

# Ottimizzare l'apprendimento collaborativo con il workshop di Moodle

Giuliana Barberis<sup>1</sup>

<sup>1</sup>Liceo Scientifico M. Curie di Pinerolo  
giuliana.barberis@gmail.com

## Abstract

Nell'insegnamento dell'informatica è quasi automatico l'utilizzo di un LMS come Moodle. Con Moodle gli studenti possono avere una traccia delle lezioni svolte fino a quel momento e l'insegnante ha a disposizione una serie di attività con le quali arricchire la propria offerta formativa.

Una delle attività di Moodle, il Workshop, offre un ambiente organizzato e automatico per la gestione del processo di apprendimento e valutazione fra pari, semplificando le operazioni di creazione e utilizzo di una griglia di valutazione, di assegnazione ai "pari" dell'esercizio da valutare, di calcolo della valutazione quantitativa.

## 1 Introduzione

Nel processo di insegnamento è importante suddividere l'argomento complessivo che ci si propone di trattare in tanti sotto-obiettivi di apprendimento chiari e raggiungibili.

Ciascun sotto-obiettivo deve essere compreso dagli studenti, anche per quanto riguarda i descrittori dei risultati attesi, che devono pertanto essere comunicati prima di cominciare la spiegazione, in seguito dovrà essere programmato un intervento valutativo, che chiuda la fase e consolidi i risultati raggiunti.

Il momento della valutazione è molto importante, soprattutto se consente allo studente di ricevere feedback personalizzati e immediati e se gli dà la possibilità di capire cosa non ha compreso dell'argomento appena concluso.

Per rendere questa fase ancora più significativa l'insegnante può organizzare un processo di peer-assessment, che da una parte obbliga lo studente a considerare in maniera critica soluzioni diverse dalla propria, dall'altra gli permette di ottenere commenti dei propri pari immediatamente dopo la consegna del proprio esercizio.

Imparare a valutare in peer-assessment induce gli studenti ad attivare abilità cognitive di alto livello come quelle di analisi, comparazione, riflessione, metacognizione.

Si ottiene così un triplice risultato:

- Ogni studente ottiene un feedback immediato al proprio lavoro, circostanziato e completo.
- Gli studenti sono stimolati a leggere e cercare di capire i percorsi logici di un pari che hanno condotto alla soluzione, certamente diversi dai propri, e acquisiscono così un'abitudine a leggere in maniera critica il codice di un programma.
- I più bravi, che fanno pochi errori nei propri programmi, imparano a correggere errori anche molto subdoli, che loro non avrebbero fatto ma dei quali è bene sapere come difendersi. I meno bravi si impadroniscono di tecniche che non avrebbero saputo progettare da soli e, dal confronto, sono motivati a cercare di superare i propri limiti.

## 2 L'importanza di imparare a interpretare programmi scritti da altri.

Per chi insegna coding, c'è una motivazione in più per attivare la valutazione fra pari per almeno qualche modulo di apprendimento.

Se suddividiamo il ciclo di vita del software in fasi (studio di fattibilità, analisi preliminare, analisi, codifica, test, manutenzione) ci rendiamo conto che la fase di scrittura di un programma non rappresenta che una parte, neppure troppo significativa, del ciclo complessivo, ciò significa che è assolutamente necessario che il codice sia ben strutturato, le variabili abbiano dei nomi chiari, l'algoritmo non faccia percorsi tortuosi, in modo da semplificare il compito a chi dovrà poi occuparsi della fase di manutenzione.

E' innegabile che ogni programmatore, e quindi ogni studente, sviluppi una propria tecnica personale di coding, ma questa caratterizzazione non deve in alcun modo inficiare la leggibilità e dunque la manutenibilità del software. Questa esigenza non è però immediatamente chiara a chi sta ancora imparando e non riconosce questa peculiarità come una proprietà importante a cui badare.

Chi programma è naturalmente orientato a fare sì che il software funzioni in tutte le condizioni, se nel corso della fase di test si evidenziano degli errori, le modifiche necessarie a correggere i malfunzionamenti dovrebbero essere integrate nella struttura del programma, invece molto spesso vengono aggiunte in modo ingarbugliato e frettoloso.

