[RELAZIONE ORALE · ITADINFO 2026]

Riprogettare la didattica dell'informatica con l'IA

Scaffolding, esempi guidati e strumenti per la programmazione al SIA

Matteo Candita
Animatore digitale e Responsabile dell'IA
ITE Gaetano Salvemini, Casalecchio di Reno (BO)
Michael Lodi · Simone Martini
Università di Bologna

[ITADINFO]
CONVEGNO ITALIANO
SULLA DIDATTICA DELL'INFORMATICA
4ª edizione
Padova, 9-11 ottobre 2026

Il contesto: dal codice al web

Studenti senza esperienza pregressa e un obiettivo di triennio chiaro

4

ore settimanali di programmazione nel primo anno del triennio

Servono sia le basi concettuali sia un linguaggio

1

Terza: JavaScript

Accessibile, eseguibile nel browser, feedback immediato, nessuna configurazione complessa

2

PHP su Apache/MySQL

Scelta consolidata negli istituti tecnici: viene mantenuta

3

Obiettivo del triennio

Realizzare un database accessibile dal web

Integrazione verticale: ogni scelta in terza prepara l'obiettivo del triennio

Il problema di partenza

Tre limiti nel modo di programmare, due nella classe

Codice non testato

Si scrive senza verificarlo in modo sistematico

Casi limite ignorati

I valori ai margini non vengono presi in considerazione

Memorizzare, non capire

La memorizzazione prevale sulla comprensione

Attenzione che cala presto

Le spiegazioni prolungate tengono poco: la lezione frontale lunga è poco efficace

Abilità molto diverse

Un ritmo unico lascia indietro i più lenti e annoia i più veloci

Esercitazioni guidate e interattive

Due componenti, entrambe necessarie: la progettazione didattica e gli strumenti

COMPONENTE DIDATTICA cosa insegnare e come costruire gli esercizi

Progressione granulare

Ogni lezione introduce una sola novità concettuale rispetto alla precedente.

Scaffolding esplicito

La lezione accompagna lo studente, passo dopo passo, verso la competenza.

COMPONENTE TECNOLOGICA con quali strumenti

Moodle con CodeRunner

- Teoria ed esercizi distribuiti come attività interattive
- Verifica automatica del codice e feedback immediato

Risultato: ogni studente procede al proprio ritmo, fino a scrivere il codice in autonomia

[ITADINFO]
CONVEGNO ITALIANO
SULLA DIDATTICA DELL'INFORMATICA

La struttura della lezione

Pochi minuti frontali, poi pratica guidata su Moodle

10–15 min frontali

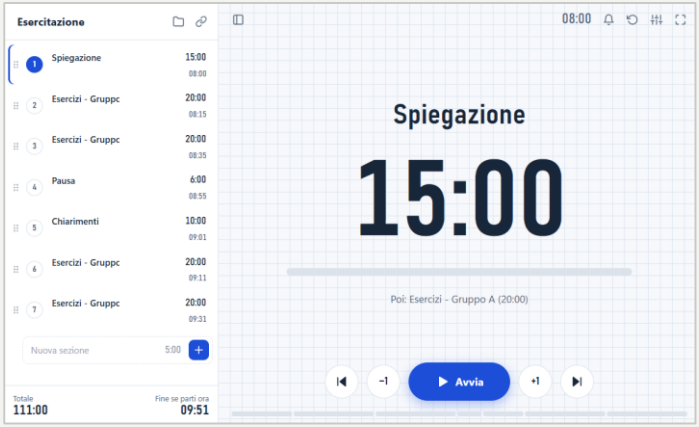
Esercitazione su Moodle con CodeRunner, ognuno al proprio ritmo

Proporzioni indicative

Necessity-based learning
Un problema che le conoscenze attuali non risolvono, una discussione con la classe, poi il nuovo costrutto come soluzione.

Ritmi disaccoppiati
Ognuno procede al proprio ritmo: nessuno resta indietro, nessuno si annoia.

Docente e ITP tra i banchi
Aiuto individuale a chi è bloccato, approfondimenti a chi ha già finito.



