INSEGNAMENTO EFFICACE DELL'INFORMATICA GRAZIE A MOODLE E CODERUNNER

Roberto Ghelli

Liceo F.Redi, Arezzo
roberto.ghelli@scuola.istruzione.it

-- FULL PAPER --

***ARGOMENTO**: Istruzione secondaria*

Abstract

A seguito di alcuni studi ed approfondimenti sulla didattica dell'Informatica nelle scuole secondarie di 2° grado si è svolta una sperimentazione volta a testare in classe l'efficacia del plugin CodeRunner di Moodle per l'apprendimento delle fondamenta di programmazione. Il plugin permette infatti di somministrare, in forma di quiz, esercizi di coding fornendo automaticamente un feedback allo studente. La sperimentazione ha visto circa 100 studenti cimentarsi con l'apprendimento delle basi della programmazione (variabili, selezioni, iterazioni, vettori, funzioni e procedure) e la sintassi del linguaggio C++ per l'intero anno scolastico. Durante tutto il periodo il docente ha osservato il processo di apprendimento mediato dalla tecnologia e calibrato alcuni ritocchi laddove necessario. A conclusione della sperimentazione i risultati dell'osservazione e della somministrazione di alcuni sondaggi hanno permesso di trarre interessanti e promettenti considerazioni sulle metodologie didattiche adottabili, l'efficacia e la sostenibilità di tale approccio.

Keywords: Metodologie didattiche, CodeRunner, coding, DigCompEdu, E-Learning.

1 INTRODUZIONE E CONTESTO

L'apprendimento delle fondamenta di programmazione comporta numerose piccole sfide, ostacoli e traguardi che ogni discente deve affrontare al fine di acquisire quelle abilità che gli permetteranno di padroneggiare strumenti e tecniche necessari per progettare e realizzare algoritmi sempre più complessi ed articolati.

Ad oggi le basi di un linguaggio di programmazione vengono insegnate in numerosi percorsi delle scuole secondarie di secondo grado, tra cui quello oggetto del presente articolo, cioè il corso di Informatica dei Licei Scientifici ad opzione Scienze Applicate, dove il tema viene trattato solitamente nel primo biennio. Si mostra di seguito come tale l'insegnamento possa essere coadiuvato da strumenti digitali specifici come il software CodeRunner, un plugin della piattaforma di E-Learning Moodle, finalizzato di somministrare domande di coding, testarne le risposte e in tempo reale e fornire un feedback immediato agli studenti.

1.1 Il plugin CodeRunner e Moodle

CodeRunner è un plugin per Moodle [1] in grado di potenziare i classici moduli "Quiz" permettendo di formulare domande la cui risposta sia l'implementazione di algoritmi in vari possibili linguaggi di programmazione, tra cui si citano Java, JavaScript, Python, C, C++, PHP (Fig. 1).

Dal punto di vista didattico-educativo CodeRunner può essere inserito senza difficoltà all'interno di una didattica tradizionale. Rappresenta un ambiente di apprendimento costruttivista [2] e virtuale dove gli studenti sono liberi di realizzare e testare possibili soluzioni ai problemi proposti.

