

**DALLA GENERAZIONE
AL GRADING:**
*UNA PIPELINE BASATA SU GENAI E
CODERUNNER PER LA VALUTAZIONE
DELLE COMPETENZE IN
PROGRAMMAZIONE 2*

Marzio Pennisi, Andrea Bussi, Roberto Pinna
DiSIT, Università del Piemonte Orientale

Background



Il Collo di Bottiglia (Docenti)

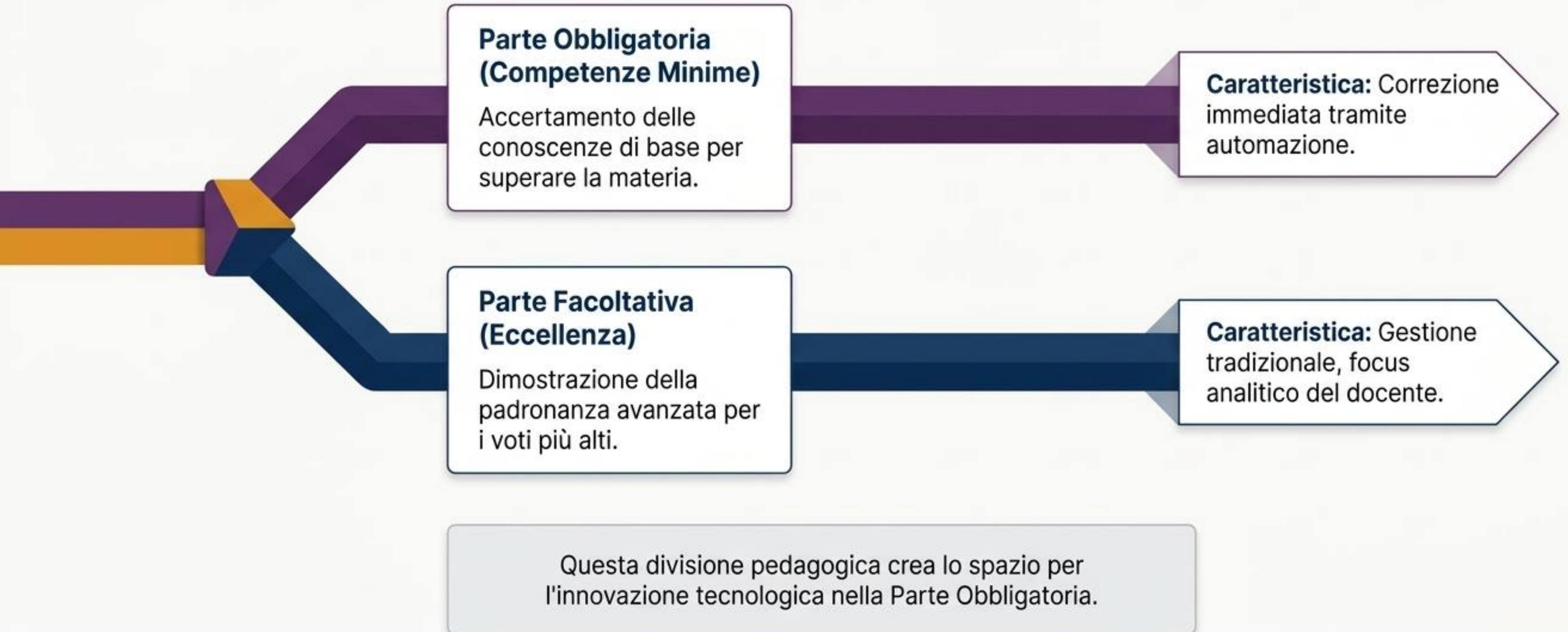
Crescente numero di studenti in Informatica vs. necessità di rilasciare esiti tempestivi. Il carico di correzione manuale genera ritardi fisiologici.

L'Impatto Psicologico (Studenti)

I tempi dilatati di correzione causano disguidi amministrativi e un forte aumento dell'ansia da prestazione nell'attesa dell'esito.

Il ritardo nella valutazione compromette sia l'efficienza didattica che il benessere degli studenti.

