pixel possano influire sulla resa di un'immagine. Per ciascuna griglia sono state poste delle domande per sviluppare negli/nelle studenti/studentesse un pensiero critico che li/le conducesse al concetto di risoluzione.

Al termine dell'attività è stata formalizzata la definizione (Figura 12).



Figura 12: Definizione di Risoluzione

Il percorso è stato concluso con riflessioni riguardanti i concetti affrontati ed eventuali dubbi.

3. La sperimentazione e valutazione delle attività

La sperimentazione è avvenuta in una classe quinta di scuola primaria composta da 22 alunni/e. Durante ciascuna lezione le fasi seguite sono state: teoria, consegna, svolgimento, discussione, conclusione. Per promuovere una didattica inclusiva la collaborazione con l'insegnante di sostegno ha permesso di:

- integrare momenti di peering education per il ripasso guidato da un pari;
- usare come misura compensativa una mascherina guida per la lettura riga per riga della griglia e, come misura dispensativa, un supporto cartaceo con codice binario prestampato per evitare il ricorso alla lavagna.

I momenti dedicati alle simulazioni (Figura 8) hanno permesso di far acquisire ulteriori conoscenze prima della relativa spiegazione teorica.

La valutazione degli esiti si è basata:

- sui feedback ricevuti dalle due insegnanti;
- sulla compilazione ex post di un questionario (Figura 13) redatto in modo da raccogliere le opinioni degli/delle studenti/studentesse su attività svolte, modalità adottate e argomenti proposti.

L'analisi delle risposte è stata riportata su un grafico (Figura 14) dove sono state inserite:

1. sull'asse delle ascisse le prime cinque domande;
2. sull'asse delle ordinate la frequenza delle risposte per ciascuna domanda.

Nella progettazione del percorso e della valutazione tramite questionari sono state seguite le linee guida della Child Protection Policy emanata dall'Università di Torino [11].

1

Classe

QUESTIONARIO FINALE

Di seguito troverai alcune domande che riguardano le idee che ti sei fatta/o sui compiti che hai svolto.

Prova a dare una risposta a tutte le domande senza pensarci troppo. Ricorda, non ci sono risposte giuste o sbagliate ma ogni risposta che darai andrà bene.

Buon lavoro!

2

1. Prima di iniziare queste attività, quanto pensavi di essere capace a imparare l'informatica? (metti una crocetta)

Per niente	Poco	Così e così	Abbastanza	Molto

3

2. Adesso che hai completato queste attività, quanto pensi di essere capace a imparare l'informatica? (metti una crocetta)

Per niente	Poco	Così e così	Abbastanza	Molto

4

3. Quanto è stato facile il programma delle lezioni per te? (metti una crocetta)

Per niente	Poco	Così e così	Abbastanza	Molto

5

4. Il modo in cui la docente ha spiegato durante le lezioni ti ha aiutato/a a comprendere meglio l'argomento? (metti una crocetta)

Per niente	Poco	Così e così	Abbastanza	Molto

6

5. Nel complesso, è stato semplice seguire le lezioni? (metti una crocetta)

Per niente	Poco	Così e così	Abbastanza	Molto

7

6. Scrivi due cose che ti sono piaciute di questi due giorni trascorsi insieme.

1. _____

2. _____

8

7. Quale argomento ti è piaciuto di più? (metti una crocetta)

☐ Il Modello colore RGB

☐ L'attività sulla rappresentazione delle immagini con le cifre binarie

☐ L'attività sulla risoluzione delle immagini

9

8. Quante stelle daresti a queste attività? (metti una crocetta)

★★★★★

Figura 13: Questionario finale

4. Conclusioni

Dall'analisi dei dati (Figura 14 e Figura 15) è emerso che:

- alcuni/e studenti/studentesse non avevano consapevolezza delle loro abilità, dubitando della propria capacità di imparare l'informatica;
- dopo questo percorso la maggior parte ha maturato la consapevolezza di riuscire ad approcciarsi a questa disciplina;
- l'approccio usato è servito alla comprensione degli argomenti, oltre ad essere considerato adeguato.

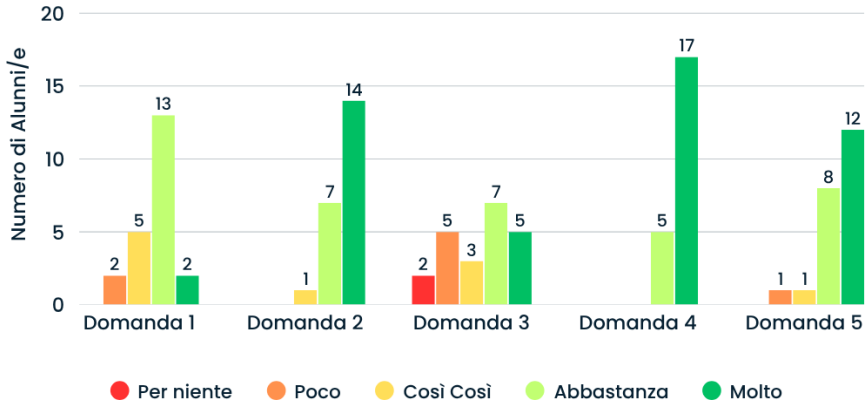


Figura 14: Analisi quantitativa delle risposte



Figura 15: Word cloud delle risposte alle domande 6, 7 e 8

Ringraziamenti

L'autrice ringrazia la Prof.ssa Sara Capecchi per le discussioni, i preziosi commenti e il supporto forniti durante la progettazione del percorso. Si ringraziano anche la classe 5^A e le insegnanti dell'Istituto Comprensivo "G. e G. Robecchi" di Gambolò (PV).

Bibliografia

- [1] Cambridge Assessment International Education. "Apprendimento attivo." Cambridge University Press & Assessment. <https://www.cambridgeinternational.org/languages/italian/Images/649897-apprendimento-attivo.pdf>, 2021. Visitato il 09-06-2025.
- [2] Capecchi, Sara, et al. "Castle and Stairs to Learn Iteration: Co-designing a UMC Learning Module with Teachers." In Proceedings of the Conference on Innovation and Technology in Computer Science Education V. 1, pp. 222–228 (2023).
- [3] Capecchi, Sara, et al. "InformaticaSenzaPC." <https://informaticasenzapc.di.unito.it/>, 2025. Visitato il 12-06-2025.
- [4] Caspersen, Michael E., Ira Diethelm, Judith Gal-Ezer, et al. "Quadro di riferimento per l'informatica nella scuola." INFORMATICS FOR ALL, <https://www.informaticsforall.org/the-informatics-reference-framework-for-school-online-it/#4>, 2022. Visitato il 09-06-2025
- [5] Computer Science Education Research Group at the University of Canterbury. "CS Unplugged: Image Representation." Visitato il 09-06-2025 <https://www.csunplugged.org/en/topics/image-representation/>
- [6] Consorzio Interuniversitario Nazionale per l'Informatica. "Proposta di Indicazioni Nazionali per l'insegnamento dell'Informatica nella Scuola." <https://www.consorzio-cini.it/images/Proposta-Indicazioni-Nazionali-Informatica-Scuola-numerata.pdf>, 2017. Visitato il 18-06-2025.

- [7] Lodi, Michael, and Simone Martini. “Coding, pensiero computazionale e informatica: quale evoluzione possibile?” Atti ITADINFO 2024 Secondo Convegno Italiano sulla Didattica dell’Informatica, a cura di S. Capecchi, B. Catania. GENOVA UNIVERSITY PRESS, pp. 20–24 (2024).
- [8] Nardelli, Enrico. “Informatica e competenze digitali: cosa insegnare?” Link&Think, <https://link-and-think.blogspot.com/2019/03/informatica-e-competenze-digitali-cosa.html> (2019). Visitato il 09-06-2025
- [9] PhET Interactive Simulations. “Visione dei Colori.” University of Colorado Boulder, <https://phet.colorado.edu/it/simulations/color-vision> (2019). Visitato il 15-06-2025
- [10] Sailer, Michael, and Lisa Homner. “The Gamification of Learning: a Meta-analysis.” *Educational Psychology Review* 32.1, Springer, pp. 77–112 (2020).
- [11] Università di Torino. “Child protection policy Unito.” https://www.unito.it/sites/default/files/dr_emanazione_child_protection_policy_unito_6570_2022.pdf , 2022. Visitato il 09-06-2025
- [12] Werthner, Hannes, Edward A. Lee, Hans Akkermans, et al. “MANIFESTO DI VIENNA PER L’UMANESIMO DIGITALE.” https://dighum.ec.tuwien.ac.at/wp-content/uploads/2019/07/Vienna_Manifesto_on_Digital_Humanism_IT.pdf, 2019. Visitato il 09-06-2025

POSTER

**PER LA PRIMARIA E
LA SECONDARIA DI PRIMO GRADO**

Giochiamo con il linguaggio binario

Anna Rita Colella¹

¹ Ministero dell'istruzione e del merito

annarita.colella@scuola.istruzione.it

Abstract

La proposta didattica qui presentata prende spunto da un'attività unplugged ideata da Tim Bell, Ian H. Witten e Mike Fellows [1] e si colloca all'interno di un percorso di introduzione ai concetti fondamentali dell'informatica nella scuola primaria. L'attività, già sperimentata dalla sottoscritta in contesti di orientamento per alunni del primo ciclo, è stata adattata per una classe quinta primaria con l'intento di proporre un'esperienza ludica finalizzata allo sviluppo di competenze nell'ambito dei dati e dell'informazione [2]. In particolare, l'attenzione è rivolta alla capacità di rappresentare informazioni complesse e familiari attraverso combinazioni simboliche, in linea con i principi della codifica e della rappresentazione digitale.

1. Introduzione

L'informatica, oggi sempre più pervasiva e profondamente intrecciata con le pratiche quotidiane, non va intesa, nel contesto didattico della scuola primaria, soltanto come mero utilizzo di strumenti tecnologici, bensì come una disciplina a cui educare sin dai primi anni del percorso scolastico. Essa rappresenta una leva fondamentale per lo sviluppo del pensiero computazionale inteso come approccio strutturato alla risoluzione di problemi mediante l'uso di metodologie di astrazione che aiutano a gestire la complessità e permettono lo svolgimento automatico di compiti di elaborazione dell'informazione [3].

L'obiettivo di un curriculum per la didattica "all'informatica"¹ nella scuola primaria è quello di guidare tutti gli alunni, al termine del percorso, verso la comprensione e l'applicazione dei principi e dei concetti fondamentali della disciplina [2].

2. Competenze

L'attività "Giochiamo con il linguaggio binario" si inserisce in questa cornice educativa, in coerenza con i traguardi per lo sviluppo delle competenze al termine della scuola primaria, delineati nel documento a cura del CINI [2]. In particolare, essa mira a sviluppare le seguenti competenze:

¹ L'espressione richiama, in forma parafrastica, all'"educazione ai media" piuttosto che con i media, evidenziando come la finalità non sia soltanto l'uso operativo degli strumenti digitali, ma la costruzione di una competenza logica e formale rispetto alla disciplina informatica.

- T-P-5- iniziare a riconoscere la differenza tra dati e informazione,
- T-P-6 - esplorare le possibilità di rappresentare dati di diversa natura (numeri, immagini, suoni, ecc.) attraverso formati differenti, anche scelti in modo arbitrario,
- T-P-7 - saper riconoscere la presenza e il ruolo dei computer nei dispositivi tecnologici utilizzati nella vita quotidiana.

3. Il Laboratorio

Per sua natura, l'informatica si presta a modalità di apprendimento attivo e laboratoriale, che non richiedono necessariamente l'impiego diretto di dispositivi digitali. Tali approcci didattici favoriscono la cooperazione tra pari, stimolano la creatività progettuale e valorizzano i processi di apprendimento, anche attraverso il riuso e l'adattamento di materiali precedentemente elaborati da altri [2]. L'attività proposta introduce i concetti di dato e informazione attraverso un laboratorio ludico da svolgere con gli alunni all'interno di un percorso didattico curricolare. L'obiettivo è consentire agli studenti di confrontarsi in modo concreto e visuale con i meccanismi logici alla base della rappresentazione dell'informazione digitale. Gli alunni vengono coinvolti attivamente in un gioco didattico che prevede la costruzione e l'utilizzo di carte "binarie", grazie alle quali possono sperimentare il linguaggio binario in modo intuitivo e motivante. Attraverso questa attività, gli studenti imparano a contare e rappresentare numeri in binario, nonché a eseguire semplici conversioni tra il sistema decimale e quello binario.

3.1 Strumenti

Per la realizzazione dell'attività sono necessari un computer o notebook, uno schermo interattivo e una lavagna collaborativa online. Inoltre, per la parte laboratoriale unplugged, sono richiesti fogli di carta formato A4, pennarelli e forbici con punte arrotondate, utili alla costruzione delle carte binarie. In alternativa, le carte possono essere predisposte o realizzate anche mediante strumenti digitali.

3.2 Metodologia

Nella scuola primaria è importante stimolare gli alunni attraverso domande legate al loro vissuto quotidiano, offrendo loro occasioni per scoprire nel concreto i concetti fondanti che stanno alla base della disciplina [2]. L'approccio proposto si fonda su una metodologia laboratoriale che utilizza attività ludiche per favorire comprensione e partecipazione e consta delle seguenti fasi:

1. **Attivazione.** In questa fase l'obiettivo è attivare le conoscenze pregresse degli alunni e stimolare la curiosità attraverso un dialogo guidato e domande stimolo, a partire dalle loro esperienze quotidiane. Alcuni esempi: "*Cosa possiamo fare oggi con uno smartphone oltre a telefonare?*", "*Come fa un computer o uno smartphone a conservare e trasmettere informazioni e messaggi?*", "*Cosa succede se lo smartphone è scarico?*". Le risposte degli alunni vengono raccolte e annotate su una lavagna virtuale collaborativa, al fine di valorizzare i contributi individuali.
2. **Introduzione all'argomento.** Si introducono i concetti di acceso/spento relativi al funzionamento dei circuiti elettronici nei dispositivi digitali, evidenziando il legame con il concetto di bit. Successivamente si chiarisce la differenza tra dato e informazione, guidando progressivamente gli studenti alla scoperta del sistema binario come linguaggio fondamentale

del computer. La spiegazione è supportata da immagini e slide, e arricchita da domande stimolo, quali: “*Come fa il computer a capire cos’è una foto, un numero o un video?*”, “*Qual è la differenza tra analogico e digitale?*”.

3. **Creazione delle carte binarie.** Si introduce formalmente il concetto di bit, inteso come rappresentazione codificata della condizione acceso/spento. Gli alunni costruiscono un set di carte, ciascuna delle quali rappresenta un bit, con valore posizionale crescente (1, 2, 4, 8, 16). Le carte diventano così uno strumento tangibile per rappresentare i numeri in binario, consentendo di contare con 5 bit da 0 a 31.
4. **Scrittura dei numeri con le carte.** L’attività ha l’obiettivo di far comprendere il valore posizionale nel sistema binario e di esercitare la rappresentazione dei numeri decimali da 0 a 31. Dopo una dimostrazione guidata, gli alunni sono invitati a rappresentare con le carte alcuni numeri assegnati (ad esempio: 6, 15, 21), verificando la correttezza della codifica.
5. **Sfida binaria.** Per consolidare gli apprendimenti, si propone un’attività a piccoli gruppi (massimo tre alunni) in forma di sfida. Gli studenti applicano la codifica binaria a numeri e date personali (ad esempio il proprio giorno di nascita o date di eventi storici) e si confrontano nel rappresentare o decodificare correttamente i numeri proposti dagli altri gruppi. La dinamica competitiva contribuisce a rafforzare la motivazione e a favorire la rapidità nell’eseguire conversioni tra sistema decimale e sistema binario.

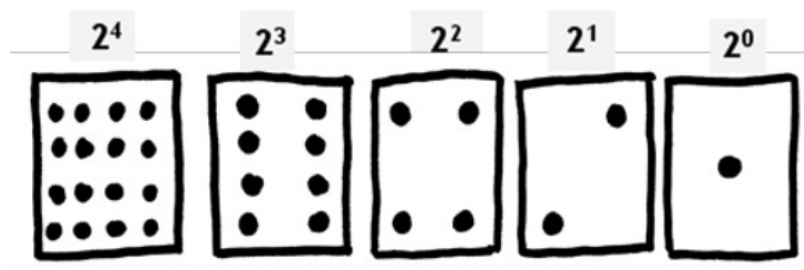


Figura 1: Costruzione delle carte binarie

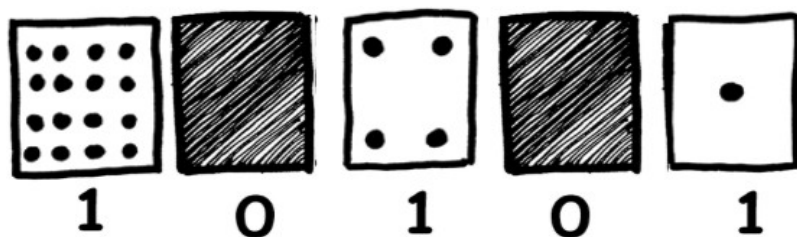


Figura 2: Rappresentazione in binario del numero decimale 21

4. Conclusione e restituzione

Nella fase conclusiva si riprendono i concetti emersi durante il laboratorio: la differenza tra dato e informazione, la distinzione tra analogico e digitale, e il ruolo del sistema binario nella rappresentazione e nell'elaborazione dei dati all'interno dei dispositivi digitali.

La riflessione viene guidata da alcune domande finali: *“Cosa abbiamo imparato oggi?”*, *“Qual è la differenza tra dato e informazione?”*, *“Qual è il linguaggio che utilizza un computer o uno smartphone per funzionare, inviare informazioni ed elaborare dati?”*.

5. Valutazione

Per monitorare e documentare lo sviluppo delle competenze oggetto del laboratorio si utilizzano strumenti di valutazione coerenti con gli obiettivi formativi dell'attività.

In particolare, per la rilevazione delle competenze del documento CINI T-P-5, T-P-6 e T-P-7, si propongono griglie di osservazione e valutazione, utilizzabili sia in itinere – durante lo svolgimento delle attività – sia in fase di restituzione, attraverso domande stimolo e momenti di riflessione guidata.

La griglia che segue è un esempio di strumento di osservazione che permette al docente di raccogliere evidenze osservabili e orientare eventuali interventi di supporto.

Osservazione delle soft skills durante la fase “sfida binaria”:

1. **Partecipazione:** interviene, prova a risolvere i compiti, non rimane passivo, rispetta le regole, accetta l'esito finale della sfida.
2. **Collaborazione:** interagisce positivamente, valorizza gli altri, evita conflitti.

Competenza	Livello 1	Livello 2	Livello 3	Livello 4
T-P-5 – Riconosce la differenza tra dati e informazione	Confonde il concetto di dato con quello di informazione; non sa fornire esempi.	Riesce a distinguere dati e informazioni e guidato dall'insegnante.	Distingue dati e informazioni con esempi semplici.	Applica in autonomia la distinzione dati/informazioni in contesti diversi, fornendo esempi pertinenti.
T-P-6 – Rappresenta dati in formati diversi	Mostra difficoltà a comprendere che un dato può essere rappresentato in più modi.	Riconosce che un numero o un simbolo può avere più rappresentazioni, se guidato.	Rappresenta correttamente dati semplici (numeri) in formato binario con le carte.	Rappresenta in modo autonomo dati diversi (es. numeri, date) in formato binario.

T-P-7 – Riconosce la presenza e il ruolo dei computer	Non collega i dispositivi di uso comune al ruolo dei computer.	Sa riconoscere alcuni dispositivi che funzionano grazie al computer, se guidato.	Identifica diversi dispositivi tecnologici quotidiani (smartphone, elettrodomestici, tablet) e ne spiega in modo semplice il ruolo.	Collega in modo pertinente il ruolo dei dispositivi digitali nelle attività quotidiane individuando anche eventuali rischi e opportunità.
--	--	--	---	---

Tabella 1: Proposta di griglia di osservazione

Bibliografia

[1] Bell, Tim, H. Witten, Ian e Fellows, Mike. “Computer Science Unplugged.” ©Bell, Witten, and Fellows, (1998).

[2] CINI. Proposta di Indicazioni Nazionali per l’insegnamento dell’Informatica nella Scuola. <https://www.consortio-cini.it/images/Proposta-Indicazioni-Nazionali-Informatica-Scuola-numerata.pdf>, 2017. Accessed: 2025-02-08

[3] Casadei, Giorgio e Lo Bello, Antonio. “Competenze e competizioni di problem solving: dal pensiero algoritmico al computational thinking.” DIDAMATICA (2012).

[4] Computer Science Education Research Group at the University of Canterbury, New Zealand. <https://www.csunplugged.org/en/>. Accessed: 2025-02-08

Scopri, crea e impara!

Paola Di Mizio¹

¹ IC Terzani

paola.dimizio@icterzani.edu.it

Abstract

Racconto di esperienza sul campo nella scuola secondaria di primo grado IC Terzani di Abbiategrasso: Corso STEM “Scopri, crea e impara!” di 25 ore, finanziato dal PNRR, A.S. 2024/2025.

La struttura del corso è stata scandita da giochi, attività laboratoriali e coding, unplugged e sulle piattaforme CODE.org, Scratch e mBlock.

Gli argomenti affrontati sono stati: codice binario, cicli, cicli annidati, istruzioni condizionali, costumi, estensioni, remix, programmazione di mBot2.

È stata sperimentata la versatilità di applicazione della programmazione a blocchi al racconto e ai laboratori su argomenti propri dell'ingegneria e della fisica (stabilità delle strutture e rotismi).

L'attività pratica e il gioco hanno consentito di veicolare concetti teorici complessi e allo stesso tempo risultavano particolarmente motivanti e coinvolgenti; la programmazione a blocchi ha catturato sin da subito l'interesse dei ragazzi.

Sono state proposte due edizioni dello stesso corso. Entrambe hanno ottenuto un riscontro molto positivo da alunni e genitori. La frequenza è stata assidua. I corsi sono stati portati a termine con profitto.

1. Contesto e struttura

Corso STEM 2024/'25 “Scopri, crea e impara!” di 25 ore. Partecipanti: 16 alunni/e delle classi prime, seconde e terze della scuola secondaria di primo grado IC Terzani di Abbiategrasso (MI). Adesione su base volontaria. Piattaforme proposte: Code.org, Scratch, mBlock.

Il corso aveva una duplice anima: quella pratica interdisciplinare e quella di introduzione alla programmazione. I due aspetti si intrecciavano a creare un humus di apprendimento collaborativo, basato sull'indagine e sulla co-costruzione delle conoscenze. I singoli incontri erano scanditi da tre momenti: il gioco, i laboratori e il coding.

L'attività pratica consentiva di veicolare i concetti teorici complessi delle STEM ed allo stesso tempo risultava particolarmente motivante e coinvolgente perché gli alunni vedevano “nascere” artefatti e prodotti digitali dal loro operato. La realizzazione di un circuito elettrico, ad esempio, faceva sperimentare la trasmissione della corrente elettrica e rendeva visibile I/O del codice binario [2]. La costruzione del Taumatropio [6], stupiva e aiutava a capire le leggi dell'ottica alla base della cinematografia, contemporaneamente, introduceva all'uso dei costumi nell'animazione dello Sprite.

Nel gioco e nell'attività motoria gli alunni interpretavano il *modus operandi* del pensiero computazionale che poi applicavano durante il Corso Rapido di Code.org [1]. Quest'ultimo ha preparato i ragazzi all'uso di Scratch [7] con il quale hanno raccontato le esperienze vissute nel laboratorio STEM pomeridiano. Scratch è stato propedeutico al laboratorio pratico di montaggio di mBot2 ed alla sua programmazione con mBlock [5]. È stato creato uno spazio di condivisione, una Classroom nella G-suite della scuola, dove venivano caricati materiali informativi e schede, link dei progetti elaborati dagli alunni e questionari di riscontro per l'insegnante e di stimolo per i ragazzi a focalizzare quanto appreso.

2. Obiettivi

- Scrivere algoritmi per descrivere semplici processi della vita quotidiana o studiati in altre discipline (sottosezioni 3.1- 4).
- Sperimentare piccoli cambiamenti in un programma per capirne il comportamento, identificarne gli eventuali difetti, modificarlo (3.1).
- Scrivere programmi che usano l'annidamento di cicli (3.2).
- Sperimentare nella creazione di contenuti digitali diversi strumenti informatici e molteplici modalità di elaborazione (3.3 - 4)
- Connettere dispositivi informatici tra di loro e con periferiche (3.4)
- Sperimentare le caratteristiche di trasversalità e complementarità della disciplina informatica per raccontare e analizzare, affrontare situazioni e fenomeni oggetto di studio nelle discipline STEM

3. Descrizione dei moduli

Di seguito vengono descritti i moduli del Corso STEM legati alla programmazione, tralasciando il racconto dei laboratori dal carattere più interdisciplinare.

3.1 Codice binario tra gioco, laboratorio e chromebook (4 ore)

Ai chromebook i ragazzi hanno scritto le prime righe di codice e si sono esercitati nel debug; hanno sperimentato il codice binario con le torce e verificato il legame tra codice binario e I/O della corrente elettrica realizzando un circuito.

Gioco – Distribuite le schede scaricate dalla piattaforma Code.org contenenti l'alfabeto in codice binario, gli alunni, muniti di torce, comunicavano i loro messaggi accendendole/spegnendole in corrispondenza degli spazi bianchi/neri della scheda. Il suono ritmico dell'accordatore online Soundcorset scandiva il tempo e serviva a calcolare per quante unità la torcia era stata spenta/accesa.

Laboratorio del circuito elettrico base – I ragazzi si sono cimentati con il concetto di istruzioni e debug unplugged: divisi in gruppi avevano a disposizione un circuito campione e le istruzioni per realizzarlo stampate in ordine sparso. Gli alunni dovevano metterle in ordine, verificare la sequenza ipotizzata tramite tutorial e infine realizzare il proprio circuito/biglietto luminoso [2].

Chromebook – Sono state svolte le lezioni 1 e 2 del Corso Rapido di Code.org, programmare con Angry birds e debugging.

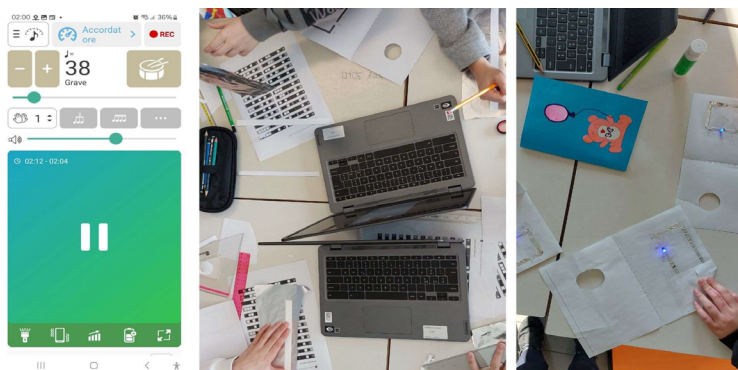


Figura 1: Soundcorset, schede con alfabeto in codice binario, circuito base e biglietto luminoso

3.2 Coordinamento e lavoro di squadra per disegnare (2 ore)

Attività unplugged su istruzioni e cicli – Gli alunni, divisi in gruppi, avevano a disposizione un solo pennarello che potevano muovere mediante dei fili, ognuno dei componenti del gruppo controllava un capo del filo. Per muovere il pennarello dovevano quindi coordinarsi ed elaborare insieme la soluzione del problema. Le figure geometriche venivano disegnate seguendo le istruzioni lette dall'insegnante, prima come sequenze semplici, poi con l'introduzione del ciclo (Fig. 2). Vincere il gruppo che era riuscito a disegnare la figura correttamente.



Figura 2: Disegno cooperativo, esecuzione di istruzioni.

Dopo l'esperienza unplugged, gli studenti hanno svolto la lezione 4 del Corso Rapido di Code.org, arrivando a creare figure complesse (Fig.3). A seguire è stato introdotto il concetto di ciclo annidato e i ragazzi hanno rivisto gli esercizi appena svolti per ridurne le righe di codice e passare alla lezione 12. Per supportare i diversi ritmi di apprendimento l'insegnante ha realizzato alcune slides da utilizzare eventualmente a casa contenenti le soluzioni richieste per ogni esercizio.

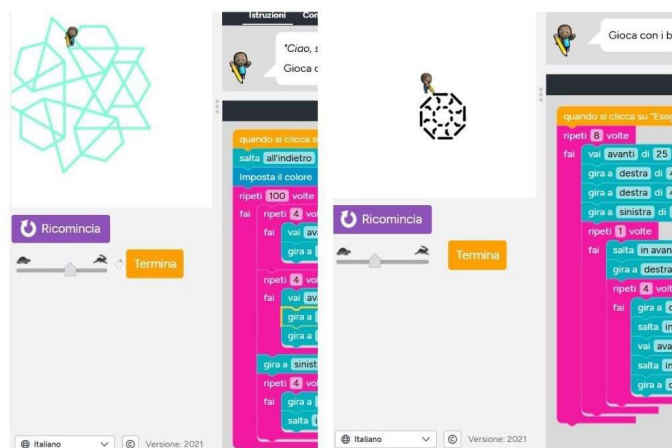


Figura 3: Figure disegnate in autonomia

L'attività di coding descritta era parte di un modulo più ampio in cui veniva sperimentata la stabilità delle strutture con attività ludiche/motorie e la costruzione di modellini [3].

3.3 Costumi, ieri come oggi è una questione di velocità (4 ore)

Laboratorio - Ogni alunno ha costruito un Taumatropio, gioco di fine '800 considerato l'antesignano del cinema. Si tratta di un disco sul quale vengono incollate due parti di un'immagine unica o le sequenze del movimento di un personaggio. Facendo ruotare velocemente il disco le immagini sembrano fondersi per via della persistenza dell'immagine sulla retina. È il principio alla base della pellicola cinematografica, sfruttato anche per creare l'illusione ottica del movimento degli Sprite: una sequenza rapida di costumi (Fig. 4).

Ballo – Gli alunni divisi in gruppi hanno progettato una breve coreografia con il sottofondo musicale preso dalla playlist di Code.org su Spotify (Fig. 5).

Chromebooks – I ragazzi hanno svolto la lezione 9 del Corso Rapido e riprodotto la coreografia progettata. [4].

È seguito un primo approccio ai costumi di Scratch e ai suoi blocchi di movimento, aspetto, situazioni e controllo.



Figura 4: Come nel Taumatropio la velocità di rotazione del disco intorno all'asse orizzontale crea l'effetto ottico dei pesci dentro l'acquario, così in Scratch i ragazzi hanno programmato i costumi dello Sprite per dare la sensazione che si muovesse

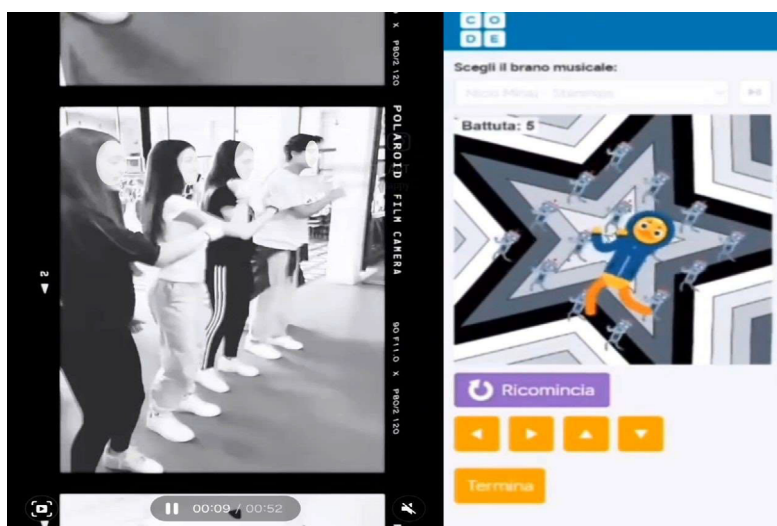


Figura 5: Ideazione di coreografie e programmazione

3.4 Scratch per raccontare e mBlock per operare (6 ore)

Compresi i cicli, le istruzioni condizionali e i costumi, i ragazzi ne hanno fatto uso in Scratch per raccontare la loro esperienza del laboratorio STEM.

Divisi in piccoli gruppi l'insegnante ha chiesto loro di scegliere un'esperienza specifica dalla quale partire: il circuito e il codice binario, il laboratorio delle forme e delle strutture stabili, il laboratorio del movimento, la visita al Museo Leonardo3.

Per incentivare il lavoro di gruppo e l'apprendimento peer to peer è stato dato un tempo di circa 30 minuti durante il quale elaborare una parte del racconto dell'esperienza scelta. Allo scadere del tempo ne veniva condiviso il link su Classroom. Un altro gruppo aveva il compito di remixare e sviluppare il racconto dei compagni. In questa maniera tutti contribuivano alle narrazioni e si arricchivano delle soluzioni di codice trovate dal gruppo che li aveva preceduti o seguiti (quando il racconto ritornava agli artefici).

Sono state sperimentate nuove estensioni come “da testo a voce”, animati gli Sprite, caricati sfondi personalizzati, disegnati oggetti.

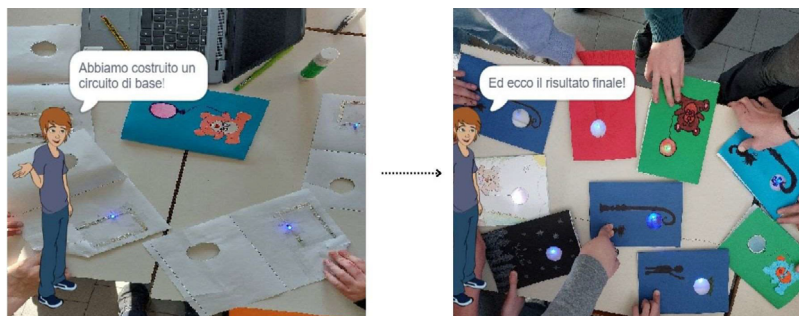


Figura 6: Il racconto della realizzazione del circuito semplice

Dopo Scratch è stato facile passare a mBlock per programmare mBot2. Gli alunni hanno montato il robottino (Fig. 7) e ne hanno programmato il movimento. L'abilità acquisita è stata verificata in modo ludico con una gara: ogni squadra doveva far tornare alla base il proprio robottino dopo aver evitato degli ostacoli (Fig. 8).

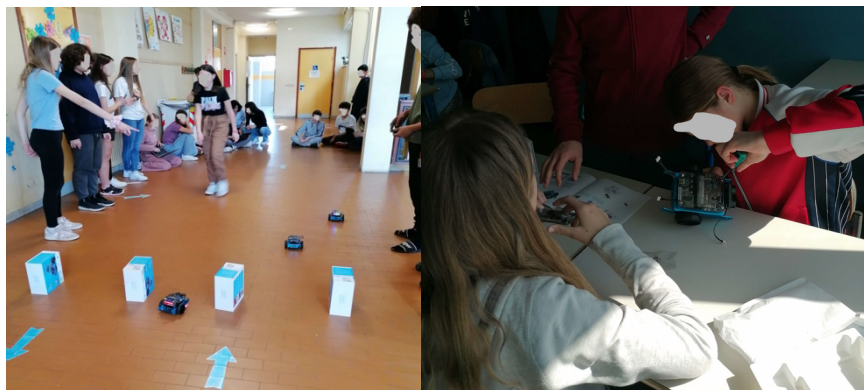


Figura 7: Challenge.

Figura 8: Montaggio di mBot2.

4. Conclusioni

Il bilancio dell'esperienza è stato molto positivo. I genitori, percependo l'entusiasmo dei ragazzi, hanno ringraziato la Scuola per aver messo in campo il corso "Scopri, crea e impara!"

È stato raggiunto l'obiettivo di dare agli studenti/esse i primi rudimenti della programmazione a blocchi e di farne scoprire le potenzialità interdisciplinari.

Ringraziamenti

Fotografie di Di Mizio Paola e Parisi Rosa.

Bibliografia

- [1] Code.org - Corso rapido, durata trimestrale: <https://studio.code.org/home>
<https://www.exploratorium.edu/tinkering/projects/paper-circuits>
- [2] Instagram, profilo ic_terzani_abbiategrosso – sperimentazione della stabilità delle figure disegnate <https://www.instagram.com/reel/DHy-laxOjCI/?igsh=bnFvNHYwamE1eGsx>
- [3] Instagram, profilo ic_terzani_abbiategrosso – costumi e coreografie, laboratori interdisciplinari <https://www.instagram.com/reel/DJBrbPeMZbo/?igsh=MTBwMm13Y3lydTl1bw==>
- [4] mBot2 programmazione on line: <https://ide.mblock.cc/>
https://www.museocinema.it/sites/default/files/downloads/servizi_educativi/taumatropio_scheda.pdf
- [5] Scratch programmazione on line: <https://scratch.mit.edu/>

Moon Camp: esplorare la Fisica con le Mani

Giuliana Gelsomino¹

¹ IC Virgilio-Gonzaga, Eboli (SA)

giuliana.gelsomino@istitutovirgilioeboli.com

Abstract

Il progetto Moon Camp: esplorare la fisica con le mani ha introdotto gli studenti della scuola secondaria ai concetti chiave dell'informatica in modo pratico e coinvolgente. Attraverso la costruzione e programmazione di un habitat lunare, i ragazzi hanno sperimentato cosa significa scrivere algoritmi, usare variabili, cicli e condizioni, fino ad automatizzare sistemi reali con Micro:bit e Arduino. La modellazione 3D e la realtà virtuale hanno reso visibile come l'informazione possa essere rappresentata e trasformata, mentre le fasi di prototipazione e debugging hanno avvicinato il lavoro del programmatore. L'informatica è stata così vissuta non solo come supporto alla scienza, ma come disciplina autonoma, capace di sviluppare pensiero computazionale, creatività e spirito critico.

1. Introduzione

Il progetto “Moon Camp: esplorare la fisica con le mani” nasce con l'obiettivo di introdurre in modo laboratoriale e motivante concetti fondamentali dell'informatica nella scuola secondaria di primo grado. L'esperienza si è svolta all'interno di un laboratorio STEM e ha integrato l'educazione scientifica con la didattica dell'informatica, ponendo al centro la comprensione e l'applicazione di concetti come algoritmi, strutture di controllo, rappresentazione dell'informazione e automazione. Attraverso la progettazione e realizzazione di un habitat lunare automatizzato, gli studenti hanno potuto applicare i principi della modellazione computazionale e della programmazione a blocchi in un contesto significativo, promuovendo il pensiero computazionale e l'approccio progettuale.