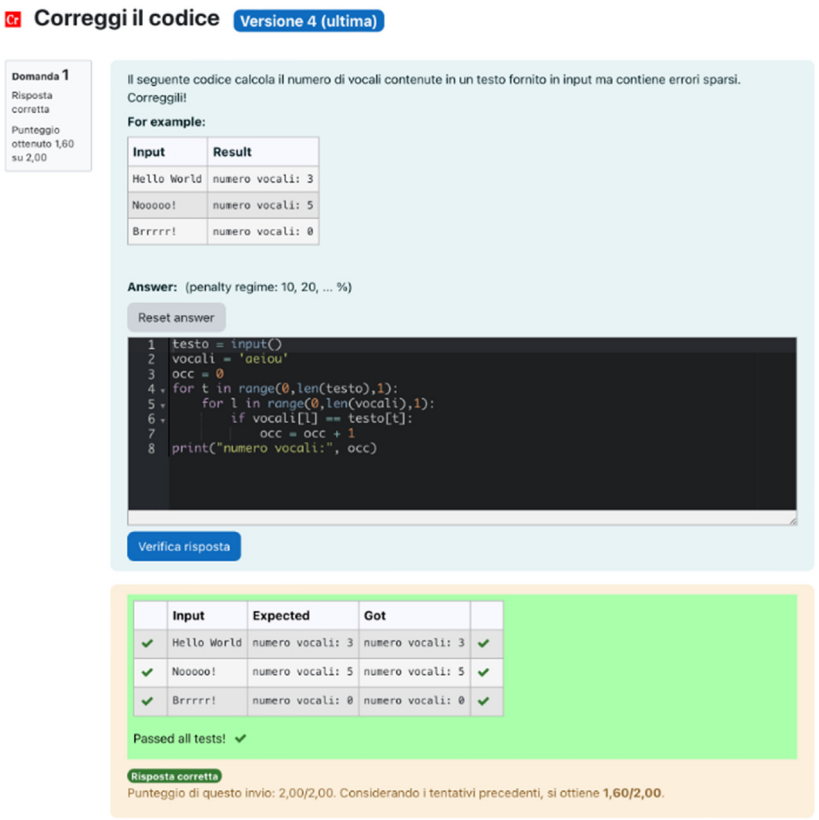


Figura 1: Esempio di quiz di coding con feedback immediato

Per approfondimenti sul plugin CodeRunner e la sua configurazione si rimanda all'articolo "Il laboratorio virtuale di coding per una didattica dei linguaggi di programmazione" di Giuliana Barberis presentato al MoodleMoot 2021 [2].

Altro aspetto di grande rilievo è il fatto che il plugin è calato all'interno dell'ambiente di E-Learning Moodle, comportando numerosi benefici qualora il percorso didattico venga progettato adeguatamente. Si pensi ad esempio all'utilità che può comportare il gestire vari contenuti in sezioni, magari in linea con i capitoli del libro di testo oppure, a titolo di esempio, sfruttare le potenzialità di tag, competenze, "gamificare" specifiche attività, offrire badge e molto altro.

Vi sono aspetti di rilievo riconducibili al modello TPACK [3] ed il framework DigCompEdu [4] anche nella gestione dei corsi e, più in generale, nella professionalità del docente [5] e [6]. Si denota infatti un legame tra Moodle e il framework DigCompEdu, che può essere visto come una leva per lo sviluppo delle competenze digitali degli insegnanti. Il framework fornisce una guida strutturata per migliorare le pratiche didattiche digitali attraverso sei ambiti di competenza, e Moodle, grazie alla sua flessibilità e alle numerose funzionalità, supporta in modo naturale questo percorso. Ad esempio, le funzionalità di Moodle, potenziate dal plugin CodeRunner, consentono di facilitare l'apprendimento attraverso quiz automatizzati e valutazioni formali e informali, favorendo l'interazione e la personalizzazione dei percorsi formativi. Questo approccio non solo rafforza la capacità del docente di gestire efficacemente gli ambienti digitali e le risorse educative, ma favorisce anche la motivazione e il coinvolgimento degli studenti, contribuendo alla realizzazione di una didattica allineata con le direttive del framework DigCompEdu. Di conseguenza, CodeRunner, integrato in Moodle, può essere visto come uno strumento utile anche per l'innovazione didattica e lo sviluppo delle competenze digitali e professionali del docente.

2 SPERIMENTAZIONE SVOLTA

La sperimentazione svolta aveva per obiettivo l'insegnamento delle fondamenta di un linguaggio di programmazione (in questo contesto il C++), quali: sintassi, variabili, selezioni ed iterazioni, funzioni e

procedure (metodi); una parte ristretta di studenti ha invece affrontato temi successivi quali: array e matrici, algoritmi notevoli di ricerca ed ordinamento, ricorsione.

Sono state coinvolte un totale di cinque classi ed ha visto impegnati circa 110 studenti per buona parte dell'anno scolastico. Le tematiche venivano trattate sia in aula (soprattutto per gli aspetti teorici) che in laboratorio informatico attrezzato (tanto per argomenti pratici quanto per l'applicazione e consolidamento delle competenze teoriche acquisite). Ad inizio anno, in fase di progettazione didattica si è scelto di sfruttare CodeRunner con il fine di agevolare gli studenti nel raggiungimento degli obiettivi disciplinari nonché migliorarsi in termini di autonomia in generale verso la materia, nell'autoregolazione, nel sentirsi più partecipanti attivi della classe. CodeRunner è stato utilizzato pertanto per una buona parte della didattica, tra cui: attività di laboratorio, lo studio autonomo a casa e buona parte delle verifiche pratiche con valutazione (vedi anche [7]).