Strutturare bene un programma richiede attenzione, precisione e tempo, bisogna che si acquisisca la consapevolezza che non si tratta di tempo perso.

Tale consapevolezza si può acquisire dovendo valutare il programma di un'altra persona, si prende così coscienza dei percorsi tortuosi che dovrebbero essere evitati e della mancanza di chiarezza nella scelta dei nomi delle variabili, nella progettazione degli oggetti, nella suddivisione in funzioni.

Allo studente che ha valutato il codice di un altro studente si dovrebbe chiedere di esprimersi anche in merito alle caratteristiche di manutenibilità, dando un punteggio alla leggibilità e alla facilità di intervento per aggiungere o modificare funzionalità. In modo da aiutare gli studenti a riconoscere le peculiarità del software di qualità.

Bisogna ricordare che il coding non si esaurisce nella progettazione dell'algoritmo e conseguente scrittura del codice, ma è molto importante anche la fase di test, e che esiste un test di tipo statico e uno di tipo dinamico. Il test di tipo statico è un test formale e teorico che non richiede che il programma venga mandato in esecuzione su dei dati di prova, bensì bisogna leggere e interpretare i percorsi del programma in modo critico, controllando che non siano presenti errori, che siano stati rispettati gli standard di qualità, che non siano possibili eventuali miglioramenti per quanto riguarda l'impegno macchina o l'impiego della RAM.

È difficile per chi ha scritto il codice essere critico e condurre un test statico veramente obiettivo, ed è anche difficile all'inizio del percorso di apprendimento riconoscere l'importanza di questa fase, per questo è molto importante introdurre nel processo di apprendimento diversi momenti in cui chi impara a programmare è costretto a leggere programmi scritti da altri in modo critico. E questi programmi non devono essere perfetti, al contrario, devono poter essere migliorabili perché sia stimolato il pensiero critico di chi legge, programmi di questo tipo li abbiamo sottomano e sono quelli scritti da altri studenti della classe, ognuno con il proprio stile.

Fagan, un ingegnere elettronico dell'IBM, introdusse già nel suo articolo del 1976 [1] le regole del processo della Fagan Inspection.

Il processo di ispezione del codice introdotto da Fagan consiste nell'esame del programma ad opera di almeno due programmatori diversi dall'autore, secondo una check-list concordata. A valle di questa fase gli "ispettori" e l'autore si devono riunire alla presenza di un moderatore e discutere i difetti riscontrati in modo da condividere le soluzioni, l'autore dovrà poi effettuare le modifiche al codice così come concordato.

Questa tecnica è utile anche nella fase di revisione del codice professionale, si tratta della code review, cioè in quella fase nella quale viene predisposta la nuova versione da pubblicare, aggiornata e corretta.

Naturalmente le tecniche di revisione del codice si sono evolute, ma applicare la Fagan Inspection, almeno in ambito didattico, è ancora importante perché consente ai programmatori “ispettori” e all’autore di condividere conoscenze, imparare nuove tecniche e riconoscere l’importanza di lavorare in team, sperimentando un’importante soft-skill.

In sostanza possiamo utilizzare le tecniche della Fagan Inspection nella didattica del coding, trasformandola nella valutazione fra pari, che avrà quindi come obiettivi l’esame delle competenze di accuratezza e consapevolezza nella scrittura di programmi, la determinazione del grado di aderenza agli standard di codifica e la definizione della capacità di rispettare le scadenze nella consegna del proprio lavoro.

### 3 Funzionamento dell’attività Workshop di Moodle.

Facendo una ricerca sulle più attuali tecniche valutative di peer-assessment, mi sono imbattuta nell’articolo “IMPROVe: sei principi research-based per realizzare attività di valutazione fra pari nei contesti formativi.” [2], dove viene presentato il modello, denominato appunto IMPROVe che si compone di sei principi di pratica.



