

Implementazione di un tool per il supporto alla valutazione automatica di progetti su GitHub

Corso di Laurea Triennale in Informatica
Tesi di Laurea

Laureando:
Matteo Azzarelli

ANNO ACCADEMICO 2017/2018



**Università
degli Studi
di Perugia**

- 1 Introduzione
- 2 GitHub
- 3 Lavori Correlati
- 4 Studio ed Implementazione
- 5 Conclusioni

Perché abbiamo sviluppato questa applicazione?

Perché abbiamo sviluppato questa applicazione?

- 1 Dissuadere gli studenti dal plagio.

Perché abbiamo sviluppato questa applicazione?

- 1 Dissuadere gli studenti dal plagio.
- 2 Scaricare i progetti assegnati su GitHub Classroom.

Perché abbiamo sviluppato questa applicazione?

- 1 Dissuadere gli studenti dal plagio.
- 2 Scaricare i progetti assegnati su GitHub Classroom.
- 3 Gestione di centinaia di progetti per svariati appelli all'anno.

Cosa è **GitHub**?

GitHub è un servizio di hosting per il controllo delle versioni.

Il controllo delle versioni consente di tenere traccia delle modifiche apportate al codice sorgente del software.

Cosa è **GitHub**?

GitHub è un servizio di hosting per il controllo delle versioni.

Il controllo delle versioni consente di tenere traccia delle modifiche apportate al codice sorgente del software.

Offre contenuti gratuiti per gli studenti **GitHub student Developer Pack**.

GitHub Classroom

Consente agli insegnanti di assegnare compiti agli studenti, facendoli approcciare a GitHub.

GitHub è basato sul software **Git**, il quale permette di:

- Tenere traccia dei cambiamenti dei file.

GitHub è basato sul software **Git**, il quale permette di:

- Tenere traccia dei cambiamenti dei file.
- Coordinare il lavoro di più persone sullo stesso insieme di file.

GitHub è basato sul software **Git**, il quale permette di:

- Tenere traccia dei cambiamenti dei file.
- Coordinare il lavoro di più persone sullo stesso insieme di file.
- Mantenere l'integrità dei files.

GitHub è basato sul software **Git**, il quale permette di:

- Tenere traccia dei cambiamenti dei file.
- Coordinare il lavoro di più persone sullo stesso insieme di file.
- Mantenere l'integrità dei files.
- Gestire flussi di lavori non lineari.

Inizializzazione repository

```
git init
```

Inizializzazione repository

```
git init
```

Clonare repository

```
git clone <url>
```

Inizializzazione repository

```
git init
```

Clonare repository

```
git clone <url>
```

Rendere effettive le modifiche

```
git commit -m 'Commento'
```

Prendere dati da remoto

```
git fetch <remote>
```


Prendere dati da remoto

```
git fetch <remote>
```

Condividere le modifiche

```
git push <remote> <branch>
```

Prendere dati da remoto

```
git fetch <remote>
```

Condividere le modifiche

```
git push <remote> <branch>
```

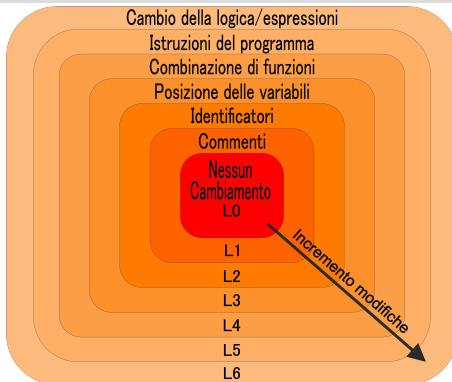
Recupero dati dal Server di origine

```
git pull
```

Definizione

Un programma che è stato prodotto da un altro e riportato con un numero esiguo di trasformazioni di routine.

1976. Alan Parker e James O. Hamblen.



Alcuni dei più famosi software di *plagiarism detection*:

SIM Software Similarity Tester

Siff Inizialmente confrontava i file di testo. Nelle versioni più recenti hanno utilizzato le fingerprint.

Plague Implementazione dei suggerimenti di Whale.

YAP Yet Another Plague. Lavora sempre con i token.

JPlag Servizio Online con un'interfaccia grafica molto funzionale. Simile a MOSS.