2. Descrizione dell'esperienza

2.1 Fase 1: Introduzione alla fisica comparativa e ai concetti fondamentali dell'informatica

Le attività sono iniziate con esperimenti pratici su gravità, peso, forza e resistenza atmosferica, utilizzando strumenti semplici come bilance, materiali di diversa massa, fogli di carta e plastilina, per stimolare l'osservazione e il problem posing. In parallelo, sono stati introdotti i concetti chiave dell'informatica: l'algoritmo come sequenza ordinata e finita di istruzioni, la nozione di variabile come contenitore di dati, l'importanza della precisione nella codifica delle istruzioni, e il ruolo dell'esecutore automatico. Gli studenti hanno prodotto i primi semplici

algoritmi in linguaggio naturale per descrivere sequenze di operazioni fisiche, evidenziando l'importanza della formalizzazione.

2.2 Fase 2: Progettazione collaborativa e modellazione digitale

Partendo da una riflessione collettiva sulle condizioni ambientali della Luna, gli studenti hanno individuato i problemi da risolvere e i vincoli progettuali. Hanno quindi scomposto il problema in sottoproblemi (protezione termica, risorse energetiche, isolamento, abitabilità) e definito requisiti computazionali da implementare. Con software di modellazione 3D come Tinkercad, hanno costruito i modelli digitali dell'habitat (Fig. 1), apprendendo a rappresentare informazioni geometriche e strutturali con strumenti digitali, secondo una logica di astrazione e modularità.

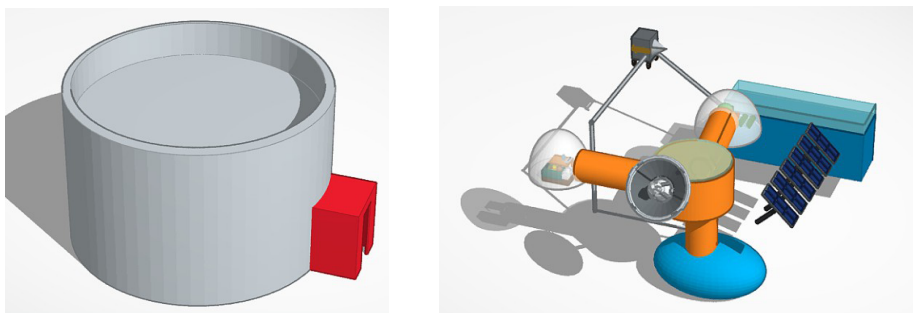


Figura 1: Rappresentazione di un modello di moon camp sviluppato in Tinkercad

2.3 Fase 3: Realizzazione pratica e making

I modelli digitali sono stati trasformati in prototipi fisici attraverso stampa 3D, incisione laser e uso di plotter. La costruzione ha richiesto l'applicazione concreta dei concetti di progettazione, test, iterazione e miglioramento, aspetti comuni anche allo sviluppo software. L'esperienza ha consolidato il pensiero algoritmico anche nella dimensione materiale: ogni fase del making ha previsto pianificazione, sequenzialità e verifica.

2.4 Fase 4: Automazione e programmazione

Il cuore informatico del progetto ha visto l'utilizzo di schede Micro:bit e Arduino per implementare sistemi automatizzati nel Moon Camp. Gli studenti hanno scritto veri e propri algoritmi in linguaggio visuale (MakeCode e Arduino IDE), che comprendevano strutture di controllo (if-else), cicli (loop), variabili e input/output.

Esempi concreti:

- algoritmi per accendere luci a LED in base alla luminosità rilevata;
- gestione automatica della temperatura tramite sensori digitali;
- sistemi di allarme programmati per reagire a eventi (apertura porta, presenza di ostacoli).

In questa fase, gli studenti hanno lavorato sull'identificazione dei dati in ingresso, sulla progettazione della logica di elaborazione, e sulla generazione di output corretti, ponendo attenzione a modularità, riuso del codice e debugging.

Questa fase ha rappresentato un vero momento di *insegnamento dell'informatica*, in cui gli studenti hanno sperimentato la relazione diretta tra dati fisici e struttura algoritmica, comprendendo la logica degli automi a stati finiti semplici.

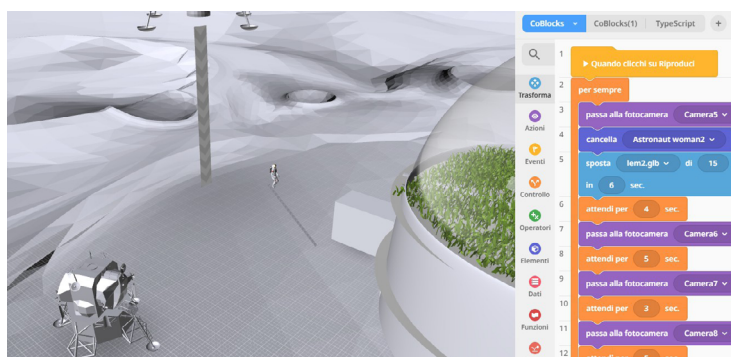
2.5 Fase 5: Esplorazione e sviluppo in realtà virtuale con Delightex

La modellazione computazionale è stata estesa alla realtà virtuale con Delightex (ex CoSpaces Edu). Gli studenti hanno ricostruito l'habitat digitale, importando i modelli 3D e programmando scenari interattivi con CoBlocks (Fig. 2), un linguaggio visuale basato su concetti informatici (eventi, oggetti, metodi, coordinate).

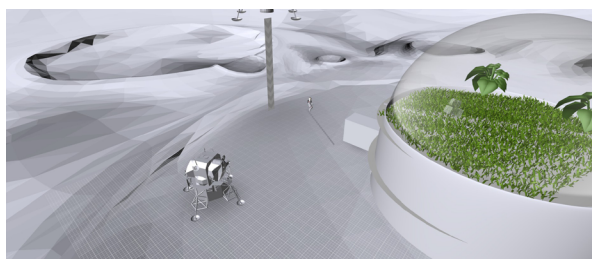
Hanno scritto codice per:

- simulare la mobilità degli astronauti;
- automatizzare il cambio giorno/notte nel paesaggio lunare;
- generare interazioni tra ambienti e avatar.

In questa fase, il concetto di *programma* come insieme di istruzioni controllate da eventi e condizioni è stato ulteriormente rinforzato.



(a)



(b)

Figura 2: Animazione del mooncamp in Delightex. a) codice di programmazione a blocchi, Coblocks; b) ambiente virtuale

2.6 Fase 6: Testing, presentazione e valutazione

I prototipi sono stati testati in laboratorio, verificando la funzionalità dei sistemi automatizzati e la resistenza delle strutture. Gli studenti hanno raccolto dati, validato il comportamento del software scritto, discusso i limiti degli algoritmi e apportato modifiche. La valutazione ha tenuto conto della correttezza logica del codice, della struttura modulare dei programmi, della qualità delle simulazioni, oltre alla comprensione dei concetti informatici coinvolti. La presentazione finale ha incluso narrazioni interattive, spiegazioni degli algoritmi e analisi delle scelte progettuali.

3. Riflessione e ricadute didattiche

Il progetto ha centrato gli obiettivi di insegnamento dell'informatica come scienza, affrontando con chiarezza concetti fondamentali quali:

1. **algoritmo** come descrizione eseguibile e verificabile;
2. **automazione** come reazione programmata a dati in ingresso;
3. **programmazione** come strumento per descrivere comportamenti;
4. **rappresentazione** digitale dell'informazione;
5. **modularità e debugging**.

L'informatica non è stata utilizzata come semplice strumento per “spiegare la fisica”, ma come oggetto di apprendimento autonomo e disciplinare, anche se inserito in un contesto interdisciplinare. La creazione di contenuti digitali, la progettazione algoritmica e la programmazione di sistemi fisici e virtuali hanno permesso di sviluppare un'autentica alfabetizzazione informatica, rafforzata da creatività, collaborazione, pensiero computazionale e spirito critico.

4. Conclusioni

Moon Camp si è rivelato un percorso efficace per integrare scienze, tecnologia e informatica, in una prospettiva disciplinare e computazionale. L'esperienza ha dimostrato come la didattica attiva, il making e la programmazione possano rendere l'insegnamento dell'informatica concreto, coinvolgente e aderente alle esigenze della scuola contemporanea.

Bibliografia

- [1] <https://www.consortio-cini.it/index.php/it/component/attachments/download/745>
- [2] https://it.eipass.com/pdf/syllabus_ministeriale.pdf
- [3] <https://www.mim.gov.it/documents/20182/8952594/Indicazioni+nazionali+2025.pdf>

FOSTEM per orientamento e sensibilizzazione all'Informatica in scuole secondarie di primo grado in Campania

Valentina Casola¹, Cristina d'Alessandro², Christian Esposito³

¹ *Università di Napoli Federico II, Italia*

casolav@unina.it

² *EITD S.r.l., Napoli, Italia*

cdalessandro@eitd.it

³ *Università di Salerno, Italia*

esposito@unisa.it

Abstract

Le scelte di studio per la formazione terziaria e universitaria sono ancora fortemente influenzate dagli stereotipi di genere e dalle convenzioni sociali, compresi gli orientamenti tradizionali all'interno delle famiglie. Il governo italiano ha fissato vari obiettivi relativi sia a una maggiore attenzione verso le materie STEM sia alla parità di genere, con l'obiettivo di ridurre il divario di genere in questi settori e promuovere una maggiore partecipazione femminile. Per poter raggiungere tali obiettivi si richiede anche agli operatori della formazione di prestare attenzione all'orientamento che dalla scuola secondaria di primo grado viene svolto verso la scuola secondaria di secondo grado, che si rifletterà anche sulla futura scelta di formazione universitaria, professionale e di opportunità lavorative. Il progetto FOSTEM, finanziato dal Fondo per la Repubblica Digitale, si concentra sulla promozione delle materie STEM (Scienze, Tecnologia, Ingegneria e Matematica) tra gli studenti del secondo e terzo anno delle scuole secondarie di primo grado (medie), fornendo loro una solida base di conoscenze STEM e sensibilizzandoli sulle opportunità di carriera nel settore industriale. Il presente rapporto mira ad illustrare le attività svolte per la formazione, la sensibilizzazione e l'orientamento della comunità educante rispetto ai temi dell'informatica, non limitandosi ad un'azione sui discenti in attività laboratoriali ma coinvolgendo anche docenti e genitori.

1. Introduzione

In Italia, sin da piccoli, i bambini e le bambine ricevono segnali diversi sul loro rapporto con la tecnologia. I maschi vengono incoraggiati a giocare con i robot, a costruire, a esplorare il mondo digitale e a “rompere per provare”. Le bambine, più frequentemente, vengono indirizzate verso giochi e ambiti più relazionali, creativi, “più umani”. Le grandi figure dell'informatica citate nei media o nei programmi scolastici – da Turing a Jobs – sono quasi sempre uomini. Le ragazze raramente vedono donne competenti e passioniste di scienza e tecnologia come

modelli a cui ispirarsi. Di conseguenza, quando si avvicina il momento di scegliere il percorso scolastico, molte di loro si sentono fuori posto anche solo a pensare a un liceo scientifico, a un istituto tecnico o a un'università di informatica. Questo squilibrio ha effetti concreti e profondi. L'informatica è uno degli ambiti professionali più richiesti, meglio retribuiti e strategici. Se le ragazze ne restano escluse o si autoescludono, si crea un doppio danno: per loro, che vedono ridursi le possibilità di realizzazione; per la società, che perde metà del talento disponibile.

In Italia, l'informatica soffre non solo di uno stereotipo di genere, ma anche di uno stereotipo economico che ne condiziona la percezione e l'accesso. Mentre i ragazzi delle famiglie più abbienti imparano a programmare, costruire robot, esplorare il pensiero computazionale già alle scuole elementari, in molti altri contesti si fatica ancora ad avere una connessione stabile in aula o un computer funzionante per ogni gruppo. Il cosiddetto digital divide in Italia non è solo infrastrutturale, ma culturale ed economico. Le scelte scolastiche ne risentono: licei scientifici, istituti tecnici a indirizzo informatico, ITS digitali sono meno frequentati da studenti provenienti da contesti svantaggiati, non per mancanza di capacità, ma per assenza di prospettiva. Paradossalmente, l'informatica potrebbe essere uno dei principali motori di mobilità sociale nel nostro Paese, visto che offre lavoro, flessibilità, competenze richieste a livello globale.

FOSTEM è un progetto sulla promozione delle materie STEM (Scienze, Tecnologia, Ingegneria e Matematica) tra gli studenti del secondo e terzo anno delle scuole secondarie di primo grado (medie), fornendo loro una solida base di conoscenze e sensibilizzando sulle opportunità di carriera in diversi settori produttivi. In considerazione anche della fascia di età degli studenti e delle studentesse, il progetto ha una forte vocazione sociale agendo sui fattori di prevenzione della dispersione scolastica e su azioni di inclusione sociale e culturale. Nell'ambito informatico, la scelta del progetto FOSTEM è di articolare le attività di formazione e di orientamento rispetto alle tematiche informatiche attorno alle metodologie didattiche basate sul Tinkering [3], in quanto modalità coinvolgente, innovativa ed efficace per lavorare con le classi della Scuola Secondaria di I grado [1]. Questa metodologia è anche adatta a contesti multi-culturali, svantaggiati e con presenza di studenti BES, contesti rappresentativi dei partner scolastici del progetto. Il tinkering rappresenta una forma di apprendimento informale in cui si impara facendo. Gli studenti lavoreranno in maniera coinvolgente su concreti casi di studio, in cui sperimenteranno le loro capacità analitiche e di problem-solving. Questo approccio del Learning-by-Doing (Imparare facendo) [5] consentirà loro di conoscere ed appassionarsi delle materie, comprendere le proprie inclinazioni, orientare le scelte future. Le attività di formazione come completamento e arricchimento di quanto già fatto nel percorso curriculare sono strutturate in una parte di didattica frontale ed una parte progettuale che include:

- introduzione alla programmazione di dispositivi hw e sw (schede elettroniche e sensori) con schede elementari di facile programmazione;
- introduzione all'intelligenza artificiale con semplici esempi e progetti specifici per bambini;
- introduzione alla programmazione di APP in contesto mobile.

Questo articolo mira a presentare i risultati preliminari del progetto, ottenuti nell'articolazione delle prime attività di formazione nella forma di summer school erogate presso quattro istituti tra le province di Napoli, Benevento e Salerno, anticamera delle attività degli altri partecipanti, da articolare durante tutto l'anno scolastico 2025/2026. Nella Sezione 2 sarà presentato il progetto e le sue attività nella prima serie di prestazioni erogate ai primi istituti coinvolti. Nella Sezione 3 presenterà i risultati intermedi ottenuti e alcune osservazioni; mentre la Sezione 4 conterrà alcune osservazioni finali e indicazioni su attività future ed evolutive.

2. FOSTEM per l'informatica

La metodologia di formazione proposta nel progetto FOSTEM si basa sul principio fondamentale del Learning-by-Doing, promosso dalla Commissione Europea. Seguendo questa metodologia didattica, gli studenti si comportano come i ricercatori [4]: fanno congetture, le verificano, imparano dai loro errori e costruiscono una base solida del sapere. La metodologia intrapresa già nelle prime attività formative erogate per la sezione di informatica, è in linea con i principi del Tinkering, oramai considerato, negli ambienti educativi a livello internazionale, un approccio innovativo per l'educazione alle discipline STEM. Nello specifico, l'alunno è incoraggiato a sperimentare, stimolando in lui l'attitudine alla risoluzione dei problemi. Tutte le attività vengono lanciate sempre sotto forma di gioco o sfida. La piattaforma alla base della metodologia è stata la scheda Micro:bit (<https://microbit.org/>), che è una schedina hardware (fornita a tutti gli studenti) che dispone di sensori, display a 25 LED e diversi connettori. Questa scheda consente l'insegnamento sia dell'informatica mediante un approccio grafico a blocchi, sia traducendo nel reale e nel Learning-by-doing l'acquisizione del concetto di algoritmo, dei principali costrutti di controllo dell'esecuzione algoritmica, come è possibile vedere nella Figura 1. I partecipanti possono avvalersi delle LIM presenti negli ambienti messi a disposizione degli istituti per il progetto, come anche pc e tablet per poter realizzare i primi programmi a blocchi per lo svolgimento di esercizi in maniera guidata e non. Inizialmente, i ragazzi sono stati diretti all'uso della piattaforma Blockly Games (<https://blockly.games/>), che consente di rappresentare algoritmi graficamente a blocchi e di poterli eseguire per risolvere semplici compiti [2], e dell'emulatore Tinkercad (<https://www.tinkercad.com/>), che consente di illustrare il funzionamento della scheda e la sua possibile connessione a sensori ed attuatori introducendo i ragazzi alla scoperta dell'Internet delle Cose e dell'automazione di spazi.

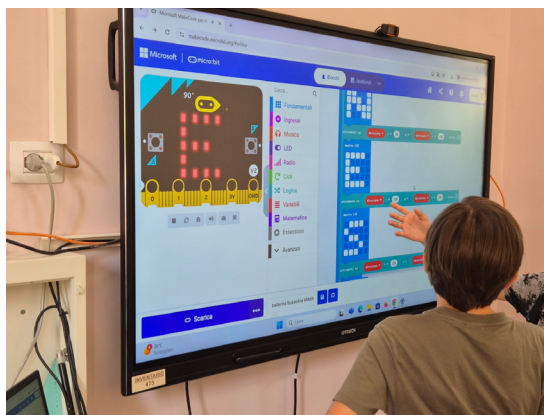


Figura 1: Insegnamento della programmazione a blocchi con la LIM

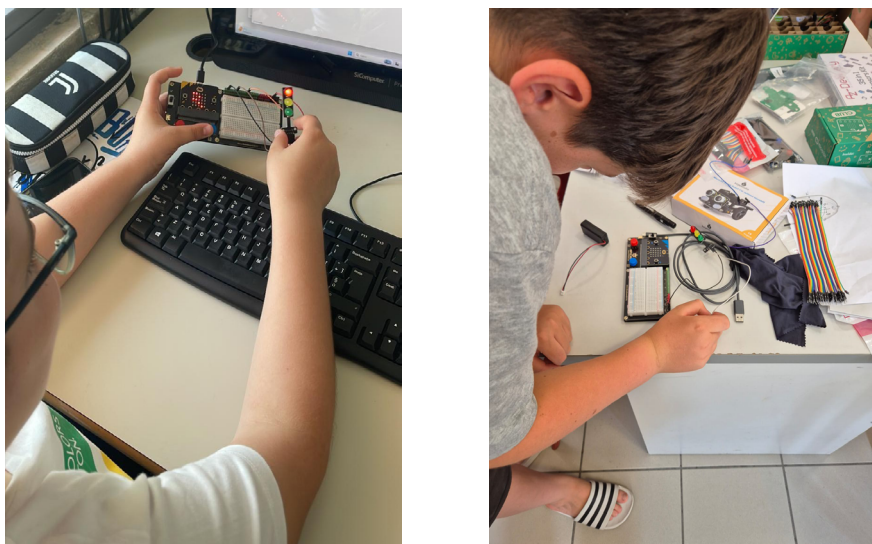


Figura 2: Esperienza hands-on dei ragazzi partecipanti alle attività di formazione



Figura 3: Condivisione progetti tra i ragazzi e conseguente collaborazione

Successivamente, una volta consolidate le conoscenze sulla connessione tra i dispositivi, i ragazzi hanno potuto sperimentare l'uso della scheda, e la sua interconnessione a sensori, attuatori, come in Figure 2, sia per la realizzazione di oggetti intelligenti, ma anche per l'assemblaggio di un primo robot da poter essere programmato per l'esplorazione dello spazio in maniera autonoma grazie all'uso di sensori.

In aggiunta, è stato integrato Scratch (<https://scratch.mit.edu/>), un ambiente di programmazione gratuito, con un linguaggio di programmazione di tipo grafico. Scratch consente

di programmare il computer per risolvere problemi e creare simulazioni, animazioni, storie interattive, grafica. In pratica usa dei mattoncini simili al Lego o ai pezzi del puzzle (gli Scratch blocks) per costruire progetti multimediali che mettono assieme immagini, suoni, video ecc. Gli studenti hanno potuto realizzare piccoli progetti di oggetti intelligenti, impiegando anche moduli di intelligenza artificiale (<https://machinelearningforkids.co.uk/>) controllati da app mobili realizzate ad hoc con App Inventor (<https://appinventor.mit.edu/>).

La scelta educativa è stata quella di un uso molto limitato della didattica frontale, preferendo un'educazione non formale ed esperienziale, con laboratori pratici e giochi educativi. In particolare, l'utilizzo di giochi è stato visto avere un impatto positivo sulla didattica, migliorando il coinvolgimento, facilitando l'apprendimento e incoraggiando i partecipanti all'azione. Tale impatto è stato misurato mediante la somministrazione di questionari di gradimento sia agli studenti che alle loro famiglie. Nel concreto, si è utilizzato di metodi e strumenti non solitamente utilizzati all'interno della didattica formale, quali laboratori esperienziali, forte coinvolgimento dei partecipanti e momenti di lavoro in gruppo parzialmente autogestiti, come visibile in Figura 3. Le attività di lezioni interattive sono state complementate da incontri con professionisti e aziende, ma anche incontri con referenti del settore educativo terziario, che hanno condiviso le loro esperienze per ispirare gli studenti, quali ITS, Università. La composizione delle classi è stata pianificata bilanciando equamente la partecipazione tra studenti e studentesse. Negli eventi di sensibilizzazione, si è puntato a supportare e incoraggiare le studentesse e le loro famiglie a conoscere le materie STEM, fornendo esempi delle possibilità di crescita lavorativa.

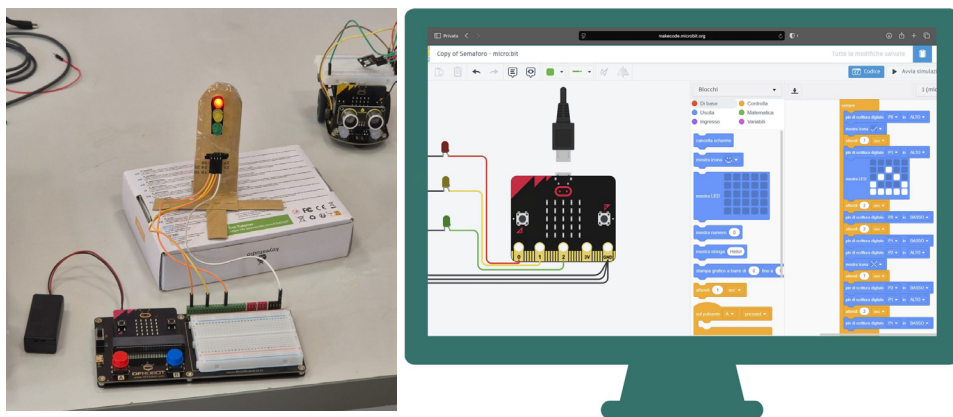


Figure 4: Esempio di realizzazione dei ragazzi: semaforo

3. Risultati e Osservazioni

In termini di risultati ottenuti, nella fase iniziale sono state organizzate 4 scuole estive, con in media 30 ore su tematiche informatiche tra Napoli, Piedimonte Matese e Pontecagnano, mentre le restanti attività saranno svolte nel prossimo anno scolastico a partire da ottobre 2025.

Sono state acquisite e distribuite a ogni scuola 15 schede Micro:bit, una serie di sensori ed attuatori Arduino e uno shield per consentire alla scheda di utilizzare il tipico wiring Arduino con breadboards per connettersi a sensori ed attuatori e un piccolo robot comandato da una scheda Micro:bit. La realizzazione di progetti con Micro:bit ha consentito di introdurre i ragazzi alla logica del codice attraverso l'esperienza diretta, come ad esempio quanto mostrato in Figura 4 dove è stato realizzato un semaforo con i led che si accendevano in successione di 2 secondi tra loro.

Scratch, invece, parla il linguaggio della narrazione e del gioco, e la realizzazione di applicativi con tale strumento ha consentito ai ragazzi di realizzare un piccolo videogioco dove Scratch ha svolto il ruolo di interfaccia per comandare i movimenti dell'avatar per il raggiungimento degli obiettivi di gioco definiti dai ragazzi, come raggiungere un determinato elemento sul campo di gioco, come in Figura 5.

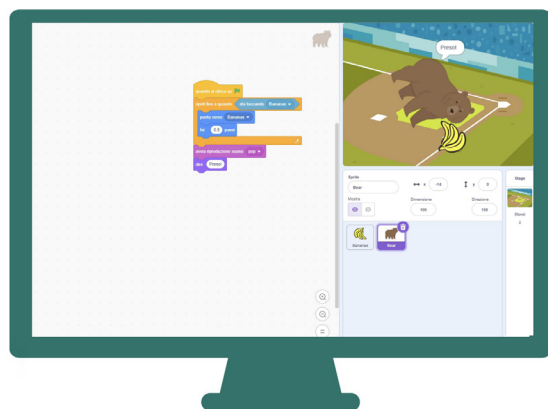


Figure 5: Gioco integrando Scratch con la scheda Micro:bit

4. Conclusioni

La prima fase del progetto FOSTEM si è composta di 4 summer school che hanno coinvolto un centinaio di ragazzi dagli 11 ai 12 anni d'età, che si sono formati all'informatica mediante l'uso di strumenti di educazione creativa come Micro:bit e Scratch. Sono state organizzate delle visite presso atenei coinvolti come forma di potenzialmente per la fase di sensibilizzazione dei ragazzi alle materie STEM. Nei prossimi mesi, attività simili saranno attivate anche negli altri istituti partecipanti, ma come una calendarizzazione più lunga durante tutto l'anno scolastico, e non concentrata a qualche settimana come nelle summer school. Il focus di queste prime attività è stato quindi i ragazzi, lasciando ai prossimi mesi altre due dimensioni di sensibilizzazione, ovvero i docenti e le famiglie.

Fondi

Le attività presentate in questo articolo sono state tutte sviluppate nell'ambito del progetto FOSTEM (codice progetto 2024 – POL –00222), un progetto selezionato e sostenuto dal Fondo per la Repubblica Digitale - Impresa sociale, avente come capofila Fondazione Città Nuova ETS, presieduta da Roberto Sanseverino (<https://www.fondazionecittanuova.it/progetti/toggle-id-4>). Il Fondo per la Repubblica Digitale è una partnership tra pubblico e privato sociale (Governo e Associazione di Fondazioni e di Casse di risparmio – Acri), che si muove nell'ambito degli obiettivi di digitalizzazione previsti dal PNRR e dal PNC ed è alimentato da versamenti delle Fondazioni di origine bancaria, alle quali viene riconosciuto un credito di imposta. Il Fondo seleziona e sostiene progetti di formazione e inclusione digitale per diversi target della popolazione come NEET, donne, disoccupati e inoccupati, lavoratori a rischio disoccupazione causa dell'automazione, dipendenti, collaboratori e volontari degli enti dell'economia sociale, studenti e studentesse delle scuole secondarie di primo e secondo grado e persone detenute. L'obiettivo è valutare l'impatto dei progetti formativi sostenuti e replicare su scala più vasta quelli ritenuti più efficaci in modo tale da offrire le migliori pratiche al Governo affinché possa utilizzarle nella definizione di future politiche nazionali. Per maggiori informazioni fondorepubblicadigitale.it

Bibliografia

- [1] Emma Caravaggi, Stefania Pagliara, Laura Leonardi, Elisa Appiani, et al. Tinkering a scuola: quando il fare consente di costruire nuovi orizzonti scientifici. *SCUOLA ITALIANA MODERNA*, 2022(Settembre 2022):22– 30, 2022.
- [2] Laura Lana and Vittorio Mazzoli. Il coding e le sue potenzialità didattiche. *Educare. it*, 21(9):98–105, 2021.
- [3] Antonella Poce, Francesca Amenduni, and Carlo De Medio. From tinkering to tinkering: tinkering as critical and creative thinking enhancer. *Journal of e-Learning and Knowledge Society*, 15(2), 2019.
- [4] Stefano Scarpa. Le pedagogie di matrice enattiva fondate sui concetti di embodied cognition e learning by doing. *CQIIA Rivista*, 35:69–91, 2021.
- [5] Alexander Skulmowski. Learning by doing or doing without learning? the potentials and challenges of activity- based learning. *Educational Psychology Review*, 36(1):28, 2024.

Informatica nella Natura: ambienti aumentati per l'orientamento verticale e la promozione delle competenze digitali

Alessia Galli¹, Lara Rollo²

¹ IIS Ettore Majorana

ing.alessiagalli@mail.com

² IC Velletri Sud-Overst

rollo.lara@gmail.com

Abstract

Il progetto si configura come un percorso laboratoriale verticale che ha coinvolto sia studenti sia docenti, come sviluppo di buona pratica, con l'obiettivo di promuovere competenze informatiche e didattiche attraverso pratiche innovative fondate sul modello TPACK (Technological Pedagogical Content Knowledge). L'esperienza svolta ha coinvolto classi di scuola primaria (quarte e quinte) e secondaria di primo grado (seconde e terze) in attività di coding e progettazione immersiva, utilizzando ambienti di realtà aumentata e virtuale per raccontare il territorio. Attraverso la piattaforma Delightex e l'integrazione di immagini generate con AI, gli studenti hanno costruito ambienti 3D interattivi e narrativi, sviluppando competenze logiche, algoritmiche e creative. I docenti partecipanti, provenienti da discipline diverse, hanno avuto l'opportunità di osservare direttamente le dinamiche di apprendimento in aula e di sperimentare in prima persona le stesse tecnologie, seguendo una logica di learning by doing e observational learning. Questa doppia prospettiva ha consentito loro di acquisire competenze progettuali e metodologiche fondate sull'interdisciplinarietà, la collaborazione tra pari, l'inclusione e l'orientamento precoce all'informatica come disciplina scientifica. La struttura verticale del progetto, che favorisce la continuità tra ordini di scuola e la co-costruzione di ambienti digitali tra studenti di età diverse, ha permesso ai docenti di riflettere su come progettare percorsi significativi e motivanti, in cui l'informatica diventa uno strumento per narrare, esplorare e valorizzare il territorio. L'interazione tra tecnologia e contesto territoriale ha reso il progetto non solo uno strumento di innovazione didattica, ma anche un'occasione concreta per potenziare il profilo professionale dei docenti nella direzione delle competenze digitali e computazionali per la didattica.

1. Introduzione

Negli ultimi anni, le tecnologie di realtà virtuale (VR) e aumentata (AR) si sono affermate come strumenti efficaci per aumentare l'engagement degli studenti e favorire un apprendimento esperienziale. Grazie alla possibilità di esplorare e programmare ambienti 3D, queste tecnologie offrono contesti motivanti per lo sviluppo del pensiero computazionale [1,2], migliorando anche la comprensione concettuale, la creatività e la capacità di astrazione [3].

Cresce inoltre l'interesse per l'applicazione della VR/AR in chiave territoriale e interdisciplinare: la costruzione di ambienti ispirati al patrimonio locale rafforza l'apprendimento e promuove cittadinanza digitale e consapevolezza ecologica, trasformando lo studente in narratore del proprio territorio [4].

In questo contesto nasce Informatica nella Natura, un progetto verticale tra scuola primaria (classi IV e V) e secondaria di primo grado, che integra programmazione a blocchi e realtà aumentata per sviluppare competenze informatiche lungo tutto il primo ciclo. Il percorso si ispira a un modello curricolare spiraleggiante [5], per garantire continuità, orientamento precoce e progressione concettuale nelle competenze chiave: costruzione di algoritmi, modellazione di dati multimediali e progettazione di ambienti interattivi.

Basato su learning by doing, pensiero computazionale e collaborazione intergenerazionale, il progetto si fonda sui principi del modello TPACK, promuovendo l'integrazione efficace tra contenuti disciplinari, tecnologie e metodologie attive [6, 8].

Infine, la partecipazione dei docenti favorisce una crescita professionale significativa: attraverso la progettazione condivisa di attività di coding e ambienti immersivi, si potenziano le competenze tecniche, didattiche e la self-efficacy nel teaching del coding [10, 11].

La scelta di incentrare il progetto sulla natura nasce dall'urgenza educativa di riconnettere le nuove generazioni con l'ambiente naturale, promuovendo consapevolezza ecologica e senso di responsabilità verso il territorio. In un'epoca segnata da crisi ambientali e crescente disconnessione dalla realtà naturale, è fondamentale coltivare un'educazione orientata alla sostenibilità e al rispetto degli ecosistemi. Coinvolgere gli studenti nella narrazione digitale della natura locale significa valorizzare il patrimonio ambientale e formare cittadini attivi e consapevoli [12].

2. La piattaforma Delightex

Delightex è una webapp progettata per il contesto educativo, che consente a studenti e docenti di modellare mondi tridimensionali, animarli tramite codice a blocchi e visualizzarli su dispositivi mobili o tramite visori immersivi. Dal punto di vista operativo, la piattaforma consente di inserire oggetti 3D, personaggi, testi, suoni e ambientazioni, con una notevole libertà creativa. I contenuti possono essere arricchiti grazie alla programmazione visuale con CoBlocks, un linguaggio ispirato a Blockly/Scratch, che permette agli studenti di animare personaggi, attivare quiz, simulare interazioni e creare vere e proprie narrazioni digitali. A questo si aggiunge la possibilità di integrare oggetti personalizzati o generati con intelligenza artificiale, come avvenuto nel progetto con Blokadelabs, che ha permesso agli studenti di costruire paesaggi e contesti naturalistici realistici. Il valore educativo della piattaforma risiede nella sua capacità di trasformare gli studenti in creatori attivi di contenuti, rendendo l'informatica concreta e visibile attraverso prodotti digitali complessi ma costruiti passo dopo passo.

3. Lo sviluppo del progetto

Nell'ambito del progetto PNRR DM65 "Scuola 4.0", è stata realizzata un'esperienza laboratoriale innovativa fondata sull'approccio TPACK, che ha coinvolto due classi della scuola primaria (quarta e quinta) e due della scuola secondaria di primo grado (seconda e terza). Il progetto è stato strutturato in tre fasi operative, tutte mirate a integrare in modo efficace le

dimensioni tecnologica, disciplinare e pedagogica dell'insegnamento, grazie alla collaborazione tra il docente esperto e il team dei docenti interni (tecnologia, matematica, italiano, arte e docenti della primaria).

3.1 Progettazione e formazione iniziale

La prima fase si è focalizzata sullo sviluppo della componente Technological Knowledge. Il docente esperto ha formato gli studenti della scuola secondaria all'utilizzo della piattaforma Delightex, guidandoli nell'esplorazione delle sue funzionalità per creare ambienti 3D utilizzando l'environment della piattaforma ma anche la generazione di immagini sferiche con l'intelligenza artificiale utilizzando la webapp Blockadelabs e infine introducendoli nella programmazione di interazioni mediante codice a blocchi. La dimensione Content Knowledge è emersa quando gli studenti, organizzati in gruppi, hanno selezionato e approfondito temi legati al territorio (come riserve naturali, parchi, coltivazioni, siti archeologici), traducendoli in contenuti digitali interattivi, quiz, cambi scena e personaggi parlanti, stimolando creatività e capacità logiche. La Pedagogical Knowledge è stata attivata attraverso un approccio cooperativo e progettuale, che ha favorito lo sviluppo del pensiero computazionale, della narrazione e della modellazione.

3.2 Avvio con la scuola primaria: esplorazione e avvicinamento

In parallelo, le classi della primaria sono state introdotte alla Technological Knowledge attraverso ambienti digitali esplorabili con schermi immersivi e tablet. La dimensione pedagogica si è concretizzata nell'uso di strategie inclusive e interdisciplinari (narrazione, rappresentazione artistica, uso di audio e testi semplificati), mentre i contenuti selezionati hanno rafforzato l'apprendimento situato. Questa fase ha permesso di costruire una prima familiarità con gli ambienti immersivi, ponendo le basi per una didattica integrata e accessibile.

3.3 Peer education e progetto condiviso

La terza fase ha rappresentato un punto di sintesi metodologica nel modello TPACK, attraverso l'attuazione del cooperative learning verticale. Gli studenti della secondaria, già competenti nell'uso delle tecnologie, hanno ricoperto il ruolo di tutor all'interno di gruppi misti (2+2), supportando gli alunni della primaria nell'inserimento di oggetti interattivi nei loro ambienti virtuali. Questa collaborazione ha promosso non solo il trasferimento di Technological Knowledge, ma anche una riflessione pedagogica sull'efficacia del learning by doing e della peer education. I contenuti prodotti hanno valorizzato il territorio locale, integrando saperi narrativi, artistici e scientifici in ambienti digitali costruiti con finalità comunicative e formative.

Il lavoro collaborativo ha dato origine a progetti condivisi, nei quali ciascun gruppo ha potuto raccontare, attraverso la tecnologia, un frammento del proprio territorio, unendo competenze narrative, artistiche, logiche e digitali. Il carattere immersivo dell'ambiente ha avuto un effetto fortemente motivante per entrambi gli ordini scolastici, incentivando l'apprendimento per scoperta e il problem solving.

4. Impatto educativo e crescita professionale dei docenti

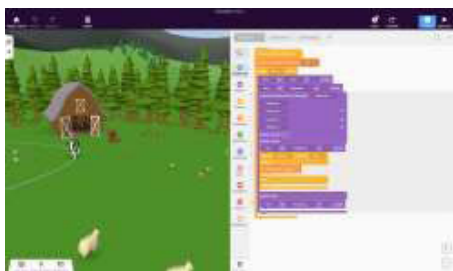
Per gli studenti dal punto di vista delle competenze informatiche, l'intero percorso ha favorito lo sviluppo di abilità fondamentali: la comprensione del concetto di algoritmo, la capacità di scomporre un problema in passi risolvibili, la progettazione e la scrittura di sequenze di istruzioni, l'uso di istruzioni condizionali, cicli e variabili e la modellazione digitale di contenuti multimediali. Il progetto ha offerto ai docenti coinvolti un'importante occasione di sviluppo professionale, consentendo loro di esplorare l'insegnamento dell'informatica attraverso approcci innovativi. Gli insegnanti della primaria hanno sperimentato l'efficacia del coding visuale per creare contenuti digitali con i più piccoli, arricchendo il proprio repertorio didattico in chiave interdisciplinare. I docenti della secondaria hanno invece approfondito l'uso educativo della realtà aumentata e virtuale, valorizzandone il potenziale per rendere l'informatica più motivante e accessibile. La collaborazione tra ordini di scuola ha infine favorito pratiche di cooperative learning verticale, rafforzando la continuità didattica e l'orientamento precoce.