Figura 1 - Modello IMPROVe

Ciascuna lettera rappresenta uno dei sei principi:

**I**nterpretare insieme i criteri di valutazione: è importante che le regole sulla base delle quali si deve formulare la valutazione siano condivise con gli studenti, e quando gli studenti abbiano già preso parte a una o più attività di peer assessment si potrebbe addirittura pensare di progettare insieme la griglia di valutazione, o di chiedere agli studenti di migliorare quella utilizzata fino a quel momento.

**M**appare gli exemplar, dove gli exemplar sono degli esempi di valutazione effettuati da altri studenti, già predisposti e forniti agli studenti perché possano trarne spunto e capire che cosa ci si aspetta da loro come valutatori.

**R**icevere e **P**rodurre feedback, cioè fornire e ricevere una valutazione descrittiva ed esaustiva degli errori riscontrati ed eventualmente i suggerimenti per la correzione.

**O**ffrire contesti formativi appropriati, in questo nostro caso specifico, ci si riferisce all’**attività Workshop di Moodle**, che rappresenta la modalità tecnica con la quale verrà organizzata ed erogata la fase di peer assessment.

**V**eicolare un nuovo ruolo docente, in effetti, come vedremo, in questo processo il docente assume il ruolo di supervisore, un ruolo quindi diverso dal suo ruolo tradizionale.

Questi sei principi sono centrali anche per quanto riguarda la valutazione fra pari nel coding, ma, essendo il compito di codificare un programma, un task con delle specificità molto particolari, considereremo questi principi, ma applicandoli nel contesto caratteristico del coding. A questo proposito l’articolo “Assessment of programming language learning based on peer code review

model: Implementation and experience report.” [3] propone un metodo molto ben strutturato per l’organizzazione della fase di peer-assessment nel caso del coding, si tratta del Peer Code Review, PCR.

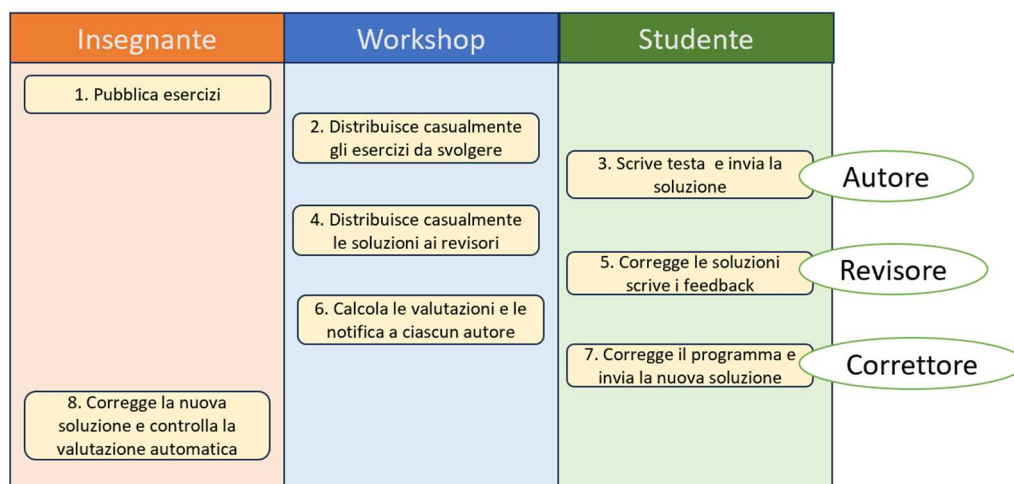
Il modello PCR coinvolge 5 ruoli: autore, revisore, correttore, insegnante e assistente, le caratteristiche di ogni ruolo sono dettagliate in tabella 1:

**Tabella 1 – i ruoli del modello PCR**

autore	Lo <b>studente</b> che scrive il programma
revisore	Lo <b>studente</b> che corregge il programma scritto da un altro studente
correttore	Lo <b>studente</b> che corregge il proprio programma sulla base delle correzioni suggerite dai commenti del revisore
insegnante	La figura che darà la valutazione finale e dovrà garantire il rispetto degli standard organizzativi e di qualità – coordinatore delle attività
assistente	Figura di ausilio alle funzioni più pratiche dell’insegnante (nel nostro caso questo ruolo viene ricoperto dall’ <b>insegnante stesso</b> )

Come si vede lo studente ricopre tre dei ruoli in tabella a seconda della fase del processo di peer assessment in cui si trova.