Soluzione Proposta



Soluzione Proposta

Parte Obbligatoria (Competenze Minime)

Accertamento delle conoscenze di base per superare la materia.

Caratteristica: Correzione immediata tramite automazione.

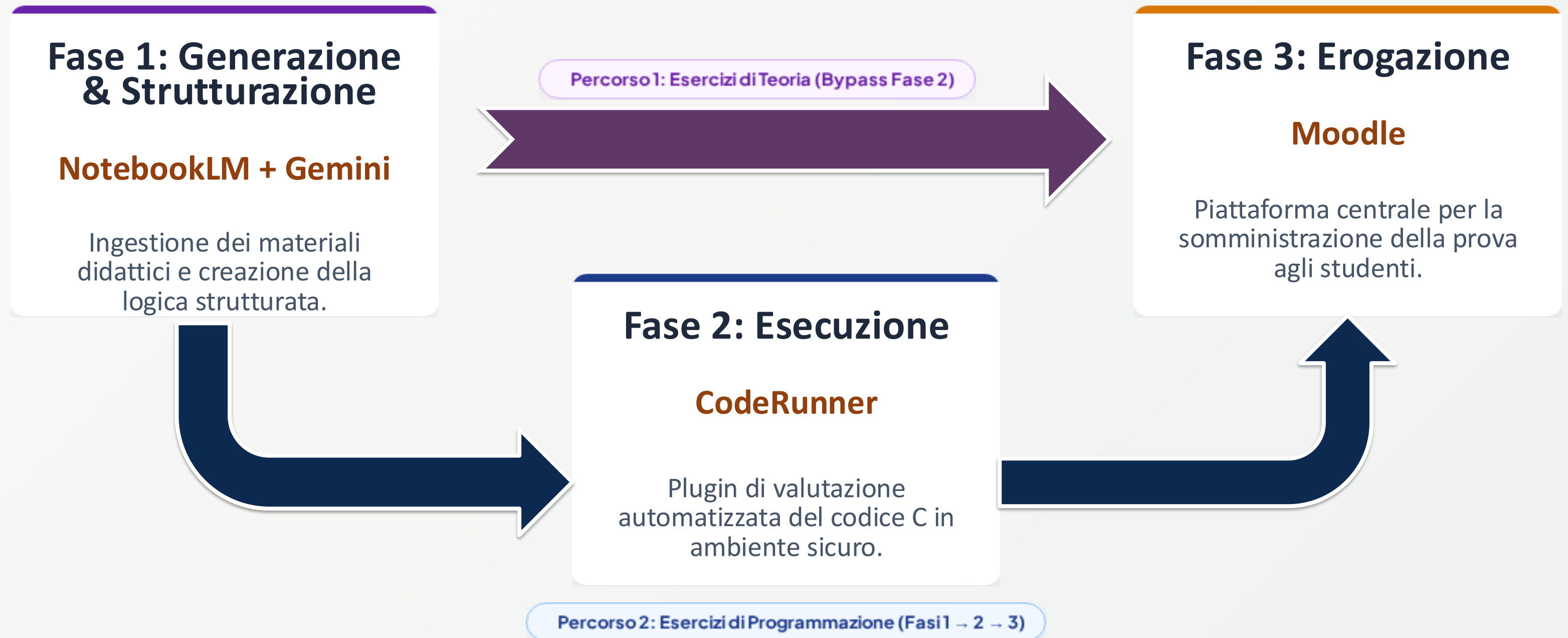
Parte Facoltativa (Eccellenza)

Dimostrazione della padronanza avanzata per i voti più alti.

Caratteristica: Gestione tradizionale, focus analitico del docente.

Questa divisione pedagogica crea lo spazio per l'innovazione tecnologica nella Parte Obbligatoria.

Pipeline



Focus attuale: Automazione della sola Parte Obbligatoria (Competenze Minime).