(a)



(b)



(c)



(d)

Figura 1: (a) Esempio di scena 3D realizzata utilizzando l'enviroment. (b) esempio di ambientazione 3D realizzata utilizzando la generazione AI. (c) Esempio di coding con CoBlocks per l'anima-zione dei personaggi. (d) Esempio di coding con CoBlocks con variabili e cicli

Il progetto si pone quindi come una buona pratica in cui l'integrazione tra realtà aumentata, coding e narrazione territoriale può rendere l'informatica accessibile e coinvolgente fin dalla primaria, facilitando l'apprendimento di concetti complessi in modo autentico.

Per i docenti, ha rappresentato un'occasione di crescita professionale concreta, permettendo di applicare il modello TPACK nella progettazione interdisciplinare e collaborativa, rafforzando competenze didattiche, tecnologiche e metodologiche in una prospettiva riflessiva, situata e interdisciplinare, nella gestione di ambienti immersivi e nell'uso del coding come strumento per l'apprendimento.

Bibliografia

- [1] Bacca, J., Baldiris, S., Fabregat, R., Graf, S., & Kinshuk (2014). *Augmented Reality Trends in Education: A Systematic Review of Research and Applications*
- [2] Dunleavy, M., & Dede, C. (2014). *Augmented Reality Teaching and Learning*
- [3] Radianti, J., Majchrzak, T. A., Fromm, J., & Wohlgenannt, I. (2020). A systematic review of immersive virtual reality applications for higher education: Design elements, lessons learned, and research agenda.
- [4] Beetham, H., & Sharpe, R. (2013). *Rethinking Pedagogy for a Digital Age*
- [5] Bruni, F., Manca, S., & Caviglia, F. (2021). *Curricolo verticale per l'informatica: modelli e strategie per la scuola di base*
- [6] Voogt, J., Fisser, P., Pareja Roblin, N., Tondeur, J., & van Braak, J. (2012). Technological pedagogical content knowledge—a review of the literature. *Journal of Computer Assisted Learning*
- [7] Di Blas, N., Paolini, P., Sawaya, S. & Mishra, P. (2014). Distributed TPACK: Going Beyond Knowledge in the Head. In M. Searson & M. Ochoa (Eds.), *Proceedings of Society for Information Technology & Teacher Education International Conference 2014*
- [8] Mishra, P., & Koehler, M. J. (2006). Technological Pedagogical Content Knowledge: A framework for teacher knowledge
- [9] Angeli, C., & Valanides, N. (2009). Epistemological and methodological issues for the conceptualization, development, and assessment of ICT–TPCK: Advances in technological pedagogical content knowledge (TPCK)
- [10] Alrawashdeh, B., Henning, G., & Chen, W. (2024). Virtual Professional Development Enhances Elementary Teachers' Coding Skills and Self-Efficacy: A Comparison of Three Models. *Journal of Technology and Teacher Education*
- [11] Yadav, A., Gretter, S., & Hambrusch, S. (2021). Preparing Teachers to Teach Computer Science: The Role of In-Service Teacher Professional Development.
- [12] Giorda, C. (2012). *Didattica del territorio. Per una geografia della formazione*. Carocci

POSTER

PER LA SECONDARIA DI SECONDO GRADO

Educare all'Intelligenza Artificiale: un percorso di conoscenza e consapevolezza

Giacomo Salvi¹

¹Istituto Istruzione Superiore "D.M. Turollo", Zogno, Bergamo

giacomo.salvi1@scuola.istruzione.it

Abstract

Questo articolo presenta un laboratorio sull'uso dell'IA generativa sviluppato presso l'IIS "D.M. Turollo" di Zogno (BG) nell'ambito del PNRR. Si descrivono le metodologie didattiche, i contenuti principali e i risultati emersi, con osservazioni utili per future sperimentazioni.

1. Introduzione

Obiettivo di questo articolo è di condividere le esperienze svolte presso l'Istituto di Istruzione Superiore "David Maria Turollo" di Zogno (BG), nell'ambito della didattica dell'Informatica, della digitalizzazione e della parità di genere nelle discipline STEM. Il contesto scolastico di riferimento è estremamente variegato in quanto l'offerta formativa si articola su 8 indirizzi; 4 Istituti Tecnici e 4 Licei. L'estensione del territorio e la sua marcata eterogeneità determinano un bacino d'utenza altamente diversificato.

L'offerta didattica proposta si inserisce nell'ambito del bando PNRR (D.M.65) [14]. In linea con i requisiti del bando, il percorso prevedeva la partecipazione di almeno 9 studenti, una frequenza minima del 75% delle attività previste per ciascun partecipante e la possibilità per ogni studente di iscriversi a più attività tra le 34 proposte.

La scelta delle tematiche dei percorsi, così come la loro quantificazione si è basata su una fase esplorativa che ha coinvolto tutto l'Istituto, con il supporto di tutto il corpo docente. Con modalità differenti si è raccolta una base di dati sufficiente ad elaborare una proposta in linea con gli interessi degli studenti e delle studentesse.

Alla base di tutte queste attività si trova la necessità di integrare i curricula e l'offerta formativa con quanto offrono le tecnologie digitali e dell'informazione [4] [7] [19] [23].

2. Il percorso

L'obiettivo del percorso Impariamo ad Utilizzare l'Intelligenza Artificiale è quello di illustrare e far comprendere agli studenti cosa sia l'IA e come padroneggiare questo strumento sfruttandone le potenzialità, ma conoscendone anche i limiti [5] [6] [8] [19]. Vista la vastità del tema, nel percorso ci si è focalizzati su strumenti di Intelligenza Artificiale Generativa, di

più rapida applicazione per gli studenti e di più immediata comprensione [4]. Gli strumenti utilizzati durante il percorso sono tutti software o piattaforme gratuiti disponibili on line.

2.1 Contesto Classe

Si è proposto un corso della durata complessiva di 10 ore, sviluppatosi, in orario extracurricolare, nei mesi di Novembre e Dicembre 2024, presso le aule ed i laboratori informatici dell'Istituto. Le lezioni sono state suddivise in 5 blocchi da 2 ore ciascuno. Il gruppo classe del corso "Impariamo ad utilizzare l'Intelligenza Artificiale" si compone di 18 alunni (5 studentesse e 13 studenti) di età ed indirizzi diversi (Figura 1).

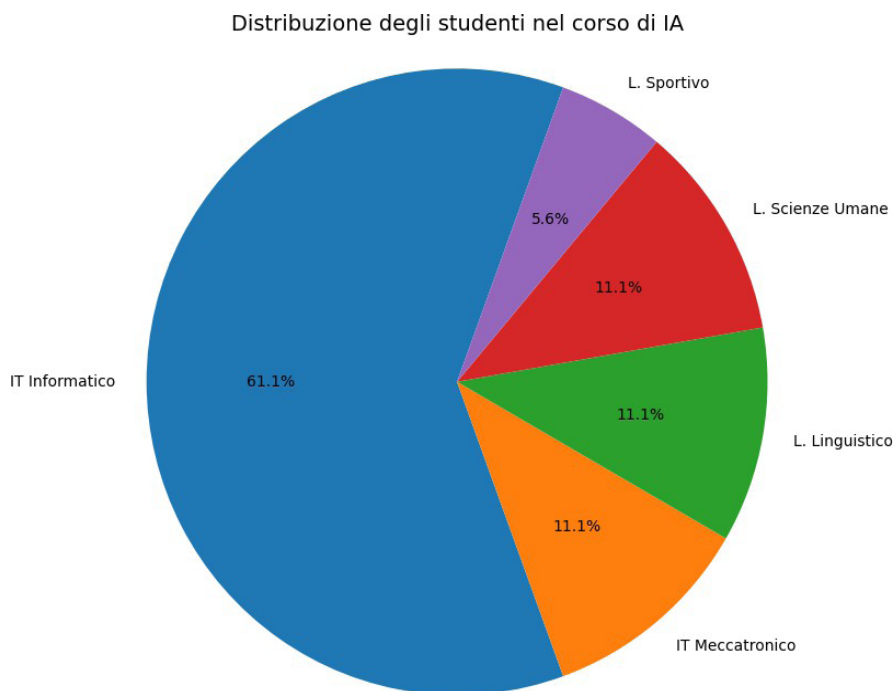


Figura 1: Distribuzione indirizzi dei partecipanti

Non sono stati previsti prerequisiti particolari per l'accesso al percorso, se non il raggiungimento del numero minimo di iscritti. Il livello di conoscenza del tema era pressoché uguale per tutti gli studenti, attestandosi su un livello medio-basso.

Sfortunatamente la partecipazione alle lezioni è stata impedita dall'elevato numero di attività (curricolari ed extracurricolari) concomitanti, attestandosi comunque al 60 - 70%.

2.2 Contenuti del percorso

Il corso è stato articolato in moduli distinti, ciascuno mirato a fornire una comprensione teorica e pratica dell'Intelligenza Artificiale e delle sue applicazioni (come indicati nelle più recenti indicazioni nazionali ed internazionali [5] [6] [8] [11] [19] [20] [23] [24]).

Introduzione teorica. Nel primo modulo è stata proposta una cornice teorica introduttiva all'IA, costruita ispirandosi all'approccio Inquiry-Based Learning (IBL), supportata da una presentazione multimediale interattiva, che ha reso più immediata la comprensione dei concetti [3] [13] [21]. In questo contesto, si è discusso il significato e la definizione di IA, introducendo gradualmente le differenti tipologie (debole, forte e superumana). Per rendere più intuitivo il funzionamento del Machine Learning, si è utilizzata una metafora basata sull'apprendimento di un bambino. Questa analogia ha aiutato gli studenti a visualizzare meglio i concetti di apprendimento supervisionato e non supervisionato, ed a comprendere la progressione verso modelli più complessi come il Deep Learning, ispirato alla struttura delle reti neurali biologiche.

Un momento particolarmente intenso è stato quello dedicato alle implicazioni etiche dell'IA, che ha stimolato una riflessione critica e partecipata. Partendo dal celebre “dilemma del carrello”, ovvero la scelta etica in situazioni di conflitto morale [1] [22], si è discusso come queste decisioni si complichino ulteriormente quando delegate a macchine autonome [25]. Gli studenti hanno ragionato su questioni come la responsabilità delle azioni dell'IA, la trasparenza/equità degli algoritmi e le potenziali conseguenze sociali dell'automazione avanzata; oltre che la tutela dei minori e l'ambito scolastico.

ChatGPT. A partire dal secondo modulo, l'approccio si è fatto pratico e laboratoriale, con un focus sui modelli di linguaggio di grandi dimensioni (LLM). In particolare, è stato approfondito l'utilizzo di ChatGPT [16], sperimentando la tecnica del prompt engineering per ottenere risposte più precise, contestualizzate ed utili. Gli studenti hanno creato e confrontato diversi prompt, ad esempio:

- Prompt generico: “Scrivi un riassunto sulla Rivoluzione Industriale”
- Prompt mirato: “Scrivi un riassunto di massimo 200 parole sulla Prima Rivoluzione Industriale, focalizzandoti su innovazioni tecnologiche e conseguenze sociali, in un linguaggio adatto a studenti delle scuole superiori.”

Questa attività ha evidenziato come una formulazione più dettagliata e consapevole produca risposte di qualità nettamente superiore.

Altre Intelligenze Artificiali Generative. Parallelamente, si è introdotto il tema della generazione di contenuti multimediali mediante IA, sperimentando in laboratorio diversi strumenti, tra cui Bing Image Creator [15], DALL-E [17] e Canva [2]. Quest'attività ha permesso di osservare in modo pratico come l'IA possa supportare la creazione visuale ed ampliare le possibilità espressive, ma anche di comprendere le differenze tra le varie piattaforme, sia in termini di qualità delle immagini generate che di tempi di risposta e complessità nella formulazione dei prompt. Per rendere più chiaro il processo, è stato proposto un prompt identico su tutte le piattaforme ed i risultati hanno mostrato differenze significative fra i vari applicativi. Questa attività, oltre alla potenza creativa degli strumenti, ha evidenziato la sensibilità del risultato alla precisione del prompt. È emerso, infatti, che piccole variazioni nella formulazione della richiesta portavano a differenze notevoli nell'output.

Laboratorio Machine Learning. Nel penultimo modulo, si è approfondito l'addestramento supervisionato con Google Teachable Machine [10], guidando gli studenti nella creazione di un semplice modello per il riconoscimento di oggetti d'arredo scolastico. Questa esperienza ha permesso di osservare concretamente il ciclo di vita di un modello di IA: dalla raccolta dei dati alla fase di addestramento, fino alla validazione ed al test del sistema.

Prova Finale. Infine, l'ultimo modulo ha avuto una duplice finalità: da un lato la valutazione delle competenze acquisite, mediante una prova pratica che ha integrato sia aspetti teorici sia esercitazioni sui vari strumenti; dall'altro la raccolta del feedback degli studenti attraverso un questionario di gradimento, utile per riflettere sull'efficacia del percorso e sulle prospettive future di integrazione dell'IA nei processi educativi.

2.3 Metodologie Didattiche

Il percorso di apprendimento è iniziato con un'esperienza diretta degli studenti, ad eccezione della prima lezione, di carattere puramente teorico-introdotivo [3]. L'intero processo ha favorito il confronto e lo scambio di opinioni tra partecipanti e formatore, creando un ambiente collaborativo e stimolante. Il corso si è sviluppato in diverse fasi didattiche:

1. Nella lezione teorica sono state adottate metodologie **induttive e interattive**, valorizzando l'eterogeneità del gruppo e arricchendo il dibattito.
2. Nella prima fase laboratoriale gli studenti hanno seguito un percorso di **apprendimento imitativo**, replicando le attività del docente per poi affrontare gradualmente compiti più autonomi e realistici [4].
3. Successivamente si sono applicate metodologie come **learn by doing, cooperative learning e problem solving**.

In quest'**ottica costruttivista**, la fase finale ha previsto un compito di realtà basato sul problem solving: gli studenti, organizzati in un unico gruppo, hanno addestrato un'IA [18].

3. Risultati e Osservazioni

Sono stati raccolti feedback dai 12 studenti che hanno raggiunto il 70% delle presenze.

Tutti hanno apprezzato il percorso, senza segnalazioni di difficoltà eccessive o di argomenti non interessanti. Tutti gli studenti hanno dichiarato di aver trovato utile il percorso, e di aver appreso molto sul funzionamento dell'IA.

- *Imparare ad usare Chat-GPT.*
- *Gli esempi pratici ed il lavoro attivo, così come la parte teorica.*
- *Imparare ad utilizzare in modo adeguato l'IA, conoscendone anche i rischi.*
- *Confrontare diversi chatbot e valutarne le differenze.*
- *Imparare a scrivere correttamente un prompt.*
- *Le proposte di miglioramento invece riguardano:*
- *Strutturazione di un percorso più esteso.*
- *Proporre casi studio relativi all'utilizzo quotidiano degli studenti.*
- *Dare un obiettivo finale al corso e valutare gli stati ex-ante ed ex-post.*

Una risposta formativa adeguata può essere un percorso di cittadinanza digitale focalizzato sull'uso consapevole dell'IA in ambito scolastico, della durata di almeno 20 ore.

4. Conclusioni

Questo lavoro presenta un progetto sperimentale di didattica dell'IA, realizzato presso l'Istituto David Maria Turoldo di Zogno (BG) nell'ambito del PNRR. Il corso Impariamo ad utilizzare l'Intelligenza Artificiale è stato concepito per introdurre i rudimenti dell'IA [19] e sensibilizzare sugli attuali limiti di questa tecnologia [11]. Sono stati descritti i contenuti didattici, le metodologie adottate e i risultati emersi dalla sperimentazione, evidenziando criticità e possibili miglioramenti. La scelta di una tematica coinvolgente ha favorito la formazione di una classe eterogenea e collaborativa, consentendo un percorso di alfabetizzazione digitale trasversale. I feedback positivi e i risultati ottenuti suggeriscono che il progetto possa ispirare iniziative analoghe in altri istituti.

Bibliografia

- [1] Jean-François Bonnefon, Azim Shariff e Iyad Rahwan. "The Social Dilemma of Autonomous Vehicles". In: *Science* 352.6293 (2016), pp. 1573–1576. doi: 10.1126/science.aaf2654.
- [2] Canva. Canva homepage. <https://www.canva.com/>.
- [3] Paul Chandler e John Sweller. "Cognitive Load Theory and the Format of Instruction". In: *Faculty of Education - Papers 8* (dic. 1991). doi: 10.1207/s1532690xci0804_2.
- [4] Commissione europea/Agenzia esecutiva europea per l'istruzione e la cultura/Eurydice. *Informatica a scuola in Europa. Rapporto Eurydice*. Luxembourg (Luxembourg): Ufficio delle pubblicazioni dell'Unione europea, 2023. doi: 10.2797/92623.
- [5] UNESCO. Assistant Director-General for Education. *Generative AI and the future of education. programme and meeting document 0000383206*. 2023. doi: 10.54675/HOXG8740(online).
- [6] UNESCO. Assistant Director-General for Education. *The ICT Competency Framework for Teachers Harnessing OER Project: digital skills development for teachers. programme and meeting document 0000383206*. France, 2022.
- [7] Consiglio dell'Unione Europea. *Raccomandazione Consiglio dell'Unione Europea del 22 maggio 2018 relativa a competenze chiave per l'apprendimento permanente* [https://eur-lex.europa.eu/legal-content/IT/TXT/PDF/?uri=CELEX:32018H0604\(01\)&from=IT](https://eur-lex.europa.eu/legal-content/IT/TXT/PDF/?uri=CELEX:32018H0604(01)&from=IT). 2018.
- [8] F.Miao e K. Shiohira for UNESCO. *AI competency framework for students*. 2024.
- [9] R. M. Gillies. "Cooperative Learning: Review of Research and Practice". In: *Australian Journal of Teacher Education* 41.3 (2016). doi: 10.14221/ajte.2016v41n3.3. url: <https://digitcult.lim.di.unimi.it/index.php/dc/article/view/254>.
- [10] Google. *Google Teachable Machine, second version*. <https://teachablemachine.withgoogle.com/> 2017.
- [11] INDIRE. *Artificial Intelligence for and by teachers, National Evaluation Report Italy*. INDIRE, 2024.
- [12] ISTAT - Istituto Nazionale di Statistica. *Popolazione residente al 1° gennaio 2024 per*

- comune. Dati relativi alla popolazione residente al 1° gennaio 2024; consultati il 17 luglio 2025. ISTAT. 2024. url: <https://demo.istat.it/app/?i=POS.5>
- [13] Diana Laurillard. *Insegnamento come scienza della progettazione. Costruire modelli pedagogici per apprendere con le tecnologie*. Traduzione italiana dell'opera originale in inglese. Milano: Franco Angeli, 2015.
- [14] Ministero dell'Istruzione e del Merito. Bando DM 65/2023. <https://pnrr.istruzione.it/avviso/dm-65-nuove-competenze-e-nuovi-linguaggi-riparto-delle-risorse/>. 2022.
- [15] Microsoft. Bing Image Creator. <https://www.bing.com/images/create/>.
- [16] OpenAI. Chat-GPT. <https://chatgpt.com/>.
- [17] OpenAI. DALL-E version 3. <https://openai.com/index/dall-e-3/>.
- [18] Francesca Pedone et al. *Valorizzazione degli stili e promozione dell'apprendimento autoregolato. Teorie e strumenti per una didattica metacognitiva*. edizioni Junior-Spaggiari edizioni, 2012.
- [19] Vuorikari R, Kluzer S e Punie Y. DigComp 2.2: The Digital Competence Framework for Citizens - With new examples of knowledge, skills and attitudes. Scientific analysis or review KJ-NA-31006-EN-N (online), KJ-NA-31006-EN-C (print). Luxembourg (Luxembourg): EU Commission, 2022. doi: 10 . 2760 / 115376(online) , 10 . 2760 / 490274(print).
- [20] Maria Ranieri. "Intelligenza Artificiale a scuola. Una lettura pedagogico- didattica delle sfide e delle opportunità." In: *Rivista di Scienze dell'Educazione* 62.1 (2024).
- [21] John Sweller. "Cognitive load theory, learning difficulty, and instructional design". In: *Learning and Instruction* 4.4 (1994). Accessed: 2025-06-20, pp. 295–312. doi: 10.1016/0959-4752(94)90003-5.
- [22] Judith Jarvis Thomson. "The Trolley Problem". In: *Yale Law Journal* 94.6 (1985), pp. 1395–1415.
- [23] Simona Tirocchi. "Digital education". In: *DigitCult - Scientific Journal on Digital Cultures* 8.2 (2024), pp. 75–89. issn: 2531-5994. doi: 10 . 36158 / 97888929589205. url: <https://digitcult.lim.di.unimi.it/index.php/dc/article/view/254>.
- [24] UNESCO. *The ICT Competency Framework for Teachers Harnessing OER Project: digital skills development for teachers. programme and meeting document* 0000383206. 2022.
- [25] Garry Young. "Using the Classic Trolley Problem to Teach AI Students and Researchers about Their Role as Moral Agents, and Why They Should Be Subject to Moral Scrutiny". In: *AI Ethics* 5 (2025), pp. 1877–1883. doi: 10.1007/s43681-024-00509-3.6

Robot, Logica e Creatività: l'esperienza “LucarelliRobot” tra STEM e futuro

Silvio Dell'Oste¹

¹ ITI “Giambattista Bosco Lucarelli” Benevento²

silvio.delloste@gmail.com

Abstract

Il mondo della scuola è sempre più chiamato a coltivare nuove competenze, e il progetto **LucarelliRobot**, realizzato presso l'Istituto “G.B.B. Lucarelli” di Benevento nell'ambito del PNRR (Progetto FUTURAMENTE), si è rivelato un'avventura didattica straordinaria per avvicinare gli studenti alla robotica educativa, alla programmazione e al pensiero computazionale.

1. Introduzione

Nel contesto del progetto PNRR “Futuramente – STEM e Digitale” promosso dal Ministero dell'Istruzione e del Merito, l'ITI “G.B.B. Lucarelli” di Benevento ha avviato l'iniziativa “LucarelliRobot”, rivolta a studenti del biennio per sviluppare competenze STEM, logico-computazionali e collaborative attraverso l'apprendimento attivo e laboratoriale. Il progetto ha previsto una combinazione di attività creative e tecnologiche: sviluppo di contenuti web (HTML e WordPress), storytelling interattivo con Scratch e robotica educativa con i kit mBot Ranger e Ultimate 2.0, programmati tramite la piattaforma mBlock.

Come accennato in precedenza, il progetto “LucarelliRobot” è stato realizzato nell'ambito del PROGETTO “FUTURAMENTE” – Codice Progetto M4C1I3.1-2023-1143-P-27708 – CUP C84D23001440001 PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) Missione 4 “Istruzione e Ricerca” – Componente 1 “Potenziamento dell'offerta dei servizi di istruzione: dagli asili nido alle Università” – Investimento 3.1 “Nuove competenze e nuovi linguaggi” – Azioni di potenziamento delle competenze STEM e multilinguistiche (D.M. 65/2023) e ha avuto come obiettivo principale quello di impegnare i partecipanti nel conoscere la robotica educativa e a conoscere nel dettaglio dei robot programmabili in possesso dell'Istituzione Scolastica, oltre che di avvicinare gli studenti alla robotica educativa, al problem solving e alla programmazione, attraverso un approccio pratico e coinvolgente. Lo scopo è ottenere un comportamento desiderato da un sistema (il robot), e nello specifico è quello di permettere al robot di riconoscere oggetti ed ostacoli in un percorso non noto e raggiungere un traguardo opportunamente segnalato.

¹ Docente di Scienze e Tecnologie Informatiche e Laboratori

² <https://www.itilucarelli.edu.it/>

2. Cornice teorica e riferimenti

Il percorso si fonda su approcci riconosciuti nella didattica dell'informatica e del pensiero computazionale:

- **Seymour Papert** e la teoria del *costruzionismo* (Papert, 1980), che sostiene l'apprendimento attraverso la costruzione di oggetti significativi;
- Il concetto di **Computational Thinking** (Wing, 2006), oggi al centro delle Indicazioni Nazionali e delle Linee guida per l'insegnamento delle STEM.

3. Obiettivi

Il progetto mirava a:

- Potenziare le competenze digitali attraverso attività pratiche;
- Sviluppare la capacità di lavorare in gruppo, risolvere problemi e progettare soluzioni;
- Favorire l'integrazione tra creatività e logica, tipica dell'informatica come disciplina scientifica;
- Avvicinare gli studenti alle discipline STEM in modo coinvolgente, inclusivo e accessibile.

Obiettivi principali:

- Sviluppare competenze trasversali nei campi della logica, dell'informatica e della meccanica
- Favorire il lavoro di gruppo, la collaborazione e la creatività
- Introdurre il linguaggio di programmazione a blocchi e le basi della robotica.
- Stimolare l'interesse verso le discipline STEM (Scienza, Tecnologia, Ingegneria e Matematica).

4. Destinatari e struttura del progetto

Il progetto ha coinvolto una ventina di studenti delle classi del primo biennio dell'istituto provenienti da tutti gli indirizzi, prevalentemente dell'indirizzo Informatica, con età compresa tra i 14 e i 16 anni. La durata complessiva è stata di 30 ore, articolate in 10 incontri da 3 ore ciascuno, con una metodologia laboratoriale basata su problem-solving, flipped classroom e cooperative learning. Le attività sono state suddivise in quattro moduli tematici:

- **Modulo 1: Web e Comunicazione Digitale**
 - Introduzione al linguaggio HTML e CSS.
 - ◊ Creazione di una pagina personale e un piccolo sito vetrina.
 - ◊ Uso guidato di WordPress per creare un blog scolastico.
- **Modulo 2: Scratch e Narrazione Interattiva**
 - Introduzione allo storytelling con Scratch.
 - Sviluppo di mini-videogiochi ed esperimenti animati.
 - Ideazione e programmazione in coppie.
- **Modulo 3: Robotica Educativa – Parte 1**
 - Montaggio e configurazione di mBot Ranger.
 - Programmazione di base con blocchi mBlock (movimento, LED, suono).
 - Test su pista e debugging.

• **Modulo 4: Robotica Educativa – Parte 2**

- Uso di sensori a ultrasuoni, luce, linea.
- Sfide a tema: "robot evita ostacoli", "seguipista", "robot messaggero".
- Programmazione con logica condizionale e cicli.

5. Metodologia e strumenti

Tutti i moduli sono stati condotti in **laboratorio di informatica**, con l'uso di:

- PC con sistema operativo Windows e software mBlock;
- Lim, microfoni, schede didattiche stampate e cartellonistica per i percorsi robotici;
- Documentazione delle fasi e cartelle condivise.

La metodologia ha seguito il ciclo "**Immagina – Progetta – Realizza – Migliora**", ispirato all'approccio del *design thinking* e del *learning by doing*.

Gli incontri si sono articolati in una prima fase teorica, seguita da un'applicazione pratica.

Le tematiche proposte, come si evince anche dal timesheet sono state:

- Presentazione del percorso.
- Cenni sul coding e pensiero computazionale.
- La mia prima pagina web: elementi di base HTML, paragrafi, collegamenti ipertestuali, immagini, tabelle.
- Principi della robotica.
- Cenni WordPress, la bacheca, gli articoli e le informazioni di profilo, creiamo e scegliamo un logo.
- Cos'è Scratch? Scopriamo il coding giocando!
 - Primi passi e interfaccia Scratch.
 - Calcolatrice con input con Scratch, storytelling con Scratch.
 - Progettiamo, creiamo e proviamo un semplice circuito elettrico e scriviamo un breve articolo.
 - Proviamo a progettare altri applicativi, altri esempi.
- mBlock - One-Stop Coding Platform for Teaching and Learning.
 - Cominciamo a conoscere mBot Ranger.
 - Esempi applicativi con mBot Ranger.
 - Cominciamo a conoscere mBot Ultimate 2.0.
 - Costruiamo un robot e lo facciamo muovere.
- Conclusioni e saluti.

6. Robotica come palestra di competenze

Il progetto aveva un obiettivo chiaro: coinvolgere attivamente gli studenti attraverso l'uso dei robot programmabili, mettendoli alla prova in contesti di problem solving. Obiettivo raggiunto, e superato, grazie a un percorso fatto di laboratori pratici e sfide appassionanti.

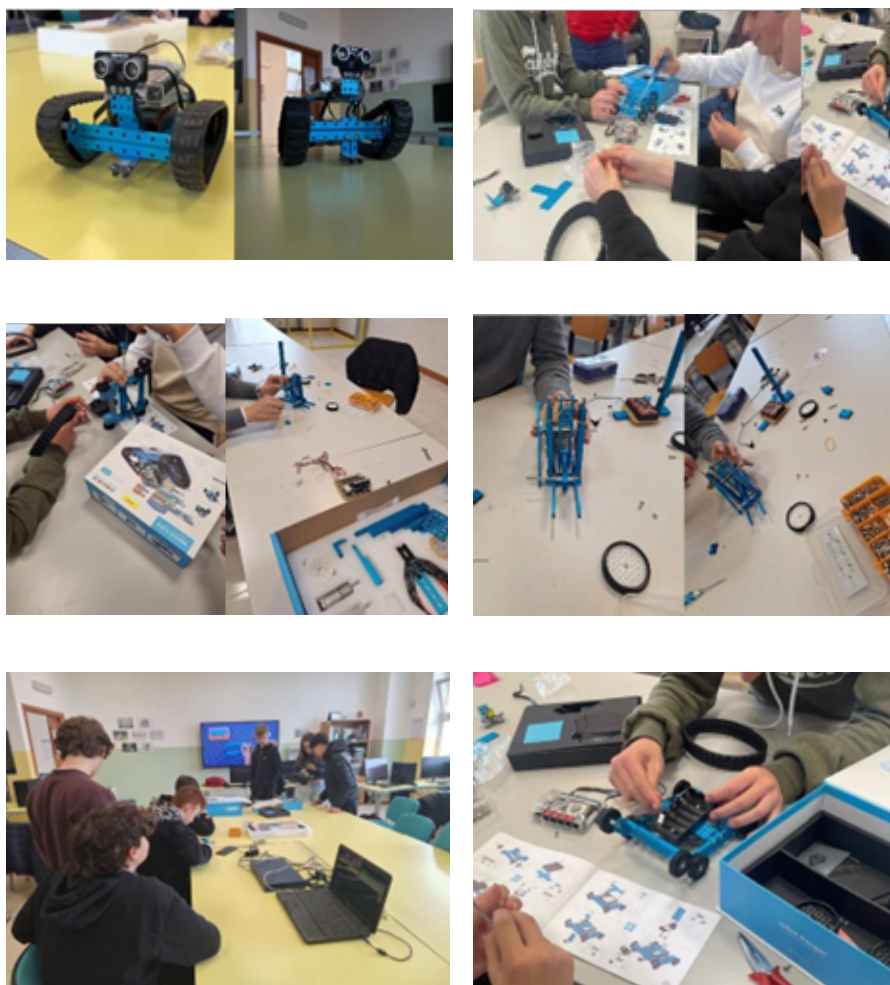


Figura 1: Foto del progetto

I robot venivano programmati per affrontare un ambiente ignoto, riconoscere ostacoli, e raggiungere un traguardo. Questo ha significato per i ragazzi apprendere davvero facendo: logica, meccanica, collaborazione, creatività. Ogni lezione si è trasformata in un'esperienza.

7. Cosa ci ha insegnato questo corso?

In rappresentanza degli studenti si presenta un piccolo contributo di uno di loro, Elia Milozzi che scrive: *“Durante questo corso di robotica il professore Dell'Oste ci ha insegnato le basi della robotica, ci ha fatto vedere come funziona scratch, un sito di programmazione semplificato dove si possono*

creare piccoli giochi e sistemi e abbiamo costruito un vero e proprio robottino telecomandato molto simpatico. Il robot si può comandare tramite il telefono ed ha un sacco di funzioni?

Gli alunni, in un contributo scritto da Gabriele Tino, dicono: *"In queste lezioni abbiamo assemblato il robot Mbot Ultimate 2.0 Catapult Ram. Esso è un piccolo robot telecomandato tramite un'applicazione, MakeBlock, capace di funzionare come una vera e propria catapulta: infatti, tramite un tasto, possiamo lanciare qualsiasi oggetto (ovviamente alla portata del robot)"*.

Ulteriori contributi sono presenti nel blog: <https://lucarellirobot.altervista.org/>.

8. Un mix vincente: HTML, Scratch, mBlock

Durante gli incontri si è partiti dalle basi del coding e del pensiero computazionale, fino ad arrivare a:

- HTML e creazione di pagine web semplici
- WordPress per imparare a comunicare attraverso un blog e per la gestione di contenuti digitali
- Scratch, con giochi e calcolatrici animate per capire la logica a blocchi, oltre che per storytelling, giochi e simulazioni logiche
- mBlock e i robot LEGO mBot Ranger e Ultimate 2.0, per la programmazione di robot con esempi concreti e costruzioni reali

Il mix tra teoria e pratica ha fatto sì che gli studenti non solo apprendessero, ma vivessero la tecnologia. Ogni laboratorio era una fucina di entusiasmo: "costruiamo, programmino, testiamo e miglioriamo". Attraverso esercitazioni pratiche, i ragazzi hanno costruito, programmato e testato robot in grado di riconoscere ostacoli e raggiungere obiettivi prefissati.

9. Risultati e impatti

Il progetto ha avuto un impatto molto positivo sugli studenti coinvolti. Sono stati raggiunti i seguenti risultati, con miglioramenti osservabili in termini di:

- Maggiore autonomia nella risoluzione dei problemi;
- Incremento del pensiero logico e algoritmico;
- Coinvolgimento attivo anche degli studenti più timidi;
- Acquisizione di vocabolario tecnico-informatico di base.
- Maggiore motivazione nell'apprendimento delle discipline tecniche.
- Aumento delle competenze digitali e di problem solving.
- Rafforzamento della capacità di lavorare in gruppo e comunicare in modo efficace.
- Partecipazione attiva e creativa alle attività laboratoriali.

La robotica educativa non è solo un modo per insegnare a programmare. È un ponte verso le competenze del XXI secolo. Stimola il pensiero critico, la creatività, la capacità di lavorare in team. E nel nostro caso ha dimostrato di essere anche una formidabile leva motivazionale. Con progetti come LucarelliRobot, la scuola si trasforma da luogo di nozioni a laboratorio del futuro.

L'esperienza ha avuto un forte impatto educativo:

- Tutti gli studenti hanno partecipato con interesse e costanza.
- Nessun episodio disciplinare, segno di forte motivazione.
- Le competenze digitali e trasversali sono risultate rafforzate.
- Il gruppo ha sviluppato spirito critico e capacità di risoluzione dei problemi.
- La robotica si è confermata una strategia vincente per l'innovazione didattica.

Sono stati raccolti dati qualitativi tramite gli elaborati finali:

- codice Scratch.
- pagine HTML.
- portale in WordPress.
- video e foto dei robot in azione.

10. Conclusioni

“LucarelliRobot” ha rappresentato un'esperienza significativa e replicabile per introdurre l'informatica come scienza e come linguaggio creativo. La multidisciplinarietà dei moduli ha permesso di sviluppare competenze trasversali, aumentando la motivazione e la fiducia degli studenti nelle proprie capacità. Nel futuro si intende rafforzare il lavoro interdisciplinare con lettere e matematica, in ottica STEAM, oltre che provare ad ampliare il progetto al triennio con componenti avanzate di intelligenza artificiale e IoT.

Parole chiave:

- Robotica Educativa
- STEM
- mBot
- Scratch
- mBlock
- Didattica innovativa
- Competenze digitali
- Problem Solving
- Coding
- Didattica laboratoriale

Ringraziamenti

Si ringraziano tutti gli studenti dell'ITI “G.B.B. Lucarelli” di Benevento che hanno partecipato al progetto “LucarelliRobot”, contribuendo con entusiasmo e impegno alla sua realizzazione. Un sentito ringraziamento va inoltre ai referenti del progetto, al tutor, al Dirigente Scolastico, alla DSGA e al personale ATA, per il supporto organizzativo e logistico che ha reso possibile lo svolgimento delle attività.

Bibliografia

- [1] Sito ufficiale e canali social dell'ITI G.B.B. Lucarelli - <https://www.itilucarelli.edu.it/>.
- [2] Scratch - <https://scratch.mit.edu/>
- [3] mBlock - <https://mblock.makeblock.com/>
- [4] Papert, S. (1980). Mindstorms: Children, Computers, and Powerful Ideas. Basic Books.
- [5] Wing, J. M. (2006). Computational Thinking, Communications of the ACM, 49(3).
- [6] MIUR. (2022). Linee guida STEM e competenze digitali (Piano "Futuramente").
- [7] D.M. 65/2023 Azioni di potenziamento STEM e multilinguismo - <https://www.istruzione.it/pnrr/>

Educational Promptization: una nuova prospettiva per l'insegnamento dell'Informatica nella scuola secondaria

Erica Perseghin¹

¹ Università degli Studi di Udine

perseghin.eric@spes.uniud.it

¹IIS Euganeo (Este, PD)

erica.perseghin@scuola.istruzione.it

Abstract

Questo articolo mette in luce opportunità e criticità dell'insegnamento dell'informatica mediante l'impiego dell'intelligenza artificiale (AI) come strumento didattico. Viene proposta un'analisi dei prompt elaborati da studenti del triennio di un istituto tecnico per la scrittura e la modellazione di esercizi di programmazione. I risultati evidenziano che l'utilizzo delle GenAI senza una formazione specifica non costituisce necessariamente un vantaggio, né per chi ha nozioni limitate né per chi possiede solide conoscenze pregresse. L'esperienza condotta sottolinea la necessità di affrontare criticamente i programmi ministeriali di informatica, valutando se e come gli strumenti generativi possano effettivamente supportare l'apprendimento. Sebbene le GenAI siano in grado di generare elementi di codice anche complessi, la loro efficacia dipende fortemente dal livello di competenza degli utenti. L'articolo conclude che un utilizzo consapevole nella scuola secondaria può offrire nuove prospettive didattiche, purché integrato in un percorso formativo che valorizzi le conoscenze tramite un'educazione al prompting in continuità con quanto già acquisito dagli studenti.