Il modello PCR descritto nell’articolo, fa riferimento ad uno specifico software per la gestione automatica del processo di peer assessment, si vedrà che l’attività workshop di Moodle costituisce un valido sostituto del software proposto da Wang perché ricopre tutte le funzionalità del software presentato con il PCR.



**Figura 2 – Cronologia e ruoli del modello PCR**

Nella figura 2 sono elencate in verticale le varie attività del processo, ciascuna attività è collocata nella colonna relativa al soggetto che se ne deve occupare: insegnante, studente o Workshop (nel caso di quelle attività che vengono gestite automaticamente da Moodle).

1. L’insegnante pubblica un certo numero di esercizi
2. Il workshop abbinava casualmente un esercizio ad ogni studente
3. lo studente scrive, testa e invia la sua soluzione (in questa fase lo studente è nel ruolo di **autore**)
4. Il workshop assegna casualmente ciascuna soluzione ad un certo numero di revisori (il numero è stabilito dall’insegnante)
5. Ciascuno studente corregge le soluzioni che gli sono state assegnate (anche anonimamente) secondo una griglia di valutazione quantitativa concordata e condivisa (in questa fase lo studente è nel ruolo di **revisore**). Alla fine della compilazione della

griglia viene calcolato e assegnato automaticamente un punteggio al programma, il calcolo dipende dalla griglia di valutazione.

6. Il workshop notifica a ciascun autore le correzioni, i feedback del suo programma e la valutazione complessiva assegnata a questa attività. La valutazione sarà composta in parte dalla media dei punteggi che gli sono stati assegnati dai suoi compagni e in parte dal punteggio che Moodle ha calcolato automaticamente al suo ruolo di valutatore.
7. Lo studente corregge il programma secondo i suggerimenti ricevuti e invia la sua nuova soluzione (in questa fase lo studente è nel ruolo di **correttore**)
8. L'insegnante controlla tutte le correzioni e i commenti e i nuovi programmi inviati e conferma o corregge la valutazione calcolata automaticamente.

L'attività Workshop di Moodle gestisce automaticamente le attività dalla 1 alla 6.

I documenti che vengono prodotti nel modello in definitiva sono: il **codice prodotto** (scritto e testato da un autore), la **griglia di correzione** che comprende i **commenti o feedback** (le idee, i suggerimenti e le critiche che un revisore propone con particolare riguardo agli standard di codifica) e il **codice corretto** (una nuova edizione del programma corretto secondo i commenti dei revisori).

Questo processo di valutazione comprende una certa interconnessione tra i ruoli, bisogna impedire che gli studenti che non rispettano le consegne, come qualità o come tempi, penalizzino i compagni che, nella cronologia del proprio task, dovrebbero partire proprio da tali consegne.

Si può ovviare ai problemi organizzativi che derivano da queste situazioni anomale prevedendo più revisori per una stessa soluzione, in questo modo ciascuno studente dovrà valutare più soluzioni e gli arriverà più di un feedback.

A questo punto l'autore farà una sintesi tra i commenti ricevuti per correggere il proprio codice.

Per evitare l'imbarazzo agli studenti di rilevare gli errori di un compagno di cui conosce l'identità, si può nascondere sia il nome dell'autore che dei revisori, anche se, in qualche sessione di peer assessment, è raccomandabile non nascondere queste informazioni, perché si possa poi organizzare quel meeting di discussione della Fagan Inspection.

Tutte le attività potranno svolgersi in un'unica mattinata, oppure potranno essere diluite nel tempo, a seconda che si tratti di una verifica, di un'esercitazione, o che il problema sia più o meno complesso.

Si può anche non assegnare un unico problema a tutta la classe, in modo da poter far correggere ad ogni studente esercizi diversi da quello che ha svolto come autore.

Il Workshop consente a ciascuno studente l'invio di uno o più file digitali (soluzione di un problema), sono queste le consegne che vengono poi ridistribuite casualmente a uno o più compagni revisori.

Le figure che si vedono successivamente corrispondono ad un'attività di peer assessment somministrata in una classe seconda, relativa ad un esercizio sulle strutture complesse in Python.