Esempi di prompt per la creazione degli esercizi di teoria

Crea un insieme di domande di teoria a risposta multipla, in un formato tale che possano essere utilizzati come esercizi su moodle. Includi gli algoritmi di ordinamento trattati, la torre di Hanoi, cenni su pile e code, domande sulla complessità, e eventuali altri quesiti interessanti che trovi sulle slide. Includi sempre l'opzione nessuna delle precedenti.

Crea 20 quesiti in cui lo studente deve individuare la riga mancante di un pezzo di codice delle funzioni viste a lezione (includi la funzione nel testo). Prendi in esame le funzioni più semplici ma fondamentali, come inserimenti in testa in coda o dopo valore, in iterazione e in ricorsione, cancellazioni in testa o in coda o dopo un elemento a valore, e così via.

Fase 2: Strutturazione e Formattazione (Gemini)

Conversione dei quesiti concettuali nel formato nativo idoneo al deposito domande di Moodle.

Gemini

Motore di strutturazione

Formato XML

Strutturazione degli esercizi nel formato nativo del plugin CodeRunner.

Soluzioni di Riferimento

Generazione del codice sorgente esatto per ogni esercizio.

Test Cases

Definizione di batterie di test a copertura delle diverse partizioni di equivalenza.

- Realizzazione di template di codice da utilizzare per la corretta esecuzione degli esercizi
 - *Lo studente si occupa solo di definire la funzione richiesta in iterazione o ricorsione*
- Identificazione di procedure per la verifica della corretta implementazione in iterazione e ricorsione

Welcome to Moodle CodeRunner

CodeRunner is a free open-source question-type plug-in for Moodle that can run program code submitted by students in answer to a wide range of programming questions in many different languages. It is intended primarily for use in computer programming courses although it can be used to grade any question for which the answer is text. Experienced users can also present users with questions formatted in HTML with additional JavaScript; almost any sort of question can be accommodated using this facility (see [the Html UI](#)).

CodeRunner is normally used in Moodle's *adaptive* quiz mode; students paste in their code in answer to each programming question and get to see their test-case results immediately. They can then correct their code and resubmit, typically for a small penalty.

For example:

Question 1

Correct

Mark 1.00 out of 1.00

Flag question

Write a Python3 function `sqr(n)` that returns the square of its numeric parameter `n`.

For example:

Test	Result
<code>print(sqr(-3))</code>	9
<code>print(sqr(11))</code>	121

Answer:

```
1 def sqr(n):
2     return n * n
```

Check

Test	Expected	Got	
<code>print(sqr(-3))</code>	9	9	✓
<code>print(sqr(11))</code>	121	121	✓
<code>print(sqr(-4))</code>	16	16	✓
<code>print(sqr(0))</code>	0	0	✓

