Partiamo dai sistemi di classificazione automatizzati. Edwards e Pérez-Quiñones della Virginia Tech hanno introdotto un sistema di classificazione automatizzato open source, Web-CAT, estensibile e personalizzabile. Molti istruttori di tutto il mondo hanno utilizzato Web-CAT lo hanno utilizzato come sistema di valutazione automatizzato. Web-CAT offre numerosi servizi a studenti e istruttori, come la presentazione di incarichi, feedback automatico basato su casi di test predefiniti, suggerimenti su come aiutare a correggere gli errori nel codice e generare voti in base al rapporto sui casi di test prodotto da JUnit-Reporter. Tuttavia, durante l'utilizzo di Web-CAT presso il Dickinson College, dipartimento di matematica e informatica, in introduzione al corso di programmazione, gli istruttori hanno spesso riscontrato difficoltà indesiderabili dalato degli studenti: gli studenti non riuscivano a capire il feedback generato dopo aver inviato i loro compiti. La maggior parte degli errori era generali e non riflettevano l’errore reale. Dall’altra parte gli istruttori avevano notato che la configurazione fosse complessa e impossibile da accedervi da remoto.

Il problema che abbiamo deciso di occuparci ha una stretta relazione con la modalità d’esame del corso di elementi di programmazione, quindi prima di passare alla sua specifica e successiva risoluzione, analizziamo, innanzitutto, come si svolge attualmente l’esame di questo corso.

## Modalità d’esame

L’esame si tiene di solito nel Laboratorio di Informatica presso l’aulario di Via Michelangelo, sito in Aversa. Ad ogni studente, prenotato all’appello d’esame, è assegnata una postazione pc con un rispettivo codice e password per effettuare il Login alla piattaforma. Gli studenti sono, in genere, liberi di scegliere se utilizzare il proprio calcolatore o quello offerto dall’università, per effettuare le prove.



Figura 20, Laboratorio di Informatica

La consegna finale dell’elaborato e la traccia d’esame prevedono l’accesso obbligatorio alla piattaforma. Ogni compilazione dell’editor dell’ambiente WebDevStudio prevede un salvataggio. Completato il compito, lo studente deve semplicemente effettuare il Logout dalla piattaforma, dopo aver fatto almeno una compilazione su di essa, senza bisogno di dover effettuare ulteriori salvataggi.

## Specifica del problema

Qual è, quindi, il nostro problema?

Il tool utilizzato durante l’esame del corso di elementi di programmazione ha la caratteristica di salvare tutte le versioni del file, anche quelle intermedie: il salvataggio avviene ad ogni compilazione. Ma nonostante ciò lo studente utilizza il tool soltanto per l’ultima compilazione la quale funge da consegna del compito, infatti dopo di essa il ragazzo esegue semplicemente il logout dalla piattaforma. Invece la stesura del codice avviene utilizzando un ambiente di sviluppo diverso, CodeBlocks, affinché possano sfruttare funzionalità non presenti sulla piattaforma come ad esempio il debugger, funzionalità che lo studente è abituato a utilizzare durante le esercitazioni o le prove svolte a casa. In questo modo il docente non ha la possibilità di monitorare le versioni intermedie realizzate dal ragazzo e analizzare più in dettaglio le conoscenze dello studente, capendo quali sono gli errori più frequenti che vengono compiuti durante un esame o per fare un’analisi più intrinseca di quella che potrebbe essere la formazione dello studente.

## Analisi dei requisiti

L’obiettivo prefissato per questo lavoro di tesi è di progettare e sviluppare una soluzione al problema precedentemente esposto. La soluzione pensata prevede attraverso CodeBlocks di:

* Scaricare la traccia del compito attraverso l’URL fornito dal docente. L’URL corrisponde a quello del repository di GitHub con all’interno la traccia del compito da svolgere dallo studente.
* Consegnare il compito attraverso la seconda voce del menù nello stesso repository in cui era presente la traccia dell’esame.

Salvare le varie versioni dell’elaborato ad ogni compilazione di CodeBlocks sfruttando git

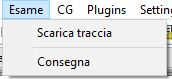


Figura 26, Plugin Esame

Cliccando sulla voce “Scarica traccia” si aprirà la seguente finestra di dialogo, come mostra la figura 26.

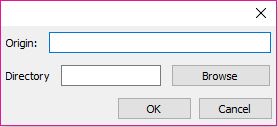


Figura 27, Finestra di dialogo "Scarica traccia"

Nel campo “Origin” bisognerà inserire l’URL del repository GitHub, invece la “Directory” corrisponde a dove l’utente desidererà di salvare il progetto clonato.

Il processo di sviluppo del software prevede l'uso di metodi, tecniche e strumenti, ma ciò non impedisce il verificarsi di errori nel prodotto in fase di sviluppo. Al fine di ridurre questi errori, ci sono VV&T - attività di verifica, convalida e test del software, che vengono utilizzate durante il ciclo di sviluppo del software. La verifica è il processo per identificare se la costruzione del software viene eseguita nel modo giusto. La convalida comporta la valutazione se il software sviluppato soddisfa i requisiti proposti dal cliente. L'attività di test prevede la scoperta della presenza di difetti, se presenti.