2.1 Organizzazione di corsi e quiz

La parte più significativa della sperimentazione ha visto coinvolte in parallelo 4 classi seconde con pregresse basi di programmazione, solo visuale. Per ogni classe è stato realizzato un corso Moodle. I corsi Moodle sono stati suddivisi in 6 argomenti fondanti: sintassi di base, variabili, selezioni, iterazioni, funzioni, vettori. Per ogni argomento sono state inserite 3 o 4 batterie di esercizi (quiz) di allenamento e consolidamento, ognuno composto da 4 o 5 domande. Ogni due argomenti è stata svolta una verifica in classe con valutazione.

I set di esercizi di coding in forma di attività quiz (Fig. 2), ad eccezione delle sole verifiche in classe, sono rimasti disponibili agli studenti per tutto il corso e potevano essere svolti più volte.

Figura 2: Esempio di corso organizzato in argomenti e quiz

Le domande non presentano penalità, dispongono di una descrizione volta a spiegare l'esercizio in modo chiaro e con un linguaggio il più possibile naturale. In pochi casi viene richiesto di scrivere codice ex-novo, più spesso di correggere o ampliare codice preesistente così da permettere allo studente di concentrarsi sulle singole abilità e competenze da rafforzare (Fig. 3, Fig. 4).

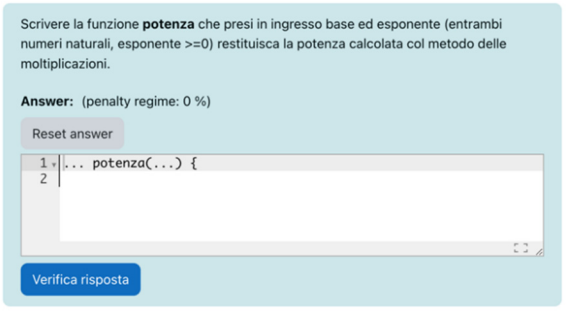
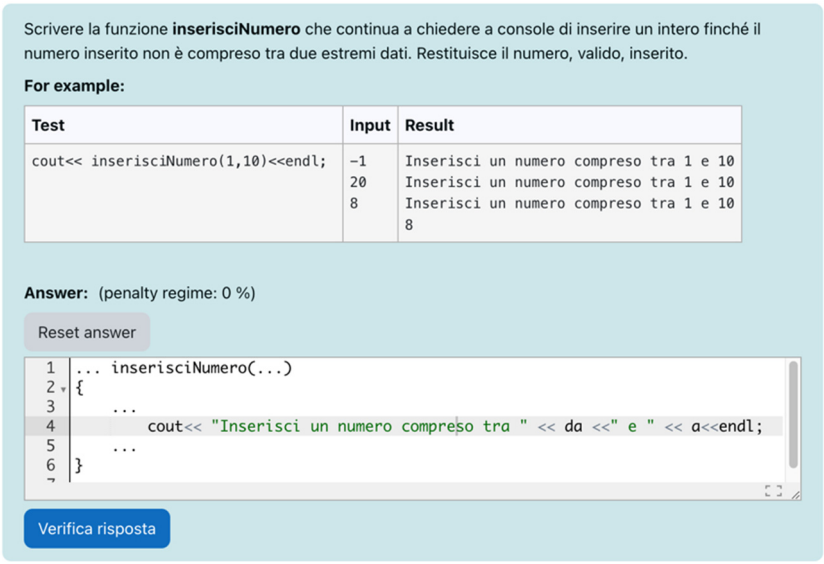


Figura 3: Esempio di domanda su Funzioni/metodi

In vari casi sono offerti dei test case e snippet di codice volti a semplificare la comprensione dell'esercizio o l'obiettivo pratico da raggiungere. (Fig. 4).

Figura 4: Esempio di domanda su Funzioni con test case.



Nel deposito di domande sono state inserite circa 120 domande, organizzate gerarchicamente, con nomi significativi (con un numero iniziale ad indicare la "temporalità" dell'argomento) e tag riferiti alle competenze a cui si riferivano al fine di semplificarne la gestione e la riusabilità (Fig. 5).