Caratteristiche dell'applicazione:

- Sviluppata in **Java**:

Caratteristiche dell'applicazione:

- Sviluppata in **Java**:
 - È cross-platform.

Caratteristiche dell'applicazione:

- Sviluppata in **Java**:
 - È cross-platform.
 - Documentazione completa.

Caratteristiche dell'applicazione:

■ Sviluppata in **Java**:

- È cross-platform.
- Documentazione completa.
- Esistenza di librerie per il nostro scopo.

Caratteristiche dell'applicazione:

- Sviluppata in **Java**:
 - È cross-platform.
 - Documentazione completa.
 - Esistenza di librerie per il nostro scopo.
- Utilizzo del pattern Model View Controller.

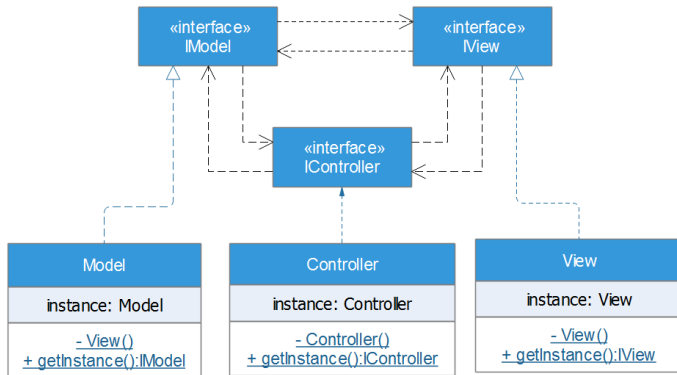
Caratteristiche dell'applicazione:

- Sviluppata in **Java**:
 - È cross-platform.
 - Documentazione completa.
 - Esistenza di librerie per il nostro scopo.
- Utilizzo del pattern Model View Controller.
- Download delle repositories di GitHub.

Caratteristiche dell'applicazione:

- Sviluppata in **Java**:
 - È cross-platform.
 - Documentazione completa.
 - Esistenza di librerie per il nostro scopo.
- Utilizzo del pattern Model View Controller.
- Download delle repositories di GitHub.
- Analisi dei progetti degli studenti tramite il servizio MOSS.

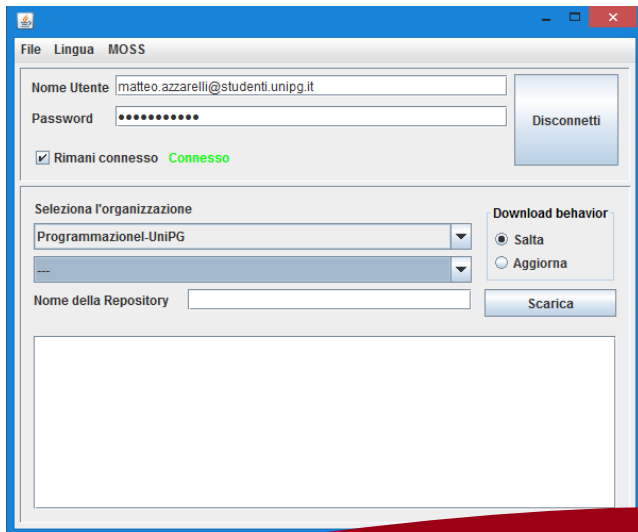
Diagramma UML della struttura del software:



Organizzazione delle repositories:



Uno sguardo all'interfaccia:

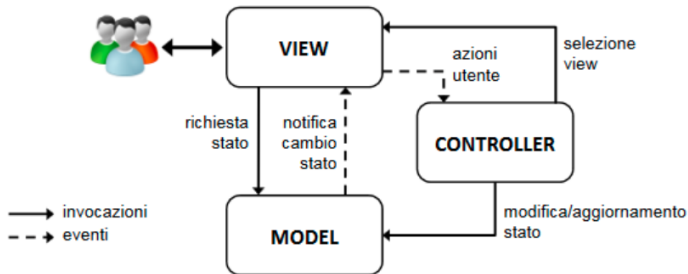


The screenshot shows a web application window titled "MOSS" with a menu bar containing "File", "Lingua", and "MOSS". The interface is divided into several sections:

- Login Section:** Includes a "Nome Utente" field with the value "matteo.azzarelli@studenti.unipg.it", a "Password" field with masked characters, a "Disconnetti" button, and a checkbox labeled "Rimani connesso" which is checked, followed by the word "Connesso" in green.
- Organization Selection:** A section titled "Seleziona l'organizzazione" containing two dropdown menus. The first dropdown is set to "ProgrammazioneI-UniPG", and the second dropdown is currently empty.
- Repository Name:** A text field labeled "Nome della Repository" is located below the organization selection dropdowns.
- Download Behavior:** A section titled "Download behavior" with two radio buttons: "Salta" (selected) and "Aggiorna".
- Download Button:** A button labeled "Scarica" is positioned below the "Download behavior" section.
- Content Area:** A large, empty rectangular box occupies the bottom half of the interface.