Passed all tests! ✓

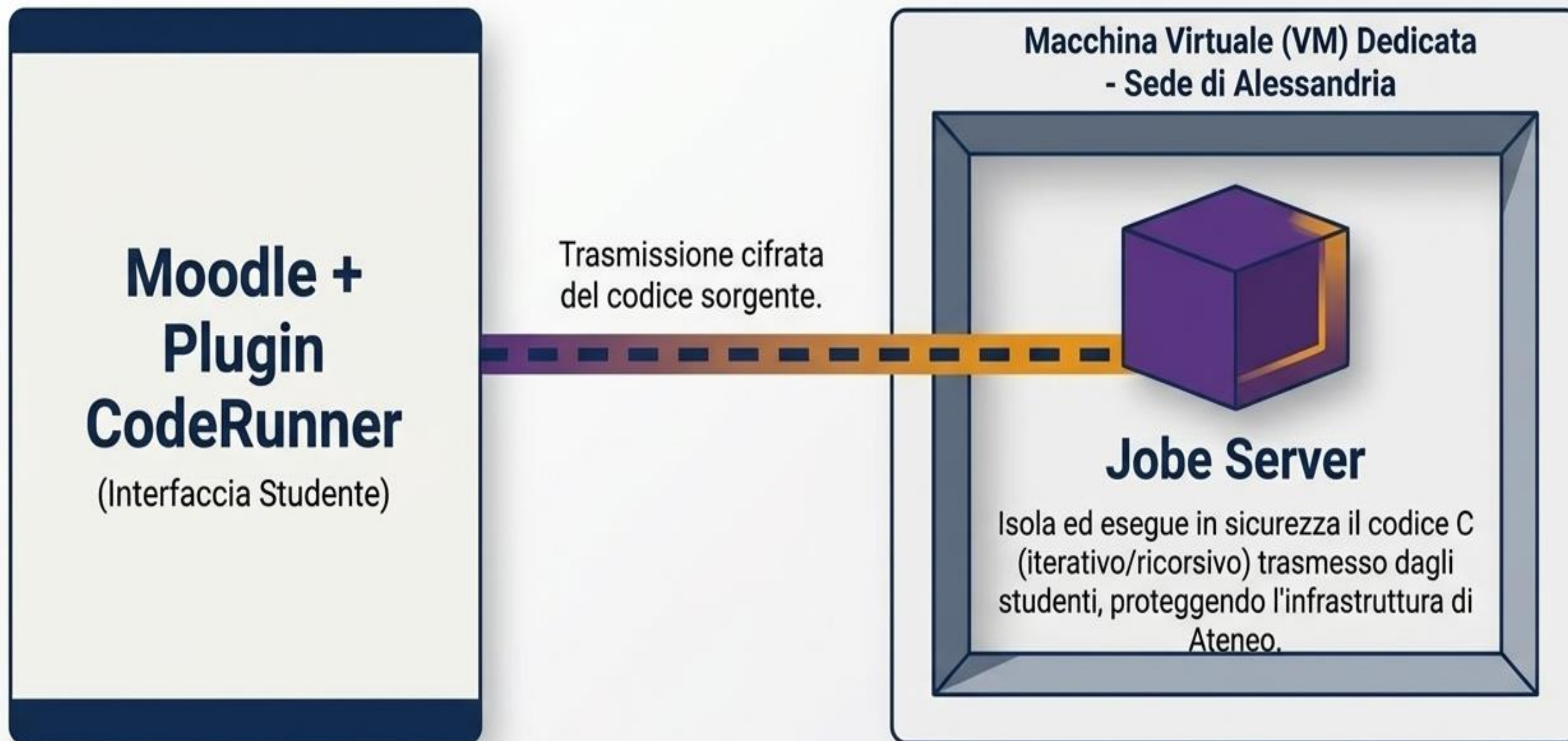
Il Paradigma Human-in-the-Loop

L'Intelligenza Artificiale propone, il Docente dispone.



Il controllo critico rimane in ogni fase nelle mani del docente, garantendo il massimo rigore metodologico.

L'Ambiente di Esecuzione



La Sperimentazione sul Campo



Sessione Estiva

Anno Accademico
corrente. Distribuita su
due appelli d'esame.



~50 Studenti

Coinvolti attivamente
nella sperimentazione live.



2 Sedi Universitarie

Esecuzione simultanea
presso i poli di
Alessandria e Vercelli.



Programmazione 2

Valutazione su
Linguaggio C e gestione
avanzata delle liste.

Risultati

Compilazione in tempo reale fluida e senza interruzioni.