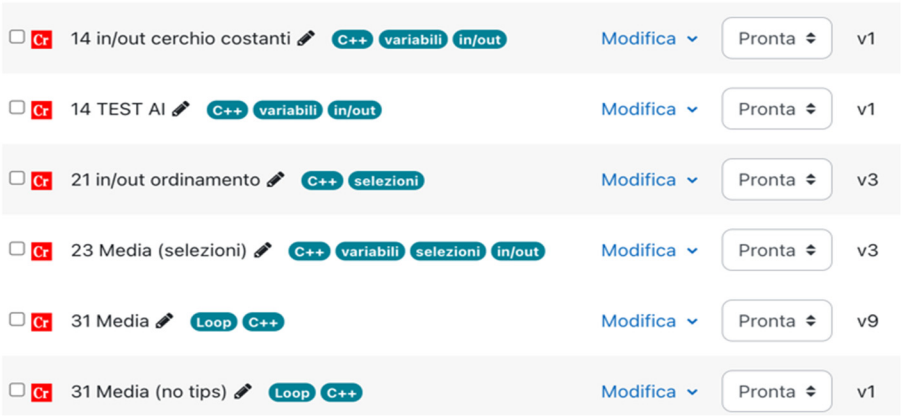


Figura 5: Alcune domande CodeRunner presenti nel deposito

2.2 La valutazione automatica sul registro

La valutazione degli esercizi di programmazione è per sua natura un potente strumento per fornire un feedback veloce e immediato agli studenti. Tra i principali pregi di questo sistema c'è la possibilità di valutare rapidamente grandi quantità di compiti, riducendo il carico di lavoro del docente e offrendo agli studenti la possibilità di correggere e migliorare il proprio codice autonomamente, con un chiaro vantaggio in termini di apprendimento attivo e auto-regolazione.

Tuttavia, durante la sperimentazione si è riscontrato che uno dei limiti più significativi di CodeRunner riguarda la sua efficacia nel valutare solo un insieme di competenze e abilità, principalmente legate alla scrittura di codice, all'applicazione di costrutti, alla risoluzione di problemi fornendo risultati talvolta rigidi come si può evincere dall'analisi dei risultati più avanti. Inoltre, è importante notare che non tutti gli studenti traggono beneficio da questo metodo di valutazione. Alcuni possono sentirsi bloccati o frustrati davanti a un test completamente automatizzato, specialmente quando l'interfaccia o i feedback sugli errori non sono sufficientemente chiari per loro. Questo può accadere in particolare con studenti che trovano difficoltà nel rapportarsi con l'ambiente digitale o che ancora necessitano di un aiuto umano per

poter mostrare le proprie abilità. Anche altre abilità molto importanti, come la comprensione teorica degli algoritmi, la capacità di spiegare concetti complessi o il ragionamento logico-deduttivo, non sono state essere pienamente misurate attraverso questo tipo di valutazione automatizzata.

Per questi motivi, la valutazione di CodeRunner ha riguardato globalmente circa il 40% delle valutazioni riportate sul registro di classe durante l'insegnamento degli argomenti oggetto di questa ricerca. Le altre modalità di verifica, ovvero il 60%, sono rimaste quelle tradizionali con prevalenza di interrogazioni orali che consentono di esplorare meglio gli aspetti concettuali e teorici della programmazione, fornendo un'opportunità per intervenire tempestivamente in caso di difficoltà individuali.

3 ANALISI DEI RISULTATI E CONCLUSIONI

Al fine di valutare in modo efficace i risultati delle attività didattiche il docente ha monitorato costantemente l'andamento del corso sulle varie classi anche con il fine di ricalibrare la didattica ed effettuare correzioni e migliorie in itinere o venire incontro a nuove esigenze. Inoltre, a fine anno scolastico è stato somministrato agli studenti un sondaggio anonimo su base volontaria a cui hanno risposto 73 allievi. Di seguito si riportano i risultati di maggior rilievo ed alcune riflessioni.

Osservando la Fig. 6, emerge come la didattica odierna sia vissuta dagli adolescenti in modo sempre più digitale e immersivo, specialmente in una disciplina come la programmazione, che è intrinsecamente legata al mondo digitale.

