

# L'ALTAIR RITORNA... E INSEGNA PYTHON CON MOODLE

**Ivano Coccorullo**

IIS Tommaso Salvini  
*info@ivanococcorullo.it*

— **COMUNICAZIONE** —

**ARGOMENTO:** *Istruzione secondaria - Valutazione dell'apprendimento a distanza*

## **Abstract**

In questo lavoro è presentata un'esperienza di insegnamento dell'informatica mediante Moodle in due corsi sulla programmazione in Python. In particolare, gli studenti coinvolti nella sperimentazione erano divisi in due gruppi: uno costituito da studenti frequentanti il liceo scientifico e uno da studenti frequentanti il tecnico informatico. L'esperienza è stata finalizzata ad evidenziare come l'adozione di strumenti digitali avanzati possa migliorare l'insegnamento e la valutazione delle discipline informatiche, rendendo il processo di apprendimento più interattivo, efficiente e personalizzato. Nei due corsi è stata, quindi, organizzata una competizione tra "programlatori" con l'ausilio di due strumenti di Moodle: CodeRunner e Level Up!. A conclusione dell'esperienza i risultati dell'osservazione in classe e della somministrazione di un sondaggio hanno permesso di trarre interessanti e promettenti considerazioni sulle metodologie didattiche adottabili, l'efficacia e la sostenibilità di tale approccio.

**Keywords** – gamification, programmazione in Python, PNRR.

## **1 INTRODUZIONE**

L'IIS Tommaso Salvini di Roma nell'ambito del Piano Nazionale di Ripresa e Resilienza (DM 65/2023) è stato destinatario di un finanziamento di circa duecentomila euro per la realizzazione di iniziative in merito all'esplorazione di nuove strategie didattiche per l'acquisizione di competenze digitali, STEM e multilinguistiche. L'istituto ha organizzato 40 percorsi dedicati agli studenti di cui 19 dedicati all'orientamento e formazione per il potenziamento delle competenze STEM, digitali e di innovazione. I corsi sono stati tenuti da docenti universitari e da docenti interni all'istituto. La commissione che ha esaminato i CV dei docenti che hanno dato la disponibilità a svolgere il ruolo di esperto nei corsi mi ha assegnato due corsi di programmazione in linguaggio Python uno per studenti del Liceo Scientifico e l'altro per studenti dell'indirizzo Tecnico Informatico.

Le competenze informatiche dei due gruppi erano molto diverse, in quanto al Liceo Scientifico non sono previste ore di informatica con esclusione dell'indirizzo Scienze Applicate in cui si svolgono due ore a settimana mentre all'indirizzo Tecnico Informatico le ore settimanali di informatica sono più di dieci.

L'iniziativa trae ispirazione da esperienze condotte soprattutto in epoca Covid quando la didattica era spesso svolta a distanza [1] e ha avuto l'obiettivo di testare l'impatto di tecnologie digitali innovative sulla didattica e sulla valutazione in ambito informatico e soprattutto di rendere l'apprendimento più coinvolgente, dinamico e su misura.

Nei due corsi pomeridiani è stata, quindi, organizzata una competizione tra "programlatori" con l'ausilio di due strumenti di Moodle: CodeRunner, uno strumento in grado di creare test di coding che si autocorreggono e levelUp attraverso cui gli studenti possono monitorare i loro progressi nel percorso assegnato e la loro posizione nella classifica della classe. L'integrazione di CodeRunner consente non solo di automatizzare la correzione dei test, ma anche di offrire agli studenti un feedback immediato e dettagliato sulle loro soluzioni. L'utilizzo di Level Up! consente il ricorso a tecniche di gamification che hanno la capacità di trasformare una partecipazione passiva al processo di apprendimento in una partecipazione attiva che rende l'apprendimento più efficace. Al termine dei due percorsi, l'analisi dei

dati raccolti, attraverso l'osservazione in aula e un questionario, ha fornito spunti significativi e incoraggianti sull'efficacia, la sostenibilità e le potenzialità delle metodologie didattiche sperimentate.