0

**Criticità Tecniche
o di Sistema.**

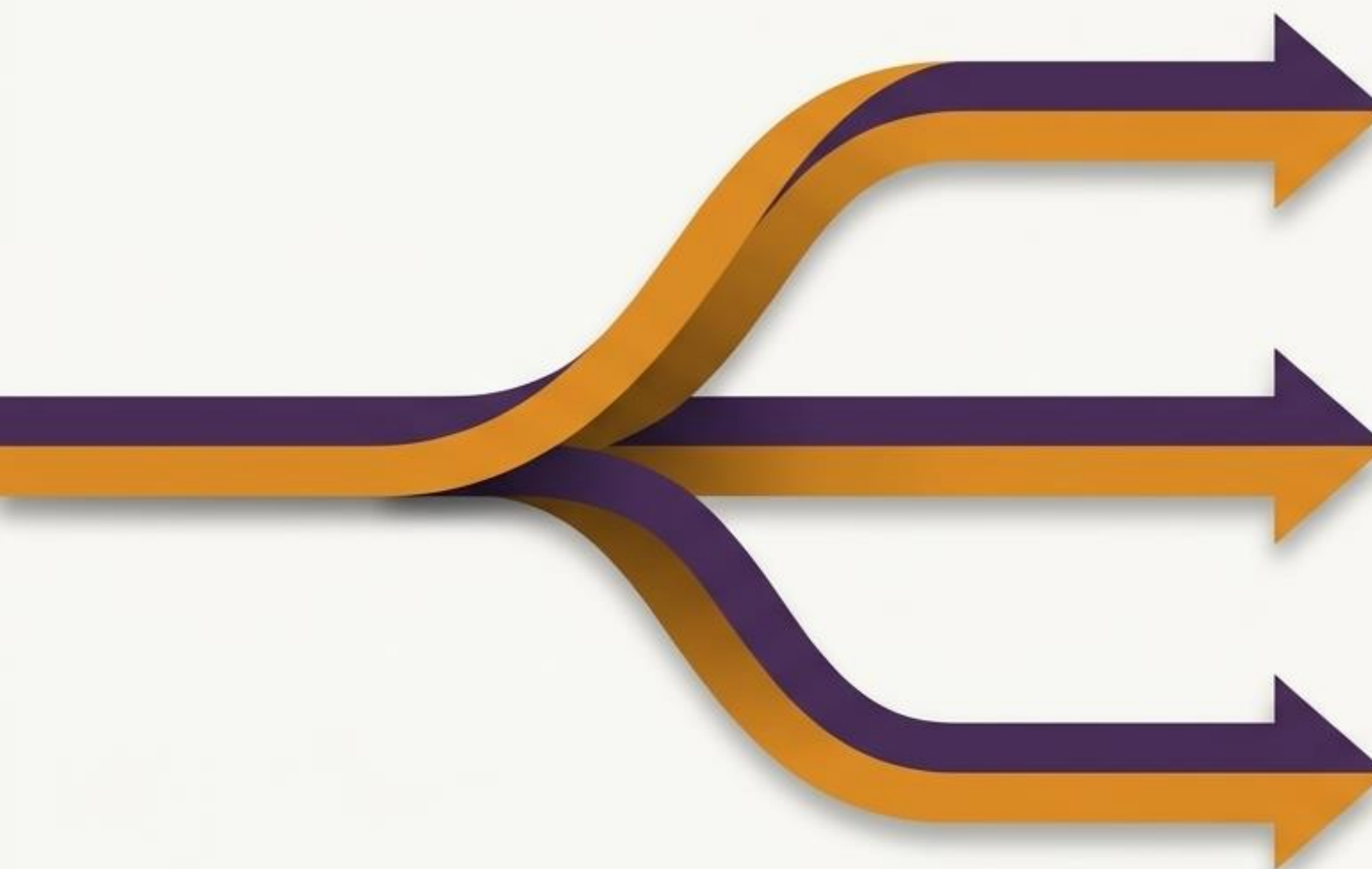
Esecuzione del codice da parte dei 50 studenti avvenuta in totale sicurezza e stabilità.

La complessa pipeline (**Moodle -> VM -> Jobe -> Moodle**) si è dimostrata **invisibile** e **perfettamente stabile** per l'utente finale.

Riduzione dell'Ansia: La consapevolezza in tempo reale di aver superato le competenze minime azzerava l'ansia dell'attesa.

Serenità Operativa: Chi affronta la parte facoltativa dispone di maggiori risorse cognitive per dimostrare le proprie competenze.

Sviluppi Futuri



Automazione dell'Eccellenza

Passaggio dalla sola parte obbligatoria alla pre-valutazione (semi-automatica) della parte facoltativa tramite test cases avanzati.

Valutazione Ibrida

Fornire al docente una base **quantitativa istantanea** per supportare e velocizzare la revisione qualitativa del codice complesso.

Adozione di Ateneo

Estendere la pipeline ad altri insegnamenti, spinti dall'interesse suscitato per l'azzeramento dei tempi di attesa.