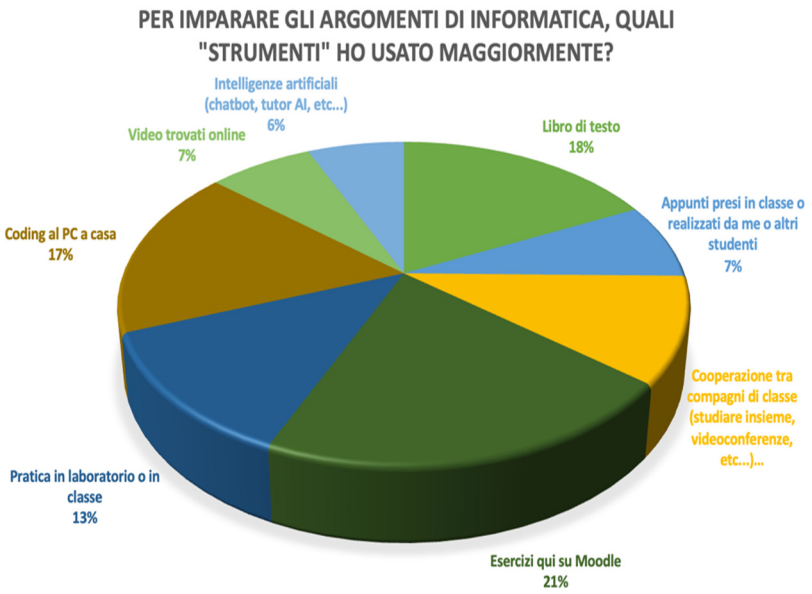


Figura 6: Principali strumenti per lo studio del coding

Sebbene il libro di testo mantenga un ruolo importante, strumenti come i video online e le intelligenze artificiali stanno acquisendo un peso crescente nell'apprendimento. È significativo notare che Moodle e CodeRunner, nonostante rappresentino una novità per gli studenti coinvolti, siano stati accolti positivamente e utilizzati in misura maggiore rispetto ad altre risorse, incluso il libro di testo.

Per brevità, si riportano in Tab.1 le principali domande e risposte in forma compatta. Dalle prime risposte della tabella (domande 1,2,3) si nota che la piattaforma è accettata di buon grado e ritenuta utile pressoché dalla totalità dei discenti che apprezzano la "novità" del feedback immediato. Mentre c'è ancora margine di miglioramento in termini metacognitivi ed autovalutativi (domande 4, 5).

L'interfaccia utente non sembra aver introdotto difficoltà (domande 12, 13 e Fig. 7) e gli studenti vorrebbero riutilizzarla in futuro (domanda 11), un aspetto non scontato, considerando che comporta esercizi a casa e tempo da dedicare allo studio, attività che non sempre rientrano tra le priorità principali per un adolescente.

N°	Domanda	Punteggio
1	(Utilità esercizi) Ritieni utile aver potuto svolgere gli esercizi sulla piattaforma?	4,2
2	(Utilità test case) Era utile avere un feedback immediato con la spunta verde/rossa sui test-case?	4,8
3	(Utilità errori compiler) Era utile avere un feedback immediato con gli errori di compilazione?	4,6
4	(Utilità per progressi) In generale la piattaforma ti ha aiutato a fare progressi?	3,8
5	(Utilità per monitorare progressi) In generale la piattaforma ti ha aiutato a MONITORARE i tuoi progressi e autovalutarti?	3,5
6	(Stimolante) Pensi che la piattaforma sia stata più stimolante rispetto alle modalità "classiche" di lezione (parte teorica, pratica in lab, etc...)?	3,8
7	(Organizzazione esercizi) Gli esercizi proposti erano in linea con quelli proposti nei compiti in classe?	4,1
8	(Errori compiler) Gli errori di compilazione forniti erano utili nel trovare bug o correggere il codice?	3,9
9	(Test case) I test case forniti erano utili nel trovare bug o correggere il codice?	4,0
10	(Utilità indicazioni) Ritieni utili il testo e le indicazioni fornite con i singoli esercizi per svolgere bene gli esercizi?	3,9
11	(Riuso in futuro) Vorresti riutilizzare per imparare tematiche simili in futuro (Coding, abilità pratiche, etc...)?	4,1
12	(UX/UI) Come valuti l'interfaccia per lo svolgimento degli esercizi?	3,9
13	(UX/UI) L'interfaccia funzionava bene? Era affidabile?	3,9

Tabella 1: Punteggio medio alle principali domande del questionario di fine corso

Dal dettaglio delle risposte risaltano anche altre informazioni utili, come nella domanda mostrata a Fig.8 che, se analizzata più nel dettaglio, mostra che quasi il 10% del campione non ha ancora fatto propria l'essenza dello strumento "compiler", pertanto, il docente dovrà fornire un adeguato supporto in merito.