## 2 L'INSEGNAMENTO DELL'INFORMATICA E LA GAMIFICATION

L'insegnamento dell'informatica deve avvalersi di tecniche che guidino gli studenti nell'acquisizione di capacità di astrazione e pensiero algoritmico. Le caratteristiche percettive e strutturali dell'apprendimento così come le descrivono le teorie costruttiviste, risultano particolarmente adatte per la didattica dell'Informatica, specie nel coding, dove, nelle attività di problem solving, è indispensabile la maturazione di un pensiero creativo computazionale [1]. Insegnare a programmare significa, prima di tutto, coltivare negli studenti il pensiero computazionale. Questa forma mentis è fondamentale per scomporre e risolvere problemi in modo logico, rappresentando il passaggio obbligato prima di tradurre una soluzione in un linguaggio di programmazione. Imparare a programmare è fondamentale per sviluppare un pensiero logico e creativo, tanto che può essere introdotto con successo già nella scuola primaria. Come segnalato da Jan Lepeltak sulla rivista Bricks nel 2019, in molti paesi europei il pensiero computazionale e il coding stanno diventando materie scolastiche obbligatorie [2]. Nonostante l'importanza dello sviluppo del pensiero computazionale nelle nuove generazioni, nei quadri orari del Liceo Scientifico non sono previste ore di informatica con esclusione dell'indirizzo Scienze Applicate in cui, in ogni caso, si svolgono soltanto due ore a settimana, diverso il discorso nei Tecnici informatici in cui le ore dedicate alle discipline informatiche sono più numerose.

I corsi sono stati svolti in orario pomeridiano, quindi, la prima difficoltà è stata quella di coinvolgere studenti reduci da 6-7 ore di didattica frontale. Dal 2019 ho messo in pratica con i miei studenti alcune esperienze di gamification nella didattica della matematica che hanno dato risultati molto incoraggianti [3]. Infatti, l'utilizzo della gamification e dei serious game nella didattica ha ampiamente dimostrato di poter cambiare il modo in cui le persone interagiscono con ciò che li circonda. L'uomo ha un impatto significativo sull'ambiente e, in virtù di questo, un'influenza positiva posta dal gioco e orientata ad una formazione che prenda in considerazione la natura e ciò che la compone, rappresenta un'interessante applicazione. La sfida della formazione nella nostra epoca potrebbe comprendere lo sfruttamento di queste leve per coinvolgere gli allievi, stimolare i loro interessi e catturare la loro attenzione.

## 3 L'ESPERIENZA

### 3.1 ALTAIR

Dall'anno scolastico 2016-2017 presso il Polo Liceale dell'IIS Tommaso Salvini di Roma ho avviato una sperimentazione volta a verificare la possibilità di utilizzare Moodle all'interno della scuola non solo come strumento didattico ma anche come strumento di organizzazione e gestione della scuola per costruire un'identità collettiva, culturale e professionale unitaria. La scelta è caduta su Moodle perché rappresenta uno strumento molto potente e versatile grazie alle sue funzioni di base estendibili tramite una biblioteca di plug-in pressoché completa. Un ulteriore vantaggio è che l'utilizzo di tale piattaforma non comporta nessun aggravio sul bilancio delle scuole, in quanto sia la piattaforma che i plug-in sono gratuitamente scaricabili dalla rete.

Dopo alcuni anni, ho sentito la necessità di rinnovare completamente la piattaforma che utilizzavo e di darle un taglio più "informatico" dovendo sviluppare corsi dedicati alla programmazione in Python e alla robotica. Il nome con cui è stata battezzata la piattaforma Altair è dedicato all'Altair 8800, uno tra i primi microcomputer disponibili sul mercato sviluppato e commercializzato nel 1975 dalla Micro Instrumentation and Telemetry Systems. Sebbene non sia il primo microcomputer in commercio (il Micral N era già venduto, anche assemblato, dal 1973), è uno dei sistemi più significativi della rivoluzione informatica, grazie al suo successo notevole per l'epoca. Il numero di gennaio 1975 della rivista Popular Electronics, presentando in copertina l'Altair 8800, diede inizio all'era dell'informatica per tutti. Prima di allora il termine "computer" era sinonimo di macchine grandi e costose che potevano permettersi solo le grandi aziende. La macchina era rudimentale, dotata in ingresso solo di una serie di interruttori e in uscita solo di una serie di LED, e inizialmente priva di ROM (all'avvio andava inserita a mano una sequenza di istruzioni generali). Tuttavia, grazie al suo costo relativamente basso, inferiore a 450 dollari nella versione in kit da montare, aprì la strada a una stagione di computer fai da te che portò a modelli più avanzati, come l'Apple I e i primi preassemblati come il Sol-20 [4].