### Esercitazione in peer Assessment sulle strutture complesse - esercizio temperature

**Consegne aperte:** lunedì, 12 dicembre 2022, 12:30  
**Consegne chiuse:** lunedì, 12 dicembre 2022, 13:34  
**Valutazioni aperte:** lunedì, 12 dicembre 2022, 13:35  
**Valutazioni chiuse:** lunedì, 12 dicembre 2022, 14:15

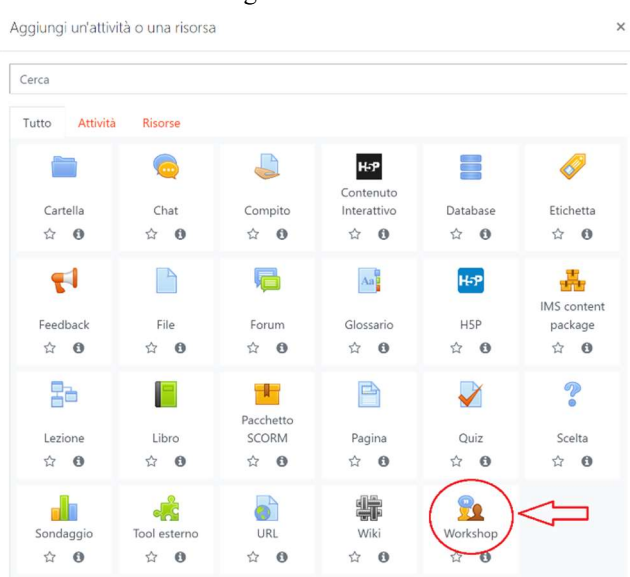
Fine				
Fase di allestimento Vai alla fase di allestimento	Fase di consegna Passa alla fase di consegna	Fase di valutazione Passa alla fase di valutazione	Fase di calcolo dei voti Passa alla fase di calcolo dei voti	Fine Fase attuale
<ul style="list-style-type: none"> <li>✓ Imposta la descrizione del workshop</li> <li>✓ Imposta istruzioni di consegna</li> <li>✓ Modifica scheda di valutazione</li> </ul>	<ul style="list-style-type: none"> <li>✓ Imposta istruzioni per la valutazione</li> <li>✓ Distribuisci consegne attesi: 19 consegnati: 14 da distribuire: 0</li> <li>⌚ Almeno un partecipante non ha ancora consegnato il proprio lavoro.</li> <li>⌚ Inizio consegne da lunedì, 12 dicembre 2022, 12:30 (248 giorni fa)</li> <li>⌚ Fine delle consegne: lunedì, 12 dicembre 2022, 13:34 (248 giorni fa)</li> <li>⌚ Le limitazioni temporali non si applicano a te.</li> </ul>	<ul style="list-style-type: none"> <li>⌚ Aperto per le valutazioni da lunedì, 12 dicembre 2022, 13:35 (248 giorni fa)</li> <li>⌚ Fine valutazioni: lunedì, 12 dicembre 2022, 14:15 (248 giorni fa)</li> <li>⌚ Le limitazioni temporali non si applicano a te.</li> </ul>	<ul style="list-style-type: none"> <li>✗ Calcola la votazione delle consegne attesi: 19 calcolati: 14</li> <li>✗ Calcola la votazione delle valutazioni attesi: 19 calcolati: 14</li> <li>✗ Fornisce una conclusione dell'attività</li> </ul>	

**Figura 3 – Pannello di controllo dell'attività Workshop di Moodle**

La figura 3 raffigura il pannello di controllo dell'attività di workshop una volta che è stata data la chiusura dell'attività.

Il pannello di controllo è fondamentale per la configurazione e il controllo delle fasi del processo di peer assessment, da lì si può controllare l'avvenuta consegna delle soluzioni, eventualmente sollecitare quelli che non sono puntuali, e dare le chiusure di ogni fase per poter passare alla fase successiva.