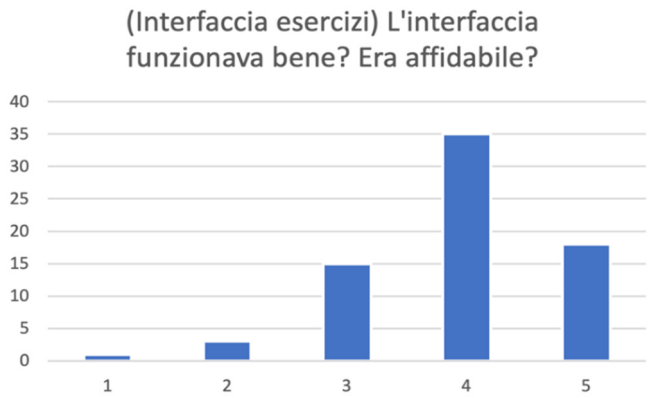


Figura 7: Riscontro sulla User Experience di Moodle e CodeRunner

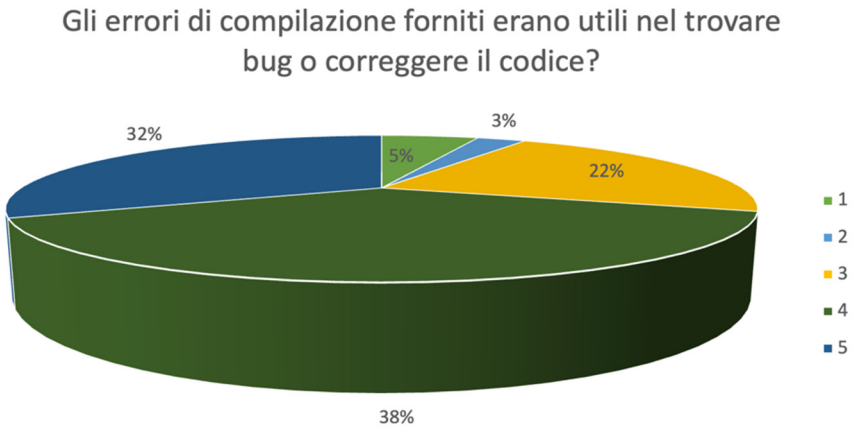


Figura 8: Riscontro sull'utilità dei feedback sugli errori di compilazione

Per ultimo agli studenti sono stati chiesti anche alcuni suggerimenti. Di seguito si riportano alcune risposte tra quelle ritenute più interessanti o frequenti.

Alla domanda "Cosa cambieresti sugli esercizi", si ha:

- *"in realtà nulla, mi sono trovato bene nel farli: li trovo completi e utili"*
- *"In caso di esercizi finalizzati alla preparazione, sarebbe stato utile avere suggerimenti in più."*
- *"In alcuni esercizi le tracce erano meno chiare e venivano capiti meglio quando si provava a fare il programma e si vedevano le test-case."*
- *"nei test case, io avrei fatto che gli errori degli spazi non incidessero sul risultato dell'esercizio"*
- *"Avrei inserito una gamma più ampia di esercizi"*

Alla domanda "Cosa non cambieresti mai", si ha (le seguenti risposte sono sostanzialmente ripetute molte volte):

- *"vorrei che la penalità continui a non esserci"*
- *"Il feedback, il debugger che aiuta a capire gli errori, come sono impostati."*
- *"La correzione immediata che si ha alla fine di ogni esercizio (segnata in rosso o verde) e la possibilità di poter ripetere il tentativo per un numero illimitato di volte"*
- *"il test case, e il fatto che forniva gli errori di compilazione in modo da correggere in maniera immediata i bug del programma"*
- *"le soluzioni dopo la consegna, utili a comprendere gli errori commessi, il fatto che non ci sia penalità, che mi aiuta ad imparare passo passo senza la paura di sbagliare. il fatto che sono simili a quelli del compito"*