Webtimer
Timer a sezioni per scandire i tempi della lezione e il pair programming

Cinque fasi per ogni esercitazione

Lo scaffolding si riduce: dalla lettura alla scrittura autonoma del codice

Scaffolding: il supporto si riduce

L'autonomia dello studente cresce

1

Teoria con domande a completamento

La teoria per iscritto, verificata con domande

2

Esempio guidato

Esempio svolto, con domande di comprensione guidata

3

Esercizi di debugging

Codice con errori da identificare e correggere

4

Completamento con GapFiller

Parti di codice da completare: espressioni, variabili, chiamate

5

Esercizio completo

Lo studente scrive il codice, verificato su test anche limite

Fin dall'inizio anche le buone pratiche: nomi delle variabili, formattazione (spaziatura e indentazione) e commenti

Un'esercitazione dal vivo

Dal mio Moodle: ciò che vedono gli studenti



1. La teoria e le domande di comprensione
2. Un completamento con GapFiller: il codice viene eseguito
3. Un esercizio con i test, casi limite compresi
4. Il feedback immediato: dal rosso al verde

L'IA per il docente

Il docente guida e delinea la progressione, l'IA produce il materiale

Il docente guida

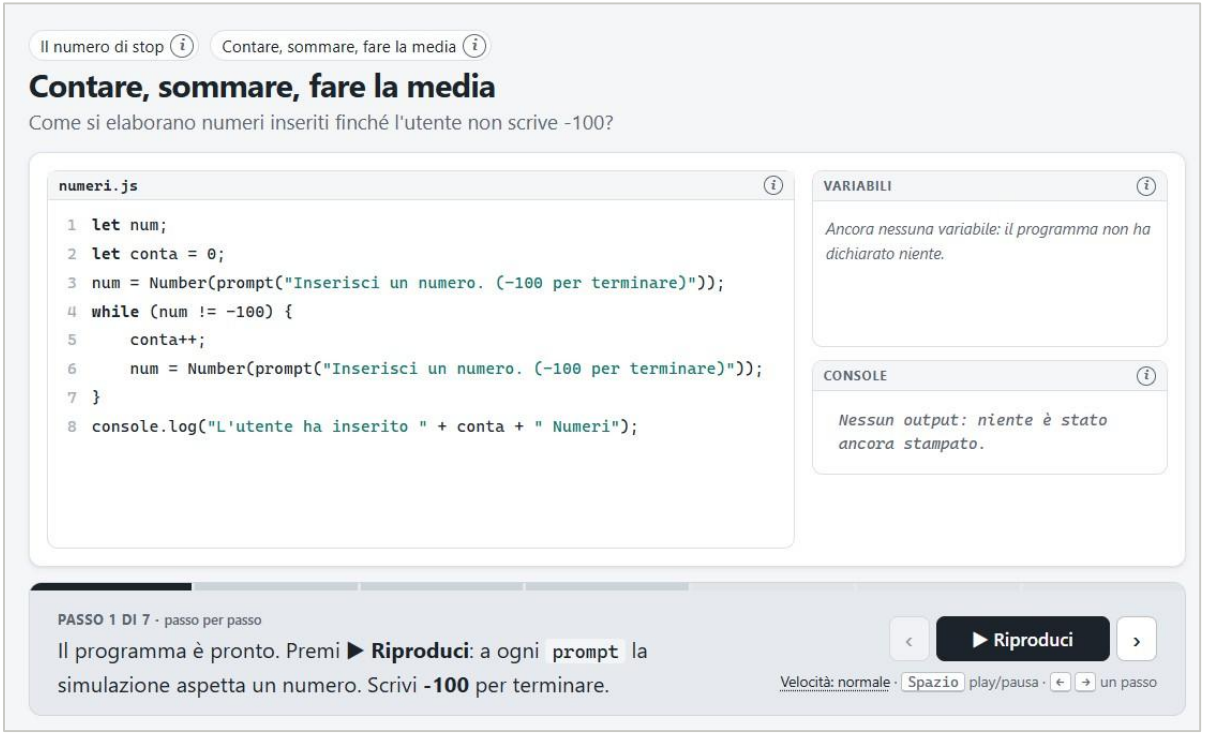
Delinea integralmente la progressione e dà indicazioni precise su ciò che serve.

L'IA produce

Genera rapidamente molti esercizi dettagliati e coerenti con le indicazioni, realizzarli manualmente non sarebbe sostenibile con il tempo.

Anche il sito didattico

Sviluppato con l'IA: visualizza passo passo l'esecuzione, lo stato delle variabili e il flusso di controllo.



