

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

Corso di Laurea Triennale in Informatica  
Tesi di Laurea

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?

- 1 Dissuadere gli studenti dal plagio.
- 2 Scaricare i progetti assegnati su GitHub Classroom.
- 3 Gestione di centinaia di progetti per svariati appelli ed esercitazioni all'anno, fornendo inoltre un'analisi su eventuali similitudini.

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 ed esercitazioni all'anno, fornendo inoltre un'analisi su eventuali similitudini.

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 ed esercitazioni all'anno, fornendo inoltre un'analisi su eventuali similitudini.

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 ed esercitazioni all'anno, fornendo inoltre un'analisi su eventuali similitudini.

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.

## GitHub Classroom

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

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.

## GitHub Classroom

Consente ai docenti 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.
- Coordinare il lavoro di più persone sullo stesso insieme di file.
- Mantenere l'integrità dei files.
- Gestire flussi di lavoro non lineari.

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 lavoro non lineari.

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 lavoro non lineari.

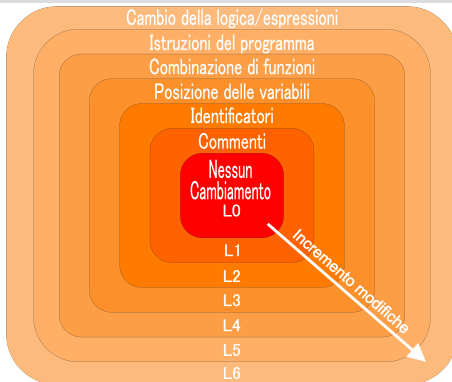
GitHub è basato sul software **Git**, il quale permette di:

- 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 lavoro non lineari.

## 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.*



## 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.

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.

Caratteristiche dell'applicazione:

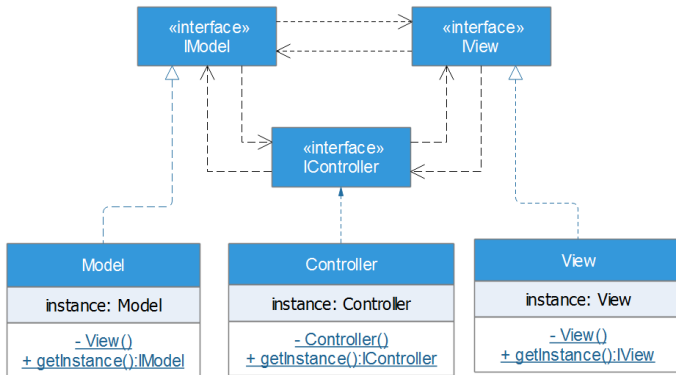
- Sviluppata in **Java**:
  - È cross-platform.
  - Documentazione completa.
  - Esistenza di librerie per il nostro scopo.
- Utilizzo del pattern Model View Controller.
- Download delle repositories di GitHub.
- Analisi dei progetti degli studenti tramite il servizio MOSS.



## 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.

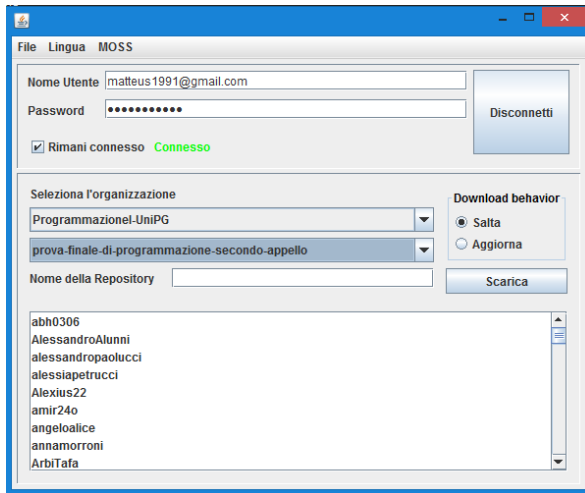
Diagramma UML della struttura del software:



## 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 "matteus1991@gmail.com", 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 "Programmazione-UniPG", and the second dropdown is set to "prova-finale-di-programmazione-secondo-appello".
- Repository Name:** A text input field labeled "Nome della Repository" which is currently empty.
- Download Behavior:** A section titled "Download behavior" with two radio buttons: "Salta" (selected) and "Aggiorna". Below this is a "Scarica" button.
- User List:** A scrollable list box containing the following usernames: abh0306, AlessandroAlunni, alessandropaolucci, alessiapetrucci, Alexius22, amir24o, angeloalice, annamorrone, and ArbiTafa.