1. Introduzione

L'evoluzione delle tecnologie educative ha portato a un rinnovato interesse verso approcci pedagogici che valorizzano il ruolo attivo dello studente nel processo di apprendimento [18] [27]. In questo contesto, l'avvento dei modelli linguistici di grandi dimensioni (LLM) ha aperto nuove possibilità alla creazione di ambienti di apprendimento intelligenti che supportano la costruzione autonoma del sapere [24] [28] [32]. Tra le tecnologie emergenti, in particolare ChatGPT, si afferma come strumento predominante:

- nell'apprendimento personalizzato per tutoraggio mirato [19], piani di studio individualizzati [10] e revisione compiti;
- nel problem-solving per creazione di scenari e risoluzione di problemi complessi [5] [7];
- nel project-based learning.

Nonostante la crescente ricerca in ambito educativo, manca ancora un consenso sulla sua efficacia nel migliorare le performance degli alunni. Diversi studi evidenziano criticità

significative come generazione di risposte errate e rischi di plagio [1], riduzione delle capacità e dell'interazione sociale [31], oltre a problemi di aggiornamento informativo che possono instillare allucinazioni. Gli LLM, addestrati su dataset generici e non specializzati, spesso producono contenuti non pertinenti per contesti educativi specifici, rendendo necessarie strategie di pre-training e fine-tuning mirate ai bisogni educativi [29].

L'indagine [4] rileva che l'84% su 1813 studenti delle scuole secondarie ha già utilizzato GenAI per creare contenuti didattici, ma solo il 18% ha ricevuto indicazioni istituzionali chiare sull'uso dell'AI. Mentre alcune università [15] [30] sviluppano linee guida specifiche, queste tardano a raggiungere i livelli scolastici inferiori, eccetto alcune indicazioni generali [14].

In questo scenario incerto, il prompt engineering emerge come pratica immediata per l'uso consapevole degli strumenti di GenAI, identificando best practices per generare risposte accurate [2] [28]. [11] sottolinea l'importanza di padroneggiare questa abilità per evitare output imprecisi e allucinazioni dannose. Gli educatori dovrebbero quindi fornire strategie di prompting efficaci per migliorare la qualità delle interazioni e favorire l'apprendimento creativo. Infatti, studi precedenti

[3] hanno dimostrato che anche studenti universitari, pur essendo programmatori esperti, possono fornire prompt minimi, includendo formule di cortesia come nell'interazione umana. Questo comportamento, pur rendendo naturale la conversazione e riflettendo strategie di role-playing preferite dagli studenti [22] genera output imprecisi [17], costi computazionali significativi [16] e possibili bias di conferma [21].

2. Metodologia Sperimentale

L'attività sperimentale è stata condotta con studenti delle classi terza, quarta e quinta di un istituto tecnico informatico, già in possesso di competenze di base in Python e Java. L'unità, svolta in 3 ore distribuite su più mattinate verso fine aprile 2024, si è articolata in due moduli: uno introduttivo, orientato agli aspetti teorici dei LLM (token, meccanismi di attenzione e strumenti visuali come BertViz o Transformer Explainer), e uno operativo, dedicato alle tecniche di prompt engineering (zero, few, one shot e Chain of Thinking (CoT)). Sono state condivise regole base per la scrittura di prompt efficaci [11] (specificità, contesto, target) e in seguito, è stata avviata una riflessione critica sui risultati generati. In funzione alle tematiche e al grado di approfondimento, l'attività è stata integrata nei percorsi curricolari (es. informatica, sistemi reti, educazione civica) o proposta in contesti extracurricolari (didattica orientativa) a seconda della classe. Infatti, in accordo con i colleghi docenti, si è scelto di non approfondire gli aspetti implementativi del codice, poiché l'istituto ha già integrato tali contenuti nella disciplina "tecnologie e progettazione di sistemi informatici e di telecomunicazioni" per le classi quarta e quinta. L'approccio didattico ha privilegiato la comprensione attraverso esempi concreti e applicazioni operative del prompt engineering, piuttosto che lo studio teorico delle architetture dei modelli. La raccolta dei prompt simulati per l'interazione con modelli GenAI è stata effettuata mediante Google Form, rendendo l'attività replicabile anche in contesti con risorse limitate, purché vi sia una preparazione di base e accesso minimo a dispositivi connessi. L'intervento mira a promuovere il confronto collettivo sulle strategie di prompting, sull'elaborazione critica dei prompt e sulla valutazione delle risposte, sviluppando competenze nell'ambito dell'AI. Studi [8] [12] dimostrano che le strategie di prompting influenzano significativamente le prestazioni nei task di programmazione, favorendo l'acquisizione di competenze tecniche e lo sviluppo di

un pensiero computazionale consapevole. L'intervento è stato quindi rivolto agli studenti del triennio per introdurre queste tecniche in una fase avanzata del percorso formativo, quando possiedono già le competenze di base necessarie.

In questo contesto, [6] propone il concetto di “Educational Promptization”, enfatizzando l'importanza dell'analisi critica delle risposte ottenute. L'uso efficace dei prompt comporta, oltre alla loro attenta progettazione, un'attività di interpretazione e di riflessione come in [20] dove si dimostra l'efficacia superiore dei prompt zero-shot nel ragionamento logico, con esempi come “Considera le probabilità di A e B” che superano prompt generici come “Descrivi passo dopo passo”. In aggiunta, l'uso corretto dei prompt può strutturare in modo sequenziale il percorso di apprendimento, supportando gli studenti nell'esecuzione graduale di compiti complessi.

Tuttavia, esiste un rischio elevato di prompt non formulati adeguatamente così come il rischio di un'accettazione non ponderata delle risposte fornite dall'AI. Gli insegnanti dovrebbero lavorare sulla risoluzione creativa dei problemi e sull'analisi metacognitiva come elementi chiave per il futuro come emerso in [25].

L'esperimento proposto ha collezionato prompt anonimi di 32 studenti del terzo e quarto anno, redatti in linguaggio naturale, in risposta a una specifica richiesta del docente. L'acquisizione dei dati è avvenuta attraverso consegne strutturate, pensate per favorire chiarezza e replicabilità. A seguito della produzione, è stata condotta un'analisi qualitativa dei prompt, applicando il modello CLEAR (Conciso, Logico, Esplicativo, Adattivo, Riflessivo), come analizzato in [9], utile a indagare la qualità e la coerenza dei prompt valutando il processo iterativo attraverso cinque indicatori esplicitati in Tabella 1. Non sono stati somministrati pre-test, in quanto tutti i partecipanti disponevano di una preparazione informatica di base. Tuttavia, la durata dell'intervento ha comportato alcuni limiti come l'impossibilità di monitorare l'evoluzione delle competenze nel tempo e l'analisi di prompt su task focalizzati solo sulla programmazione.

Indicatore	Livello 1 (Inziale)	Livello 2 (In sviluppo)	Livello 3 (Competente)	Livello 4 (Avanzato)
CONCISO	Linguaggio generico, terminologia imprecisa	Prompt concisi, terminologia tecnica limitata	Prompt ben strutturati, sintesi efficace e terminologia appropriata	Prompt precisi, linguaggio tecnico esperto e massima efficienza comunicativa
LOGICO	Concetti sconnessi, nessuna strutturazione, approccio casuale	Strutturazione parziale, alcune relazioni logiche identificate	Sequenza logica chiara, relazioni concettuali ben definite	Architettura logica complessa, interconnessioni evidenti
ESPLICATIVO	Richieste vaghe	Output parzialmente mirati, alcune imprecisioni	Output precisi, minimizzazione delle risposte non pertinenti	Eliminazione di ambiguità e output ultra-specifici
ADATTIVO	Nessuna personalizzazione	Adattamenti minimi, flessibilità limitata	Buona personalizzazione e adattamento al contesto	Personalizzazione avanzata, adattamento dinamico continuo

RIFLESSIVO	Nessuna riflessione metacognitiva, interazione superficiale	Riflessione limitata, analisi parziale del processo	Riflessione strutturata, analisi accurata delle scelte	Riflessione profonda
-------------------	---	---	--	----------------------

Tabella 1: Tabella di valutazione tramite modello CLEAR

In base alla Tabella 1, un esempio di prompt di Livello 1 è “Questo codice non funziona, puoi correggerlo?” mentre un esempio di livello 4 è “Analizza gli errori nella funzione X. Determina la tipologia di errore ed implementa una soluzione che filtri i valori. Gestisci le eccezioni con try- catch, valida l’input prima del calcolo e restituisci un messaggio di errore tramite la funzione X. Spiega le tue scelte”. La programmazione assistita dai LLM offre nuove opportunità didattiche, ma resta insufficiente nel gestire la complessità dei problemi reali. Per questo, è essenziale l’intervento umano nella definizione dei compiti, nella progettazione dei prompt e nella valutazione qualitativa del codice prodotto.

3. Analisi Collaborativa e Sviluppo del CT

L’analisi collaborativa di un codice corrotto e mai visto stimola competenze chiave del critical thinking come astrazione, scomposizione in sotto-problemi, debugging, generalizzazione e iterazione. Il processo opera tramite un meccanismo iterativo (Q&A), in cui si sviluppa il problem solving attraverso l’identificazione di pattern, l’applicazione di soluzioni note e la costruzione di nuove strategie. In questo contesto, il prompt engineering agisce come strumento metacognitivo e sociale, promuovendo collaborazione tra pari e riflessione tramite processi iterativi simili al CoT, rendendo lo studente agente attivo dell’apprendimento.

L’analisi dei prompt rivela al docente la padronanza disciplinare, gli interessi e i processi metacognitivi dello studente. Contemporaneamente, gli studenti sviluppano competenze di valutazione critica delle risposte AI, identificando errori e ridefinendo i prompt per ottimizzare i risultati. Il docente assume un ruolo di scaffolding [23], facilitando la creazione di prompt coerenti e guidando la riflessione sugli output anche tramite il modello CLEAR.

Come illustrato in Tabella 2, oltre la metà dei prompt raccolti (dataset reperibile al link) risultava poco chiara omettendo elementi fondamentali come il linguaggio richiesto, il tipo di dato in input o l’utilizzo di una funzione per verificare se il numero era primo. I test condotti con GenAI mostrano infatti che l’efficacia dei prompt dipende sia dalla padronanza del linguaggio sia dalla precisione delle istruzioni fornite.

Durante l’esercitazione è emersa anche una scarsa definizione dei criteri di correttezza: richieste generiche come “non usare formule specifiche”, risultavano troppo vaghe per orientare correttamente il modello. L’analisi dei prompt tramite il modello CLEAR ha evidenziato livelli medi tra 1 e 2, con criticità ricorrenti nei seguenti indicatori:

- Conciso: prevale un linguaggio colloquiale, talvolta ambiguo;
- Logico: la struttura delle istruzioni è mancante o solo parzialmente coerente;
- Esplicativo: gli output richiesti sono spesso indefiniti o sottintesi, senza vincoli espliciti.

Consegna	Supponi di voler utilizzare una GenAI per creare un problema di programmazione in Python. Il problema dovrebbe richiedere all'utente di fornire un numero intero e, successivamente, di determinare se tale numero è primo, restituendo True o False. Come scriveresti il prompt?					
Esempi di Prompt di Studenti Non Chiari o Incompleti	C (Conciso)	L (Logico)	E (Esplicitivo)	A (Adattivo)	R (Riflessivo)	Note
Crea un Python che prenda in input un numero e risponda se è primo o meno, senza considerare la condizione di vero o falso	2	1	1	1	1	L'output atteso non è chiaro: manca il contesto e non specifica vincoli né obiettivi.
Scrivi un codice che prende un numero intero in input e rispondi con true se è primo se no false	2	2	1	1	1	Formula incompleta: manca gestione casi limite e struttura della funzione. Sintassi approssimativa.
#genera un codice che prende in input un numero intero e restituisci true se è un numero primo e false se non lo è	2	2	1	1	2	Privo di gestione di casi limiti o vincoli tecnici.
Dato un numero, scrivi un programma che restituisca True se il numero è primo, altrimenti False	2	2	1	1	1	Mancano specifiche tecniche
Python. Dato un numero intero restituisci se true o false	1	1	1	1	1	Linguaggio telegrafico, ambiguità nella struttura e assenza di istruzioni chiare

Tabella 2: Valutazione di prompt secondo il modello CLEAR in base a un esempio di consegna che mira ad analizzare l'efficacia di un prompt su un task in Python.

Il modello CLEAR dimostra che una mancanza di specifiche tecniche, contesto e parametri di performance impedisce una corretta interpretazione da parte della GenAI ed allo stesso tempo, riflette una debole metacognizione da parte degli studenti. Un esempio di prompt opportunamente riformulato e strutturato secondo gli indicatori di Tabella 1 potrebbe essere il seguente: “Scrivi un programma in Python che chieda all'utente di inserire un numero intero tramite input(). Il programma deve contenere una funzione parametrica chiamata num_primo(num) che verifichi se il numero è primo. Se lo è, deve restituire True, altrimenti False. Assicurati di gestire anche i casi limite, come numeri minori o uguali a 1, e inserisci dei commenti nel codice per spiegare la logica utilizzata.”

Consegna	Immagina di avere un codice Python di grandi dimensioni che presenta problemi di lentezza nell'esecuzione. Scrivi un prompt per chiedere alla GenAI come ottimizzarlo in termini di performance, mantenendo la correttezza del codice.
Esempi di Prompt di Studenti Non Chiari o Incompleti	Ottimizza il codice rendendolo più veloce
	Ottimizza l'algoritmo copiato in precedenza e rendilo più performante e dinamico
	Questo script è troppo lento a eseguire. Come posso ottimizzarlo rendendolo più veloce?
	Fai in modo che venga eseguito più velocemente
	Elabora il seguente script dando un output veloce ma chiaro e conciso
	Rendilo più chiaro, semplice e breve

Tabella 3: Esempio di consegna che mira a richiedere maggiore efficienza di uno script in Python

Un’ulteriore analisi è stata condotta con un task più complesso che ha messo in evidenza difficoltà significative poiché si richiedeva una maggiore capacità di analisi, competenze tecniche e precisione espressiva come emerge dalla Tabella 3. Grand parte degli studenti¹ non è riuscito a formulare prompt completi e precisi. Le richieste risultavano spesso generiche, ambigue o carenti di vincoli tecnici. Un esempio di prompt riformulato, costruito secondo gli indicatori di Tabella 1, potrebbe essere il seguente:

Analizza il seguente codice Python e prova a ottimizzarlo migliorando la complessità temporale, limitando l’uso della memoria a X MB e utilizzando l’ambiente di sviluppo XX. Mantieni la stessa funzionalità del programma e cerca di renderlo leggibile, fornendo anche dei benchmark prima e dopo l’ottimizzazione per confrontare le prestazioni.

In sintesi, il prompt proposto, allineato alle conoscenze degli studenti, dovrebbe specificare l’ambiente di esecuzione, il tempo previsto, il tipo di elaborazione richiesta e la dimensione dei file di input, in modo da consentire un confronto efficace tra le soluzioni prodotte.

L’analisi attraverso il modello CLEAR mette in luce come la capacità di elaborare prompt efficaci e specifici sia ancora limitata tra gli studenti, che tendono a formulare richieste generiche o poco strutturate, faticando a comprenderne pienamente il funzionamento e compromettendo la qualità dell’output prodotto. Da queste criticità emerge il potenziale della GenAI nel favorire apprendimento personalizzato e collaborazione, se adeguatamente guidata. Tuttavia, un’integrazione efficace richiede interventi formativi mirati, aggiornamento continuo e attenzione agli aspetti etici [13].

¹ Dataset disponibile al link https://uniudamce-my.sharepoint.com/:x:/g/personal/123767_spes_uniud_it/EQM4eFSdSHpJr2rtTjQz20sBwbSqiqs9DjHiox1LYkQ0w?rttime=rqya2gD23Ug

4. Conclusione

L'indagine ha valutato l'uso dei prompt nell'insegnamento dell'informatica nel triennio di un istituto tecnico superiore, valutando le competenze di base degli studenti mediante scenari diversificati con prompt teorici e pratici. Il processo ha permesso la diffusione di buone pratiche per la progettazione di prompt efficaci, seguito da un'analisi critica delle produzioni degli alunni, considerando le dinamiche reali dell'interazione uomo-GenAI. Le consegne sono state formulate come istruzioni chiare e concise, simili a quelle dei manuali didattici, affinché i prompt fossero progettati per essere facilmente verificabili e analizzabili sia da studenti che docenti. I risultati evidenziano il potenziale dei LLM nel fornire supporto adattivo in tempo reale, migliorando la definizione degli obiettivi e la pianificazione educativa. Questi esperimenti costituiscono un invito all'azione per la comunità di ricerca educativa a intensificare l'esplorazione dei LLM nei contesti formativi.

Tuttavia, l'implementazione dei LLM presenta ancora criticità significative. Le preoccupazioni riguardano privacy, accesso diseguale agli strumenti, uso responsabile da parte di studenti e docenti [14] [26] e questioni etiche (bias nei modelli, necessità di trasparenza nello sviluppo e impatti ambientali). Inoltre, l'utilizzo commerciale comporta costi di abbonamento per piattaforme specifiche come ChatGPT Edu e Claude for Education.

Con educational promptization, si estende il concetto di CT come processo di astrazione, analisi e interazione con sistemi AI al fine di trovare delle soluzioni che trascendono l'informatica tradizionale [9]. In questo processo, lo studente formula problemi, crea prompt, ridefinisce output basati sulle risposte AI, collabora e trova soluzioni per raggiungere risultati desiderati.

L'analisi dimostra che le GenAI non dovrebbero essere implementate arbitrariamente, ma utilizzate secondo il livello della classe e il modello di apprendimento, risultando particolarmente efficaci negli approcci problem-based. Naturalmente, si considera ancora necessaria una valutazione regolare da parte degli insegnanti per prevenire la dipendenza tecnologica e promuovere il mantenimento delle conoscenze. Emerge però la crescente necessità di potenziare le iniziative educative per consentire un'interazione efficace ed etica con le nuove AI.

Bibliografia

- [1] Alier, Marc, Francisco José García-Peñalvo, and Jorge D. Camba. "Generative Artificial Intelligence in Education: From Deceptive to Disruptive." *International Journal of Interactive Multimedia and Artificial Intelligence* 8(5):5 (2024).
- [2] Cain, William. "Prompting Change: Exploring Prompt Engineering in Large Language Model AI and Its Potential to Transform Education." *Springer TechTrends Issue 1/2024*. (2024).
- [3] Denny, Paul, Sami Sarsa, Arto Hellas, and Juho Leinonen. "Automatic Generation of Programming Exercises and Code Explanations Using Large Language Models". In *Proceedings of the 2022 ACM Conference on International Computing Education Research - Volume 1 (ICER '22)*, Vol. 1. Association for Computing Machinery, New York, NY, USA, 27–43. (2022). <https://doi.org/10.1145/3501385.3543957>
- [4] Generazioni Connesse <https://www.generazioniconnesse.it/site/it/2025/01/02/sid-safer-internet-day-2025/>. Accesso: 26 maggio 2025.

- [5] Ghadeer, Sawalha, Imran Taj, and Abdulhadi Shoufan. "Analyzing Student Prompts and Their Effect on ChatGPT's Performance." *Cogent Education* 11.1 (2024). DOI: 10.1080/2331186X.2024.2397200.
- [6] Haugsbakken, Halvdan, and Marianne Hagelia. "A New AI Literacy for the Algorithmic Age: Prompt Engineering or Educational Promptization?" 4th International Conference on Applied Artificial Intelligence (ICAPAI) (2024): 1–8. DOI: 10.1109/ICAPAI61893.2024.10541229.
- [7] Hieke, Keuning, et al. "Goodbye Hello World: Research Questions for a Future CS1 Curriculum." 24th Koli Calling International Conference on Computing Education Research (2024). <https://doi.org/10.1145/3699538.3699591>
- [8] Hou, Wenpin, and Zhicheng Ji. "Comparing Large Language Models and Human Programmers for Generating Programming Code." *Advanced Science* 12.8 (2025). DOI: 10.1002/advs.202412279.
- [9] Hsu, Hsiao-Ping. "From Programming to Prompting: Developing Computational Thinking Through Large Language Model-Based Generative Artificial Intelligence." *TechTrends* (2025). <https://doi.org/10.1007/s11528-025-01052-6>
- [10] Kehui, Tan, Jiayang, Yao, Chenyou Fan, and Yu, Song. "ELF: Educational LLM Framework of Improving and Evaluating AI Generated Content for Classroom Teaching". *J. Data and Information Quality* (2025). <https://doi.org/10.1145/3712065>
- [11] Lin, Zhiwei. "How to Write Effective Prompts for Large Language Models." *Nature Human Behaviour* (in press, 2024). <https://osf.io/r78fc/download/?format=pdf>
- [12] Maity, Subhankar, Deroy, Aniket and Sarkar, Sudeshna. "Can large language models meet the challenge of generating school-level questions?" *Computers and Education: Artificial Intelligence*, 8, 100370. (2025). <https://doi.org/10.1016/j.caeai.2025.100370>
- [13] Martini, Simone. "Teaching Programming in the Age of Generative AI". *Proceedings of ITiCSE 2024, Association for Computing Machinery, New York, NY, USA*, 1–2. (2024). <https://doi.org/10.1145/3649217.3653527>
- [14] Miao, Fengchun, and Wayne Holmes. *Guidance for Generative AI in Education and Research*. UNESCO, 2023.
- [15] MIT "Guidance for use of generative AI tools". <https://ist.mit.edu/ai-guidance>. Accesso: 26 maggio 2025.
- [16] New York Times. "Saying 'Thank You' to ChatGPT Is Costly. But Maybe It's Worth the Price." <https://www.nytimes.com/2025/04/24/technology/chatgpt-alexa-please-thank-you.html>. Accesso: 26 maggio 2025.
- [17] Oppenlaender, Jonas, Rhema Linder, and Johanna Silvennoinen. "Prompting AI Art: An Investigation into the Creative Skill of Prompt Engineering." *arXiv preprint* (2024). <https://doi.org/10.48550/arXiv.2303.13534>
- [18] Piaget, Jean. "Cognitive Development in Children: Piaget Development and Learning." *Journal of Research in Science Teaching* 2 (1976): 176–186. <https://doi.org/10.1002/tea.3660020306>
- [19] Radhika, Makharia et al., "AI Tutor Enhanced with Prompt Engineering and Deep Knowledge Tracing," 2024 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI), Gwalior, India, pp. 1-6. (2024) doi: 10.1109/IATMSI60426.2024.10503187.

- [20] Ryu, Jongwon, Jungeun Kim, and Junyeong Kim. "A Study on the Representativeness Heuristics Problem in Large Language Models." *IEEE Access* 12 (2024): 147958–147966. <https://doi.org/10.1109/ACCESS.2024.3474677>
- [21] Sarsa, Sami, Denny, Paul, Prather, James, Becker, Brett, Finnie-Ansley, James, Hellas, Arto et al. "Computing Education in the Era of Generative AI." *Communications of the ACM*, 67(2), 56–67. (2024). <https://doi.org/10.1145/3624720>
- [22] Shao, Yunfan, Linyang Li, Junqi Dai, and Xipeng Qiu. "Character-LLM: A Trainable Agent for Role-Playing". In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13153–13187, Singapore. (2023).
- [23] Shimizu, Amanda, Yoshiko, Blaine, Smith, Burriss, Sarah K., Hundley, Melanie, and Pendergrass, Emily. "Multimodal composing with generative AI: Examining preservice teachers' processes and perspectives". *Computers and Composition*, 75, 102896. (2025). ISSN 8755-4615. <https://doi.org/10.1016/j.compcom.2024.102896>
- [24] Snow, Lauren, Nixon, Nia and Lin, Yiwen. "Catalyzing Equity in STEM Teams: Harnessing Generative AI for Inclusion and Diversity". *Policy Insights from the Behavioral and Brain Sciences*, 11(1), 85-92. 2024. <https://doi.org/10.1177/23727322231220356>
- [25] Tassoti, Sebastian. "Assessment of Students' Use of Generative Artificial Intelligence: Prompting Strategies and Prompt Engineering in Chemistry Education." *Journal of Chemical Education* 101.6 (2024): 2475–2482. DOI: 10.1021/acs.jchemed.4c00212
- [26] UNESCO. L'etica dell'IA: Modellare il futuro delle nostre società. 2023 <https://www.unesco.it/wp-content/uploads/2023/11/Brochure-su-Raccomandazione-UNESCO-sullIntelligenza-Artificiale.pdf> Accesso: 26 maggio 2025.
- [27] Vygotskij, Lev Semënovič. *Pensiero e linguaggio*. Giunti, 2010.
- [28] Wang, Shen, Tianlong Xu, Hang Li, Chaoli Zhang, Joleen Liang, Jiliang Tang, Philip S. Yu and Qingsong Wen: "Large Language Models for Education: A Survey and Outlook". (2024) <https://doi.org/10.48550/arXiv.2403.18105>
- [29] Xing, Wei, et al. "The Use of Large Language Models in Education." *International Journal of Artificial Intelligence in Education* (2025). <https://doi.org/10.1007/s40593-025-00457-x>
- [30] Yu, Ji Hyun, An, Yunjo and James, Shadarra. "Investigating the higher education institutions' guide- lines and policies regarding the use of generative AI in teaching, learning, research, and administration". *International Journal of Educational Technology in Higher Education*, 22, 10. (2025).f. <https://doi.org/10.1186/s41239-025-00507-3>
- [31] Zhang, Peng, and Gary Tur. "Empowering Learner Agency Through e-Portfolio Co-Design." *Grupo Kiobos Editorial* (2024).
- [32] Zapata, Gabriela C. C., et al. "The Role of AI Feedback in University Students' Learning Experiences." *Ubiquitous Learning: An International Journal* 18.2. (2024). <https://doi.org/10.18848/1835-9795/CGP/v18i02/1-30>

“Chi ha paura di un LLM”: un approccio psico-didattico all’integrazione di un LLM nelle lezioni di informatica al triennio

Michele Iacobellis¹

¹ Colamonico - Chiarulli

michele.iacobellis@gmail.com

Abstract¹

Da quando i modelli di intelligenza artificiale generativa sono stati resi disponibili al grande pubblico, nel 2023, abbiamo potuto osservare con attenzione i principali caso d’uso attuati inconsapevolmente dai ragazzi d’ogni età, frequentanti la scuola dell’obbligo e oltre. Tra questi, il caso d’uso “vincente” è stato certamente quello della “mente surrogata”. Gli LLM si sono dimostrati abili nel legarsi ai recettori di un’apparente *accidia*, alimentata da insicurezze e fragilità variegate, diffusa caratteristica di molti degli studenti frequentanti gli istituti tecnici ad indirizzo informatico. La scarsa destrezza nell’affrontare le difficoltà dello studio, la debole o inesistente capacità di gestire la frustrazione del fallimento momentaneo, la diffusa credenza che l’errore sia il male assoluto, portano migliaia di studenti a delegare lo sforzo cerebrale al GPT di turno, perdendo capacità di pensiero in favore del surrogato digitale. Per questo, *il surrogato digitale, come individuo, deve entrare a far parte della classe e diventarne l’ineguagliabile modello di riferimento, da imitare asintoticamente.*

1. Introduzione

L’intelligenza artificiale generativa mantiene da tre anni circa la sua caratteristica di “disruption”. Questa, al momento, si ripercuote non tanto su aspetti lavorativi, commerciali e di mercato, quanto piuttosto sulle abitudini umane stesse, dando luogo ad evidenti alterazioni d’abito, dunque di natura squisitamente antropologica². È ormai acquisito, per personale osservazione diretta, che i ragazzi hanno colto immediatamente, in modo conscio o inconscio, la possibilità di “barare”, di impiegare l’AI generativa per poter svolgere compiti e attività in modo più rapido, meno oneroso in termini di energie psichiche e intellettive: il minor sforzo con il massimo risultato³.

Questa pratica si è consolidata nel tempo, è stata acquisita come prassi ed abitudine⁴, consente di “vincere” nelle tipiche dinamiche scolastiche, ma espone, come ho rilevato sul campo,

1 Nessuna porzione di testo di questo articolo è stata scritta con un LLM o AI.

2 Il garante privacy è intervenuto in questo senso (si veda [10]).

3 “[...] *nearly 80% of students felt using an LLM was [...] cheating [...]*” (si veda [11], p. 2).

4 Si veda [12] per una statistica relativa ai “*college students*” da me ipotizzata estendibile agli studenti di secondaria di secondo grado, sulla base delle osservazioni dirette mie e dei colleghi.

a gravi rischi nelle circostanze competitive esterne al contesto scuola, ma ad esso legate (gare di programmazione competitiva, challenge di cyber sicurezza, ecc.). In aggiunta, conseguenza più odiosa, incide negativamente sulle capacità cognitive⁵ del giovane, proprio nel periodo in cui l'essere umano è più reattivo agli sforzi compiuti per accrescere conoscenze, competenze ed abilità.

2. Il prologo motivazionale

“Il *pathos* è l'inizio del pensiero. L'intelligenza artificiale è *apatica*, vale a dire senza *pathos*, senza *passione*. Essa *calcola*.”⁶

Il problema fondamentale del docente è attivare nei propri studenti le energie psichiche loro necessarie per poter fornire *attenzione* e *sforzo*, durante l'ora di lezione, cercando di tenere tali energie attive anche per il lavoro a casa, oramai un'utopia.

Il *primo punto di innesto* di un LLM è proprio questo: lo studente si trova in un certo stato emotivo e intellettuale, da cui, *spendendo energia* prevalentemente psichica, deve muoversi lungo un percorso di varia difficoltà, per giungere all'obiettivo di apprendimento che il docente ha fissato.

In generale, l'intensità dello sforzo richiesto è nota, mediamente omogenea per tutti gli studenti, stimata dal docente che progetta l'attività. Ogni studente, tuttavia, risponde in modo differente e, inoltre, parte da una composizione di stati interiori diversa. L'intelligenza artificiale generativa (LLM) si intromette *distruttivamente* nel processo di passaggio dal punto di partenza all'obiettivo, poiché riduce di fatto lo *sforzo effettivo* e lo *sforzo percepito*. Lo studente, in relazione ad un'attività non-spontanea:

- non impara a riconoscere e gestire la composizione di stati d'animo di partenza,
- non impara ad uscirne per poter affrontare l'attività assegnatagli,
- non impara ad attribuire un costo energetico e temporale ad un task.

Emergono tre asserti fondamentali che governano i primi cinque o dieci minuti della vita degli studenti in aula, all'inizio dell'ora di lezione. Tuttavia, questi asserti sono applicabili anche allo studente che, a casa, deve decidere quanto tempo dedicare ai compiti, piuttosto che al resto.

È ipotizzabile che esistano diversi tipi di studente che qualificiamo come “*prototipo*” (differenziati per attitudini, grado di motivazione, indole, tipo psicologico, ecc.). Tra questi scegliamo quello che potrebbe presentarsi inizialmente problematico, a causa di una temporanea carenza di motivazione, ma che dispone di energia psichica latente, risvegliabile e reindirizzabile mediante un preliminare momento motivazionale, prima del corpo della lezione.

Questo studente “*prototipo*” appare generalmente immerso in un “*blob*” che rappresenta un confuso e denso stato emotivo in cui *noia*, *pigrizia*, *accidia*, *disinteresse*, *identità indefinita* (a cui si associa un'autostima dai contorni confusi) e *inerzia* si combinano costruttivamente, intrappolando il soggetto all'interno di una capsula invisibile, come in Figura 1.

⁵ Si veda [13], articolo di recente pubblicazione, accolto in modo controverso.

⁶ Byung-Chul Han, “Le non cose”, p. 46. Corsivi dello stesso autore del libro.

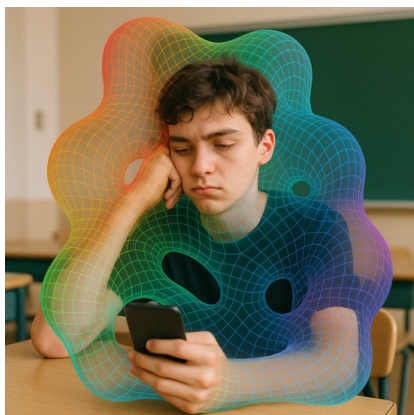


Figura 1 – Immagine che rappresenta uno studente nello stato di partenza⁷

Una volta assegnato il task, quanto è difficile “*lasciare*” questo stato (ed il relativo strumento di auto-appagamento) per entrare in attività, per innescare il meccanismo virtuoso che conduce all’acquisizione di abilità, competenze e conoscenze! Il costo energetico è alto, l’innescio è ancor più impegnativo. Ecco i tre asserti fondamentali, “*osservati*” dal punto di vista dello studente:

- Io imparo a *riconoscere* la composizione di stati d’animo di partenza,
- Io imparo ad *uscirne*,
- Io imparo ad *attribuire* un costo all’atto.

Tuttavia, *quanta consapevolezza di queste tre operazioni possono possedere i nostri studenti?* Quanto sinora descritto si riferisce alla relazione a tre {*docente, studente, attività*}. Esiste tuttavia un ulteriore aspetto della dinamica educativa che riguarda la tupla {*studente, classe, attività*} ed ha a che fare prevalentemente con il *processo di acquisizione* della propria identità e la costruzione dell’immagine di sé all’interno della comunità dei compagni (o colleghi) di classe.

In questo senso, il *secondo punto di innesto* di un LLM è localizzato psicologicamente nel luogo mentale in cui, nello studente, interferiscono costruttivamente (in termini ondulatori) gli stati d’animo di *invidia, frustrazione, incertezza, orgoglio, paura, vergogna*⁸. In questo scenario, lo studente percepisce la “*mente surrogata*” come avente un ruolo “*salvifico*”, poiché rappresenta quell’espansione di capacità intellettuale che consente all’individuo (in preda agli effetti della sovrapposizione dei suddetti stati d’animo) di superare i concorrenti quanto basta, per godere di ammirazione e rispetto (o almeno per non passare da “*perdente*”). La selezione di questi stati d’animo non è casuale:

- L’*invidia* per i successi del compagno concorrente⁹,
- La *frustrazione* causata dai necessari ed inevitabili fallimenti del momento,
- L’*incertezza* delle proprie capacità (a cui si lega un fisiologico senso di inadeguatezza),
- L’*orgoglio* sempre in pericolo,
- La *paura* di sbagliare e dunque di essere giudicato dagli altri (e dal docente),
- La *vergogna* dell’esprimersi male in presenza dei compagni.

⁷ Le foto presenti in questo articolo sono generate ex-nihilo mediante GPT-4o.

⁸ Un elenco dettagliato ed esaustivo delle emozioni umane in [9].

⁹ Si veda [16] per una indagine su invidia e bullismo in alcune scuole italiane.

Gregory Bateson, in *“Verso un’Ecologia della Mente”*¹⁰, presenta un interessante modello antropologico, valido per l’individuo e per le nazioni, che qui utilizziamo come framework di analisi, secondo cui esiste una classificazione delle relazioni umane, a varie scale, secondo le coppie seguenti:

Autorità – Sottomissione,
Assistenza – Dipendenza,
Esibizionismo – Ammirazione.

Benché il discorso sia molto articolato, ritengo appropriato evidenziare qui come la relazione che gli studenti sviluppano con il docente possa ritenersi del tipo *“Autorità – Sottomissione”*, indipendentemente dal fatto che sia il docente stesso a strutturarla in questo senso. Invece, le relazioni tra compagni di classe possono rientrare in tutte e tre le categorie indicate, con le ultime due, a mio giudizio, spesso più probabili della prima. Un LLM può dunque essere percepito dall’*“assistente”* come un valido strumento per gestire una relazione tra umani di tipo *“Assistenza – Dipendenza”*, quantomeno per ridurre il proprio carico di lavoro da *“assistente”*. Di contro, diventa essenziale ipotesi per l’*“esibizionista”*, che alimenta le proprie sicurezze *“barando”* con il GPT di turno.

Duole constatare tuttavia che si instauri una relazione robusta tra lo studente e la *“mente surrogata”*. Relazione che è certamente *“Assistenza – Dipendenza”*, dunque estremamente pericolosa, perché attuata potenzialmente nella sfera occulta del giovane (ciascuno agisce in una sfera pubblica, una privata, una occulta perché intima e problematica¹¹) e legata a doppio filo alla costruzione delle fondamenta dell’autostima, della percezione del sé nella comunità classe e, non da ultimo, delle capacità cognitive.



Figura 2 – Il groviglio emotivo in cui versano, non sempre consapevoli, i nostri ragazzi

3. L’integrazione con la didattica

Le consapevolezza acquisite sul piano psicologico, un continuo dialogo emotivo con gli studenti e l’osservazione diffusa per cui i nostri studenti si relazionano continuamente e senza coscienza con le *“menti surrogate”* degli LLM, ci costringono a fare una continua premessa di metodo ad inizio lezione: *“Cari ragazzi, la vostra condizione attuale è, con alta probabilità, rappresentata da questa immagine semplice ed efficace...”* (si veda Figura 3).

¹⁰ G. Bateson, *“Verso un’Ecologia della Mente”*, p. 103-140.

¹¹ Vincere barando è un atto problematico, che sottopone la psiche a stress emotivi. Si veda [17].

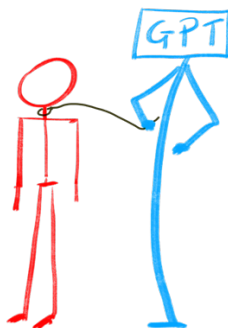


Figura 3 – Una icona che esprime l’attuale relazione tra studente e LLM

Il nostro scopo è liberare lo studente dal legaccio psichico, governando direttamente le interazioni con gli LLM a lezione in aula, ma soprattutto in laboratorio (se la materia ne prevede l’impiego). Nello specifico, abbiamo attuato un insieme di esperienze costruttive e interessanti fatte con una classe terza (e parzialmente una quinta) ad indirizzo informatico.