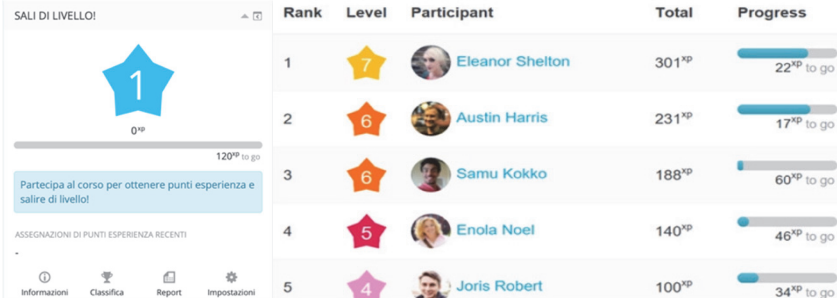
Utilizzando le attività messe a disposizione da Moodle è stata sviluppata una sezione sulla piattaforma Moodle in cui è stato implementato un percorso con quesiti informatici che gli alunni devono superare per conseguire il premio. Tutte le Attività e le Risorse di Moodle prevedono di stabilire alcune condizioni che permettono l'accesso soltanto agli studenti che le soddisfano.

Nei due corsi pomeridiani è stata, quindi, organizzata una competizione tra “programmatori” con l'ausilio di due strumenti di Moodle: CodeRunner, uno strumento in grado di creare test di coding che si autocorreggono e levelUp attraverso cui gli studenti possono monitorare i loro progressi nel percorso assegnato e la loro posizione nella classifica della classe.

### 3.2 LEVEL UP!

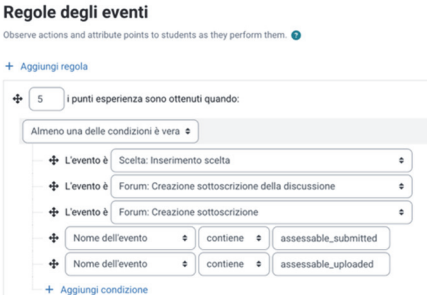
Per rendere più interattivo il percorso è stato utilizzato il plug-in “Level Up!”. Gli alunni mediante “Level Up!” possono monitorare i loro progressi nel percorso assegnato e la loro posizione nella classifica della classe. Il plug-in esiste in una versione gratuita ed una a pagamento contenente una serie di funzioni precompilate. In figura 1 e 2 sono riportate alcune immagini tratte dal plug-in Level Up!

Il plug-in permette di stabilire delle regole in base alle quali vengono assegnati i punteggi. Per incoraggiare gli alunni sono stati premiati, seppur in maniera ridotta, anche i tentativi non riusciti. Per implementare tali regole è stato necessario uno studio approfondito della metodologia di gestione degli eventi in Moodle. In verità nella versione a pagamento del plug-in tali regole possono essere implementate in maniera più agevole. Un aspetto fondamentale della competizione risiede nel fatto che al termine di ogni attività si riceve un punteggio e non un voto, per evitare di suscitare ansia o frustrazione, spesso legati alla votazione in classe.



Rank	Level	Participant	Total	Progress
1	7	Eleanor Shelton	301 <sup>XP</sup>	22 <sup>XP</sup> to go
2	6	Austin Harris	231 <sup>XP</sup>	17 <sup>XP</sup> to go
3	6	Samu Kokko	188 <sup>XP</sup>	60 <sup>XP</sup> to go
4	5	Enola Noel	140 <sup>XP</sup>	46 <sup>XP</sup> to go
5	4	Joris Robert	100 <sup>XP</sup>	34 <sup>XP</sup> to go

Figura 1 - immagini tratte dal plug-in Level up!



**Regole degli eventi**  
Observe actions and attribute points to students as they perform them.

+ Aggiungi regola

5 i punti esperienza sono ottenuti quando:

Almeno una delle condizioni è vera

- + L'evento è Scelta: Inserimento scelta
- + L'evento è Forum: Creazione sottoscrizione della discussione
- + L'evento è Forum: Creazione sottoscrizione
- + Nome dell'evento contiene assessable\_submitted
- + Nome dell'evento contiene assessable\_uploaded

+ Aggiungi condizione

Figura 2 - immagini tratte dalla sezione regole del plug-in Level up!