Il nuovo sito di visualizzazione: codice, variabili e console, un passo alla volta

L'IA per lo studente: il tutor

Un chatbot generico dà soluzioni troppo avanzate: il tutor guida con ciò che si è già visto

Tutor Google Gemini (Gem): configurato con gli appunti del corso, limitato agli strumenti già affrontati; scelto per la conformità alle norme scolastiche sulla privacy

Lo studente incolla la traccia

Il tutor non svolge l'esercizio e non scrive codice: offre di scomporlo in step logici, scegliere la struttura di controllo, scrivere lo pseudocodice.

Lo studente incolla il proprio codice

Il tutor diagnostica l'errore, ne spiega la causa e guida la correzione passo per passo.



Il tutor non scrive soluzioni complete: guida il ragionamento, non lo sostituisce

Per una lettura interessante: paper del 2023 sui ruoli che l'IA può assumere per la didattica (Mollick & Mollick) · arxiv.org/abs/2306.10052

Prompt di JS-Tutor ›

[ITADINFO]
CONVEGNO ITALIANO
SULLA DIDATTICA DELL'INFORMATICA

Demo dal vivo: il tutor JS

Stessa traccia incollata in Gemini non personalizzato e in JS-Tutor

TRACCIA DELL'ESERCIZIO

Realizzare un programma che chiede all'utente di inserire i voti presi da cinque studenti in un compito, memorizzandoli uno dopo l'altro in un array; al termine stampare il messaggio "Il voto più alto è: " seguito dal voto più alto tra quelli inseriti.

Gemini non personalizzato

- Quale linguaggio sceglie?
- Scrive subito la soluzione?
- Usa una funzione già pronta per il massimo?

JS-Tutor

- Offre scelte: scomporre in step, struttura di controllo, pseudocodice
- Non scrive il codice al posto dello studente

[ITADINFO]
CONVEGNO ITALIANO
SULLA DIDATTICA DELL'INFORMATICA

Risultati

Due classi terze SIA, obiettivi più ambiziosi degli anni precedenti

42

studenti in due classi terze SIA

36/42

con sufficienza piena, nonostante l'aumento di difficoltà

8/10

mediana del giudizio complessivo (media 7,6), 34 rispondenti su 42

Obiettivi ampliati: esercitazioni finali più complesse e verifiche più dettagliate, che valutano correttezza funzionale e qualità formale del codice, con recupero in itinere.

Coinvolgimento e replicabilità: mediana 8/10 anche per chiarezza delle spiegazioni e materiale digitale; colleghi che hanno adottato i materiali riportano risultati soddisfacenti (verifica informale).

[ITADINFO]
CONVEGNO ITALIANO
SULLA DIDATTICA DELL'INFORMATICA

Conclusioni e prospettive

L'IA è un abilitatore essenziale, ma resta molto da misurare

Cosa mostra il percorso

- Progressione granulare, scaffolding esplicito e testing automatico portano a buoni risultati di apprendimento e gradimento
- Senza l'IA un singolo docente difficilmente produrrebbe i materiali necessari
- Il modello è indipendente dal linguaggio: da JavaScript in terza è stato esteso anche a SQL, HTML (Form) e PHP con connessione al DB

Prospettive

Tecnologie in evoluzione: difficile prevedere come insegneremo tra qualche anno

Prima quinta: quest'anno completiamo la nuova programmazione; vedremo come arriviamo in fondo rispetto ai programmi precedenti

Già in uso a scuola: da approfondire come condividerlo con altre scuole e ufficializzarlo

Da misurare: effetto dell'IA, lavoro del docente, uso del tutor

Grazie. Domande?

[ITADINFO]
CONVEGNO ITALIANO
SULLA DIDATTICA DELL'INFORMATICA

Materiali citati nel paper

Esercitazione «Early return»

Mock-up con contenuti e struttura

matteocandita-gif.github.io/demo_moodlecoderunner/esercitazione_early_return.html

Visualizzatore passo passo

Sito didattico per la notional machine

matteocandita-gif.github.io/demo_moodlecoderunner/js_demo/

Tutor Gemini (Gem)