LLM CHE INTERROGA

Abbiamo fatto in modo che un LLM interrogasse lo studente, ponendogli specifiche domande, come indicato dal docente, e, se richiesto, dando un giudizio sulla qualità delle risposte e correggendone il tiro se necessario. Nella nostra esperienza, abbiamo impiegato la versione commerciale di *Open.AI Chat-GPT o3*, con abbonamento Plus, ottenendo risultati soddisfacenti sotto il profilo della qualità delle domande, la personalizzazione della difficoltà, la costruzione delle risposte correttive. L’esperimento è stato effettuato utilizzando la funzione di sintesi e riconoscimento vocale mediante smartphone (dunque non in modo testuale scritto, ma nella modalità dialogata verbale interattiva). L’AI ha mostrato in modo autonomo un tono accomodante e poco rigido che poi il docente ha deliberatamente modificato in *perentorio*, *rigoroso* e *severo*, ottenendo un’interazione studente-docente “*vecchio stampo*”, in cui la novità è che il docente è ovviamente artificiale. L’effetto complessivo sulla classe e sullo studente che si è sottoposto al test è stato di vivo interesse e diffusa perplessità mista a stupore, dato l’effetto novità.

POLARIZZAZIONE DEGLI APPRENDIMENTI

Siamo poi partiti dall’assunto che gli studenti, interagendo continuamente con gli LLM, siano prevalentemente esposti ai contenuti da essi prodotti, piuttosto che a testi ufficiali strutturati.

Inoltre, abbiamo assunto che i ragazzi non dispongano ancora delle opportune e adeguate capacità di redazione dei prompt. Per tale ragione, si è stabilito di “*polarizzare*” gli apprendimenti degli studenti non tanto sulla loro capacità di sintesi (creare codice durante le lezioni di informatica), quanto piuttosto sulla loro capacità di **interpretare efficacemente** gli output dell’AI.

L’anno scolastico è stato pertanto strutturato per affrontare il tema “*linguaggio di programmazione*” secondo una declinazione squisitamente linguistica (come se studiassimo la lingua di un popolo, il “*popolo cibernetico*”), acquisendo i concetti di più alto livello, di fatto **indipendenti** dal linguaggio scelto (secondo quindi una classica tripartizione in sintassi, semantica e pragmatica):

- Paradigma *comandante/ esecutore*,
- *Asserto*, inteso come affermazione di senso compiuto di molteplice natura,
- *Asserto dichiarativo*,
- *Asserto interrogativo*, inteso come espressione booleana valutabile in termini di verità e falsità,
- *Asserto imperativo*,
- Il simbolo “=” e le differenze rispetto alla matematica (immersione temporale del simbolo, piuttosto che atemporale). Dunque, gli studenti hanno imparato a leggere e comprendere, *piuttosto che a scrivere*.

RINFORZO STRUMENTALE

Nei momenti di verifica scritta, si è deciso di concedere l'uso di un LLM a scelta, abbinato tuttavia ad una modalità a **token economy**: veniva richiesto allo studente di incrociare gli output dell'intelligenza artificiale con i soli concetti, costrutti e pratiche affrontati nel corso delle lezioni teoriche e nelle esercitazioni; la presenza di simboli, di costrutti e concetti non affrontati a lezione sull'elaborato finale, avrebbe comportato la **decurtazione**, sul punteggio complessivo, di **0.25** punti per ogni errore rilevato (=elemento non affrontato, presente nell'elaborato).

Questo ha consentito di mettere gli studenti in uno stato di *intensa attività*, accompagnato da *moderata ansia*, che ha consentito loro di fare uno sforzo reiterato di richiamo alla memoria di tutti i concetti effettivamente affrontati a lezione.

3.1 Elementi di riflessione

Nell'esperienza del “**rinforzo strumentale**”, in una prima consegna dell'elaborato, gli studenti hanno preferito omettere gran parte del codice piuttosto che incorrere nella sanzione prevista secondo la token economy: gli elaborati erano prevalentemente in “*bianco*”. Successivamente, i ragazzi hanno compreso il meccanismo e orientato lo sforzo mnemonico nella direzione dell'incrocio tra output del LLM e conoscenze possedute. I migliori con discreto successo, gli altri, con risultati non del tutto soddisfacenti. In questo contesto emergono le seguenti categorie di studente:

- Colui che ha timore e vergogna di dichiarare di utilizzare un LLM,
- Colui che copia abilmente ogni simbolo, parola e frase in modo rigoroso (pochi studenti),
- Colui che non è in grado di copiare verbatim gli output, omettendo importanti elementi sintattici (molti studenti).

Si osserva con soddisfazione che gli studenti H e DSA hanno mostrato di trarre beneficio e vantaggi nell'impiego di un LLM, a supporto dello studente da solo, ma anche se accompagnato dal docente di sostegno.

Le affermazioni, lo stile, il contenuto dell'articolo, delle lezioni e delle immagini sono il frutto della lettura critica delle opere in bibliografia. Molte altre hanno contribuito, di cui si omette per ragioni di spazio il titolo.



Figura 4 - L'obiettivo

Bibliografia

- [1] Byung-Chul Han, “Le non cose”, Super ET.
- [2] Yuval Noah Harari, “Nexus”, Saggi Bompiani.
- [3] C. G. Jung, “Energetica Psicica”, Bollati Boringhieri.
- [4] C. G. Jung, “Tipi Psicologici”, Bollati Boringhieri.
- [5] J. Bruner, “La mente a più dimensioni”, Editori Laterza.
- [6] A. Guasparri, “Il primate della moda”, FrancoAngeli.
- [7] G. Bateson, “Verso un’Ecologia della Mente”, Adelphi.
- [8] O. Lizzini, “Avicenna”, Carocci Editore.
- [9] <https://www.studiotrevisani.it/emozioni-lelenco-delle-101-emozioni/?amp=1>
- [10] <https://www.rainews.it/articoli/2025/07/ia-il-garante-privacy-protagonista-di-una-rivoluzione-antropologica-ma-attenzione-allapproccio-df931619-721c-434f-ad32-d8c662a8c221.html>
- [11] Paustian, Slinger, “Students are using large language models and AI detectors can often detect their use”, <https://doi.org/10.3389/feduc.2024.1374889>
- [12] <https://www.intelligent.com/nearly-1-in-3-college-students-have-used-chatgpt-on-written-assignments/>
- [13] Kosmyrna et al., “Your Brain on ChatGPT: Accumulation of Cognitive Debt when Using an AI Assistant for Essay Writing Task”, <https://doi.org/10.48550/arXiv.2506.08872>
- [14] Stadler, M., Bannert, M., & Sailer, M. (2024). “Cognitive ease at a cost: LLMs reduce mental effort but compromise depth in student scientific inquiry”. *Computers in Human Behavior*, 160, 108386. <https://doi.org/10.1016/j.chb.2024.108386>
- [15] Lo Presti et al., “IA per lo Sviluppo delle Abilità Creative: Un’Analisi dello Stato dell’Arte delle Teorie e degli Approcci Attuali”, doi: 10.14668/QTimes_16324
- [16] <https://www.giornaledipsicologia.it/indagine-invidia-e-il-bullismo/>
- [17] <https://www.psychologytoday.com/us/blog/surviving-your-childs-adolescence/202102/cheating-in-high-school-a-response-to-student>

GEM FLEXNAO: Tecnologia in movimento per il benessere del domani

Veronica Cavicchi^{1}, Matteo Bettini¹, Nikolai Carnaghi¹, Amir Shams Eddin¹, Saverio Venturi¹, Ejuxchin Hoxha¹, Astik Gantam¹, Letizia Galletta¹*

¹ Liceo Scientifico Statale “A. Calini”, Laboratorio di robotica GEM, Brescia, Italia

veronica.cavicchi@liceocalini.edu.it

Abstract

GEM FLEXNAO è un progetto multidisciplinare sviluppato dal team GEM del Liceo Scientifico di Stato “A. Calini” di Brescia per la NAO Challenge. L’obiettivo è realizzare un sistema conversazionale capace di integrare un robot umanoide, l’intelligenza artificiale e un’interfaccia multimediale per migliorare l’accessibilità, la motivazione e il benessere fisico. Al centro dell’esperienza c’è NAO, programmato per registrare l’audio dell’utente, inviarlo a un server, comprendere quanto detto, generare una risposta coerente e, se richiesto, visualizzare un video tramite Raspberry Pi. Il sistema, interamente sviluppato in Python, si fonda su due web server e sull’uso del modello AI llama3- groqtool-use con funzioni esterne personalizzate. GEM FLEXNAO è un esempio di come, in ambito scolastico, sia possibile progettare ambienti intelligenti modulari, replicabili e sostenibili.

1. Introduzione

Il progetto nasce per promuovere un’alleanza innovativa tra robotica, scienze motorie e sostenibilità [1]. NAO è programmato con Choregraphe per interagire vocalmente con l’utente. GEM FLEXNAO nasce come progetto integrato per la NAO Challenge con l’intento di esplorare il potenziale della robotica conversazionale al servizio del benessere umano. L’interazione è interamente basata sulla voce: l’utente parla, NAO ascolta, comprende, risponde e può anche mostrare un video in base alla richiesta. Il progetto si è sviluppato con un forte approccio multidisciplinare, coinvolgendo studenti con competenze tecniche, creative e comunicative. La nostra ambizione è contribuire a una visione futura dove tecnologia e benessere corporeo si fondono in sistemi accessibili, motivanti e replicabili. **REPORT SCIENTIFICO E TECNICO** (bit.ly/NAO-IA-WebSocket-gem-flexnao). L’esperienza è raccontata nel blog del GEM ([Ecomentality.it](https://ecomentality.it)).

* Referente scientifico e docente responsabile.

2. Architettura del sistema

L’ecosistema si basa su tre componenti principali:

1. **NAO**: il robot umanoide programmato in Choregraphe per registrare l’audio, inviarlo al server e fornire la risposta (bit.ly/coreographe-gem-flexnao). [5]
2. **Web Server 1 (PC)**: gestisce l’audio, il riconoscimento vocale e l’elaborazione AI; riceve i file audio, li elabora con la libreria *Python SpeechRecognition*, interroga l’IA *llama3-groq-tool-use* (bit.ly/github-naocomms-gem-flexnao). [4]
3. **Web Server 2 (Raspberry Pi)**: visualizza video su richiesta ricevendo comandi da una funzione *PlayVideo* [2] dell’IA(bit.ly/github-naovideo-gem-flexnao).

Componente	Tecnologia
Robot umanoide	NAO V6, programmato con Choregraphe
Webserver 1	Python, SpeechRecognition, llama3-groq-tool-use
Webserver 2	Raspberry Pi 3, server
Librerie	Python Flask HTTP, JSON, OpenAI Tools

Tabella 1: Tecnologie principali impiegate nel progetto

3. Funzionamento del sistema

Pipeline conversazionale. Il flusso interattivo si articola in 7 passaggi:

1. NAO attiva il microfono, rileva il parlato e registra
2. Il file audio viene inviato al Webserver 1 come byte.
3. Il server riconosce e converte l’audio in testo tramite *SpeechRecognition*.
4. Il testo è letto da NAO per confermare la comprensione.
5. Il testo viene rispedito al server per generare una risposta, cioè l’input viene inviato al modello AI *llama3-groq-tool-use*; l’IA elabora la risposta o esegue funzioni.
6. La risposta può essere:
 - una frase coerente;
 - una funzione esterna come *PlayVideo*(video name) (che riproduce un video su Raspberry Pi) o *RandomNumber* (min, max) (che genera numeri casuali per quiz);
7. NAO pronuncia la risposta e, se richiesto, il Raspberry visualizza il video.

4. Scelte progettuali

- Linguaggio **Python** per flessibilità, compatibilità e semplicità.
- Comunicazione via **HTTP**, standard e interoperabile.
- Librerie **open source** per l’elaborazione vocale.
- Uso di *llama3-groq-tool-use* con *function calling* per interazione evoluta.
- **Separazione modulare** tra AI e video per miglior scalabilità.

5. Esperienza sul Campo

L'esperienza, multidisciplinare, ha coinvolto studenti con competenze tecniche, creative e comunicative in attività pratiche di programmazione informatica per attività sportive rieducative e apprendimento STEM. Il progetto ha beneficiato della preziosa collaborazione con il partner scientifico e sportivo Forza e Costanza Brescia – Fisioterapia sportiva. [3]. Il flusso interattivo prevedeva:

- Interazione vocale tra utente e robot.
- Elaborazione IA della richiesta.
- Risposta vocale e visualizzazione di video educativi.

6. Applicazioni e contesto

Il progetto valorizza la robotica al servizio del benessere, con attenzione ad inclusività, motivazione e accessibilità. Le applicazioni spaziano dal supporto alla fisioterapia post-operatoria, all'educazione motoria nei bambini, alla prevenzione dell'ipomobilità negli anziani.

- **Riabilitazione post-trauma:** supporto motivazionale alla fisioterapia e alla postura e guida vocale.
- **Attività sportiva in palestra per bambini e anziani:** interazione empatica e accessibile attraverso attività sportive e ludico-educative.
- **Educazione STEM:** uso didattico per apprendimento attivo.
- **Ginnastica posturale in eventi, fiere e scuole:** NAO come guida robotica.

La presenza del modello AI, integrato in una pipeline software fluida e modulare, apre la strada a scenari futuri di: **Telemedicina** assistita da robot, **Allenatori virtuali** intelligenti, **Tutor educativi** personalizzati, **Interfacce vocali interattive** in ambienti terapeutici.

7. Brand Identity e Visione

Il nostro logo rappresenta una testa umana stilizzata con foglie, a simboleggiare l'equilibrio tra crescita personale, tecnologia e natura. Il colore verde è scelto per sottolineare l'impegno verso sostenibilità e innovazione (<https://www.ecomentality.it/2025/05/22/brand-identity-gem-flexnao/>, INSTAGRAM: @GEMFLEXNAO). I nostri valori sono innovazione e interdisciplinarietà; accessibilità, sostenibilità e inclusione; eccellenza scientifica e crescita.

8. Team e strumenti

8.1 Team GEM – Liceo Calini, Brescia:

Nome	Ruolo
Matteo Bettini	Team leader, programmatore, videomaker
Amir Shams Eddin	Social media manager
Ejuxhin Hoxha	Graphic designer

Letizia Galletta	Concept developer
Nikolai Carnaghi	Programmatore
Saverio Venturi	Videomaker
Astik Gautam	Blogger

Tabella 2: Membri del Team GEM FLEXNAO e ruoli ricoperti

Referente scientifico e docente responsabile: **Prof.ssa Veronica Cavicchi**.

8.2 Strumenti

- **Hardware** – 8 notebook, NAO Robot V6, Raspberry Pi 5 8Gb, microfono, display.
- **Programmazione** – Python, Choregraphe, LaTeX e Overleaf, modelli AI llama3-groq-tool-use, GitHub, HTML.
- **Progettazione grafica e digitale** – Adobe InDesign, Adobe Photoshop, Adobe Express, Canva, VistaCreate.
- **Divulgazione, promozione ed editing video** – WordPress, YouTube, Instagram, Google Work-Space, Renderforest, Da Vinci Resolve, ClipChamp.

9. Conclusione

L’idea si fonda sulla tecnologia come alleata del benessere, della motivazione e dell’inclusione.

“GEM FLEXNAO: tecnologia in movimento per il benessere del domani.”

GEM FLEXNAO rappresenta un esempio di come tecnologie avanzate possano essere integrate efficacemente nella didattica scolastica e nell’ambito motorio, promuovendo inclusività, motivazione e benessere fisico (Video YouTube: NAO in palestra).

Il progetto GEM FLEXNAO rappresenta un esempio di applicazione del Project-Based Learning (PBL) e della metodologia Hackathon, in linea con quanto previsto dalla Proposta di Indicazioni Nazionali per l’Insegnamento dell’Informatica nella Scuola, che evidenzia l’importanza di apprendere l’informatica attraverso attività progettuali, collaborative e interdisciplinari, capaci di stimolare il pensiero computazionale e le competenze di problem-solving degli studenti. [1] In particolare, il progetto ha coinvolto gli studenti nella realizzazione concreta di un sistema conversazionale integrato con robotica e intelligenza artificiale, favorendo la comprensione delle tecnologie emergenti e delle loro implicazioni etiche, sociali e ambientali. Secondo il framework UNESCO sull’Intelligenza Artificiale, è essenziale che gli studenti siano coinvolti attivamente come co-creatori di tecnologie AI, sviluppando una mentalità critica e orientata all’inclusività, alla sostenibilità e alla responsabilità sociale. GEM FLEXNAO ha adottato questo approccio, incoraggiando i partecipanti ad applicare creativamente le competenze STEM, affrontare sfide autentiche e sviluppare una forte consapevolezza etica, preparando così gli studenti a essere protagonisti consapevoli e competenti in un mondo sempre più guidato dall’AI [6].

10. Estratto di codice Python (semplificato)

```

from webapp import App
HOST = "0.0.0.0"
PORT = 31417

app = App(HOST, PORT)

import os
import threading
# Endpoint per far vedere un video
@app.post("/")
def showvideo ( payload : dict ) :
# Il nome deve essere passato
# nel json
videoname:str = payload [ "video" ]
# Se il file non esiste mandiamo un 404
if not os.path.isfile (f"{videoname}.mp4 " ) :
    return 404, "Video_name_not_found"
# Altrimenti creiamo un nuovo thread
# che fa vedere il video usando vlc
t = threading.Thread (
    target = os.system ,
    args = (f'vlc --quiet '{videoname}.mp4' --fullscreen",))
t.start ()
return 200, "Video_is_now_playing"

# Un endpoint per vedere i video disponibili
@app.get("/videos")
def getvideos(payload:dict):
    # restituisci tutti i video
    # che possono essere riprodotti
    return 200,[x.removesuffix(".mp4") for x in os.listdir()
if os.path.isfile (x) and x.endswith (".mp4")]

# Un endpoint che fa vedere la pagina test
@app.get("/")
def return HomePage ( payload : dict ) :
    with open ("client.html") as f:
        h = f.read()
        return 200, h

app.start ()

```

Listing 1: Server Python per gestione video

Bibliografia

- [1] CINI. Proposta di indicazioni nazionali per l'insegnamento dell'informatica nella scuola. <https://www.consortio-cini.it/index.php/it/component/attachments/download/745>, 2017.
- [2] Python Software Foundation. Speechrecognition library. <https://pypi.org/project/SpeechRecognition>, 2025.
- [3] Liceo Calini GEM. Ecomentality.it. <https://ecomentality.it>, 2021–2025.
- [4] Groq. llama3-groq-tool-use model documentation. <https://groq.com>, 2025.
- [5] SoftBank Robotics. Nao robot. <https://us.softbankrobotics.com/nao>, 2025.
- [6] UNESCO. AI competency framework for students. <https://unesdoc.unesco.org/ark:/48223/pf0000391105>, 2024.

Team: Matteo Bettini, Nikolai Carnaghi, Amir Shams Eddin, Saverio Venturi, Ejuxhin Hoxha, Astik Gautam, Letizia Galletta.

Referente: Prof.ssa Veronica Cavicchi.

Coach: Camilla Trebeschi.

Scuola: Liceo Scientifico di Stato "A. Calini", Laboratorio di robotica GEM.

Partner scientifico e sportivo: Forza e Costanza Brescia - Fisioterapia sportiva.

Programmazione C++ applicata alla stampa 3D: un'esperienza interdisciplinare per l'apprendimento dei cicli iterativi nella scuola superiore

Sonia Guerci^{1}, Federico Florit^{1**}*

¹ *Istituto Tecnico Antonio Zanon di Udine*

sonia.guerci@itzanon.it federico.florit@itzanon.it

Abstract

Questo articolo presenta un'esperienza didattica interdisciplinare che ha integrato l'apprendimento dei cicli iterativi in C++ con la comprensione delle stampanti 3D attraverso la generazione di file G-code. L'attività, realizzata in una classe terza di un Istituto Tecnico Economico (Sistemi Informativi Aziendali), si è basata sul principio del *Necessity-Based Learning*, motivando gli studenti a creare G-code tramite C++.

Il percorso didattico ha previsto una fase iniziale di comprensione dei comandi G-code e l'utilizzo di visualizzatori online per simulare il movimento della testina di stampa. Successivamente, gli studenti hanno automatizzato la creazione di file G-code per forme geometriche via via più complesse, partendo da un quadrato e arrivando a un bicchiere, includendo la geometria analitica delle coniche per la realizzazione di figure piane composte da rette e archi di circonferenze. Questa progressione ha permesso di sperimentare il valore dell'automazione e affinare le capacità di debugging tramite feedback visivo.

I risultati hanno evidenziato un significativo miglioramento nell'apprendimento dei cicli iterativi, nella manipolazione di file di testo e nell'applicazione del pensiero computazionale a problemi reali, confermando l'efficacia di contesti applicativi autentici e tecnologicamente rilevanti per l'insegnamento della programmazione.

1. Introduzione

L'insegnamento della programmazione nella scuola superiore richiede spesso di trovare contesti applicativi che motivino gli studenti e rendano concreti i concetti astratti. Questo contributo descrive un'esperienza didattica innovativa in una classe terza di un Istituto Tecnico Economico (Sistemi Informativi Aziendali). L'obiettivo è rendere la programmazione meno astratta: per farlo, l'insegnamento dei cicli iterativi in C++ è stato collegato al funzionamento delle stampanti 3D, attraverso la generazione di file G-code. I file G-code sono file di testo che contengono sequenze di codici (G0, G1, G2, ecc.) seguiti da parametri numerici che vengono impiegati nella definizione del percorso macchina su macchine a controllo numerico, tra cui

* Docente di Informatica.

** Docente di Matematica.

anche le macchine da stampa 3D [1]. Il progetto nasce da una collaborazione tra docenti di informatica e matematica, e mira a far approfondire agli studenti la geometria analitica e la programmazione C++ tramite la creazione di G-code. Questa attività integra perfettamente gli obiettivi di entrambe le discipline.

2. Metodologia e Obiettivi

L'attività è stata progettata seguendo i principi del Necessity-Based Learning [2] (apprendimento basato sulla necessità), metodologia che mira a creare situazioni in cui gli studenti percepiscono la necessità di acquisire nuove competenze per risolvere problemi concreti e significativi. Per quanto riguarda il contesto di attuazione e i partecipanti, l'esperienza didattica si è svolta in orario curricolare presso l'Istituto Tecnico Antonio Zanon di Udine, coinvolgendo una classe terza dell'indirizzo Sistemi Informativi Aziendali composta da 17 studenti con un'età media di 16 anni. L'attività ha coinvolto tutti gli studenti, i quali possedevano già le conoscenze di base della programmazione C++ (incluse variabili, gestione di input/output e strutture condizionali) ed elementi di geometria analitica del piano (come l'equazione della retta e le coordinate cartesiane), acquisiti come parte integrante del programma curricolare. Non era invece richiesta alcuna esperienza pregressa con la stampa 3D o il G-code.

Per valutare l'efficacia didattica, è stato adottato un approccio combinato. L'osservazione diretta degli studenti durante lo svolgimento delle attività e la raccolta dei loro commenti spontanei hanno fornito una prima analisi del processo di apprendimento. Questa fase è stata poi integrata dal monitoraggio degli esiti delle verifiche somministrate dopo il termine dell'attività, volti a validare il raggiungimento degli obiettivi a distanza.

2.1 Obiettivi disciplinari specifici per la Matematica

- Consolidamento del concetto di retta nel piano cartesiano.
- Apprendimento del concetto di conica nel piano cartesiano (in particolare circonferenza).
- Applicazione dei concetti di geometria analitica (punti, rette, circonferenze) a problemi reali.

2.2 Obiettivi disciplinari specifici per l'Informatica

- Comprensione e utilizzo dei cicli iterativi (FOR, WHILE).
- Manipolazione di file di testo tramite stream di input e stream di output.
- Parametrizzazione di algoritmi attraverso variabili.
- Sviluppo di competenze di debugging.
- Applicazione del pensiero computazionale a problemi reali.

2.3 Competenze trasversali

- Comprensione del funzionamento di base delle tecnologie di stampa 3D.
- Capacità di tradurre specifiche geometriche in istruzioni procedurali.

3. Svolgimento dell'Esperienza

Il progetto ha adottato un approccio in due fasi. Nella prima fase matematica si è lavorato sulla geometria analitica delle coniche e su come ricavare i valori necessari alla realizzazione di figure piane, composte da rette e archi di circonferenze, tramite i comandi base G-code (G0, G1, G2, G3, ecc). Nella seconda fase informatica, gli studenti hanno implementato algoritmi per la generazione automatica dei comandi necessari alla generazione di solidi tridimensionali.

3.1 Matematica

La prima fase prevedeva un lavoro in gruppi di 2/3 studenti. A ogni gruppo è stata assegnata una figura piana (simile alla Figura 1) dotata delle dimensioni reali di stampa e scomponibile in pochi elementi.

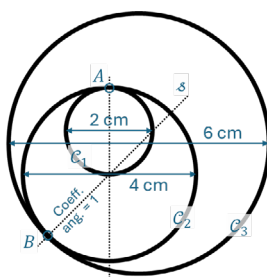


Figura 1: Esempio di figura piana da cui derivare i comandi G-code

Gli studenti, guidati da una lista di richieste fornita dal docente, hanno utilizzato le loro conoscenze di geometria analitica per determinare le equazioni degli elementi costituenti la figura assegnata, le posizioni dei centri degli archi di circonferenza presenti e le coordinate dei punti di intersezione (ed eventualmente di tangenza) tra archi/segmenti differenti. Gli studenti, con l'aiuto dell'insegnante, hanno in seguito tradotto i risultati trovati in un G-code di stampa della figura su singolo layer. I codici dei vari gruppi sono stati uniti dal docente in un unico G-code ed è stata effettuata la stampa con la stampante disponibile presso l'Istituto. In Figura 2 è riportata l'anteprima di stampa.

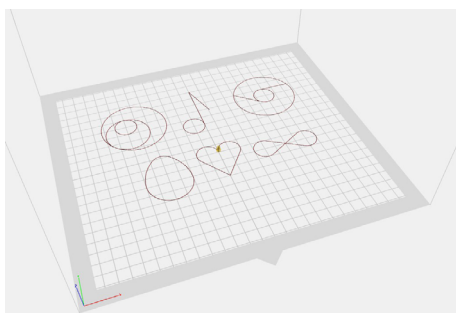


Figura 2: Anteprima di stampa del codice sviluppato dagli studenti nella fase di matematica

3.2 Informatica

Dopo aver approfondito il funzionamento della stampante 3D e generato i primi file G-code 3D tramite Tinkercad e uno slicer, si è passati a crearli manualmente, coadiuvati da visualizzatori G-code online come NC Viewer. Strumenti come questo permettono non solo di scrivere direttamente le istruzioni G-code, ma anche di simulare il movimento della testina della stampante, offrendo una comprensione più profonda di come le istruzioni si traducano in movimenti fisici (Figura 3).

Il passo successivo è stato quello di automatizzare la creazione dei file G-code scrivendo programmi console in C++ che permettessero all'utente di specificare forme e dimensioni desiderate. Il percorso didattico ha seguito una progressione graduale partendo da forme geometriche semplici. Inizialmente gli studenti hanno parametrizzato la creazione di un quadrato, definendo le coordinate dei vertici in funzione del punto iniziale e del lato.

Dopo aver illustrato la sintassi del costrutto iterativo e richiamato i comandi essenziali per la manipolazione dei file di testo, l'insegnante ha facilitato l'implementazione dell'algoritmo in modo partecipato, scrivendo il codice alla lavagna sulla base delle decisioni collettive degli studenti per creare un cubo (ottenuto ripetendo un quadrato su piani successivi, Listato 1), e fornendo suggerimenti e chiarimenti solo se strettamente necessario.

```

1      const float E_COEFF = 0.03135; //quanto PLA per mm
2
3      // Parametri da acquisire in input
4      float x0, y0, lato, z = 0.0;
5      // ... gestione dei parametri di input
6
7      ofstream fout("cubo.gcode");
8      fout << "G90" << endl; // Imposto modalita' coordinate
9
10     // Mi posiziono nel punto iniziale (Es. G0 X5 Y5)
11     fout << "G0 X" << x0 << " Y" << y0 << endl;
12
13     z = 0; // siamo sul piano ad altezza zero
14     while (z < lato){
15         // Lato 1
16         fout << "G1 X" << x0 << " Y" << y0 + lato; // coord. finali
17         fout << " E" << E_COEFF * lato << endl; // estrusione
18         // ... altri lati
19
20         z = z + 0.2; // devo salire ogni giro di 0.2
21         fout << "Z" << z << endl; //Salgo fino a z
22     }

```

Listato 1: Esempio di ciclo While per la generazione di layer

Questa fase è stata fondamentale perché gli studenti hanno sperimentato direttamente il valore pratico del ciclo: l'automazione ha sostituito la scrittura manuale di decine di comandi identici. Sono stati quindi introdotti i comandi G2/G3 per la creazione di figure composte da archi di circonferenza, con il calcolo della lunghezza dell'arco e la gestione dei parametri aggiuntivi I, J che rappresentano l'offset del centro dell'arco rispetto al punto di partenza. Gradualmente, il progetto si è evoluto verso forme più complesse, culminando nella creazione del file G-code per un bicchiere, che ha richiesto l'implementazione di algoritmi parametrici per la creazione delle pareti svasate e del fondo del bicchiere.

Il visualizzatore G-code online NC Viewer ha permesso la verifica immediata dei risultati tramite il caricamento del file generato dagli studenti, permettendo loro di vedere immediatamente l'effetto del loro codice, identificare errori nella logica del programma e sviluppare capacità di debugging attraverso feedback visivo.

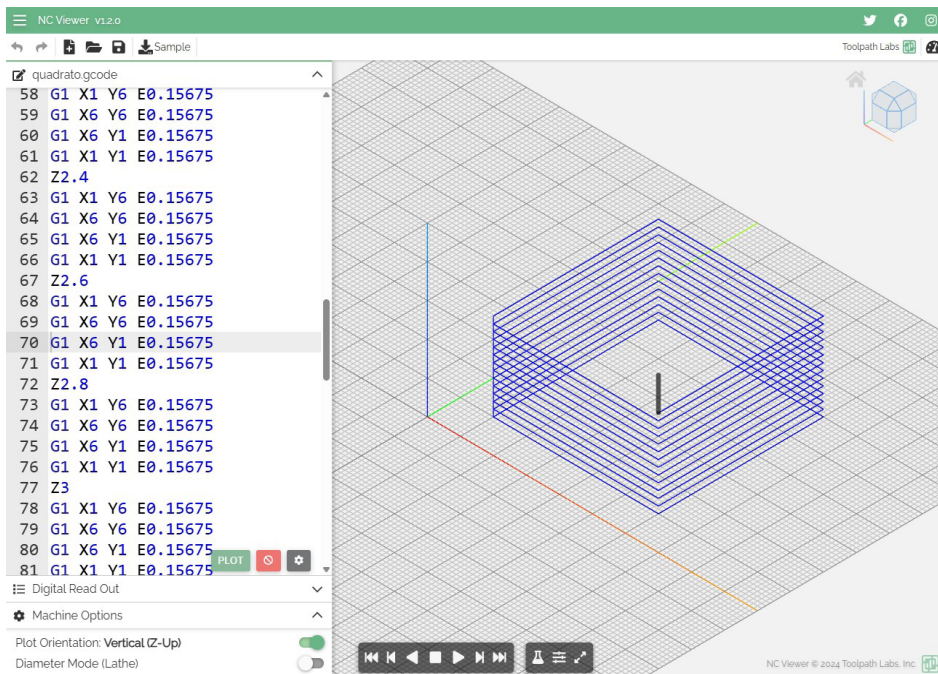


Figura 3: NC Viewer in azione: G-code di un cubo e la simulazione del percorso di stampa

4. Risultati e Osservazioni

L'esperienza didattica ha prodotto risultati positivi sia sul fronte dell'apprendimento disciplinare sia sullo sviluppo delle competenze trasversali, indicando l'efficacia del contesto applicativo proposto. Relativamente agli obiettivi specifici di Informatica, le verifiche svolte dopo il termine dell'attività hanno mostrato un buon livello di comprensione e utilizzo dei cicli iterativi. La maggior parte degli studenti ha acquisito una solida comprensione dei cicli iterativi e della loro corretta applicazione. La manipolazione di file di testo attraverso stream di input e output è stata consolidata dagli studenti, come testimoniato dalla correttezza dei file G-code generati. Dal punto di vista matematico, i lavori di gruppo e l'analisi dei percorsi G-code hanno rivelato un consolidamento dei concetti di retta e l'apprendimento della conica (in particolare la circonferenza) nel piano cartesiano. Gli studenti hanno applicato con successo i concetti di geometria analitica per determinare le coordinate e i parametri necessari alla realizzazione delle figure.

Le osservazioni dirette durante le attività hanno evidenziato un chiaro coinvolgimento e una buona motivazione degli studenti, riconducibili alla necessità di risolvere problemi di stampa

3D. La capacità di debugging ha mostrato un miglioramento osservabile: il feedback visivo immediato fornito dai visualizzatori G-code online ha permesso agli studenti di identificare e correggere gli errori nei loro algoritmi, rendendo il debugging un'attività più gestibile e coinvolgente. Gli studenti hanno inoltre sviluppato una comprensione del valore dell'automazione, sperimentando il passaggio da un lavoro manuale e ripetitivo alla creazione di algoritmi efficienti. Le competenze trasversali, come la comprensione del funzionamento di base delle tecnologie di stampa 3D e la capacità di tradurre specifiche geometriche in istruzioni procedurali, sono state acquisite e applicate nella creazione dei solidi tridimensionali.

5. Conclusioni

Il progetto ha dimostrato come l'utilizzo di contesti applicativi autentici e tecnologicamente rilevanti possa significativamente migliorare l'apprendimento dei concetti fondamentali della programmazione. Gli elementi chiave del successo sono stati la progressione didattica graduale partendo da forme geometriche elementari, la motivazione contestuale fornita dal collegamento con tecnologie moderne e la verifica immediata attraverso visualizzatori online che ha trasformato il debugging in un'attività visuale e coinvolgente.

L'esperienza conferma che l'apprendimento dei concetti astratti della programmazione, come i cicli iterativi, è più efficace quando emerge da necessità concrete vissute dagli studenti. L'approccio che parte dal concreto per arrivare all'astrazione e permette sperimentazione diretta prima dell'implementazione automatica si è rivelato particolarmente efficace per consolidare la comprensione dei meccanismi iterativi e delle tecniche di manipolazione dei file.

Questa esperienza può essere replicata in altri contesti scolastici con risorse tecnologiche di base (computer, ambiente di sviluppo del linguaggio scelto, accesso a visualizzatori G-code online e una stampante 3D, anche di fascia economica).

I prerequisiti per gli studenti includono una conoscenza di base della programmazione (variabili, strutture di controllo semplici) e della geometria analitica. Sebbene non indispensabile, la pregressa familiarità con la manipolazione dei file di testo può ridurre il carico cognitivo, facilitando il processo di apprendimento. L'interdisciplinarietà tra informatica e matematica è un elemento chiave per il successo, ma l'approccio può essere adattato anche in contesti con un focus più specifico su una delle due discipline.

Bibliografia

- [1] RepRap Community. G-code - RepRap. <https://reprap.org/wiki/G-code/it>, 2025.
- [2] Marco Sbaraglia, Michael Lodi, and Simone Martini. Apprendimento della programmazione guidato dalla necessità: il Necessity Learning Design. In *Atti del Convegno Italiano sulla Didattica dell'Informatica, ITA- DINFO 2023*, pages 190–195. Università degli Studi di Bari Aldo Moro. <https://www.itadinfo.it/2023/ attidelconvegno/ATTI-ITADINFO-2023.pdf>.

UNIUD Game Jam: Esperienza Formativa e di Orientamento attraverso lo Sviluppo di Videogiochi

Biagio Tomasetig¹, Alessandro Forgiarini¹, Massimiliano Pascoli¹, Christian Corrò¹, Fabio Buttussi¹

¹HCI Lab, Dipartimento di Scienze Matematiche, Informatiche e Fisiche,

Università degli Studi di Udine, Udine, Italia

{biagio.tomasetig, alessandro.forgiarini, massimiliano.pascoli, christian.corro, fabio.buttussi}@uniud.it

Abstract

Lo sviluppo collaborativo di videogiochi, durante la prima edizione della UNIUD Game Jam, ha permesso di unire formazione tecnica e orientamento universitario per gli studenti delle scuole superiori. Il percorso formativo è stato strutturato in due fasi: un corso introduttivo di 16 ore sullo sviluppo di videogiochi tramite il motore di gioco Unity, seguito da una game jam di 72 ore svolta in parallelo alla competizione internazionale Ludum Dare 57. La metodologia didattica adottata è stata quella del project-based learning, basata sulla creazione di progetti in gruppo. I partecipanti sono stati suddivisi in gruppi cercando di bilanciare gli interessi e le competenze dei partecipanti. Tutti i gruppi hanno completato lo sviluppo dei propri videogiochi, pubblicandoli alla competizione, confrontandosi così in un contesto internazionale.

1. Introduzione

Il crescente abbandono degli studi universitari¹ e il disallineamento tra le aspettative degli studenti e la realtà dei percorsi formativi [4] motivano una continua innovazione nelle attività didattiche e di orientamento. Anche i corsi dell'area informatica, sebbene offrano buone prospettive occupazionali, sono a volte percepiti come prevalentemente teorici e non allineati agli interessi pratici degli studenti, creando difficoltà nell'orientamento e nella motivazione a intraprendere o proseguire un percorso di studi in questo ambito [13, 5].

L'ambiente di apprendimento, la percezione di utilità del corso e diversi fattori psico-sociali, come l'auto-efficacia percepita dello studente, sono cruciali per il completamento con successo del percorso [5, 8, 4]. Inoltre, gli studenti, pur essendo nativi digitali [1], mostrano difficoltà nel collegare le competenze tecnologiche informali con i contenuti accademici formali [6, 10].

In questo contesto, metodi come game-based learning [12], learning by doing [14], collaborative learning [11], problem-based learning [9] e peer learning [2], trovano nel project-based learning (PBL) [15] una sintesi particolarmente efficace. Il PBL offre agli studenti la possibilità di osservare i risultati concreti del proprio lavoro [7, 3] sperimentando il potenziale pratico e creativo dell'informatica e auspicabilmente rafforzando la loro motivazione verso gli studi informatici.