Alla domanda "Hai qualche altro suggerimento per migliorare la piattaforma?", si ha:

- *"il fatto che la piattaforma segna errore anche quando il codice scritto dall'utente differisce con quello preimpostato anche solo di uno spazio è abbastanza fastidioso, a volte depistante. non so se è possibile rendere più "morbida" la correzione da parte della piattaforma, in caso io lo farei"*
- *"magari la funzione (facoltativa) di evidenziare nel programma stesso i possibili errori di scrittura."*
- *"Dopo un tot. di tentativi fornire un indizio o dritta su come svolgere l'esercizio"*
- *"secondo me sarebbe bello avere un interfaccia un pò piu carina a livello estetico perchè aiuterebbe a lavorare meglio"*

- *"evidenziare in rosso direttamente nel body del programma dove è situato l'errore commesso"*

3.1 Conclusioni

Lo strumento CodeRunner ha raggiunto pienamente gli obiettivi attesi, contribuendo in modo significativo al raggiungimento dei traguardi didattici prefissati. Gli studenti hanno utilizzato la piattaforma in modo proattivo e con serenità, affrontando le tante sfide dell'apprendimento della programmazione e migliorando le proprie competenze. Tuttavia, è emerso che ci sono alcune aree che necessitano di migliorie. Ad esempio, sarebbe utile affinare la gestione dei feedback per rendere più chiari alcuni aspetti del codice e considerare un approccio più "flessibile" per gli errori formali, come quelli legati a piccoli dettagli sintattici (ad esempio spazi o formattazione) magari realizzando nuovi prototipi di domanda CodeRunner.

Un ulteriore beneficio di questa metodologia è la riduzione della ripetitività del lavoro del docente, che può dedicare più tempo ad attività di progettazione e realizzazione di contenuti didattici, riflessione e condivisione tra pari. Questo approccio consente di sviluppare pratiche didattiche più mirate, in grado di valorizzare le diverse potenzialità degli studenti, favorendo un ambiente di apprendimento che incoraggia la collaborazione e il coinvolgimento attivo, stimolando il pensiero critico e l'autonomia nell'apprendimento.

Nonostante la valutazione automatica abbia un valore aggiunto importante, essa non può sostituire completamente il giudizio qualitativo e l'empatia del docente. L'integrazione con momenti di valutazione orale e attività collaborative continua a essere cruciale per garantire una gestione ottimale della classe e delle sue dinamiche nonché un'analisi più completa delle competenze degli studenti, specialmente quelle legate alla comprensione teorica, alla capacità di ragionamento logico e alla comunicazione delle soluzioni proposte. In sintesi, CodeRunner, inserito nel contesto di un ambiente di apprendimento digitale come Moodle, rappresenta un valido alleato per l'insegnamento della programmazione, pur richiedendo un'integrazione armonica con altre modalità didattiche per un'efficace formazione a tutto tondo.

Riferimenti bibliografici

[1] Homepage del plugin CodeRunner https://moodle.org/plugins/qtype_coderunner

[2] Barberis G. *Pagine da MoodleMoot Italia 2021* - Atti del Convegno-, pp. 13-24. https://www.aium.it/pluginfile.php/9957/mod_data/content/9628/Pagine da MoodleMoot Italia 2021 - Atti del Convegno- 1064.pdf

[3] Harrington, R. A., Driskell, S. O., Johnston, C. J., Browning, C. A., & Niess, M. L.. *Technological Pedagogical Content Knowledge*, In TPACK (2019), https://doi.org/10.4018/978-1-5225-7918-2.ch016

[4] https://scuolafutura.pubblica.istruzione.it/fr/didattica-digitale/strumenti-e-materiali/digcompedu

[5] Ghelli R., Manzanedo M.S. *Pagine da MoodleMoot Italia 2023* - Atti del Convegno-, "*Formazione docenti sugli strumenti open source per l'insegnamento dei linguaggi di programmazione*"

[6] Ranieri M. *Pagine da MoodleMoot Italia 2023* - Atti del Convegno-, "*Competenze digitali per insegnare: dal framework europeo alla applicazioni su Moodle*"

[7] Ranieri M. "*Competenze digitali per Insegnare*" Carrocci Editore, 2022, pp. 97-108