▼ Esercizio 1

QPy 1

Acquisisci il punteggio

Condizioni per l'accesso: Aver ottenuto o superato la valutazione richiesta in QPy 1

Figura 3 - immagini tratte dalla sezione degli esercizi

Come si evince dalle figure 2 e 3, il plugin Level up! assegna 5 punti a chi effettua “la scelta” che, a sua volta, ha come condizione per l’accesso l’aver ottenuto o superato la valutazione richiesta nell’esercizio in cui entra in gioco il plugin coderunner. Questo piccolo stratagemma consente di assegnare il punteggio a chi riesce a ottenere una valutazione positiva nello svolgere l’esercizio di programmazione.

### 3.3 CODERUNNER

Il plugin coderunner consente di creare un ambiente di sviluppo, con editor e compilatore o interprete, integrato in Moodle, almeno per i linguaggi di programmazione più diffusi. Il plugin è stato creato da Richard Lobb (University of Canterbury, New Zealand) e Tim Hunt (The Open University, UK). La piattaforma Moodle con la sua installazione si trasforma in un laboratorio di Informatica. Si possono assegnare esercizi di programmazione di difficoltà variabile e progressiva, che possono essere svolti da tutti gli studenti, con qualsiasi dispositivo personale, purché abbia un browser. Lo studente è chiamato a realizzare praticamente l’esercizio di coding, potendo far verificare anche molte volte la propria soluzione dalla piattaforma stessa, in modo da ottenere un feedback che gli consentirà di essere guidato alla soluzione ottimale per tentativi successivi. Se lo studente svolge correttamente l’esercizio guadagna, grazie all’integrazione dei due plugin, punti utili a scalare la graduatoria. L’attività quiz di Moodle, installando coderunner, si arricchisce di un nuovo tipo di domanda, denominato appunto CodeRunner. Scegliendo il tipo di domanda CodeRunner si può mettere in un quiz un quesito che chieda di scrivere del codice in uno fra molti linguaggi di programmazione disponibili: C, C++, Java, Nodejs, Octave, Pascal, PHP, Python e Sql. Nelle figure seguenti sono riportate le schermate relative ad uno degli esercizi creati mediante Coderunner: nella prima è riportato il testo dell’esercizio da svolgere, nella seconda un possibile svolgimento corretto e nella terza uno dei test cases che il plugin utilizza per testare la bontà della soluzione proposta dallo studente.

The screenshot displays the Moodle CodeRunner interface for a Python question. The question text asks for a program that reads the number of rows and columns, then reads elements row by row to form a matrix. The answer field contains a Python script that implements this logic. The test cases section shows a table with the following data:

Test case	Standard Input	Expected output
Test case 1	2 2 1	Inserisci il numero di righe: 2 Inserisci il numero di colonne: 2 Inserisci l'elemento della matrice: 1 Inserisci l'elemento della matrice: 2 Inserisci l'elemento della matrice: 3 Inserisci l'elemento della matrice: 4 La matrice inserita è: 1 2 3 4

Figura 4 - Immagini tratte dalle impostazioni della domanda Coderunner

Quando lo studente trova una domanda di questo tipo nel suo quiz, ha a disposizione una finestra per digitare il codice nella quale deve scrivere la sua soluzione (figura 5). Cliccando sul tasto “verifica risposta” viene mandato in esecuzione il codice scritto dallo studente e il suo output viene confrontato con i risultati attesi, lo studente riceve quindi un feedback (Figura 5) che gli consente di confrontare quello che ha ottenuto (Got) con il risultato atteso (Expected). Se non appare immediatamente evidente la differenza tra il risultato atteso e quello che lui ha ottenuto con il suo programma, può cliccare sul tasto “Show differences”.

The screenshot shows a Moodle question interface. On the left, there's a sidebar with 'Domanda 1', 'Risposta corretta', 'Punteggio max.: 10,00', and buttons for 'Contrassegna domanda', 'Modifica domanda', and 'v2 (ultima versione)'. The main area contains the question text: 'Creare in Python il programma per mostrare a schermo una serie di numeri. In particolare il programma deve: chiedere all'utente di inserire un numero intero mostrare a schermo tutti i numeri compresi tra 1 ed il numero inserito dall'utente'. Below this is a 'For example:' section with a table:

Input	Result
5	Inserisci un numero intero: 5 1 2 3 4 5

Below the table is a code editor with the following code:

```

1 num1 = int(input("Inserisci un numero intero: "))
2 for i in range (1,num1-1,1):
3     print(i)

```

At the bottom, there are two comparison tables. The top one is green and shows a positive feedback (green checkmark) for the input '5', where the 'Expected' and 'Got' columns both show the same output. The bottom one is red and shows a negative feedback (red X) for the input '5', where the 'Expected' column shows the same output but the 'Got' column shows a different output (missing the number 5).