1 https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Early_leavers_from_education_and_training

Le game jam, maratone di sviluppo finalizzate alla creazione di videogiochi in un tempo limitato (24–72 ore), possono rappresentare un valido strumento che integra i principi del PBL attraverso la realizzazione concreta di prototipi, il lavoro collaborativo e la condivisione di competenze complementari tra pari. Queste esperienze permettono agli studenti di sperimentare ruoli tecnici diversi (programmatore, designer, modellatore e gestore del progetto), facilitando l'auto-orientamento verso il percorso formativo che riterranno più adatto. Inoltre, questo tipo di esperienza risulta rilevante anche in prospettiva professionale. Infatti, l'industria videoludica è un settore in crescita e richiede competenze multidisciplinari che possono essere sviluppate anche attraverso la partecipazione alle game jam.

Nel caso specifico dell'orientamento verso i diversi corsi triennali in ambito informatico offerti dall'Università degli Studi di Udine (Informatica; Internet of Things, Big Data & Machine Learning; Tecnologie Digitali e Comunicazione per le Industrie Creative), questa metodologia potrebbe facilitare l'esplorazione delle diverse competenze peculiari di ciascun percorso formativo. Alla luce di queste considerazioni, il Dipartimento di Scienze Matematiche, Informatiche e Fisiche dell'Università degli Studi di Udine nell'ambito delle attività di orientamento PNRR e del "Progetto Lauree Scientifiche", ha organizzato la "UNIUD Game Jam".

L'iniziativa ha coinvolto studenti delle classi terze, quarte e quinte delle scuole superiori del Friuli Venezia Giulia ed era aperto anche a studenti universitari di corsi che trattavano argomenti affini. L'evento si è svolto in parallelo alla competizione internazionale Ludum Dare 57² ed è stato progettato per essere frequentato anche da studenti con limitate conoscenze informatiche, permettendo la partecipazione di studenti provenienti da diversi indirizzi scolastici.

2. Metodologia

Il percorso si è articolato in due fasi. La prima ha previsto un corso introduttivo di 16 ore sullo sviluppo di videogiochi con il motore di gioco Unity³ (Figura 1). Tale corso è stato suddiviso su due settimane in quattro lezioni da quattro ore ciascuna, svolte presso i laboratori del dipartimento. Ogni lezione prevedeva per ciascun argomento affrontato una spiegazione frontale da parte di uno dei docenti e un'attività pratica in cui gli studenti hanno applicato subito in Unity i concetti appresi con il supporto del docente della lezione assieme ad un altro docente. Il programma ha incluso: interfaccia di Unity, gestione dei modelli 3D e materiali, movimento tramite input utente, motore fisico, audio, interfaccia utente e organizzazione del flusso di gioco. L'ultima lezione si è conclusa con la formazione dei gruppi di lavoro sulla base di un'auto-valutazione delle competenze individuali in programmazione, design di meccaniche di gioco e creazione di asset (immagini, suoni, modelli 3D), e considerando gli interessi degli studenti. Il processo di auto-valutazione è servito a bilanciare le competenze tra i gruppi, considerati i diversi background formativi di provenienza degli studenti.

La seconda fase del percorso si è svolta in parallelo alla competizione internazionale Ludum Dare 57 presso il Lab Village⁴. I partecipanti hanno avuto a disposizione 72 ore complessive, di cui 20 in presenza, per realizzare un videogioco da sottoporre sulla piattaforma della competizione seguendo il tema "Depths", scelto collettivamente dai partecipanti della competizione internazionale. Durante questa fase, i docenti hanno offerto assistenza agli studenti, che

2 <https://ludumdare.com/>

3 <https://unity.com/>

4 <https://www.uniud.it/it/territorio-e-societa/uniud-lab-village/scopri-smact3>

potevano chiedere sia chiarimenti sulle lezioni affrontate nella prima fase sia consigli su come approcciarsi all'implementazione delle loro idee. La figura 2 mostra i gruppi durante l'attività di sviluppo nello spazio collaborativo del Lab Village. A conclusione del percorso, i progetti sono stati sottomessi alla competizione internazionale.

Per fornire ai partecipanti un riscontro al termine del percorso e incentivare ulteriormente la collaborazione e la qualità dei progetti, i docenti hanno implementato anche un sistema di valutazione interno. Tutti i docenti più due esperti hanno assegnato un punteggio da 1 a 5 per ognuno dei seguenti criteri di valutazione: aderenza al tema, divertimento, grafica, audio, originalità, collaborazione tra i membri, più una valutazione generale. La classifica finale è stata ottenuta calcolando la media aritmetica dei punteggi assegnati dai valutatori per ogni criterio e poi facendo la media tra tutti i criteri per ottenere il punteggio finale.

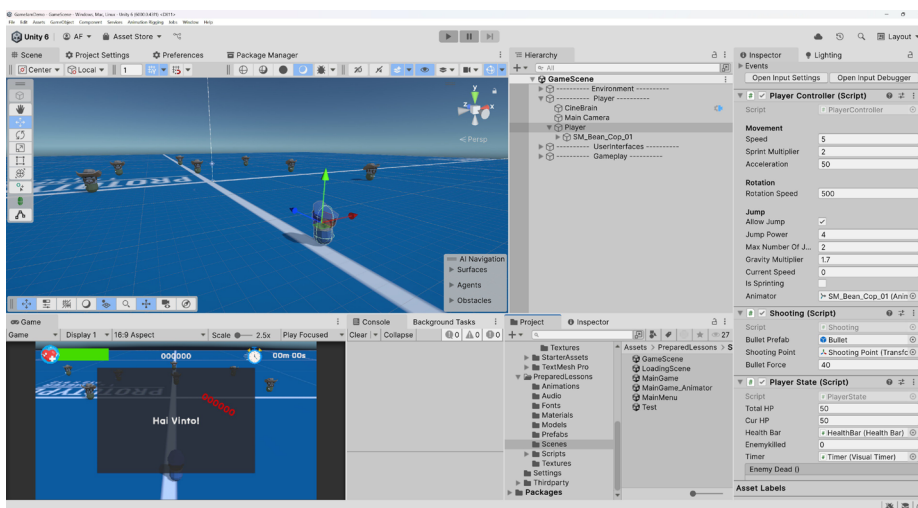


Figura 1: Schermata del motore di gioco Unity durante una delle lezioni.

3. Risultati

Al corso hanno partecipato 33 studenti delle scuole superiori della regione (ITI “Bearzi”, ISIS “A. Malignani”, ITG “G. G. Marinoni”, Liceo Scientifico “G. Marinelli”, ITE “A. Zanon”, tutti di Udine; Liceo Scientifico “Galilei” di Gorizia) e alcuni studenti dei percorsi universitari. Alla game jam hanno partecipato 28 di questi studenti (27 dalle scuole superiori e uno universitario), divisi in sette gruppi da quattro partecipanti ciascuno.

Tutti i gruppi hanno completato e sottomesso con successo i loro videogiochi, confrontandosi così con oltre 4000 sviluppatori da tutto il mondo. La Tabella 1 riporta i collegamenti ai videogiochi realizzati e testimonia la varietà creativa e tecnica raggiunta.

La classifica interna ha assegnato: il primo posto a “Deep in the mine”, un’avventura di un minatore che scende nelle profondità terrestri per raccogliere minerali, dovendo gestire le scorte di ossigeno; il secondo posto a “Lost in Depth”, un’esperienza riflessiva di un astronauta che

recupera dei frammenti luminosi per riportare la luce in un universo morente; il terzo posto, a pari merito, a due progetti, “Hell Pizza”, dove un pizzaiolo deve difendere una pizza da spiritelli infernali, e “Deep in the Internet”, in cui si controlla un cursore evitando banner malevoli del deep web per raggiungere un download.

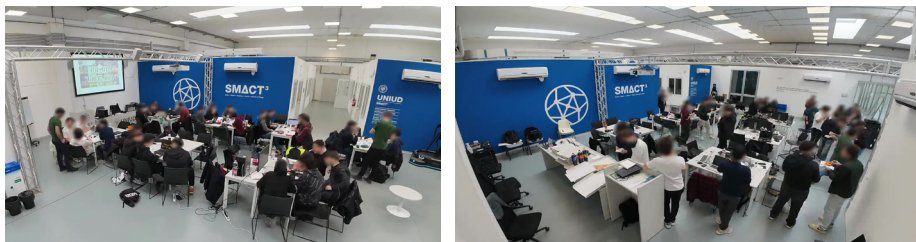


Figura 2: Studenti e docenti durante la game jam al Lab Village.

Titolo videogioco	Link
Lost in depth	https://ldjam.com/events/ludum-dare/57/lost-in-depth
Deep in the internet	https://ldjam.com/events/ludum-dare/57/deep-in-the-internet
Galactic wasteland	https://ldjam.com/events/ludum-dare/57/galactic-wasteland
Hell pizza	https://ldjam.com/events/ludum-dare/57/hellpizza
Deep sigma	https://ldjam.com/events/ludum-dare/57/deepsigma
Deep in the mine	https://ldjam.com/events/ludum-dare/57/deep-in-the-mine
Vulcanus Escape	https://ldjam.com/events/ludum-dare/57/vulcanus-escape

Tabella 1: Link ai giochi caricati dagli studenti partecipanti alla game jam

Nella classifica della Ludum Dare 57, “Deep in the mine” si è posizionato al 575-esimo posto su 1599 sottomissioni (36-esimo percentile). Questo risultato è particolarmente rilevante considerando che nessun membro del gruppo aveva precedenti esperienze nello sviluppo videoludico.

4. Conclusioni

La prima edizione della “UNIUD Game Jam” si è conclusa positivamente con la sottomissione di tutti e sette i videogiochi sviluppati durante l’evento alla competizione internazionale Ludum Dare 57. Durante l’evento, gli studenti hanno partecipato attivamente alle attività proposte. L’approccio basato sul PBL, unito alla classifica interna e alla partecipazione a una competizione internazionale, potrebbe aver favorito il mantenimento di questo interesse durante tutto il percorso, portando tutti e sette i gruppi al completamento dello sviluppo e stimolando la collaborazione tra membri con competenze diverse.

L'iniziativa ha offerto agli studenti, provenienti da percorsi formativi diversi, l'opportunità di avvicinarsi all'informatica in modo pratico e multidisciplinare, applicando in un contesto concreto le conoscenze e competenze acquisite sia nei propri percorsi di studio sia durante la prima fase del progetto. L'esito dell'evento indica che, nonostante i limiti di tempo, i partecipanti sono stati in grado di applicare quanto appreso nella creazione di un videogioco completo, sviluppando al contempo competenze trasversali come il team building e la gestione del lavoro collaborativo. Ai fini dell'orientamento, il corso introduttivo ha proposto argomenti legati a paradigmi di sviluppo, linguaggi di programmazione e strumenti professionali comunemente affrontati nei corsi universitari di informatica. Questa esperienza ha consentito agli studenti di esplorare praticamente alcuni contenuti che potrebbero incontrare nel loro futuro percorso accademico, favorendo al contempo la familiarizzazione con l'ambiente universitario.

Tra le limitazioni dell'iniziativa è emersa l'organizzazione dell'evento su più settimane, che ha reso complessa la partecipazione per gli istituti fuori regione. Questo aspetto verrà considerato nella progettazione di future edizioni in giorni consecutivi per favorire una più ampia partecipazione.

Ringraziamenti

Gli autori ringraziano Ester Orlandi per il supporto tecnico e Nicola Zangrando per aver condiviso la sua esperienza maturata in precedenti edizioni della game jam internazionale. Parte del lavoro di Alessandro Forgiarini è sostenuta da una borsa di dottorato del Ministero dell'Università e della Ricerca (PNRR M4C1 Inv. 4.1 D.M. 351/2022).

Bibliografia

- [1] Sue Bennett, Karl Maton, and Lisa Kervin. The 'digital natives' debate: A critical review of the evidence. *British Journal of Educational Technology*, 39(5):775–786, 2008.
- [2] David Boud, Ruth Cohen, and Jane Sampson. Peer learning and assessment. *Assessment & evaluation in higher education*, 24(4):413–426, 1999.
- [3] Cheng-Huan Chen and Yong-Cih Yang. Revisiting the effects of project-based learning on students' academic achievement: A meta-analysis investigating moderators. *Educational Research Review*, 26:71–81, February 2019.
- [4] Michail N. Giannakos, Trond Aalberg, Monica Divitini, Letizia Jaccheri, Patrick Mikalef, Ilias O. Pappas, and Guttorm Sindre. Identifying dropout factors in information technology education: A case study. In *2017 IEEE Global Engineering Education Conference (EDUCON)*, pages 1187–1194, April 2017.
- [5] Michail N. Giannakos, Ilias O. Pappas, Letizia Jaccheri, and Demetrios G. Sampson. Understanding student retention in computer science education: The role of environment, gains, barriers and usefulness. *Education and Information Technologies*, 22(5):2365–2382, September 2017.
- [6] Patricia M. Greenfield. Technology and Informal Education: What Is Taught, What Is Learned. *Science*, 323(5910):69–71, January 2009.
- [7] Hannam University and Myeong-Hee Shin. Effects of Project-based Learning on Students'

- Motivation and Self-efficacy. *English Teaching*, 73(1):95–114, March 2018.
- [8] Barbara M. Kehm, Malene Rode Larsen, and Hanna Bjørnøy Sommersel. Student dropout from universities in Europe: A review of empirical literature. *Hungarian Educational Research Journal*, September 2019.
- [9] D A Kilroy. Problem based learning. *Emergency Medicine Journal*, 21(4):411–413, July 2004.
- [10] K-W Lai, Ferial Khaddage, and Gerald Knezek. Blending student technology experiences in formal and informal learning. *Journal of computer assisted learning*, 29(5):414–425, 2013.
- [11] Timothy J. Nokes-Malach, J. Elizabeth Richey, and Soniya Gadgil. When Is It Better to Learn Together? Insights from Research on Collaborative Learning. *Educational Psychology Review*, 27(4):645–656, December 2015.
- [12] Marina Papastergiou. Digital Game-Based Learning in high school Computer Science education: Impact on educational effectiveness and student motivation. *Computers & Education*, 52(1):1–12, January 2009.
- [13] Ilias O. Pappas, Michail N. Giannakos, and Letizia Jaccheri. Investigating Factors Influencing Students’ Intention to Dropout Computer Science Studies. In *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education*, pages 198–203, Arequipa Peru, July 2016. ACM.
- [14] Roger C. Schank, Tamara R. Berman, and Kimberli A. Macpherson. *Learning by Doing. In Instructional- Design Theories and Models*. Routledge, 1999.
- [15] Gavin Tierney, Rochelle Urban, and Gina Olabuenaga. Designing for Equity: Moving Project-Based Learning from Equity Adjacent to Equity Infused. *Interdisciplinary Journal of Problem-Based Learning*, 18(1), 2024.

Verso una didattica della comunicazione con l'AI

Francesco Lombardi¹

¹IISS Volta De Gemmis, Bitonto Bari

francesco.lombardi.sg28391@iissvoltadegemmis.edu.it

Abstract

Il presente contributo propone un'analisi critica dell'impatto dell'Intelligenza Artificiale Generativa (AI) non soltanto in termini di incremento della produttività degli studenti, ma soprattutto rispetto allo sviluppo delle competenze informatiche all'interno di attività didattiche che ne prevedono l'uso come ambiente computazionale. L'attenzione è focalizzata in particolare sui contesti a forte componente laboratoriale, nei quali l'integrazione dell'AI si configura come catalizzatore di pratiche progettuali e sperimentali, in linea con i principi del Project-Based Learning (PBL). L'articolo intende offrire spunti teorici e applicativi per una didattica innovativa, orientata alla formazione di competenze digitali avanzate e critiche.

1. Introduzione

L'obiettivo del presente articolo è analizzare in modo critico l'impatto dell'utilizzo dell'Intelligenza Artificiale Generativa (AI) non solo sulla produttività degli studenti, ma principalmente sulle competenze informatiche sviluppate attraverso attività didattiche in cui l'AI viene utilizzata come ambiente computazionale e non solo come strumento ausiliario. Particolare attenzione è rivolta ai contesti a forte componente laboratoriale, dove l'integrazione dell'AI assume il ruolo di catalizzatore di pratiche di tipo progettuale e sperimentale, tipiche del Project-Based Learning (PBL). L'intento è indagare se e in che modo gli strumenti basati sull'AI possano contribuire a rafforzare le pratiche e i concetti dell'informatica nella scuola secondaria, supportando percorsi di apprendimento attivo centrati su problem solving, modellazione e validazione delle soluzioni. Non si tratta, quindi, di verificare l'efficacia dell'AI come tecnologia educativa, ma di analizzarne l'impiego come contesto operativo in cui si sviluppa il pensiero computazionale.

Quest'ultimo è qui inteso come la capacità di:

- *analizzare e formulare un problema,*
- *scomporlo in sotto-problemi,*
- *individuare gli elementi rilevanti su cui intervenire,*
- *definire e rappresentare le relazioni tra tali elementi (modellazione),*
- *organizzare le informazioni in strutture logiche,*
- *costruire, attraverso l'astrazione e la generalizzazione, algoritmi complessi o soluzioni automatizzabili.*

Nel lavoro con l'AI, lo studente agisce in un contesto simile a quello del debugging analizzando l'output generato, riflettendo sulla coerenza delle risposte, riformulando prompt o condizioni iniziali e iterando il processo secondo il ciclo tipico dello sviluppo computazionale:

prova – errore – apprendimento – ripetizione

Tale processo, oltre a stimolare un apprendimento attivo di tipo esperienziale, porta a integrare diverse dimensioni cognitive centrali nell'informatica, come:

- *Attivo vs Riflessivo*
- *Sensoriale vs Intuitivo*
- *Visivo vs Verbale*
- *Sequenziale vs Globale*

2. Contesto disciplinare

L'esperienza, svolta nella disciplina TPSIT in una classe del secondo biennio dell'indirizzo Informatica, ha previsto un'attività PCTO articolata in due fasi: una formazione asincrona su HTML e JavaScript tramite videolezioni e una fase progettuale in cui gli studenti hanno sviluppato applicazioni client-side. L'obiettivo era applicare competenze informatiche in un contesto realistico, favorendo l'analisi del problema, la modellazione e il coding. L'eventuale uso dell'AI è stato orientato alla scrittura e verifica del codice, stimolando riflessione critica, debugging e pensiero computazionale, in coerenza con gli obiettivi disciplinari e lo sviluppo di competenze professionali ICT.

La traccia da sviluppare era:

STEAM Who?: applicazione con l'obiettivo di identificare, tramite domande e risposte, un personaggio noto appartenente alle discipline STEAM (Science, Technology, Engineering, Arts, Mathematics).

Al fine di valutare l'impatto concreto dell'uso dell'AI, è stato richiesto a tutti gli studenti di utilizzare uno strumento di Intelligenza Artificiale Generativa (nello specifico ChatGPT modello 3.5) durante lo sviluppo del proprio progetto. Il periodo di svolgimento delle attività progettuali si è esteso per circa sei mesi, con inizio il 1° gennaio e conclusione il 7 giugno 2025.

3. Analisi dei tempi di consegna

- Entro il 15 febbraio 2025, solo 5 studenti (22,72%) avevano consegnato un progetto funzionante. Questi studenti hanno mostrato una maggiore autonomia, propensione alla sperimentazione e motivazione intrinseca.
- Da metà febbraio a fine aprile non sono pervenute nuove consegne, nonostante il docente avesse assegnato task intermedi e sollecitazioni, come revisioni parziali o incontri di verifica.
- Nel mese di maggio, in prossimità della scadenza finale, si è registrata un'impennata nelle consegne: 13 studenti (59,09%) hanno consegnato il proprio elaborato.
- Quattro studenti (18,18%) non sono riusciti a completare il progetto entro la scadenza. In questi casi, si è osservata una combinazione di scarsa motivazione, difficoltà organizzative e, talvolta, sopravvalutazione delle capacità dell'AI.

4. Analisi qualitativa di casi esemplari

- **Studente 1** [1] mostra sin da subito l'ambizione di emulare il comportamento e l'estetica dell'applicazione Akinator. Dopo aver analizzato modelli esistenti, ha utilizzato l'AI come supporto per scomporre il problema in sotto-obiettivi funzionali e per strutturare il progetto secondo una logica modulare. L'interazione con l'AI lo ha guidato nella modellizzazione del flusso di domande e risposte, nell'ottimizzazione dei prompt e nella validazione degli output, attraverso un confronto critico tra le risposte generate e il comportamento atteso. Il lavoro sul debugging tramite console del browser, svolto con il docente, ha rafforzato la comprensione del codice e introdotto buone pratiche come il responsive design e la gestione ordinata dei file per tipologia.

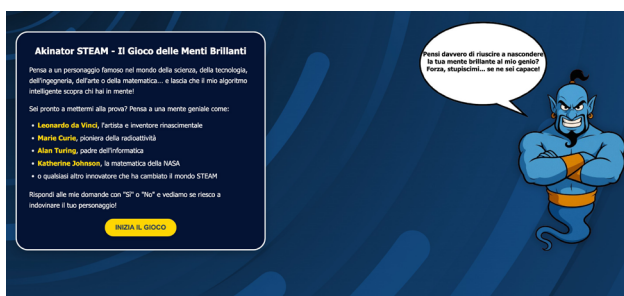


Figura 1: Sviluppo di Akinator STEAM

- **Studente 2** [2], ha realizzato un elaborato di livello intermedio, distinguendosi per la capacità di modellare in modo coerente l'interfaccia e la struttura dell'applicazione. L'uso dell'AI lo ha supportato nella definizione gerarchica degli elementi grafici, nell'organizzazione modulare del codice e nella sperimentazione di soluzioni per implementare effetti visivi avanzati. L'applicazione dei vincoli (20 personaggi STEAM) è stata gestita come sotto-problema da integrare nel progetto complessivo, mostrando attenzione alla coerenza tra contenuti e struttura funzionale. Il confronto con il docente e l'adozione dei suggerimenti ricevuti hanno favorito un processo iterativo di verifica e raffinamento degli output, consolidando le abilità di progettazione e controllo della qualità.



Figura 2: Sviluppo traccia STEAM Who?

- **Studente 3** [3], pur non avendo soddisfatto pienamente tutti i requisiti della traccia, ha mostrato solide competenze nell'analisi critica del problema e nella valutazione dei vincoli progettuali. È stato l'unico a interrogarsi sulla coerenza logica del sistema, osservando che un numero limitato di personaggi comprometteva la significatività delle domande. A partire da questa riflessione, ha modellizzato un'alternativa funzionale: un'interfaccia in cui l'utente seleziona il personaggio prima di rispondere. Questa modifica ha richiesto la riorganizzazione del flusso interattivo e l'adattamento del codice, mostrando capacità di scomposizione del problema, progettazione autonoma e ricerca di soluzioni realistiche, anche in deroga al modello iniziale.

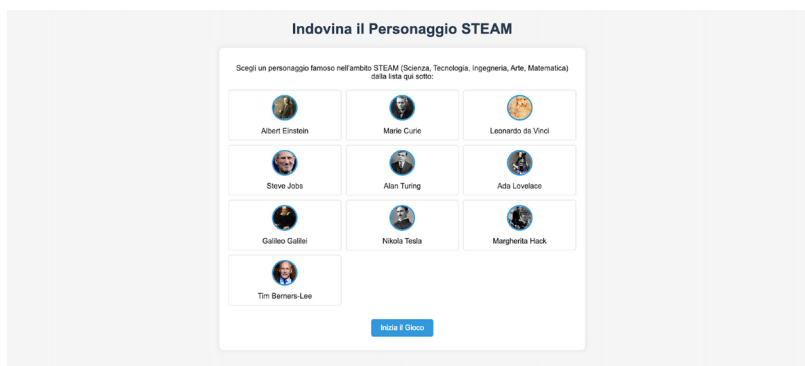


Figura 3: Sviluppo traccia STEAM Who?

Studente 4 [4], ha realizzato un'applicazione essenziale, strutturata attorno a una sequenza fissa di 5 domande finalizzate alla selezione tra un numero limitato di personaggi. L'assenza di scomposizione strutturata del problema e la mancata modellizzazione del flusso interattivo hanno ridotto il progetto a una logica lineare e poco adattabile. L'interfaccia è stata curata in modo minimale, con modifiche superficiali ai colori, senza considerazioni progettuali approfondite. L'uso dell'AI si è limitato all'esecuzione di richieste puntuali, senza un'elaborazione critica o una riflessione sugli output. L'interazione con il docente non ha prodotto miglioramenti sostanziali, evidenziando una limitata attivazione di processi di revisione e autovalutazione.

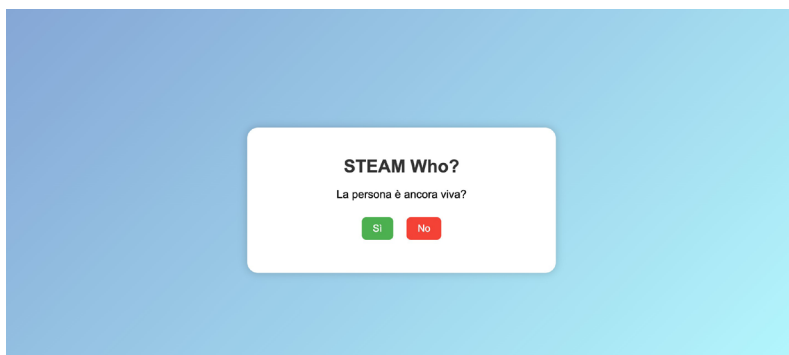


Figura 4: Sviluppo traccia STEAM Who?

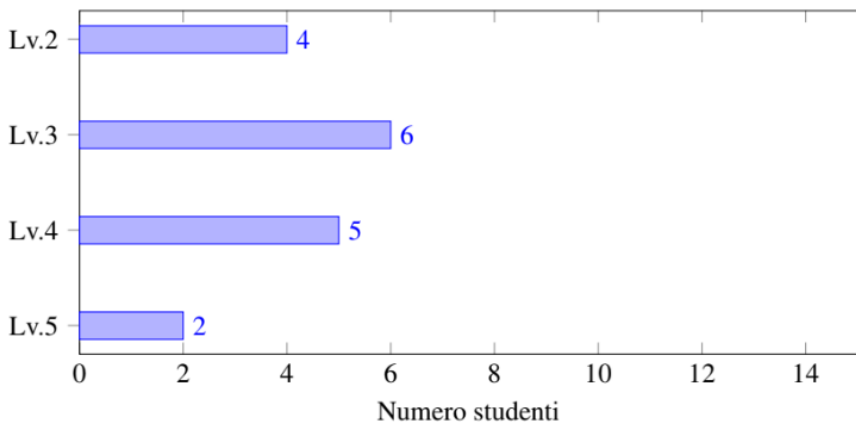


Figura 5: Numero di studenti suddivisi per livello di qualità del prodotto

5. Competenze acquisite attraverso l'AI

L'attività ha favorito competenze trasversali quali la capacità di risoluzione dei problemi, il ragionamento logico, la riflessione sull'errore e la capacità di apprendere dall'esperienza, secondo uno stile di apprendimento attivo/esperienziale.

Declinando l'esperienza tendendo presente il modello di Felder-Silverman [5], l'interazione con l'AI può essere interpretato come un ciclo iterativo di:

- *formulazione del prompt (azione),*
- *ricezione del risultato (osservazione),*
- *valutazione e correzione (riflessione),*
- *reformulazione (nuova azione).*

L'esperienza progettuale ha favorito lo sviluppo di concetti fondamentali dell'informatica, come la rappresentazione dell'informazione, l'interazione uomo-macchina e l'automazione, stimolando negli studenti capacità di astrazione, modellazione e riflessione critica tipiche del pensiero computazionale.

6. Conclusioni

L'integrazione dell'Intelligenza Artificiale Generativa nel contesto didattico ha rappresentato un'opportunità concreta per sviluppare competenze trasversali e tecnico-disciplinari di rilievo. L'utilizzo di strumenti come ChatGPT ha favorito nei discenti l'acquisizione di abilità legate al problem solving e al ragionamento logico, elementi centrali del pensiero computazionale. L'interazione con l'AI ha inoltre promosso un approccio attivo all'apprendimento, in cui lo studente è stato chiamato a formulare richieste efficaci (prompting), interpretare criticamente le risposte, valutarne la coerenza rispetto agli obiettivi progettuali e riformulare le richieste in

modo iterativo. Questo ciclo ha contribuito allo sviluppo della capacità di analisi, dell'autonomia operativa e della metacognizione, rendendo l'AI non solo uno strumento di supporto tecnico, ma un facilitatore per l'esplorazione di strategie di apprendimento più consapevoli e riflessive.

L'esperienza ha mostrato che l'Intelligenza Artificiale Generativa, pur rappresentando un supporto concreto, non sostituisce le competenze degli studenti, né tantomeno il ruolo del docente. Gli studenti hanno utilizzato l'AI in modo fortemente personalizzato: alcuni come strumento strategico per risolvere problemi e sviluppare idee complesse; altri in modo passivo, affidandosi acriticamente ai suggerimenti generati. Infine, l'introduzione dell'AI generativa ha sollevato interrogativi metodologici e valutativi: come riconoscere l'effettivo apporto dello studente in presenza di contenuti generati? Come valutare la qualità del processo e non solo del prodotto? L'esperimento ha aperto spunti di riflessione preziosi, confermando la necessità di aggiornare i modelli didattici affinché includano strumenti di AI come parte integrante, consapevole e critica dell'esperienza educativa.

Bibliografia

- [1] Url al prodotto studente 1. https://cortomaltese83.github.io/24_25/ITADINFO/PCTO/STUDENTE_1/. A.S. 2024/25.
- [2] Url al prodotto studente 2. https://cortomaltese83.github.io/24_25/ITADINFO/PCTO/STUDENTE_2/. A.S. 2024/25.
- [3] Url al prodotto studente 3. https://cortomaltese83.github.io/24_25/ITADINFO/PCTO/STUDENTE_3/. A.S. 2024/25.
- [4] Url al prodotto studente 4. https://cortomaltese83.github.io/24_25/ITADINFO/PCTO/STUDENTE_4/. A.S. 2024/25.
- [5] Richard M. Felder and Linda K. Silverman. Learning and teaching styles in engineering education. *Engineering Education*, 78(7):674–681, 1988..

Una raccolta di modelli in linguaggio Python per la didattica delle applicazioni di ML

Giuliana Barberis¹

¹ Liceo M. Curie di Pinerolo

giuliana.barberis@gmail.com

Abstract

Questo articolo presenta un percorso didattico innovativo sull'Intelligenza Artificiale (AI) e il Machine Learning (ML), specificamente progettato per studenti delle scuole secondarie di secondo grado. Basato su un corso già erogato e pubblicato su una piattaforma Moodle, il materiale didattico è disponibile con licenza Creative Commons e copre un ampio spettro di argomenti: partendo dagli aspetti etici e le precauzioni d'uso dell'AI, per poi approfondire le sue applicazioni nello studio e il suo funzionamento intrinseco, sottolineandone i limiti; vengono inoltre introdotti concetti fondamentali come big data, data science e diverse tipologie di ML e addestramento. L'articolo si focalizza in particolare su esempi pratici di ML fornendo anche il relativo codice Python, iniziando da un modello di regressione lineare con dati simulati per facilitare la comprensione della validazione del modello, per poi passare a un'applicazione su dati reali ("I soldi fanno la felicità?"). Successivamente, l'articolo descrive l'algoritmo KNN, dimostrandone l'applicazione allo stesso problema e al famoso dataset degli iris. Viene descritto un approccio progressivo che introduce concetti complessi come la valutazione del modello, l'overfitting e l'underfitting attraverso esempi pratici. L'obiettivo è fornire agli educatori e agli studenti un percorso chiaro e pratico per comprendere e applicare i principi fondamentali del ML.

1. Introduzione

Questo articolo si basa su parte del materiale didattico di un corso di intelligenza artificiale rivolto a studenti delle scuole secondarie di secondo grado. Il corso ha avuto come obiettivo la diffusione tra gli studenti di strumenti e tecniche per l'utilizzo dell'AI: inizialmente sono stati presentati gli aspetti etici e le precauzioni d'uso; successivamente sono state trattate le applicazioni che possono essere di ausilio allo studio; poi è stato dato largo spazio all'esposizione dell'Intelligenza Artificiale dal punto di vista del suo funzionamento sottolineandone le limitazioni, in quanto non è un agente che esercita un pensiero autonomo bensì un processo che elabora un input e produce un output.

Sono stati introdotti i concetti di:

- big data;
- data science;
- tipi di Machine Learning;
- tipi di addestramento.

Infine, sono state introdotte alcune applicazioni classiche per la didattica del ML con relativo codice in Python (con l'ausilio della piattaforma Colab).

I modelli sono stati presentati con le relative applicazioni di esempio, in ordine di “difficoltà” crescente, in un percorso didattico pensato per permettere l'introduzione di concetti anche complessi come la valutazione del modello, l'overfitting, l'underfitting e la backpropagation nel momento in cui l'applicazione potesse renderne più chiara la spiegazione:

- Modelli di regressione lineare
 - Modello con dati simulati
 - Modello “I soldi fanno la felicità?”
- Modelli KNN
 - Trasformazione in KNN del modello “I soldi fanno la felicità?”
 - Modello per la classificazione degli iris
 - Modello “Pubblicità Mirata sui Social: Chi Acquista Davvero?”
 - Modello “Predizione del successo degli studenti”
- Modelli K-Means
 - Modello di “Clustering per la personalizzazione dei percorsi di apprendimento”
- Modelli Naive Bayes
 - Gioco “Sentiment Analysis sulle recensioni dei film”
 - Modello “Filtro antispam”
 - Modello “Previsione diabete”
 - Modello “Predizione tumore al seno”
- Modelli PCA:
 - Modello “Riduzione della Dimensionalità (PCA) sui Dati di Analisi del Sangue”
- La rete neurale artificiale
 - Gioco “Facciamo rete”
 - Modello “Riconoscimento di cifre scritte a mano”
 - Modello “Classificazione di immagini di capi di abbigliamento”

Tutto il materiale è pubblicato su una piattaforma Moodle all'indirizzo: <https://www.informaticabitabit.it/moodle/course/view.php?id=30> (Barberis, 2025) liberamente fruibile con licenza creative commons, dove possiamo trovare tutte le attività, i contenuti e i codici (notebook di Colab) citati.

Il corso si è articolato in 12 lezioni da 2h 30 minuti ciascuna (gennaio/aprile 2025), tutte le lezioni sono state svolte nel laboratorio di informatica dove ogni studente ha potuto utilizzare la propria postazione.

Hanno frequentato le lezioni di questo corso 18 studenti, 17 del liceo scientifico delle scienze applicate e 1 del liceo scientifico normale: 2 studenti di prima, 1 di seconda, 6 di terza, 9 di quarta. Tutti gli studenti avevano voglia di programmare e passione per il coding, ma erano a livelli di competenza molto diversi tra loro, la difficoltà maggiore è stata quindi quella di rendere le lezioni comprensibili a tutti, per questo si è scelto un approccio il più possibile pratico: privilegiando la metodologia “learning by doing”.

In questo articolo analizziamo i primi due modelli (regressione lineare e KNN) e il percorso didattico proposto, illustrando la strategia adottata soprattutto per la parte di coding.

	Argomento	Metodologia
1	AI Act e AI generativa, presentazioni, immagini, video e schemi	Nella prima parte della lezione è stata introdotta la problematica della regolamentazione dell'AI: AI Act e sue applicazioni. Nella seconda parte è stato dato ampio spazio alla sperimentazione degli studenti rispetto alle applicazioni di AI che consentono la creazione di presentazioni, immagini e schemi
2	La programmazione dei prompt, può la AI riconoscersi?	In questa lezione è stato svolto un lavoro a gruppi: ciascun gruppo ha selezionato 5 testi generati da un chatbot e 5 testi scritti da umani (articoli di giornale, testi di esercizi proposti da un insegnante, testi recuperati da pagine web, purché non provenienti da libri famosi e di sicura provenienza umana), cercando di mantenere uno stile omogeneo tra i testi, variando temi come narrativa, descrizione, e contenuto informativo. Ogni gruppo ha poi presentato i testi in ordine casuale ad un chatbot diverso, chiedendo di indicare per ciascun testo: se il testo fosse stato scritto da un'AI o da un umano. L'obiettivo era quello di cercare di capire quanto fosse possibile per l'AI identificare la provenienza di un testo.
3	Introduzione al Machine Learning	Inizialmente sono stati forniti dall'insegnante, con l'aiuto di una presentazione, i concetti di: big data, tipi di apprendimento, fasi di realizzazione di una applicazione di machine learning, funzionamento generale di un'applicazione con apprendimento supervisionato. In seguito, è stato spiegato il funzionamento del "test di Turing" e gli studenti hanno sperimentato praticamente il meccanismo mettendo alla prova la capacità di ChatGPT o altro chatbot di imitare il linguaggio umano: in ogni gruppo di tre studenti: uno studente aveva il ruolo di interrogatore (lo studente che pone le domande), uno di "chatbot" (lo studente che utilizza un chatbot per ottenere risposte), uno di "umano" (lo studente che risponde in autonomia, cercando di sembrare un chatbot). L'obiettivo era vedere quante volte l'interrogatore riusciva a distinguere l'umano dal chatbot.
4	Il problema dei pregiudizi umani insegnati all'AI	All'inizio della lezione è stato introdotto il problema dell'addestramento e la necessità di avere a disposizione grandi quantità di dati organizzati in modo adeguato. Sono stati mostrati alcuni video tra i quali spezzoni significativi del documentario "Coded bias". Nella seconda parte della lezione la classe è stata suddivisa in due gruppi, ciascun gruppo ha scelto un argomento (da una lista proposta dall'insegnante ma con la possibilità di fare proposte originali, ad esempio "uso dell'Intelligenza Artificiale nell'Educazione: è giusto che gli educatori utilizzino l'AI per preparare le proprie lezioni?"). Ciascun gruppo si è diviso in due parti una parte ha preparato un debate a favore dell'argomento e una parte contro, secondo le regole consuete del debate.