Il **Model View Controller** è uno dei pattern architetturali.

- **Model**: modello dei dati dell'applicazione.
- **View**: rappresenta la grafica e interattiva del model.
- **Controller**: logica di controllo che garantisce il modello dei dati.



MOSS (Measure Of Software Similarity) è un sistema automatico per determinare la similarità di programmi.

Input Accetta gruppi di documenti.

Output Restituisce un'insieme di pagine HTML contenenti le coppie di documenti simili.

MOSS (Measure Of Software Similarity) è un sistema automatico per determinare la similarità di programmi.

Input Accetta gruppi di documenti.

Output Restituisce un'insieme di pagine HTML contenenti le coppie di documenti simili.

Può analizzare molti linguaggi di programmazione, come ad esempio:

- C, C++ e C#
- Java
- Il linguaggio naturale e molti altri.

Con poche fingerprint ottiene ottimi risultati. Ciò implica maggiore efficienza.

Algoritmo:

- 1 Costruisce una mappa delle fingerprint per tutti i documenti.

Con poche fingerprint ottiene ottimi risultati. Ciò implica maggiore efficienza.

Algoritmo:

- 1 Costruisce una mappa delle fingerprint per tutti i documenti.
- 2 Per ogni documento viene effettuata una nuova fingerprint, ottenendo una lista di fingerprint per ciascun documento. Ora ogni documento d può contenere fingerprint di molti altri documenti d_1, d_2, \dots .

Con poche fingerprint ottiene ottimi risultati. Ciò implica maggiore efficienza.

Algoritmo:

- 1** Costruisce una mappa delle fingerprint per tutti i documenti.
- 2** Per ogni documento viene effettuata una nuova fingerprint, ottenendo una lista di fingerprint per ciascun documento. Ora ogni documento d può contenere fingerprint di molti altri documenti d_1, d_2, \dots .
- 3** La lista delle fingerprint viene raggruppata per documento e poi vengono fatte le coppie di documenti (d, d_1) , (d, d_2) . Queste coppie vengono ordinate per numero di fingerprint uguali.

Generare le fingerprint:

- Dividere un documento in k-grams, cioè una sottostringa di lunghezza k.

Generare le fingerprint:

- Dividere un documento in k-grams, cioè una sottostringa di lunghezza k .
- Utilizza una funzione hash per ogni k-gram e selezioniamo tra essi qualche sottoinsieme, che sarà la fingerprint.

Generare le fingerprint:

- Dividere un documento in k-grams, cioè una sottostringa di lunghezza k .
- Utilizza una funzione hash per ogni k-gram e selezioniamo tra essi qualche sottoinsieme, che sarà la fingerprint.

Generare le fingerprint:

- Dividere un documento in k-grams, cioè una sottostringa di lunghezza k.
- Utilizza una funzione hash per ogni k-gram e selezioniamo tra essi qualche sottoinsieme, che sarà la fingerprint.

Fingerprint

L'insieme di **fingerprint** è un piccolo sottoinsieme dell'insieme di tutti i "k-grams hashes".

Esempio k-gram

A do run run run, a do run run

(a) un po di testo [The Crystals. Da do run run, 1963]

adorunrunrunadorunrun

(b) il testo senza alcune caratteristiche irrilevanti

adoru dorun orunr runru unrun nrunr runru
unrun nruna runad unado nador adoru dorun
orunr runru unrun

(c) la sequenza di 5-grams derivata dal testo

MOSS mette a disposizione un webservice accessibile tramite vari script e librerie.

Per il nostro progetto abbiamo utilizzato la libreria Java **MOJI** che consente l'accesso al servizio.

MOSS mette a disposizione un webservice accessibile tramite vari script e librerie.