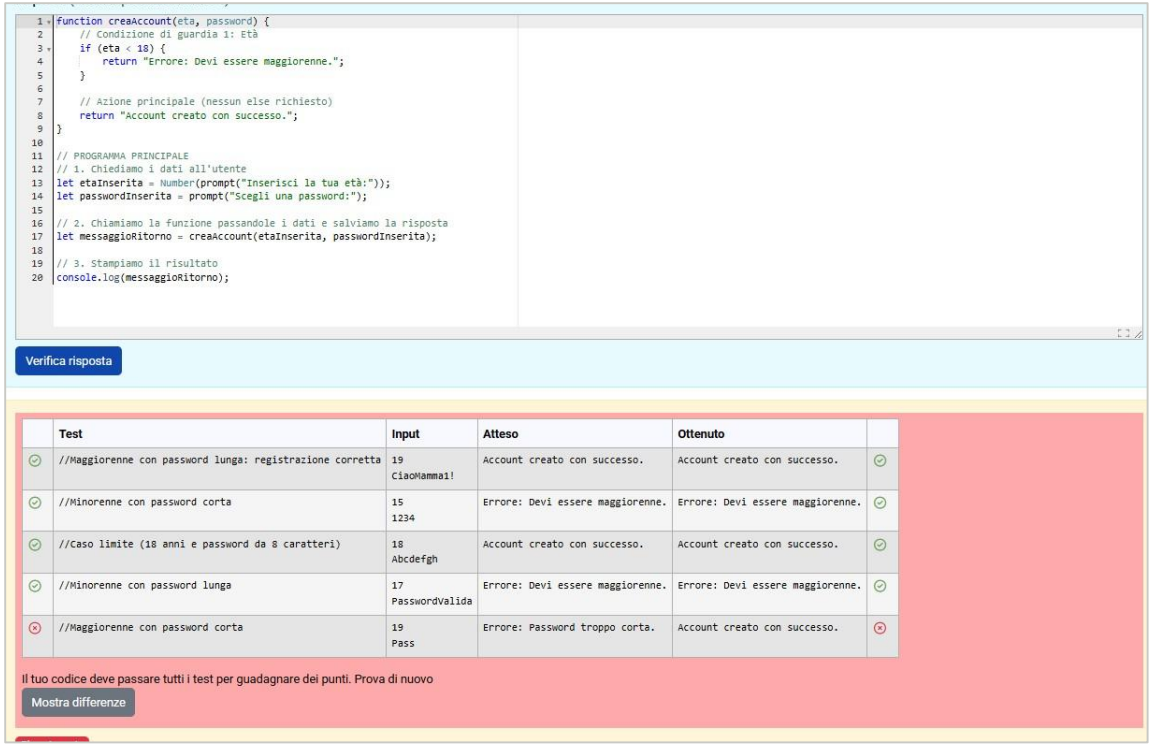
Il tutor IA personalizzato

gemini.google.com/gem/10LGVNjoGQO8fw22XYy7FV73mbc6ocCjj

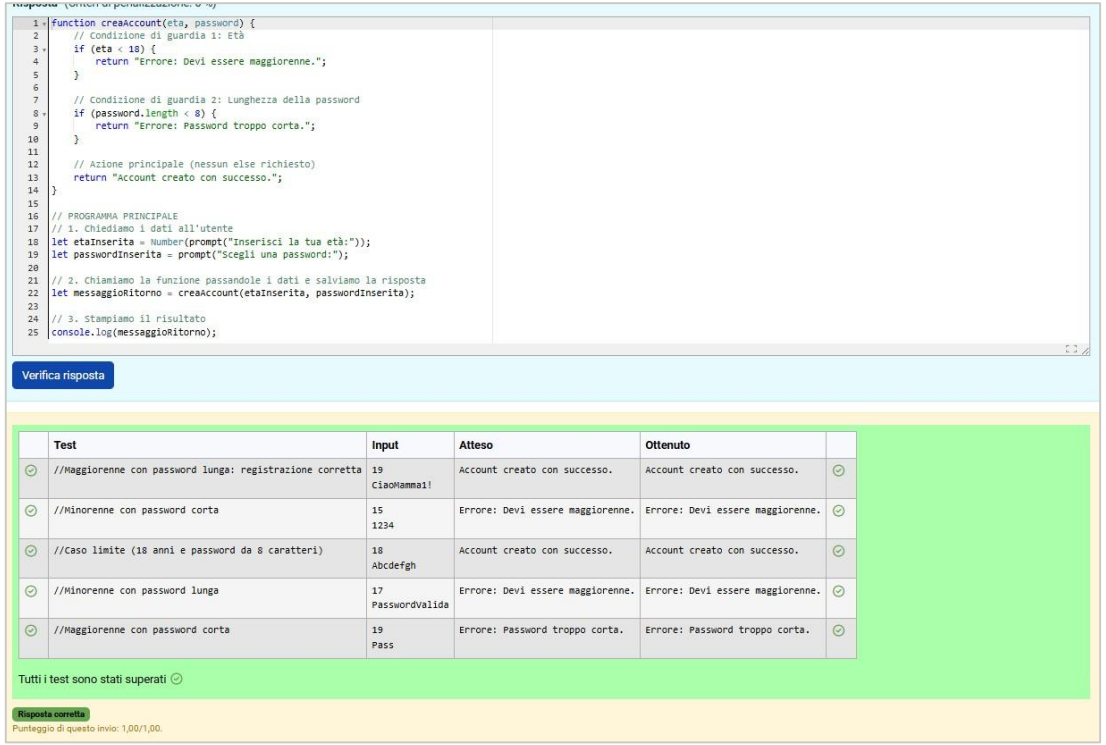
Matteo Candita · Michael Lodi · Simone Martini
matteo.candita@salvemini.bo.it