5	Python Colab - un modello di regressione lineare	All'inizio della lezione si è svolto il debate preparato nella lezione precedente. Successivamente è stato sperimentato l'addestramento di un modello di ML con la piattaforma "The teachable machine" e l'insegnante ha presentato l'uso della piattaforma Colab per scrivere ed eseguire programmi in Python attraverso un esempio pratico di modello di regressione lineare.
6	"I soldi fanno la felicità?"	Gli studenti sono stati divisi in gruppi, ciascun gruppo ha adattato il modello di regressione lineare visto al termine della lezione 5 a un dataset composto dalle informazioni di "prodotto interno lordo pro-capite" per nazione e dall' "indice di soddisfazione di vita" per nazione (un valore in $[0, 10]$), l'obiettivo era rispondere alla domanda "i soldi fanno la felicità?".
7	Il modello KNN	L'insegnante ha inizialmente illustrato la tecnica KNN sul modello di classificazione degli iris, successivamente la classe è stata divisa in gruppi, ogni gruppo aveva l'obiettivo di adattare il modello KNN al problema "Pubblicità Mirata sui Social: Chi Acquista Davvero?" basato su un dataset contenente dati sugli utenti di un sito di social networking, per ciascun utente sono presenti i dati di stipendio, età e genere e se ha acquistato un prodotto cliccando su un annuncio pubblicitario del sito.
8	Il modello Naive Bayes sull'identificazione dei messaggi di spam	L'insegnante ha presentato all'inizio della lezione il concetto di "sentiment analysis", successivamente la classe è stata suddivisa in gruppi e ha sperimentato un gioco di "sentiment analysis sulle recensioni dei film", classificando le recensioni dei film secondo parole chiave scelte appositamente. L'insegnante ha infine illustrato il funzionamento dell'applicazione del modello Naive Bayes sull'identificazione dei messaggi di spam.
9	La rete neurale biologica e la rete neurale informatica	Sono stati presentati agli studenti due video interattivi, uno sul funzionamento della rete neurale biologica e uno sulla rete neurale informatica, alla fine della lezione è stato presentato il gioco "facciamo rete con l'AI".
10	La Backpropagation: il perceptrone di Rosenblatt e il modello Mnist	All'inizio della lezione l'insegnante ha presentato il funzionamento del perceptrone e il significato degli iperparametri e del bias attraverso un programma in python (Colab). Il modello è stato applicato anche al riconoscimento dei numeri scritti a mano nel dataset MNIST.
11	Il modello Fashion Mnist	Il modello MNIST è stato generalizzato nel modello "fashion MNIST" per la classificazione di immagini di capi d'abbigliamento mediante una rete neurale.
12	Modelli LLM - configuriamo un semplice chatbot	È stato preso in considerazione l'utilizzo di un risponditore automatico per gestire automaticamente le domande degli utenti relative al funzionamento di una scuola o di un piccolo comune, la classe è stata divisa in gruppi, ad ogni gruppo è stato chiesto di configurare un chatbot con uno strumento automatizzato come "botpress".

2. Modelli di regressione lineare

I modelli di regressione lineare sono forse quelli che meglio promuovono il tipo di approccio semplice che permette per entrare gradualmente in argomento.

Partiamo da un esempio “simulato” costruendo un dataset costituito da coppie di valori, che potrebbero rappresentare per esempio la media della statura dei genitori e la statura del figlio, o l'umidità e la temperatura dell'aria, o la distanza dal mare e il prezzo al mq di un immobile.

L'obiettivo è quello di ottenere un dataset a partire da una funzione regolare aggiungendo un po' di “rumore”, addestrare un modello di regressione lineare con una parte del data set, valutare il modello con la parte restante e dimostrare che si è ritrovata la funzione di partenza.

Apparentemente questo metodo potrebbe sembrare un circolo vizioso, ma è funzionale all'idea di presentare il funzionamento di un modello di ML in modo da renderlo indipendente dall'incertezza dei dati reali.

In questa prima fase si considera il dataset costruito a partire dai punti di una funzione, scegliamo per esempio $f(x) = 3x^2 + 2x + 1$ nell'intervallo $[-10, 10]$ e aggiungiamo del “rumore” casuale, utilizziamo questi dati per addestrare un modello di regressione lineare.

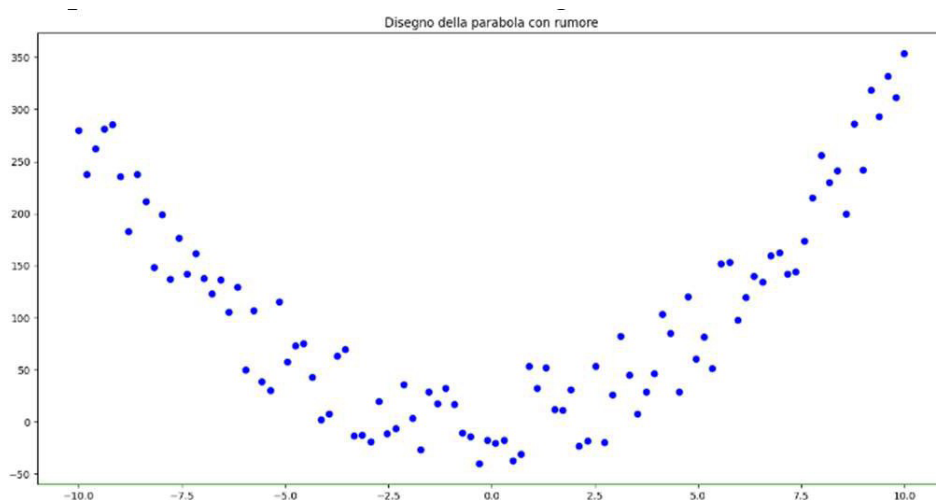


Figura 1: Punti del dataset di addestramento

Si tratterà di un modello predittivo, quindi, una volta addestrato, se gli sottoponiamo un valore di X non compreso nel dataset, esso sarà in grado di predire l'ordinata corrispondente.

Utilizziamo la libreria sklearn di Python che contiene molti comandi utili alla realizzazione di ML, in questo caso ci serviamo del modulo linearRegression, il modello troverà i punti che si posizionano il più vicino possibile a tutti i punti del dataset: il punto rosso (la nostra predizione) viene “spostato” finché la somma dei quadrati delle distanze da n punti limitrofi non è minima.

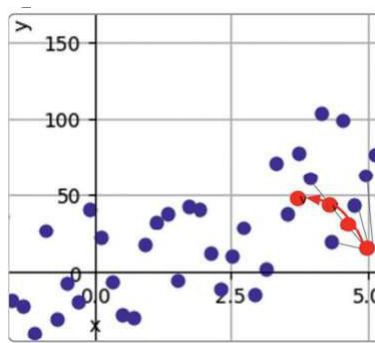


Figura 2: Regressione

Il modulo `linearRegression` prevede che la relazione sia lineare, in questo caso sappiamo che è quadratica e quindi dobbiamo trasformare la variabile `X` in modo da catturare la non linearità aggiungendo un termine quadratico alla funzione di regressione.

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

# Step 2: Addestramento del modello
# Prepariamo i dati per la regressione lineare
# La regressione lineare non può gestire direttamente il termine quadratico,
# quindi dobbiamo espandere la feature X
X_poly = np.vstack((X, X**2)).T

# Addestriamo il modello
model = LinearRegression()
model.fit(X_poly, y)
```

Figura 3: Codice Python per l'addestramento

Ora possiamo valutare il modello che abbiamo addestrato, generiamo altri punti della parabola con rumore, sottoponiamo questi valori al nostro modello che ci darà la sua predizione sul valore di ciascuna `X_poly_v` del dataset di test.

```
# Step 3: Valutazione del modello
f = open("dati1.txt", "r")
X_valuta = np.array([])
y_valuta = np.array([])
for line in f:
    line = line.strip()
    dati=line.split(", ")
    X_valuta = np.append(X_valuta, float(dati[0]))
    y_valuta = np.append(y_valuta, float(dati[1]))
X_poly_v = np.vstack((X_valuta, X_valuta**2)).T
y_pred_v = model.predict(X_poly_v)
```

Figura 4: Codice Python per la valutazione

Abbiamo ottenuto così un array di valori previsionali (y_pred_v), per ciascun valore abbiamo anche il valore esatto (y_valuta) perché si tratta di addestramento supervisionato e abbiamo anche la parabola di partenza. Disegniamo sul piano cartesiano la predizione, il dataset e la parabola di partenza per verificare che siano concordi, e lo devono essere perché i dati sono simulati:

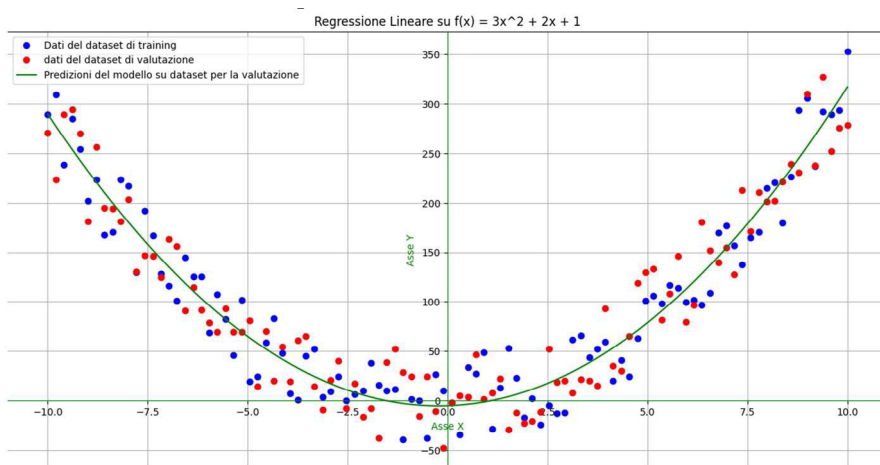


Figura 5: Confronto tra dataset di training e di valutazione

Per avvicinarsi ad un'applicazione pratica, realizziamo un modello di regressione su dati reali, il modello deve rispondere alla domanda "I soldi fanno la felicità", l'esercitazione è tratta da un libro fondamentale per la didattica dell'AI, si tratta di "Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow", Concept, Tools and Techniques to Build Intelligent Systems (O'Reilly, 2021). In questo caso avremo il nostro dataset di test composto dalle informazioni di "prodotto interno lordo pro-capite" per nazione e dall' "indice di soddisfazione di vita" per nazione (un valore in $[0, 10]$)

Addestriamo il modello con una parte del dataset, valutiamolo con la parte restante e facciamo predire al modello l'indice di soddisfazione di vita partendo dal prodotto interno lordo per le nazioni per le quali non abbiamo l'indice di soddisfazione.

Il dataset di training e di valutazione sono già stati realizzati a partire dai dati che si possono raggiungere ai link:

- prodotto interno lordo pro-capite: <https://homl.info/4>, che rimanda al sito dell'OECD
- indice di soddisfazione di vita: <https://homl.info/5> che rimanda al sito dell'International Monetary Fund

Il notebook Colab di questo modello si trova nel corso Moodle (Barberis, 2025) allegato a questo articolo, si otterrà questo risultato:

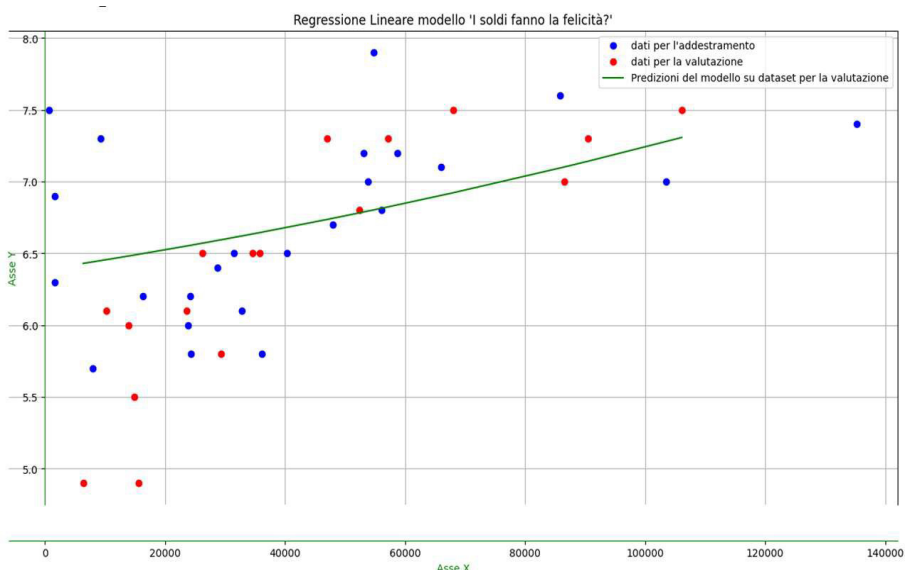


Figura 6: Modello "I soldi fanno la felicità?" con regressione

Come si vede dal grafico la nuvola dei punti (rossi e blu) non è approssimata in modo molto preciso dalla retta di regressione (in verde), per valutare l'errore del modello, calcoliamo l'errore quadratico medio; poi calcoliamo la varianza dei dati (y_{valuta}) e confrontiamo la varianza con l'errore: se l'errore calcolato $<$ varianza, il modello sta spiegando bene i dati, se l'errore calcolato $\approx$ varianza o l'errore calcolato $>$ varianza, il modello non sta catturando bene la relazione tra input e output.

```

▶ # Calcoliamo il Mean Squared Error (MSE)
mse = mean_squared_error(y_valuta, y_pred_v)
print(f'Mean Squared Error: {mse}')
# Calcoliamo la varianza dei dati di output
var_y = np.var(y_valuta, ddof=1) # ddof=1 per avere la varianza campionaria
print(f'Varianza di y_valuta: {var_y}')

➡ Mean Squared Error: 0.47345236629591614
   Varianza di y_valuta: 0.7113235294117646

```

Figura 7: Valutazione dell'errore

In questo caso l'errore è minore della varianza, ma non c'è molta differenza fra i due valori e dunque forse si può trovare un modello di ML migliore per la previsione.

3. Modelli KNN

L'algoritmo k-nearest neighbors (KNN) è un modello di ML ad apprendimento supervisionato non parametrico, che utilizza la prossimità per effettuare previsioni o clustering di un singolo punto sul piano, possiamo per esempio strutturare il dataset di apprendimento come coppie di valori che sia possibile disegnare su un piano cartesiano.

L'obiettivo dell'algoritmo KNN è quindi identificare i K vicini più prossimi di un dato punto, in modo da poter assegnare un'etichetta di classificazione a quel punto, si parte dal presupposto che punti simili possano essere trovati vicini l'uno all'altro, KNN decide la classe del nuovo punto basandosi sulla maggioranza delle classi di questi K vicini.

Per determinare quali punti sono più vicini a un determinato punto X, è necessario calcolare la distanza tra X e gli altri punti: si possono usare diversi metodi di calcolo, detti metriche.

Qui di seguito la formula per la distanza di Minkowski, dove il parametro p ne rappresenta la "sensibilità", x_i e y_i sono le componenti di X:

$$\left(\sum_{i=1}^n |x_i - y_i|^p \right)^{1/p}$$

- $p = 1$, metrica di Manhattan (geometria del taxi o distanza di isolato)
- $p = 2$, metrica euclidea (più comune: è la misura del segmento tra i due punti)

KNN non necessita di un vero e proprio addestramento, il problema è solo quello di determinare il K migliore nell'ottica di avere la previsione più accurata.

Proviamo ad applicare il KNN sul problema "I soldi fanno la felicità?" per mostrare che ad un problema possiamo applicare algoritmi diversi per poi scegliere quello che ci da risultati migliori. Dobbiamo semplicemente sostituire il modulo `sklearn.LinearRegression` con il modulo `sklearn.neighbors` e utilizzarlo per l'addestramento, in questo caso abbiamo provato con $K = 9$ (parametro `n_neighbors`):

```
import sklearn.neighbors
from sklearn.metrics import mean_squared_error
model = sklearn.neighbors.KNeighborsRegressor(n_neighbors=9)
# riorganizza il vettore X in modo che abbia una colonna e
# tante righe quante erano gli elementi X
X_resaped = X.reshape(-1, 1)
model.fit(X_resaped, y)
```

Figura 8: Il modello KNN

Valutando l'errore del modello calcolando l'errore quadratico medio, otteniamo un valore di media = 0.376 e di varianza = 0.711, che migliora il risultato ottenuto con il modello di regressione lineare, come dimostra il grafico di confronto tra il dataset di addestramento, quello di valutazione e le predizioni, le linea verde ha un andamento che sembra andare a “approssimare” meglio la nuvola di punti del dataset:

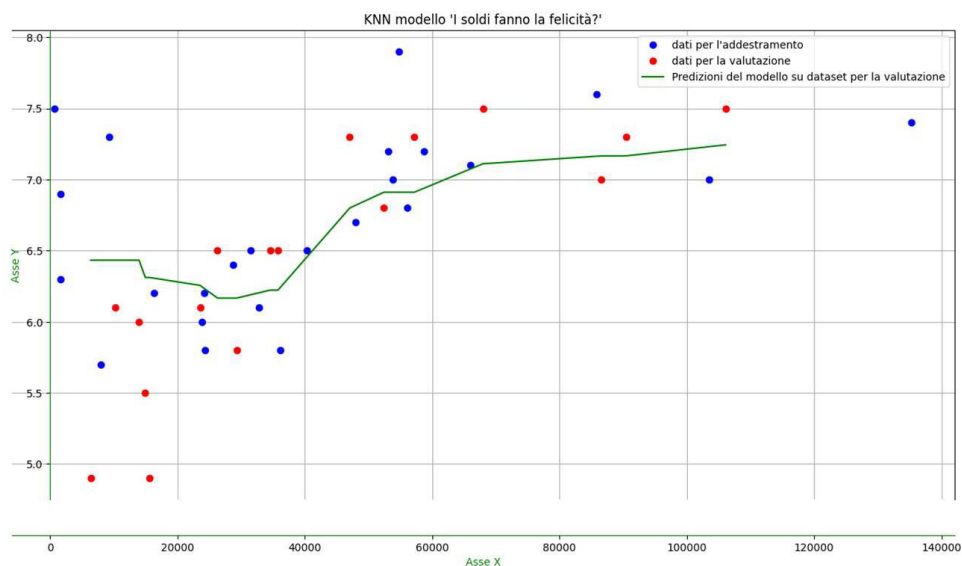


Figura 9: Modello “I soldi fanno la felicità?” con KNN ($K=9$)

A questo punto si può osservare che, banalmente, potremmo pensare di aumentare molto il parametro K per cercare l'approssimazione migliore. Potremmo quindi fare una prova con $K=30$, ottenendo un valore di media = 0.652 e di varianza = 0.711 che peggiora le prestazioni del modello:

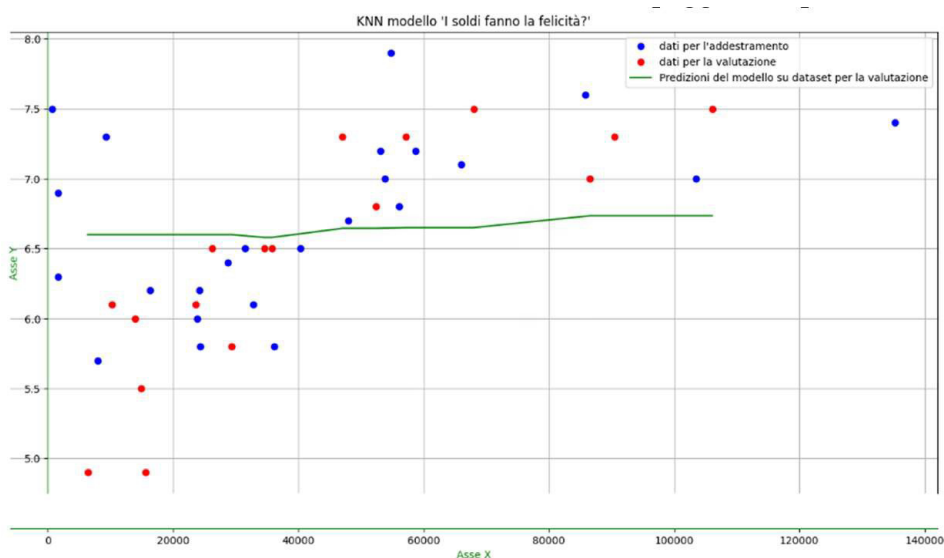


Figura 10: Modello “I soldi fanno la felicità?” con KNN ($K=20$)

In questo caso, infatti, l'approssimazione si avvicina al risultato ottenuto con la regressione lineare. A questo punto del percorso didattico viene bene introdurre i concetti di **overfitting** e **underfitting**, possiamo infatti osservare che per piccoli valori di K (es. $K=1$) l'algoritmo classifica un nuovo punto semplicemente assegnandogli la classe del suo singolo vicino più prossimo.

Questo rende il modello molto sensibile al rumore nei dati, la decisione di classificazione è basata su un'unica osservazione che potrebbe essere atipica, e siamo nel caso dell'overfitting.

Per valori grandi di K (es. K vicino alla dimensione del dataset) la decisione di classificazione per un nuovo punto sarà basata su un'ampia “media” delle classi dei vicini. Il confine decisionale diventa molto più liscio e generico e siamo nel caso dell'underfitting.

Un'applicazione del KNN molto famosa nella didattica è quella della “Classificazione degli iris”, tutti i libri di testo e numerosi articoli ne presentano una propria versione, l'obiettivo è quello di classificare gli iris, in 3 categorie: Versicolor, Setosa, Virginica.



Figura 11: un iris

Il modello è comunemente utilizzato per la didattica, tanto che i dati sono disponibili direttamente nella libreria Python sklearn: abbiamo una serie di informazioni etichettate: larghezza e lunghezza del sepal, larghezza e lunghezza del petalo per 150 fiori, per ciascuno dei quali abbiamo l'indicazione di quale tipo di iris si tratti: ci sono 50 Iris Versicolor, 50 Iris Setosa e 50 Iris Virginica.

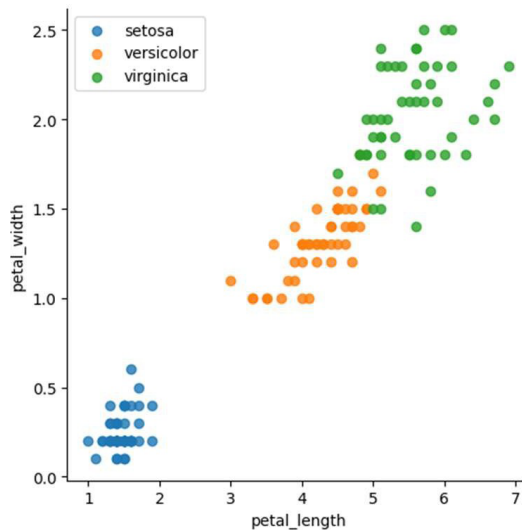


Figura 12: Dataset degli iris

Esaminiamo questo modello per fornire agli studenti l'esempio di un metodo per trovare il parametro K ottimale: utilizzando GridSearchCV infatti faremo variare K fra 1 e 31 e vedremo per quale valore il modello approssima meglio. Classificheremo i K secondo due metriche:

- **uniform:** a tutti i K punti più vicini all'esempio che stiamo cercando di classificare viene assegnato lo stesso peso o la stessa importanza
- **distance:** che indica che i vicini vengono pesati in base alla loro distanza dall'esempio da classificare. I vicini più vicini avranno un peso maggiore.

```
from sklearn.model_selection import GridSearchCV
k_range = range(1,31)
weight = ['uniform','distance']
param_grid = dict(n_neighbors=k_range, weights=weight)
neigh = neighbors.KNeighborsClassifier()
grid = GridSearchCV(neigh,param_grid,cv=10,scoring='accuracy')
grid.fit(X, y)
# visualizziamo il miglior punteggio e il parametro migliore
print(grid.best_score_)
print(grid.best_params_)