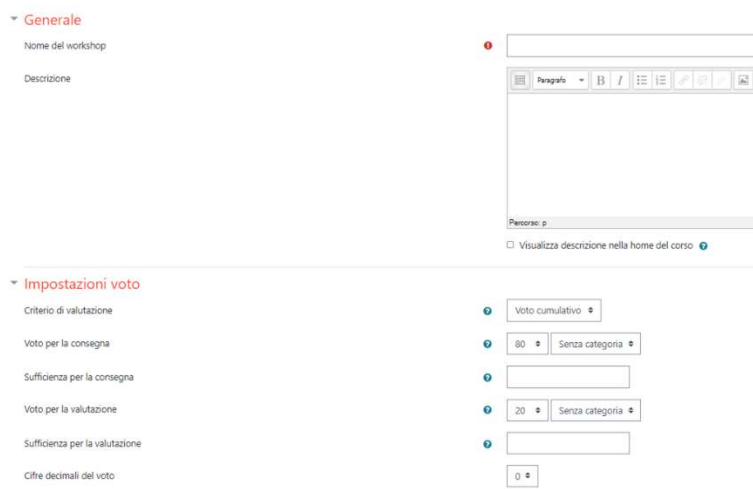
Possiamo vedere per immagini quali sono i parametri che possiamo configurare per realizzare una attività analoga. Innanzitutto nell'immagine successiva si vede come si inserisce l'attività:



**Figura 4 – Inserire l'attività Workshop**

Nell'immagine 5 invece vediamo quali parametri configurare, molti parametri dell'attività Workshop corrispondono ai parametri comuni a tutte le attività di Moodle ("disponibilità", "impostazioni comuni", "condizioni per l'accesso", "tag" e "competenze") ma ci sono parametri caratteristici che si devono definire solo per il Workshop.

Molto importante è la definizione del "criterio di valutazione", che può essere: "voto cumulativo," "commenti", "numero di errori" o "rubric"; per ottenere sia una valutazione "numerica" sia permettere ai revisori la formulazione di feedback testuali bisogna scegliere voto cumulativo.



**Figura 5 – Alcuni parametri dell'attività Workshop**

In questo caso l'insegnante fornisce una griglia di valutazione (che si dovrà inserire successivamente nell'attività Workshop) elencando una serie di item o descrittori e dando ad ognuno un peso e una percentuale di raggiungimento.

Lo studente, rispondendo in modo quantitativo a una serie di domande, ottiene automaticamente la valutazione delle soluzioni che gli sono state assegnate.

Un'altra scelta molto importante è determinare quanta parte del voto sia riservata alla soluzione (ruolo autore) e quanta alla valutazione (ruolo revisore), se si accetta la proposta 80-20, il voto complessivo sarà composto per l'80% dalla valutazione della soluzione, derivante dalla valutazione ricevuta dai compagni, mentre per il 20% sarà "valutata la valutazione" dello studente nel ruolo di revisore.

La "valutazione sulla valutazione" è importante perché induce gli studenti ad un comportamento corretto quando ricoprono il ruolo di revisori, essa viene calcolata con una funzione statistica sulla base di tutte le valutazioni ricevute da ciascun item.

Le "impostazioni della consegna", quelle della "valutazione" e la "descrizione" nella parte generale sono fondamentali per una corretta comprensione del processo, in queste finestre bisogna scrivere le indicazioni per gli studenti in modo che sia chiaro cosa devono fare in ciascuna delle fasi.

Nell'esercitazione assegnata la descrizione del problema comprendeva sia il testo dell'esercizio da svolgere, sia una sintetica indicazione dei goal da raggiungere:

Questa esercitazione è divisa in due parti:  
 nella prima dovrai svolgere e inviare un esercizio sulle strutture complesse  
 nella seconda parte ti saranno assegnate 3 consegne tra quelle dei tuoi compagni  
 e dovrai valutarle, avrai a disposizione una griglia di valutazione che renderà  
 automatico il calcolo della valutazione

**Testo del problema**  
**esercizio temperature**  
 Creare un programma per il caricamento delle temperature medie rilevate in **10 città italiane**:

- caricare le informazioni in una struttura dizionario o in due liste parallele (chiedendo in input ciascuna città e rispettiva temperatura che può non essere un numero intero) **4 punti**  
 visualizzare tutte le informazioni caricate come:  
 città: temperatura  
 città: temperatura  
 .....
- trovare e visualizzare la media delle temperature su tutte le città arrotondata a due cifre decimali **1 pto.**
- trovare e visualizzare la città con la temperatura massima **2 pti**
- visualizzare in un elenco le città con la temperatura più alta della media **2 pti**

**Figura 6 - La descrizione del problema**

Ciascun programma è stato valutato da tre studenti, di conseguenza a ciascuno studente sono arrivati tre feedback diversi, per facilitare la correzione agli studenti valutatori, quando sono stati assegnati i programmi da valutare è stato anche fornito un esempio di soluzione.

La griglia di valutazione viene impostata, descrivendone gli "elementi" o item come in figura 7.

Elemento 1

Descrizione

Il programma "gira" senza errori di sintassi:

- non ha errori di sintassi e parte. 5
- gira correggendo un errore di sintassi 4
- gira correggendo due errori di sintassi 3
- contiene più di due errori di sintassi 2

Persono p

Tipo Punteggio

Punteggio massimo

5

Peso

1

Elemento 2

Elemento 3

**Figura 7 - Impostazione della griglia di valutazione**

Ogni elemento ha una descrizione e gli si deve attribuire un punteggio massimo e un peso nell'ambito della valutazione complessiva.

E' consigliabile fornire il significato di ogni punteggio per facilitare chi valuta, si possono definire quanti elementi quanti si vuole, ma, per non complicare troppo la fase di valutazione è importante che non ci siano troppi item, per ogni item lo studente può scrivere un commento a motivazione del punteggio assegnato (feedback), questa informazione può essere resa obbligatoria, come prescritto da uno dei principi di pratica del modello IMPROVe.

In figura 8 si può vedere come si presenta la Griglia al momento della compilazione.

Scheda di valutazione

**Elemento 1**

Il programma "già" senza errori di sintassi:

- non ha errori di sintassi e parte. 5
- già correggendo un errore di sintassi 4
- già correggendo due errori di sintassi 3
- contiene più di due errori di sintassi 2

Voto per Elemento 1

Commento su Elemento 1

**Elemento 2**

E' stato impostato un ciclo di richiesta di città e temperatura e gli input sono:

- il codice è presente e da i risultati attesi ?
- il codice è presente già ma ha errori di logica (per esempio non carica 10 elementi, non è stata scelta una delle due strutture indicate, i numeri non sono float ...) 5-7 pts
- il codice è presente, completo, ma non gira perché contiene errori di sintassi 3-4
- manca il codice per questa richiesta o è presente solo parzialmente: 1-2

Voto per Elemento 2

Commento su Elemento 2

**Elemento 3**

Le informazioni sono state visualizzate in forma tabellare come richiesto

- il codice è presente e da i risultati attesi 3
- il codice è presente già ma ha errori di logica 2
- il codice è presente ma non gira perché contiene errori di sintassi 1
- manca il codice per questa richiesta: 0

Voto per Elemento 3

Commento su Elemento 3

**Figura 8 - La Griglia di Valutazione**

Una volta che tutti gli studenti abbiano consegnato la propria soluzione (si può impostare un blocco per cui il sistema non accetta consegne in ritardo), l'insegnante apre la fase di valutazione, in questa fase il workshop distribuisce casualmente a ciascuno studente il numero stabilito di file da correggere.

Man mano che gli studenti compilano la griglia di valutazione il pannello di controllo dell'insegnante mostra i progressi del processo. Quando tutti hanno valutato tutto, l'insegnante passa alla fase di calcolo dei voti e ottiene la situazione complessiva come in figura 9, dove si vede chiaramente il calcolo del voto della consegna, che è una media tra i punteggi assegnati da ciascuno studente; nella colonna finale invece si vede la valutazione assegnata allo studente come valutatore.