Per il nostro progetto abbiamo utilizzato la libreria Java **MOJI** che consente l'accesso al servizio.

Questa libreria permette di:

- 1 Connettersi al server con il proprio numero utente.

MOSS mette a disposizione un webservice accessibile tramite vari script e librerie.

Per il nostro progetto abbiamo utilizzato la libreria Java **MOJI** che consente l'accesso al servizio.

Questa libreria permette di:

- 1 Connettersi al server con il proprio numero utente.
- 2 Inviare la cartella contenente tutti i progetti degli studenti.

MOSS mette a disposizione un webservice accessibile tramite vari script e librerie.

Per il nostro progetto abbiamo utilizzato la libreria Java **MOJI** che consente l'accesso al servizio.

Questa libreria permette di:

- 1 Connettersi al server con il proprio numero utente.
- 2 Inviare la cartella contenente tutti i progetti degli studenti.
- 3 Inviare la cartella contenente il modello del progetto.

MOSS mette a disposizione un webservice accessibile tramite vari script e librerie.

Per il nostro progetto abbiamo utilizzato la libreria Java **MOJI** che consente l'accesso al servizio.

Questa libreria permette di:

- 1 Connettersi al server con il proprio numero utente.
- 2 Inviare la cartella contenente tutti i progetti degli studenti.
- 3 Inviare la cartella contenente il modello del progetto.
- 4 Ricevere il link per visualizzare la pagina HTML dei risultati.

Struttura cartella dei progetti

```
solution_directory
|- student1
    |- main.c
    |- ...
|- student2
    |- ...
|- student3
    |- ...
```

Moss Results

Wed Jul 4 13:31:37 PDT 2018

Options -l c -d -m 10

ESAME PROGRAMMAZIONE 1

[[How to Read the Results](#) | [Tips](#) | [FAQ](#) | [Contact](#) | [Submission Scripts](#) | [Credits](#)]

File 1	File 2	Lines Matched
C:/ProgrammazioneI-UniPG/prova-finale-di-programmazione-secondo-appello/██████c98/ (96%)	C:/ProgrammazioneI-UniPG/prova-finale-di-programmazione-secondo-appello/██████s2/ (97%)	1035
C:/ProgrammazioneI-UniPG/prova-finale-di-programmazione-secondo-appello/██████ni4/ (74%)	C:/ProgrammazioneI-UniPG/prova-finale-di-programmazione-secondo-appello/██████elli/ (78%)	776
C:/ProgrammazioneI-UniPG/prova-finale-di-programmazione-secondo-appello/██████ce/ (96%)	C:/ProgrammazioneI-UniPG/prova-finale-di-programmazione-secondo-appello/██████trdo/ (97%)	669
C:/ProgrammazioneI-UniPG/prova-finale-di-programmazione-secondo-appello/██████iv/ (85%)	C:/ProgrammazioneI-UniPG/prova-finale-di-programmazione-secondo-appello/██████p0/ (92%)	831
C:/ProgrammazioneI-UniPG/prova-finale-di-programmazione-secondo-appello/██████etta/ (84%)	C:/ProgrammazioneI-UniPG/prova-finale-di-programmazione-secondo-appello/██████etti/ (86%)	759
C:/ProgrammazioneI-UniPG/prova-finale-di-programmazione-secondo-appello/██████etto/ (76%)	C:/ProgrammazioneI-UniPG/prova-finale-di-programmazione-secondo-appello/██████sae/ (80%)	722
C:/ProgrammazioneI-UniPG/prova-finale-di-programmazione-secondo-appello/██████c0/ (80%)	C:/ProgrammazioneI-UniPG/prova-finale-di-programmazione-secondo-appello/██████sae/ (78%)	729

C:\Programmazione\UniPG\prova-finale-di-programmazione-secondo-appello

C:\Programmazione\UniPG\prova-fine-di-programmazione-secondo-appello\ [redacted] etta

22 of 23

L'applicazione è stata testata sul campo riportando risultati positivi.

Futuri ampliamenti:

- Download dei risultati di MOSS (MOSS li elimina dopo 14 giorni).
- Analisi statica dei progetti (es. Valgrind), per fornire un aiuto al docente con una pre-valutazione del progetto.
- Ottenere il nome e il cognome degli studenti dal file ReadMe delle repositories e costruire dinamicamente un gestionale degli strumenti.