0.9800000000000001
{'n_neighbors': 13, 'weights': 'uniform'}
```

Figura 13: Codice per trovare il K migliore

I nostri risultati mostrano che il punteggio migliore è 0,98 quando k è impostato su 13 e pesa come uniform. Per confermare le competenze acquisite fino ad ora possiamo studiare applicazione del modello KNN al problema “Pubblicità mirata sui social, chi acquista davvero?”, per il quale utilizziamo il dataset che si trova a questo indirizzo: <https://www.kaggle.com/datasets/rakeshrau/social-network-ads>. Il dataset contiene alcuni dettagli degli utenti (400) di un social network (stipendio, età e genere), in particolare l’informazione se un utente ha acquistato un prodotto cliccando sull’annuncio che gli è comparso sulla pagina che stava leggendo.

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19	19000	0
1	15810944	Male	35	20000	0
2	15668575	Female	26	43000	0
3	15603246	Female	27	57000	0
4	15804002	Male	19	76000	0
5	15728773	Male	27	58000	0
6	15598044	Female	27	84000	0
7	15694829	Female	32	150000	1
8	15600575	Male	25	33000	0
9	15727311	Female	35	65000	0

Figura 14: Dataset Social Network Ads

Il problema è quello di prevedere se un dato utente acquisterà un prodotto online, conoscendone l’età, il genere e lo stipendio.

Sul corso allegato a questo articolo [1] è presente la proposta di un’attività per risolvere questo problema con il modello KNN, trovando il valore migliore di K tra 1 e 40.

Possiamo fare un’altra applicazione interessante del modello KNN sul dataset “Predizione del successo degli studenti”, che troviamo a questo indirizzo: <https://archive.ics.uci.edu/dataset/320/student+performance>

Studiare questo modello può essere interessante perché può fare da tramite per introdurre la categoria di modelli K-Means; nel corso Moodle (Barberis, 2025) possiamo trovare attività e notebook Colab che ci possono essere utili per introdurre altri tipi di ML: Naive Bayes, PCA, Deep Learning, in un percorso didattico completo.

Bibliografia

- [1] Barberis, G. (2025). Intelligenza artificiale per tutti. Tratto da [informaticabitabit](https://www.informaticabitabit.it/moodle/course/view.php?id=30): <https://www.informaticabitabit.it/moodle/course/view.php?id=30> O’Reilly. (2021). Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow. Aurelien Geron.

Dal concept al gioco: una testimonianza sull'insegnamento della programmazione attraverso il game making nei coding camp ibridi

Corrado Cristallo¹, Alessandro Pagano¹, Veronica Rossano¹

¹ Dipartimento di Informatica – Università degli Studi di Bari

c.cristallo6@studenti.uniba.it alessandro.pagano@uniba.it veronica.rossano@uniba.it

Abstract

Il presente contributo descrive un'esperienza didattica condotta nell'ambito del progetto europeo OSCAR, volta a promuovere l'apprendimento della programmazione tra studenti delle scuole superiori attraverso un coding camp ibrido dedicato allo sviluppo di serious games con il game engine GDevelop. L'iniziativa ha coinvolto 26 studenti tra i 15 e i 19 anni in un percorso breve ma intensivo, articolato in quattro giornate (due in presenza e due da remoto), con l'obiettivo di coniugare l'insegnamento di concetti fondamentali dell'informatica a una dimensione progettuale creativa e collaborativa. La metodologia adottata si ispira al costruzionismo, valorizzando il "fare" come strumento per un apprendimento profondo. I risultati della sperimentazione, valutati tramite osservazione, questionari e analisi dei prodotti finali, hanno evidenziato luci e ombre: da un lato, la validità dell'approccio e l'efficacia del supporto visivo offerto dai video tutorial; dall'altro, criticità legate alla durata limitata del percorso, alla composizione dei gruppi e alla collocazione temporale dell'iniziativa. Il paper si conclude con alcune raccomandazioni per future implementazioni e riflessioni sul potenziale del game making come leva didattica per l'insegnamento dell'informatica.

1. Introduzione

L'insegnamento della programmazione nella scuola secondaria di secondo grado rappresenta oggi una sfida didattica sempre più urgente. Nonostante il riconoscimento del valore formativo del pensiero computazionale [5], l'approccio prevalente nei curricula scolastici resta eccessivamente teorico, con scarsi collegamenti a contesti motivanti o a esperienze concrete. Gli studenti faticano a percepire la rilevanza e l'applicabilità della programmazione nella vita reale, e ciò può tradursi in una scarsa motivazione e in risultati formativi modesti.

Nel tentativo di superare questo scollamento, il progetto europeo OSCAR ha offerto l'opportunità di sperimentare un nuovo modello didattico attraverso un coding camp ibrido, rivolto a studenti tra i 15 e i 19 anni. Il focus è stato posto sullo sviluppo di serious games, ovvero videogiochi con finalità educative, utilizzando il game engine GDevelop. L'esperienza, qui narrata in prima persona, intende fornire un contributo metodologico alla riflessione sull'insegnamento dell'informatica come disciplina scientifica, fondata non solo su contenuti tecnici, ma anche su creatività, progettualità e collaborazione.

2. Premesse didattiche e scelte metodologiche

Il cuore pedagogico dell'esperienza è stato il principio del costruzionismo [3], secondo cui gli studenti apprendono meglio costruendo artefatti significativi in contesti sociali e motivanti. In quest'ottica, progettare e sviluppare un videogioco non è un semplice esercizio tecnico, ma un processo complesso che coinvolge competenze trasversali: dalla narrazione alla logica, dalla gestione del progetto al lavoro di gruppo. Il videogioco diventa così il pretesto per un apprendimento integrato, profondo e personale. La scelta del game engine è ricaduta su GDevelop, una piattaforma open source gratuita, con un'interfaccia visuale intuitiva che consente di realizzare giochi 2D senza scrivere codice. Questa caratteristica lo rende particolarmente adatto a contesti educativi con studenti alle prime armi. A differenza di altri engine analizzati (Godot, GameMaker), GDevelop (Fig.1) si distingue per la semplicità d'uso, l'accessibilità via browser e una libreria di asset integrata. I criteri considerati nella selezione includevano la gratuità, la facilità d'uso, il supporto alla collaborazione e la disponibilità di funzionalità didatticamente rilevanti [2]. Per accompagnare l'apprendimento tecnico, sono stati realizzati sei video tutorial, ciascuno incentrato su un costrutto fondamentale della programmazione: variabili, oggetti, condizioni, cicli, funzioni e interfacce utente. I video seguivano una struttura coerente: introduzione del concetto teorico, collegamento con GDevelop e dimostrazione operativa. La realizzazione è avvenuta tramite OBS Studio e DaVinci Resolve per il montaggio video e Audacity per la pulizia del sonoro. Questo materiale è stato reso disponibile ai partecipanti durante il camp per supportare il lavoro in asincrono.

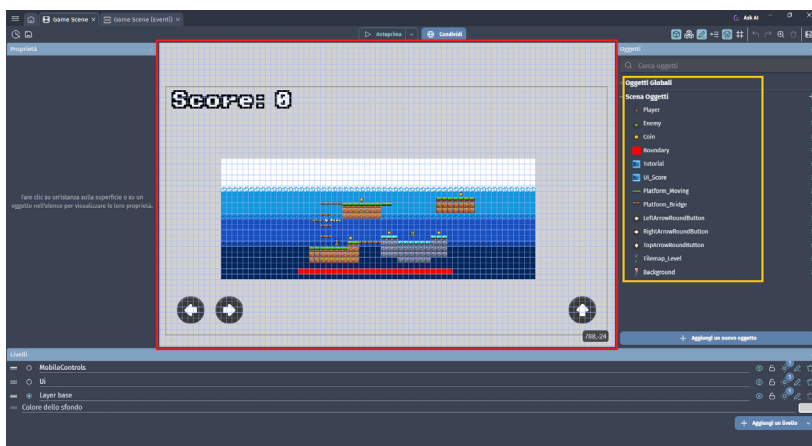


Figura 1: Schermata dell'editor di scena (in rosso) di GDevelop con libreria degli oggetti (in giallo)

3. La struttura del coding camp

Il coding camp si è svolto in quattro giornate, per un totale di dodici ore, distribuite su due settimane. La prima e l'ultima giornata si sono tenute in presenza, presso il Dipartimento di Informatica dell'Università degli Studi di Bari, mentre le due centrali si sono svolte in modalità remota. Tale struttura ha consentito di sfruttare i punti di forza di entrambe le modalità: l'interazione sociale e la coesione iniziale della presenza, la flessibilità e la continuità del remoto [1].

Durante la prima giornata, gli studenti hanno ricevuto una panoramica sul game design, sviluppando in gruppo il concept del proprio serious game. Le attività hanno incluso la definizione dell'obiettivo educativo, la costruzione dello storyboard e l'individuazione delle meccaniche principali. I gruppi, formati arbitrariamente in base alla distribuzione delle classi partecipanti, erano composti da tre o quattro membri. Questa eterogeneità forzata si è poi rivelata un elemento critico per la dinamica collaborativa.

Le giornate da remoto hanno rappresentato il cuore operativo dell'esperienza. Inizialmente previsto su Microsoft Teams, il lavoro è stato ostacolato da limiti tecnici legati agli account scolastici. Gli studenti hanno quindi adottato spontaneamente piattaforme alternative come Google Meet e Discord, dimostrando resilienza e capacità di problem solving. Durante questi incontri, i partecipanti hanno seguito collettivamente i video tutorial e lavorato allo sviluppo del prototipo, assistiti da tutor che ruotavano tra le "stanze virtuali".

L'intervallo di cinque giorni tra la terza e la quarta giornata ha permesso agli studenti di lavorare in autonomia, rivedere i video e consolidare le competenze. La giornata finale, nuovamente in presenza, è stata dedicata alla presentazione dei progetti, alla rifinitura dei dettagli tecnici e alla discussione collettiva delle scelte progettuali.

4. Sfide incontrate

Nonostante la cura progettuale, l'esperienza ha evidenziato alcune criticità rilevanti. Sul piano tecnico, l'impossibilità di utilizzare agevolmente Microsoft Teams ha reso necessaria un'improvvisazione che ha generato una certa frammentazione dell'esperienza. L'accesso ai tutorial non è stato uniforme e alcuni studenti hanno riferito difficoltà nel seguire in modo lineare le attività proposte. La composizione dei gruppi, pur pensata per favorire la diversità interna, si è scontrata con un'omogeneità effettiva (tutti gli studenti provenivano dallo stesso istituto, salvo un caso) e con una certa disaffezione iniziale. Alcuni partecipanti hanno mostrato scarso coinvolgimento, probabilmente legato al fatto che la partecipazione era stata imposta all'ultimo momento. Inoltre, la collocazione del camp nel mese di maggio ha inciso negativamente sulla motivazione: il carico scolastico elevato e l'imminenza delle vacanze estive hanno limitato l'energia disponibile da parte degli studenti.

Infine, il tempo a disposizione si è rivelato insufficiente per consentire un reale approfondimento delle competenze tecniche e una piena realizzazione dei progetti. La durata complessiva di 12 ore, seppure distribuita su più giornate, ha imposto tempi serrati, non sempre compatibili con le esigenze di apprendimento degli studenti meno esperti.

5. Risultati e osservazioni

La valutazione dell'esperienza si è articolata su tre livelli: osservazione dei comportamenti durante le sessioni, somministrazione di un questionario post-attività, e analisi degli elaborati finali attraverso un framework strutturato. Quest'ultimo ha permesso di valutare criteri quali la coerenza educativa del gioco, la definizione del pubblico target, la presenza di narrazione, la progettazione dei livelli e l'equilibrio tra interfaccia utente e logica computazionale [6].

I risultati hanno restituito un quadro sfaccettato. Su otto team, solo tre hanno presentato un prototipo giocabile corredato da una presentazione strutturata. Gli altri gruppi si sono limitati a presentazioni teoriche o concept incompleti. In termini quantitativi, il livello di soddisfazione si è attestato su valori medi (Fig. 2), mentre la sicurezza percepita nell'uso degli strumenti è risultata variabile (Fig. 3).

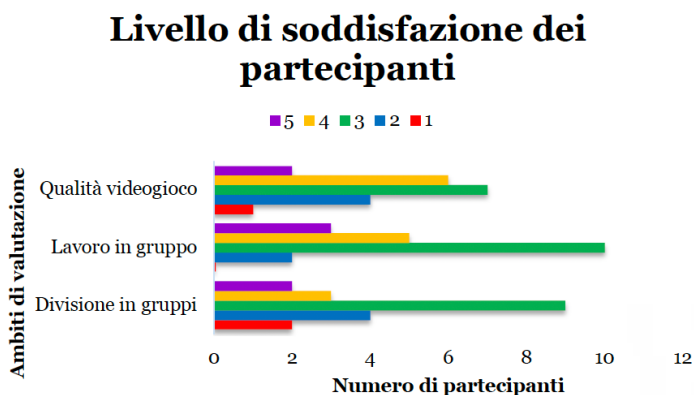


Figura 2: Grafico del livello di soddisfazione dei partecipanti estratto dal questionario

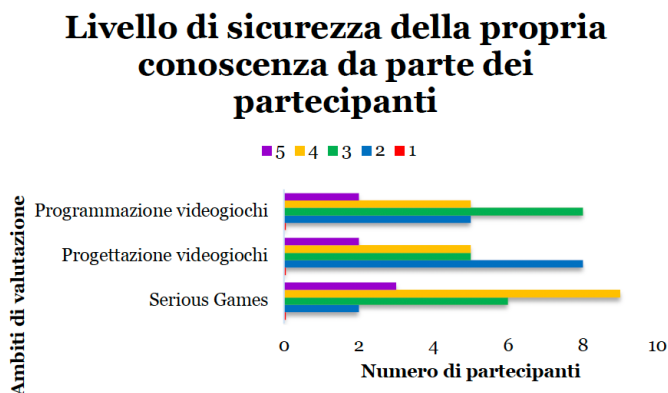


Figura 3: Grafico del livello di sicurezza della propria conoscenza da parte dei partecipanti estratto dal questionario

Tra i progetti emersi, spicca quello sviluppato dal gruppo contenente l'unico studente esterno al contesto scolastico principale, a conferma di quanto l'eterogeneità possa stimolare dinamiche più produttive. Le osservazioni dei tutor hanno evidenziato che l'engagement aumentava nei momenti laboratoriali e calava durante le spiegazioni teoriche, confermando l'efficacia di un approccio hands-on.

6. Riflessioni e prospettive future

L'esperienza del coding camp OSCAR suggerisce alcune direzioni di miglioramento. In primo luogo, la durata: 12 ore non sono sufficienti per affrontare seriamente un ciclo completo di progettazione, sviluppo e riflessione. Una durata minima di 20–25 ore, distribuite su un periodo più lungo, permetterebbe una maggiore maturazione delle idee e una migliore assimilazione dei contenuti tecnici. In secondo luogo, la selezione dei partecipanti: per ottenere un reale valore educativo e creativo, è importante coinvolgere studenti motivati e con background diversi. Ciò richiede un processo di candidatura volontaria e una comunicazione anticipata con le scuole [4].

Infine, l'organizzazione temporale: evitare i periodi scolastici congestionati, prediligere contesti extracurricolari o estivi, e prevedere momenti di follow-up per consolidare i risultati ottenuti. Nonostante gli ostacoli, l'utilizzo di GDevelop ha confermato la validità di strumenti visuali e accessibili come ponte tra astrazione teorica e realizzazione concreta. L'ibrido, se ben progettato, può rappresentare una modalità efficace per promuovere competenze tecniche e soft skill, come il lavoro di squadra e la resilienza [1].

7. Conclusione

Insegnare informatica attraverso il game making non è solo una strategia didattica innovativa, ma una vera e propria visione pedagogica: quella di un apprendimento centrato sul fare, sul costruire insieme, sul dare forma concreta a idee astratte. L'esperienza del coding camp OSCAR ha dimostrato come, anche in presenza di vincoli logistici e limiti strutturali, sia possibile creare spazi significativi di apprendimento autentico. Il gioco, quando ben progettato, non distrae: coinvolge, orienta, educa. E restituisce alla programmazione quel fascino originario di strumento creativo e trasformativo.

Ringraziamenti

Questo lavoro è stato finanziato dal progetto OSCAR, numero 101132432 ERASMUS-EDU- 2023-PI-FORWARD, finanziato dall'Unione Europea. Le opinioni espresse sono esclusivamente quelle dell'autore (o degli autori) e non riflettono necessariamente quelle dell'Unione Europea. Né l'Unione Europea né l'autorità concedente possono esserne ritenute responsabili.

Bibliografia

- [1] Fronza, I., Corral, L., & Iaccarino, G. (2024). Nurturing hybrid work literacy in upper secondary schools: Selecting the best hybrid work configuration for coding camps. In *Proceedings of the 2024 IEEE Frontiers in Education Conference (FIE)* (pp. 1–9). IEEE. <https://doi.org/10.1109/FIE61694.2024.10893177>
- [2] Kafai, Y. B., & Burke, Q. (2015). Constructionist gaming: Understanding the benefits of making games for learning. *Educational Psychologist*, 50(4), 313–334. <https://doi.org/10.1080/00461520.2015.1124022>
- [3] Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books.
- Porras, J., Happonen, A., Knutas, A., Khakurel, J., Ikonen, J., & Herala, A. (2019). Code camps and hackathons in education: Literature review and lessons learned. In *Proceedings of the 52nd Annual Hawaii International Conference on System Sciences (HICSS)* (pp. 7750–7759). IEEE Computer Society. <https://doi.org/10.24251/hicss.2019.933>
- [4] Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35. <https://doi.org/10.1145/1118178.1118215>
- [5] Katsantonis, M. N., Kotini, I., Fouliras, P., & Mavridis, I. (2019). Conceptual framework for developing cyber security serious games. In *Proceedings of the 2019 IEEE Global Engineering Education Conference (EDUCON)* (pp. 872–881). IEEE. <https://doi.org/10.1109/EDUCON.2019.8725061>

Servizi di Rete

Daniela Decembrino¹

¹ ITT Panetti Pitagora - Bari

daniela.decembrino@gmail.com

Abstract

Sfruttando il podcasting, vengono presentati i Servizi di Rete dedicando a ciascuno di essi un episodio del podcast intitolato mnemonicamente “Servizi di Rete” e rinvenibile sulla piattaforma Spotify all’ indirizzo <https://open.spotify.com/show/7bO67diw8BGnzI7btHas1A> BITCOIN, Blog, BYOD, chat, DNS, e-commerce, e-learning, e-government & e-democracy, e-health, e-mail, emoticon, Cloud Computing, FAQ, File Sharing, firma digitale, Forum, Home Banking, hosting & housing, IA a servizio dei motori di ricerca, Mailing List, motori di ricerca, multiplayer sono i servizi di Rete indagati durante le prime due edizioni del podcast.

1. Introduzione

I “Servizi di Rete” sono studiati in concomitanza con le architetture di rete nell’ ambito della disciplina di insegnamento “Sistemi e reti”. Queste conoscenze sono proprie di quanto proposto durante il quinto anno di Istituto Tecnico Tecnologico indirizzo Informatica e telecomunicazioni articolazione Informatica. L’ idea progettuale innovativa consiste nel coinvolgere l’intero gruppo classe nella registrazione di un episodio del podcast intitolato appunto “Servizi di Rete” rinvenibile gratuitamente sulla piattaforma Spotify al link <https://open.spotify.com/show/7bO67diw8BGnzI7btHas1A> . Seguendo un ordine sostanzialmente alfabetico vengono indagati uno per uno i servizi di Rete e per ciascuno di essi un singolo studente viene impegnato nella registrazione vocale dell’episodio monotematico assegnato. Ogni servizio viene idealmente descritto attraverso una “icona” che ne “ricorda” il significato. Ricorrendo anche all’ Intelligenza Artificiale, è stato creato il logo specifico dell’episodio dedicato ai Motori di Ricerca che, attualmente, sono coadiuvati appunto dalla Intelligenza Artificiale stessa.

2. Podcasting: come il podcast può intervenire in soccorso alla didattica curricolare

I podcast, in ambito formativo, costituiscono un valido supporto alla diffusione di contenuti e si rendono adatti anche nel garantire l’inclusività proprio per la loro natura audio— in qualsiasi momento, con qualsiasi dispositivo, in qualsiasi luogo un podcast può essere ascoltato e riascoltato dall’ utente e ancor più da un discente. Punto di forza del progetto didattico è l’aver permesso che discenti potessero veicolare contenuti pensati per se stessi e per altri discenti

nell'ottica della peer education (educazione tra pari). Aver reso pubblici i contenuti proposti, autoprodotti, fa sì che anche l'utente in qualche modo interessato possa essere ascoltatore e fruitore del lavoro prodotto. Quindi, oltre che in ambito squisitamente didattico, il campo d'azione si estende alla crescita personale, professionale e aziendale, collocando il prodotto podcast in quello che viene definito il *longlife learning* cioè apprendimento durante tutto l'arco della vita. Ulteriore punto di forza del progetto in essere rappresenta l'aver coinvolto l'intero gruppo classe assegnando a ciascuno studente, al massimo in un paio di casi due studenti, un determinato servizio di Rete di cui approfondirne storia, significato, usi e prospettive future. Nel contempo è stato individuato un "podcaster" responsabile della pubblicazione on line dei contenuti audio e testuali descrittivi e un logo artist responsabile della creazione dei loghi specifici. La didattica tradizionale cede il passo alla didattica progettuale e all'uso di nuove frontiere della comunicazione cioè il podcast per l'appunto. Stimoli creativi, lavoro di squadra, ricerca-azione diventano volano per giovani che si affacciano alla new technology come protagonisti attivi piuttosto che come spettatori passivi.

3. L'Intelligenza Artificiale

L'Intelligenza Artificiale, croce e delizia dei tempi attuali, fa la sua apparizione anche tra i servizi di Rete. Sono i motori di ricerca, servizio di Rete molto utilizzato dagli internauti, che stanno via via lasciando sempre più spazio all'Intelligenza Artificiale.

La domanda sorge spontanea: Ma saranno i motori di ricerca soppiantati totalmente dalle Intelligenze Artificiali?

Ai posteri l'ardua sentenza.

Al momento appare evidente che molti motori di ricerca offrono la possibilità ai propri utenti di poter ottenere sostanzialmente gratuitamente le risposte cercate fornite anche da una Intelligenza Artificiale. Su molti smartphone Android viene proposta l'Intelligenza Artificiale AI Overview, così come Google search si accompagna a Gemini e via scorrendo.

Come ignorare una simile situazione? Come eludere questi cambiamenti?

Solo fino a pochi anni addietro sarebbe stato possibile affrontare lo studio dei motori di ricerca a prescindere dalle Intelligenze Artificiali, ma, allo stato delle cose, sarebbe assolutamente anacronistico proporre lo studio di un servizio di Rete quale quello dei motori di ricerca prescindendo dall'impiego che gli stessi motori di ricerca fanno delle Intelligenze Artificiali.

Così la didattica si evolve e segue il passo dei tempi: da un canto usa uno strumento comunicativo nuovo quale il podcast per condividere contenuti e veicolare gli apprendimenti, dall'altro persegue l'evoluzione delle stesse nuove tecnologie che al tempo stesso si fanno contenuto e strumento d'uso.

Nello specifico l'episodio di riferimento è quello udibile al seguente link <https://open.spotify.com/show/7bO67diw8BGnzI7btHas1A> e il logo ad esso relativo è creato facendo ricorso a una Intelligenza Artificiale.



Figura 1 : L'IA al servizio dei motori di ricerca

3.1 I vantaggi di una didattica innovativa

L'uso di metodologie didattiche innovative soprattutto rispetto ai contesti attuali rappresenta garanzia di successo formativo in considerazione del fatto che l'era attuale è quella in cui il messaggio è più veloce del messaggero. Pensare di cristallizzare la didattica soprattutto delle discipline che ruotano intorno all'informatica significherebbe negare l'evidenza. Evolvono gli strumenti e evolvono i contenuti. Gli uni sono intrecciati con gli altri. Avviluppati gli uni agli altri. Il primo nemico della didattica è il suo immobilismo e questo è maggiormente avvertito se l'oggetto è di natura informatica.

3.2 Le scatole cinesi

Tra i servizi di Rete ci è il podcast. Usando la metodologia del podcasting per veicolare lo studio dei servizi di Rete significa anche usare la parte per il tutto quindi un podcast per spiegare il podcast. Ci sta!

4. Parole e musica

Ogni episodio del podcast si compone di varie parti:

- jingle introduttivo;
- introduzione all'episodio e al servizio di Rete oggetto dell'episodio stesso;
- jingle intermedio;
- esposizione dell'episodio quindi trattazione sintetica del servizio di Rete considerato;
- jingle conclusivo.

Tutti gli intermezzi musicali sono tratti da siti che rendono disponibili gratuitamente i contributi musicali proposti. La voce narrante costituisce il “filo rosso” di tutto il podcast e introduce ogni episodio. I tempi sono scanditi in funzione della corposità dei contenuti proposti. La narrazione del singolo contenuto è affidata allo studente che ha approfondito quello specifico servizio di Rete.

4.1 La matrice delle responsabilità

Come in ogni progetto che si rispetti, anche per la realizzazione di un podcast è necessario esplicitare chi fa cosa, ragion per cui viene redatta una matrice delle responsabilità in cui sono indicati i nomi dei soggetti coinvolti nell’azione progettuale e il compito loro affidato

La matrice delle responsabilità viene idealmente proposta mediante la tabella sottostante.

Nome / servizio	servizio-1	servizio-2	servizio-3	...	servizio-n
nome1					X
nome2		X			
nome3			X		
...					
nomen	X				

Tabella 1: Matrice delle responsabilità

Conclusioni

Aver utilizzato una metodologia didattica innovativa attiva, quale quella del podcasting, ha sicuramente incuriosito gli studenti che, affascinati dalle fasi della progettazione, hanno messo in campo processi di apprendimento dinamici e cooperativi aprendosi alla divulgazione ad extra oltre che ad intra.

Bibliografia

[1] Andrew, Tanenbaum. Reti di calcolatori, Pearson, 2023 . ISBN 8891 91 53 1 9.

[2] Cristianini, Nello. ‘ La scorciatoia. Come le macchine sono diventate intelligenti senza pensare in modo umano’ il Mulino, 2023 . ISBN 978 -8 8 -1 5 -29983 -3 .

[3] Cristianini, Nello. ‘ machina sapiens. L’ algoritmo che ci ha rubato il Segreto della conoscenza’ il Mulino, 2024. ISBN 978 -8 8 -1 5 -3 8446-1 .

[4] Giordani, Alessandro, and Mari, Luca. “L’ Intelligenza Artificiale di Platone. Il pensiero, i chatbot e noi. ‘ il sole 24 ore, 2025 . ISBN 979-1 2-5484-4397

[5] Gatto, Romeo, and Mari. “iPod e iTunes. Musica, video, podcast e foto in tasca’. APOGEO, 2006. ISBN 88 -503 -221 3 -5

POSTER

PER OGNI ORDINE E GRADO

EduPKG: un’Ontologia per la didattica STEM

Paolo Campanelli¹, Alessandro Marcelletti¹, Barbara Re¹

¹Università degli Studi di Camerino, Sezione di Informatica, Camerino, Italia

{paolo.campanelli, alessandro.marcelletti, barbara.re}@unicam.it

Abstract

L’educazione STEM svolge un ruolo fondamentale per formare cittadini tecnologicamente competenti e preparare le nuove generazioni alle professioni del futuro. Le discipline scientifiche e, in particolare, l’informatica trovano applicazioni concrete nella vita quotidiana e nei contesti professionali, contribuendo allo sviluppo di pensiero critico, creatività e capacità di problem-solving. Strumenti come coding e robotica educativa favoriscono il pensiero computazionale e la collaborazione, ma la loro integrazione nei curricula risulta spesso frammentaria e priva di una strategia condivisa. Per questo motivo proponiamo EduPKG, un’ontologia educativa per supportare la progettazione di attività STEM. EduPKG consente di allineare attività didattiche con bisogni formativi, risorse e vincoli temporali, favorendo la selezione di esperienze significative e realizzabili. L’approccio valorizza competenze chiave come la data literacy e le “4C” (Creatività, Collaborazione, Comunicazione e Pensiero Critico). L’ontologia è stata sviluppata seguendo il metodo Methontology e validata attraverso un’indagine con esperti dimostrando la sua efficacia nel supportare la progettazione di attività STEM.

1. Introduzione

Nell’era digitale, l’educazione STEM svolge un ruolo fondamentale nel formare cittadini tecnologicamente competenti e preparare gli studenti alle professioni del futuro. Scienza, Tecnologia, Ingegneria, Matematica e, in particolare, Informatica, sono sempre più presenti nella vita quotidiana, dai dispositivi intelligenti ai contesti professionali. Sviluppare il pensiero critico e le competenze pratiche sin dalla giovane età permette a studenti e studentesse di affrontare con successo un mercato del lavoro orientato all’innovazione [11]. Un approccio efficace per promuovere l’interesse verso le STEM è l’uso del coding e della robotica educativa, che rappresentano applicazioni dirette dell’informatica e coinvolgono attivamente gli studenti nel pensiero computazionale, nella creatività e nella risoluzione pratica dei problemi [16]. L’integrazione di questi strumenti nei primi cicli scolastici favorisce la familiarità con il digitale, stimola la collaborazione e sviluppa un pensiero flessibile, essenziale per affrontare le sfide tecnologiche future. Un punto di forza dell’educazione STEM è la sua natura interdisciplinare, che consente di collegare diverse materie in un’ottica applicativa [2]. Le attività STEM stimolano inoltre la creatività, il pensiero logico e la collaborazione tra pari, potenziando competenze chiave per le discipline scientifiche e tecnologiche [10, 12, 17]. Nonostante i benefici, l’educazione STEM è ancora in fase di sviluppo, soprattutto nella scuola primaria e secondaria

di primo grado [3]. Negli ultimi vent'anni, in Europa sono stati fatti importanti progressi nell'integrazione delle STEM nei curricula scolastici, grazie a politiche e iniziative dedicate [5, 9]. Tuttavia, permangono criticità significative [8]. In particolare, manca una strategia condivisa per la selezione e l'attuazione delle attività STEM. Gli insegnanti spesso faticano a scegliere metodologie, risorse e obiettivi adeguati [7]. In ambito informatico, questa difficoltà si traduce talvolta in un uso frammentario e non sistematico degli strumenti digitali, con progetti avviati senza una reale comprensione dei bisogni pedagogici e tecnologici degli studenti, riducendone l'efficacia [4].

Per rispondere a queste sfide, la Sezione di Informatica dell'Università degli Studi di Camerino ha lavorato ad EduPKG, un'ontologia educativa pensata per supportare la progettazione delle attività STEM, allineandole ai bisogni degli studenti, alle risorse disponibili e ai vincoli di tempo. Il lavoro estende la metodologia presentata in [1], concentrandosi nel supporto alla creazione di attività formative. EduPKG facilita infatti la selezione di attività coinvolgenti e realizzabili, promuove competenze chiave come il problem-solving e la data literacy, e valorizza le 4C: Creatività, Collaborazione, Comunicazione e Pensiero Critico. L'ontologia è stata sviluppata seguendo il metodo Methontology [6] e validata attraverso un'indagine con esperti dimostrando la sua efficacia nel supportare la progettazione di attività STEM.

2. Ontologia

In questa sezione descriviamo l'ontologia EduPKG, illustrandone la struttura in termini di componenti, classi, proprietà e relazioni. L'ontologia EduPKG è progettata per fornire un modello strutturato utile all'organizzazione di risorse educative, attività didattiche, metodi di insegnamento e competenze. Questa ontologia consente una rappresentazione chiara e riutilizzabile della conoscenza, facilitando l'adattamento e la personalizzazione delle esperienze di apprendimento nei diversi livelli scolastici. L'ontologia è stata definita attraverso un processo iterativo basato sull'analisi dei bisogni educativi e delle tecnologie disponibili, seguendo l'approccio Methontology. L'ontologia EduPKG segue una struttura gerarchica, raggruppando gli elementi in diverse categorie. Ogni classe descritta nella Tabella 1 in EduPKG rappresenta un concetto chiave nel processo di insegnamento e apprendimento, collegato tramite relazioni semantiche, in modo da garantire che i contenuti educativi siano logicamente organizzati.

Classe	Descrizione	Sottoclassi
ActivityType	Classifica le attività di apprendimento in base agli approcci pedagogici.	CollaborativeActivity, CreativeActivity, ExplorationActivity, HandsOnActivity, ProblemSolvingActivity
Competency	Definisce le competenze sviluppate tramite le attività STEM, in linea con DigiComp.	Collaboration, Communication, Creativity, CriticalThinking, DigitalContentCreation, InformationAndDataLiteracy, ProblemSolving, Safety

LearningLevel	Classifica le attività in base al livello scolastico degli studenti.	<i>Nessuna</i>
Resource	Definisce i materiali didattici a supporto delle attività.	AdvancedResource, DigitalResource, EntryLevelResource, PhysicalResource
Subject	Classifica i contenuti educativi per disciplina, favorendo connessioni interdisciplinari.	Humanities, STEM
TargetType	Definisce il pubblico destinatario di un pacchetto educativo.	FullClass, JointClass, KindergartenTeacher, MixedAbilityGroup, PrimaryTeacher, SecondaryTeacher, SmallGroup
TeachingMethod	Definisce le strategie pedagogiche ispirate alle teorie costruttiviste.	CollaborativeMethod, ExperientialMethod, GamifiedMethod, PassiveMethod
TimeAllocation	Definisce la durata prevista delle attività per la pianificazione.	<i>Nessuna</i>

Tabella 1: Classi dell'ontologia EduPKG con relative sottoclassi

L'ontologia EduPKG si basa su una rete strutturata di relazioni semantiche descritte nella Tabella 2 che collegano le diverse classi. Ogni proprietà è definita secondo OWL (Web Ontology Language), assicurando coerenza e leggibilità automatica.

Proprietà	Dominio	Codominio
adoptsTeachingMethod	ActivityType	TeachingMethod
isDesignedFor	ActivityType	TargetType
isPartOfSubject	ActivityType	Subject
isSuitableForLearningLevel	ActivityType	LearningLevel
requiresTimeAllocation	ActivityType	TimeAllocation
targetsCompetency	ActivityType	Competency
usesResource	ActivityType	Resource

Tabella 2: Proprietà oggetto delle attività di apprendimento

EduPKG include inoltre un insieme di data properties descritte nella tabella 2, che forniscono attributi descrittivi alle varie entità. A differenza delle proprietà oggetto, le data properties collegano un'entità a valori letterali.

Proprietà	Dominio	Codominio	Descrizione
hasDescription	owl:Thing	xsd:string	Fornisce una descrizione testuale di qualsiasi elemento dell'ontologia.
hasDifficultyLevel	Resource	xsd:string (Base, Advanced)	Classifica le risorse in base al livello di difficoltà.
hasMaterialType	Resource	xsd:string (Digital, Physical)	Distingue le risorse tra digitali e fisiche.
hasDuration	Time/Allocation	xsd:decimal	Specifica la durata dell'attività (in minuti o ore).

Tabella 3: Proprietà dei dati dell'ontologia EduPKG

3. Implementazione

Questa sezione presenta i dettagli relativi all'implementazione dell'ontologia EduPKG. L'ontologia è stata sviluppata utilizzando Protégé [15], un editor open-source di ontologie ampiamente utilizzato, e validata tramite strumenti di inferenza. Per dimostrarne una possibile applicazione, viene fornito un esempio di utilizzo tramite una query SPARQL. L'ontologia implementata è accessibile online¹.

Analisi formale tramite strumenti di inferenza. L'ontologia EduPKG è stata analizzata utilizzando i reasoner *Pellet* [14] e *HermiT* [13], al fine di garantire la coerenza logica, la corretta classificazione e l'adeguata definizione dei vincoli tra proprietà oggetto. Il processo di inferenza ha confermato l'integrità della gerarchia di classi, eliminato ridondanze semantiche e garantito un corretto collegamento tra attività di apprendimento, risorse, metodi didattici e competenze. Non sono state rilevate incongruenze o errori, confermando che EduPKG è strutturalmente solida e semanticamente coerente per l'integrazione in sistemi educativi intelligenti.

Esempio di utilizzo tramite query SPARQL. Di seguito è presentato un esempio di utilizzo dell'ontologia EduPKG, come visibile nella Lista 1, attraverso l'uso delle query SPARQL. SPARQL è un linguaggio di interrogazione per la manipolazione e il recupero di dati in formato RDF (Resource Description Framework). In Protégé, SPARQL può essere utilizzato per testare la struttura e la funzionalità dell'ontologia, consentendo un'efficiente estrazione e analisi dei dati educativi in base a relazioni e vincoli definiti. Uno degli obiettivi fondamentali dell'ontologia EduPKG è permettere il recupero di attività educative specifiche per disciplina, affinché i docenti possano accedere rapidamente a piani didattici strutturati e adatti ai diversi contesti scolastici. Per dimostrare questa funzionalità, è stata eseguita una query SPARQL per recuperare tutte le attività associate alla disciplina *Tecnologia* (riga 5) e destinate al livello scolastico *Scuola Primaria* (riga 6). La query include anche la descrizione (riga 7), il tempo necessario (riga

¹ <https://github.com/PaoloCampanelli/EduPKG-Data>

8) e le risorse associate (riga 9), in modo da restituire tutti i componenti essenziali dell’attività formativa.

Listing 1: Query per attività di apprendimento nella disciplina Tecnologia alla scuola primaria

```

SELECT ? activity ( STR (?description ) AS ?cleanDescription
)
  ?timeAllocation ?resource
WHERE {
  ?activity a :ActivityType ;
  :isPartOfSubject :Technology ;
  :isSuitableForLearningLevel :PrimarySchool ;
  :hasDescription ?description .
  OPTIONAL { ?activity :requiresTimeAllocation ?timeAllocation
}
  OPTIONAL { ?activity :usesResource ?resource }
}

```

L’esecuzione della query ha restituito con successo tutte le attività classificate nel dominio Tecnologie per la scuola primaria, confermando che le classificazioni per materia e livello scolastico sono state correttamente applicate. I risultati hanno mostrato che tutte le attività recuperate erano correttamente associate alla disciplina Tecnologia e ad un livello di apprendimento definito, assicurando contenuti adeguati all’età. Inoltre, i vincoli opzionali hanno permesso il recupero, ove disponibile, delle informazioni relative alla durata e alle risorse utilizzate.

4. Validazione

Per garantire la reale applicabilità di EduPKG, è stato condotto un processo di validazione che ha coinvolto esperti del settore educativo². Insegnanti e docenti universitari hanno esaminato l’ontologia, offrendo preziosi feedback per migliorarne l’allineamento con le attività in classe e la pianificazione di progetti STEM.

Per valutare la rilevanza pratica e l’usabilità di EduPKG, è stato somministrato un questionario a docenti con esperienza diretta nell’insegnamento STEM e nell’uso di tecnologie educative. L’indagine mirava a esplorare le difficoltà incontrate nella pianificazione delle attività STEM, le aspettative rispetto a EduPKG e i possibili ostacoli nell’adozione di uno strumento di questo tipo. I partecipanti includevano insegnanti della scuola dell’infanzia, primaria e secondaria, con un’esperienza didattica compresa tra i 5 e oltre i 30 anni. Tutti avevano già implementato attività STEM utilizzando tecnologie educative come robotica, piattaforme di coding e attività unplugged. Un riepilogo dei risultati del questionario è riportato di seguito, mentre i dati completi sono disponibili online². Prima della compilazione del questionario, è stato chiesto di individuare le principali difficoltà nella pianificazione delle attività STEM. Sono emersi tre ostacoli principali: vincoli di tempo (50%), accesso limitato alle tecnologie educative (71,4%) e difficoltà di adattamento delle attività a classi eterogenee (35,7%). Questi dati confermano l’esigenza di un sistema strutturato che supporti i docenti nell’identificazione e personalizzazione delle attività STEM. Uno degli aspetti principali analizzati è stato l’interesse dei docenti verso un sistema come EduPKG. I risultati, illustrati nella Figura 1, confermano

² <https://github.com/PaoloCampanelli/EduPKG-Data>

che un sistema basato su ontologie potrebbe fornire un valido supporto alla strutturazione e implementazione di lezioni STEM. Un altro aspetto rilevante è stato l’analisi delle caratteristiche attese dai docenti e delle difficoltà previste nell’implementazione del sistema. Uno degli obiettivi del questionario era capire se EduPKG venisse percepito come uno strumento utile nella pratica didattica. Tutti i partecipanti hanno riconosciuto il potenziale valore di un sistema basato su ontologie per la pianificazione dell’educazione STEM, anche se le motivazioni variavano in base al contesto scolastico e alla familiarità con le tecnologie. Le principali prospettive sono riportate nella Tabella 4.

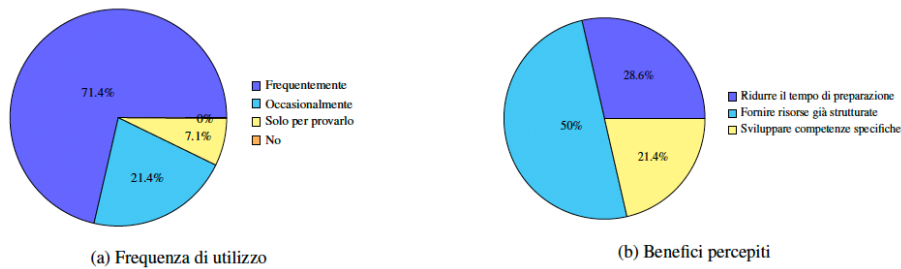


Figura 1: Aspettative nell'utilizzo di EduPKG e supporto alla pianificazione

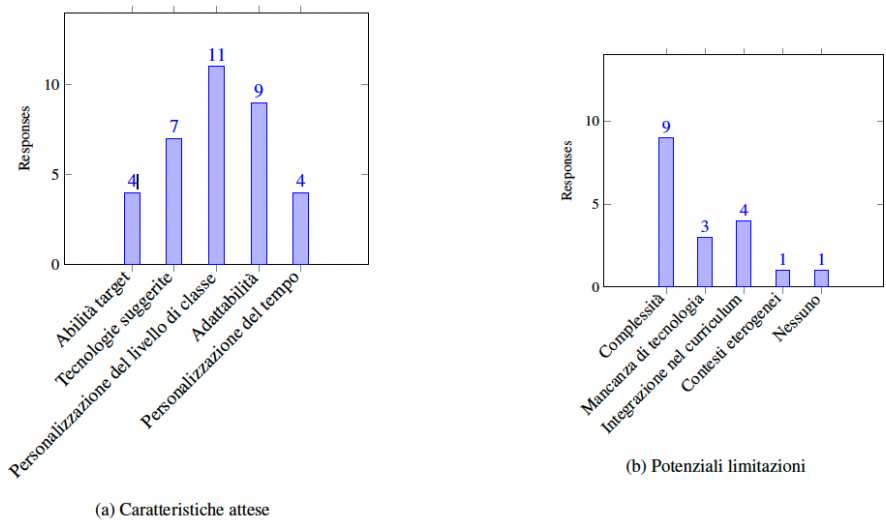


Figura 2: Caratteristiche attese e potenziali limitazioni

Prospettiva	Sintesi
Integrazione in aula	Deve integrarsi facilmente con le metodologie già in uso.
Accessibilità STEM	Supporto per docenti senza formazione STEM specialistica.
Applicabilità alla prima infanzia	Utile anche in contesti educativi prescolari.
Adattabilità ai contesti	Deve adattarsi a diversi ambienti e bisogni scolastici.
Personalizzazione dell’insegnamento	Supporta percorsi su misura per ogni studente.
Apprendimento pratico	Favorisce esperienze di apprendimento applicate.
Coinvolgimento e motivazione	Potenzia l’engagement, ma richiede valutazioni critiche.
Rafforzamento degli apprendimenti	Consolida le competenze STEM in modo strutturato.
Supporto didattico strutturato	Riduce i tempi di pianificazione con risorse pronte all’uso.
Sviluppo di competenze	Promuove abilità non affrontate dalla didattica tradizionale.

Tabella 4: Prospettive principali sull’integrazione di EduPKG

In sintesi, i risultati del questionario evidenziano il potenziale di EduPKG nel rafforzare l’insegnamento STEM attraverso la pianificazione didattica, l’allineamento ai quadri normativi e il supporto all’apprendimento personalizzato. Per un’adozione efficace sarà però fondamentale semplificare l’interfaccia, garantire compatibilità con i programmi scolastici e adattabilità a diversi contesti educativi.

5. Conclusioni

EduPKG nasce per supportare i docenti nella pianificazione di attività STEM efficaci, colmando la mancanza di strumenti strutturati. Attraverso un approccio basato sulle competenze e l’integrazione delle 4C, l’ontologia favorisce un’educazione STEM accessibile e coinvolgente. In futuro, lo sviluppo di un’applicazione intuitiva renderà EduPKG più fruibile, permettendo la personalizzazione delle attività. Saranno inoltre esplorate applicazioni pratiche come collaborazioni scuola-impresa e percorsi di apprendimento personalizzati, per avvicinare ulteriormente teoria e pratica didattica.

Bibliografia

[1] Barbara Re Beatrice Strappa Alessandro Marcelletti, Sara Pettinari. Una metodologia per l’integrazione del coding nell’educazione scolastica. In ITADINFO 2024 Secondo Convegno Italiano sulla Didattica dell’Informatica. 2024.

[2] M. U. Bers. Coding as a Playground: Programming and Computational Thinking in the Early Childhood Classroom. Routledge, 2018.

[3] Stefania Bocconi, Augusto Chioccariello, Giuliana Dettori, Anusca Ferrari, and Katja

- Engelhardt. Developing computational thinking in compulsory education-implications for policy and practice. Technical report, Joint Research Centre (Seville site), 2016.
- [4] Jessica R Chittum, Brett D Jones, Sehmuz Akalin, and Asta B Schram. The effects of an afterschool stem program on students' motivation and engagement. *International journal of STEM education*, 4:1–16, 2017.
- [5] European Commission. Resetting education and training for the digital age, 2020.
- [6] M. Fernández-López, A. Gómez-Pérez, and Natalia Juristo Juzgado. Methontology: From ontological art towards ontological engineering. In *AAAI97 Spring Symposium*. American Association for Artificial Intelligence, 1997. OEG.
- [7] Mohammad Alamgir Hossain, James Deehan, and Leanne Gibbs. Unveiling the pedagogical approaches in stem classroom: A scoping review. *International Journal of Learning, Teaching and Educational Research*, 23(12):1–22, 2024.
- [8] Wai Man Vivienne Leung. Stem education in early years: Challenges and opportunities in changing teachers' pedagogical strategies. *Education Sciences*, 13(5), 2023.
- [9] Ministero dell'Istruzione dell'Università e della Ricerca. Piano nazionale scuola digitale, 2015.
- [10] S. Papert. *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books, 1980.
- [11] M. Resnick. *Lifelong kindergarten: Cultivating creativity through projects, passion, peers, and play*. MIT Press, 2017.
- [12] R. Pea S. Grover. Computational thinking in k–12: A review of the state of the field. *Educational Researcher*, 42(1), 38-43, 2013.
- [13] Robert DC Shearer, Boris Motik, and Ian Horrocks. Hermit: A highly-efficient OWL reasoner. In *Owled*, volume 432, page 91, 2008.
- [14] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. Pellet: A practical owl-dl reasoner. *Journal of Web Semantics*, 5(2):51–53, 2007.
- [15] Ramakrishnan Sivakumar and PV Arivoli. Ontology visualization protégé tools—a review. *International Journal of Advanced Information Technology (IJAIT) Vol*, 1, 2011.
- [16] J. M. Wing. Computational thinking. *Communications of the ACM*, 49(3), 33-35, 2006.
- [17] Q. Burke Y. B. Kafai. *Connected code: Why children need to learn programming*. MIT Press, 2014.

VIDEO DIVULGATIVI

ST-EMotionaLearning

Francesco Mario Pio Damiani¹, Daniela Troia¹

¹ IC G. Rodari – D. Alighieri - Marconi

francescomariopiodamiani@gmail.com

dnltr71@gmail.com

Abstract

Il video “ST-EmotionaLearning” illustra il percorso di innovazione intrapreso dall’Istituto Comprensivo Rodari-Alighieri-Marconi, volto a favorire lo sviluppo del pensiero computazionale nelle alunne e negli alunni tramite una progettazione verticale che coinvolge tutti gli ordini scolastici, dalla scuola dell’infanzia alla secondaria di primo grado. Ciò avviene attraverso la revisione del curriculum digitale e iniziative formative rivolte sia al corpo docente sia agli alunni. Il curriculum digitale, strutturato secondo il quadro di riferimento DigComp 2.2 e articolato in tre aree – tecnologia digitale, pensiero computazionale e cittadinanza digitale – pone particolare enfasi sull’area del pensiero computazionale. Gli obiettivi di apprendimento sono stati definiti sulla base delle indicazioni per il curriculum di informatica elaborate dal CINI, con l’intento di valorizzare oltre le competenze tecnologico-digitali, i contenuti e le finalità specifiche della disciplina informatica. All’interno di questo contesto, il video intende mettere in evidenza come sia possibile pianificare un curriculum di informatica continuo e coerente, a partire dallo sviluppo del pensiero algoritmico nella scuola dell’infanzia fino alla promozione del pensiero computazionale nei gradi successivi. Il percorso didattico prevede attività unplugged con il corpo e di robotica educativa con oggetti programmabili per i bambini della scuola dell’infanzia, cui seguono l’introduzione dei concetti di istruzione, sequenza e ciclo nella scuola primaria e secondaria di primo grado, anche mediante l’utilizzo della programmazione visuale a blocchi.

Obiettivo parallelo è la promozione di competenze trasversali fondamentali per la crescita personale e professionale: il lavoro di squadra in contesti di pair working e robotica, la resilienza e la gestione dell’errore attraverso il debugging, la creatività nella progettazione di soluzioni digitali e nello storytelling, nonché il pensiero critico nella riflessione sui comportamenti tecnologici responsabili. Le esperienze unplugged e narrative hanno favorito curiosità e collaborazione; sfide legate a tematiche quali la salvaguardia degli oceani e la sana alimentazione, realizzate tramite kit di robotica e realtà aumentata, hanno incentivato creatività e problem solving; le sperimentazioni con intelligenza artificiale e machine learning hanno stimolato un approccio critico e consapevole all’uso del digitale. In tutte le attività proposte, il riconoscimento dell’errore come opportunità e l’approccio laboratoriale hanno reso centrali lo sviluppo dell’autoefficacia e della resilienza negli alunni.

Il linguaggio segreto dei computer: 0 e 1!

Maria Grazia Giannoccaro¹

¹ IC "Marconi Oliva" - Locorotondo (Ba)

giannoccaromag@gmail.com

Abstract

Obiettivi. L'educazione informatica di base (Computational Thinking) è cruciale per preparare le nuove generazioni a comprendere il mondo digitale. Tuttavia, concetti fondanti come il codice binario possono risultare astratti e difficili da assimilare per un pubblico di non specialisti, in particolare bambini e adolescenti. Questo lavoro presenta un approccio didattico innovativo per la divulgazione del codice binario, utilizzando la produzione di un video breve (massimo 3 minuti) come strumento principale.

Metodologia. Il video è stato progettato seguendo i principi di divulgazione scientifica "peer-to-peer", con un linguaggio semplice, un ritmo dinamico e un'estetica visiva accattivante, pensata per il pubblico di riferimento, gli alunni della Scuola Primaria.

La metodologia si è basata su alcuni cardini.

- Scomposizione del concetto: il concetto di "bit" e "byte" è stato tradotto attraverso analogie visive e sonore, come l'interruttore "ON/OFF" e la costruzione di parole.
- Visualizzazione creativa: l'uso di animazioni, grafiche stilizzate e illustrazioni personalizzate ha reso l'informazione tangibile e memorabile.
- Integrazione multimediale: il progetto ha sfruttato l'uso di piattaforme di editing video accessibili (Canva) e l'integrazione di contenuti generati da intelligenza artificiale (immagini e audio) per dimostrare l'efficacia di strumenti moderni nella produzione di materiale didattico.

Risultati attesi. Il video mira a demistificare il codice binario, rendendolo un concetto intuitivo e non intimidatorio. Si prevede che l'approccio ludico e visivamente stimolante del video aumenterà l'engagement del pubblico giovane, migliorando la comprensione e l'interesse verso i fondamenti dell'informatica. La presentazione al convegno non solo mostrerà il prodotto finale, ma discuterà anche la metodologia di progettazione e le sfide affrontate, offrendo un modello replicabile per futuri progetti di divulgazione.

ITADINFO è il convegno italiano dedicato alla didattica dell'informatica, un luogo in cui scuola e ricerca si incontrano per condividere idee, pratiche e visioni.

La terza edizione, ospitata a Salerno nel 2025, ha raccolto contributi che mostrano come l'informatica possa diventare non solo una disciplina da insegnare, ma anche una chiave per leggere e comprendere il mondo.

Il volume raccoglie esperienze concrete, percorsi innovativi e riflessioni che spaziano dall'intelligenza artificiale alla modellazione della realtà, fino all'uso integrato di strumenti analogici e digitali. Tra relazioni scientifiche, racconti dal campo, poster, laboratori e video divulgativi, emerge un panorama ricco e accessibile, utile a chi insegna già la materia e a chi si avvicina ora.

Più che atti di un convegno, queste pagine rappresentano un invito a esplorare nuovi modi di fare scuola, a valorizzare il pensiero critico e creativo degli studenti e a costruire una comunità aperta, inclusiva e sempre in dialogo.

€ 59,00