Report dei voti del workshop

Nome / Cognome	Consegna	Ultima modifica	Voti ricevuti	Voto per la consegna (su 80)	Voti dati	Voto per la valutazione (su 20)
André	Consegna impariale	modificato lunedì, 12 dicembre 2022, 13:28	67 (17) < Sofía 51 (19) < Diego 53 (20) < Beatrice	57	67 (20) > Samuele 51 (17) > Matia 53 (16) > Ilenia	18
Bruno	webby	modificato lunedì, 12 dicembre 2022, 13:28	25 (18) < Sofía 15 (20) < Estelle 28 (17) < Ilenia	23	15 (18) > Estelle 25 (20) > Rakhman 44 (15) > Mihal	18
Sofía	Esercizio temperature	modificato lunedì, 12 dicembre 2022, 13:22	11 (18) < Estelle 17 (20) < Beatrice 17 (20) < Ivan	15	67 (17) > André 25 (18) > Bruno 40 (17) > Rakhman	17
Diego	consegna esercizio temperature	modificato lunedì, 12 dicembre 2022, 13:22	73 (20) < Emanuele 78 (20) < Ivan 114 < Mihal	76	51 (18) > André 35 (20) > Emanuele 16 (19) > Matilde	20
Emanuele	esercizio temperature	modificato lunedì, 12 dicembre 2022, 13:20	35 (20) < Diego 40 (20) < Beatrice 20 (13) < Rakhman	32	73 (20) > Diego 22 (20) > Matilde 31 (17) > Rakhman	19

**Figura 9 - Valutazione finale**

Gli studenti si sottopongono sempre molto volentieri a questo tipo di esercitazioni, soprattutto se lo scopo ultimo è poi quello di sostenere una verifica organizzata in questo modo, il meccanismo è sfidante e stimola la partecipazione e l'impegno.

Dalle discussioni sulla valutazione di questo tipo di attività che ho avuto con gli studenti coinvolti ho rilevato:

1. che preferiscono quando i programmi che devono essere valutati sono anonimi, mentre non è importante che i revisori siano anonimizzati.



2. che è importante che la griglia di valutazione sia stata provata prima in qualche altra esercitazione e che gli item siano condivisi.

Naturalmente l'attività Workshop non è limitata all'applicazione del modello PCR, può essere utile in tutte quelle situazioni in cui si voglia innescare un processo di peer assessment, per esempio in tutti i casi in cui si voglia far valutare agli studenti la presentazione fatta da ciascuno studente/gruppo a tutta la classe.

In questo caso la Griglia di valutazione conterrà gli item secondo i quali ciascuno studente dovrà valutare la presentazione, come esemplificato in figura 10.

**Criteria di valutazione**

Rubrica di Valutazione							
Gruppo valutato							
Gruppo che valuta							
Studente							
Criteria	Indicatori	Punti 1	Punti 2	Punti 3	Punti 4	Punti 5	
Selezione organizzata dei materiali	I contenuti scelti sono significativi rispetto al tema?						scegli un punteggio
Grado di rielaborazione personale	Emerge una rielaborazione personale dei contenuti?						scegli un punteggio
Utilizzo del linguaggio specifico	Viene utilizzato un linguaggio corretto con un'opportuna scelta di termini specifici?						scegli un punteggio
Originalità della soluzione	Sono stati scelti collegamenti non banali per spiegare approfondire i concetti chiave?						scegli un punteggio
Impiego documentato delle fonti	Le fonti sono state dichiarate in modo corretto?						scegli un punteggio
Rispetto delle consegne	Sono stati rispettati i vincoli quantitativi e di corretto funzionamento delle consegne?						scegli un punteggio
Estetica della presentazione	La presentazione ha uno stile curato, organizzato e chiaro.						scegli un punteggio
Punteggio complessivo					Su 35 punti		

**Figura 10 - Griglia di valutazione per il peer assessment di una presentazione**

## Bibliografia

- [1] M. Fagan, «"Design and Code Inspections to Reduce Errors in Program Development."», 1976.
- [2] G. V. Serbati N., «IMPROVe: sei principi research-based per realizzare attività di valutazione fra pari nei contesti formativi.» *Form@re, Firenze University Press*, 2019.
- [3] L. H. F. Y. J. Y. L. Y. Wang Y., «Assessment of programming language learning based on peer code review model: Implementation and experience report.» *Computer & Education, Elsevier*, 2012.