Il feedback di CodeRunner

Esercitazione «Early return e refactoring»: schermate reali di Moodle



Soluzione parzialmente errata: il quinto caso di test fallisce e lo studente deve correggere il codice



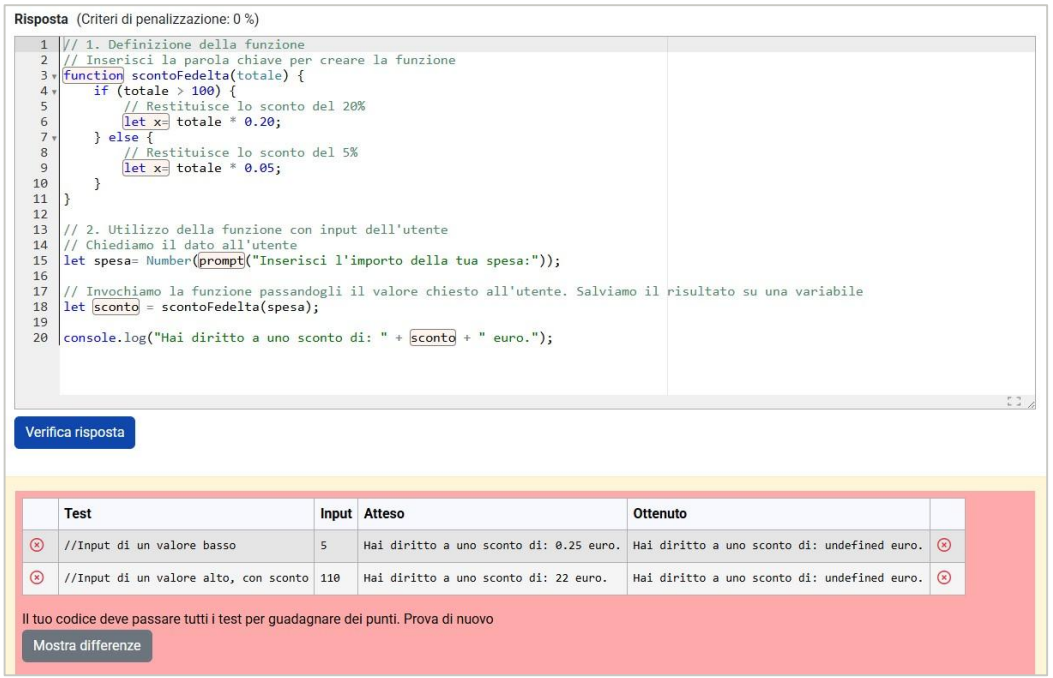
Soluzione corretta: tutti i test superati, il pannello diventa verde, punteggio pieno

GapFiller: completare ragionando

Un completamento di codice che viene eseguito, non confrontato



La struttura è già presente: lo studente inserisce function, return, prompt e dà il nome alla variabile



Risposta errata: senza return la funzione restituisce undefined e i test falliscono

Si verifica il risultato a runtime: serve ragionare, non memorizzare