Figura 5 - Finestra per digitare il codice, esempio di feedback positivo e negativo

## 4 RISULTATI

### 4.1 Punti di forza e di debolezza

L'esperienza di didattica dell'informatica ha mostrato risultati complessivamente positivi, evidenziando come l'approccio adottato favorisca un maggiore coinvolgimento degli studenti e una partecipazione più consapevole alle attività proposte. In particolare, gli alunni hanno dimostrato un interesse crescente verso i contenuti, accompagnato da un miglioramento nella comprensione dei concetti fondamentali e da una più solida capacità di applicarli in contesti pratici. La metodologia ha inoltre contribuito a creare un ambiente di apprendimento dinamico e collaborativo, in cui il feedback immediato e la possibilità di interazione continua hanno rafforzato la motivazione e la coesione del gruppo classe. La competizione che si è creata grazie al plugin Level up! ha stimolato gli alunni a mettersi in gioco per superare gli altri programmatori e raggiungere l'agognato premio che consisteva in un uovo di cioccolata.

Naturalmente non mancano alcune criticità: la gestione tecnica dei plug-in, la necessità di un monitoraggio costante da parte del docente e l'impegno richiesto per progettare attività equilibrate con due gruppi così eterogenei, rappresentano aspetti che richiedono tempo, competenze e una pianificazione accurata. Tuttavia, i risultati ottenuti indicano che tale investimento è ampiamente ripagato in termini di coinvolgimento e di efficacia formativa.

### 4.2 La voce degli alunni

Per valutare la percezione degli utenti sul percorso è stato somministrato un questionario di valutazione, uno strumento semi-strutturato composto da domande a risposta multipla e aperte per raccogliere suggerimenti. I risultati raccolti nel questionario sono positivi e quindi, incoraggianti. Gli studenti hanno molto apprezzato i due corsi richiedendo all'Istituto di organizzare un ulteriore corso di programmazione in Python più avanzato. Per il nuovo corso ho utilizzato le stesse modalità organizzative dei due corsi base.

## 5 CONCLUSIONI

L'esperienza descritta ha confermato come l'integrazione di strumenti digitali avanzati e di tecniche di gamification possa rappresentare una strategia didattica efficace per l'insegnamento dell'informatica nella scuola secondaria. L'utilizzo congiunto di Moodle, CodeRunner e Level Up! ha permesso di creare un ambiente di apprendimento stimolante, interattivo e personalizzato, capace di valorizzare la partecipazione attiva degli studenti e di rendere il processo di valutazione più immediato e trasparente. L'approccio sperimentato ha mostrato vantaggi significativi sia in termini di motivazione sia di rendimento: gli studenti hanno reagito positivamente al feedback automatico e immediato e alla dimensione ludica della competizione, percependo l'apprendimento come un percorso di crescita personale più che come una successione di prove valutative. La sfida proposta ha favorito la collaborazione, l'autonomia e il desiderio di migliorarsi, elementi chiave nello sviluppo del pensiero computazionale. In prospettiva, l'esperienza dell'Altair potrà essere estesa e consolidata attraverso la creazione di moduli didattici progressivi, integrando ulteriori strumenti di analisi automatica e di apprendimento adattivo. L'obiettivo è continuare a sperimentare metodologie che rendano l'informatica non solo una disciplina da apprendere, ma un linguaggio attraverso cui comprendere e interpretare il mondo digitale.

### Riferimenti bibliografici e sitografici

- [1] Barberis G. Il laboratorio virtuale di coding per una didattica dei linguaggi di programmazione efficace anche a distanza. *Bricks* n.3, (2022), pp. 86-100.
- [2] Lepeltak J. *In molti paesi europei il pensiero computazionale/coding diventa una materia scolastica obbligatoria. Bricks n.1, (2019), pp. 155-157.*
- [3] Coccorullo I. Gamification in un liceo scientifico di Roma utilizzando Moodle - M4Edu. *Atti del MoodleMoot Italia*, (2019), pp. 59-63.
- [4] *Altair8800*, [https://it.wikipedia.org/wiki/Altair\\_8800](https://it.wikipedia.org/wiki/Altair_8800)