L'attività di test dovrebbe essere presente durante tutto il ciclo di sviluppo e prevede le seguenti fasi di test:

1. unit test, in cui vengono testate le unità più piccole del software, come procedure, funzioni o metodi e classi;

2. test di integrazione, in cui viene testata l'integrazione tra moduli, componenti o altre forme di unità del software;

3. test di sistema, in cui il sistema viene testato nel suo insieme. I test applicati in questa fase includono stress test, test di sicurezza, test di carico, tra gli altri.

Oltre a queste tre fasi, esiste anche il test di regressione eseguito quando il programma viene modificato ed è necessario garantire che i test precedenti siano ancora applicati e anche testare i nuovi requisiti introdotti dalla modifica.

Il problema della valutazione automatica e semiautomatica è stato evidenziato anche in passato ed è stata suggerita, negli ultimi anni, una notevole quantità di lavoro innovativo per superare il problema. Gli strumenti possono essere controllati e utilizzati attraverso Internet. Il docente e gli studenti possono essere in qualsiasi parte del mondo e possono comunicare liberamente attraverso il sistema. Il sistema aiuta a rendere vivo lo scenario di apprendimento a distanza. Questo rappresenta un miglioramento cruciale per qualsiasi grande istituto o università

GitHub Classroom, in primo luogo, dà la possibilità di creare automaticamente repository per studenti, monitorare i compiti nella dashboard e valutare quest’ultimi con strumenti di valutazione automatica grazie all’integrazione con strumenti di terze parti come i test automatizzati, risparmiando, come abbiamo già osservato, molto tempo al docente che può essere utilizzato a favore degli studenti.

Il funzionamento di GitHub Classroom consente agli studenti di accettare un compito con un solo link, in modo da poter accedere direttamente al materiale che sia per un esame o per un’esercitazione. Oppure il docente ha la possibilità di invitare gli studenti facenti parte di un gruppo ad un archivio condiviso con all’interno gli elementi da utilizzare per uno scopo specifico fissato dal docente. Gli studenti pubblicheranno il proprio elaborato e il docente quando vuole valutarli ha l’opportunità di scaricare i compiti direttamente sul proprio computer da GitHub Classroom.

Nel caso si tratti di un’esercitazione GitHub Classroom può essere utile anche per i seguenti motivi: l’insegnante può vedere quando gli studenti accettano il compito e può accedere in qualsiasi momento al loro lavoro. Questo può essere conveniente per osservare a che punto si trovano gli studenti grazie al controllo di versione o addirittura aiutarli a risolvere problemi nel caso incontrassero difficoltà.

Il tool utilizzato durante l’esame del corso di elementi di programmazione ha la caratteristica di salvare tutte le versioni del file, anche quelle intermedie: il salvataggio avviene ad ogni compilazione. Ma nonostante ciò lo studente utilizza il tool soltanto per l’ultima compilazione la quale funge da consegna del compito, dopo di essa il ragazzo esegue semplicemente il logout dalla piattaforma. Invece la stesura del codice avviene utilizzando un ambiente di sviluppo diverso, CodeBlocks, affinché possano sfruttare funzionalità non presenti sulla piattaforma come ad esempio il debugger, funzionalità che lo studente è abituato a utilizzare durante le esercitazioni o le prove svolte a casa. In questo modo il docente non ha la possibilità di monitorare le versioni intermedie realizzate dal ragazzo e analizzare più in dettaglio le conoscenze dello studente, capendo quali sono gli errori più frequenti che vengono compiuti durante un esame o per fare un’analisi più intrinseca di quella che risulta essere la formazione dello studente.

Ad ogni studente, prenotato all’appello d’esame, è assegnata una postazione pc con un rispettivo codice e password per effettuare il Login alla piattaforma

Spesso però lo studente in fase di esame non utilizza il tool messo a disposizione dal docente ma preferisce utilizzare CodeBlocks con il quale ha molta più familiarità grazie al maggior numero di funzionalità da esso offerte come il debugger e utilizza la piattaforma solo per l’ultima compilazione che funge da consegna.

Quindi dov’è che nasce il problema?

Il tool offerto durante l’esame del corso di elementi di programmazione ha la caratteristica di salvare tutte le versioni del file, anche quelle intermedie: il salvataggio avviene ad ogni compilazione. Questo aspetto è molto utile al docente per effettuare indagini e comprendere al meglio lo studente e la sua formazione ma spesso non viene sfruttato poiché la stesura avviene sfruttando CodeBlocks. Il risultato di questa scelta quindi comporta il mancato salvataggio delle versioni presenti nel mezzo e l’impossibilità da parte del docente di svolgere analisi intrinseche e dettagliate della classe.

Obiettivo di questa tesi è progettare e sviluppare un’estensione dello strumento che lo studente utilizza durante le sue esercitazioni. Attraverso questo strumento è consentito, all’allievo, partecipare alla prova d’esame grazie ad un’integrazione con il sistema di consegna e valutazione utilizzato dal docente.