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.

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.

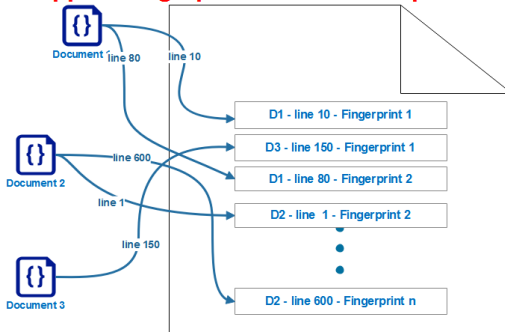
Può analizzare molti linguaggi di programmazione, come ad esempio:

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

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

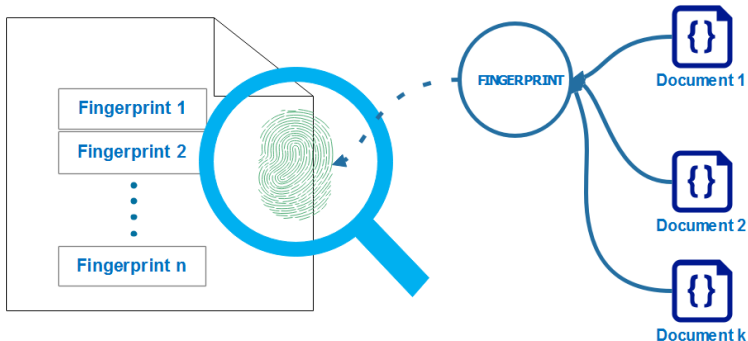
Algoritmo:

**Indice che mappa le fingerprint alle locazioni per tutti i documenti**



1) Costruisce un indice che mappa le fingerprint alle locazioni per tutti i documenti.

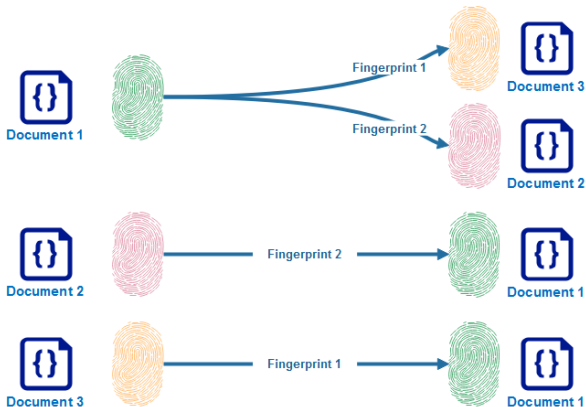
**Per ogni documento vengono effettuate nuovamente le fingerprint e ricercate nell'indice creato precedentemente**



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  $d_1, d_2, \dots$



## Copie dei documenti



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:

- Divide un documento in k-grams, cioè in sottostringhe di lunghezza k

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

- Utilizza una funzione hash per ogni k-gram e selezioniamo tra essi qualche sottoinsieme, che sarà la fingerprint.

Generare le fingerprint:

- Divide un documento in k-grams, cioè in sottostringhe di lunghezza k

## 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

- Utilizza una funzione hash per ogni k-gram e selezioniamo tra essi qualche sottoinsieme, che sarà la fingerprint.

Generare le fingerprint:

- Divide un documento in k-grams, cioè in sottostringhe di lunghezza k

## 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

- Utilizza una funzione hash per ogni k-gram e selezioniamo tra essi qualche sottoinsieme, che sarà la fingerprint.

Generare le fingerprint:

- Divide un documento in k-grams, cioè in sottostringhe di lunghezza k

## 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

- Utilizza una funzione hash per ogni k-gram e selezioniamo tra essi qualche sottoinsieme, che sarà la fingerprint.

Generare le fingerprint:

- Divide un documento in k-grams, cioè in sottostringhe di lunghezza k

## 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

- Utilizza una funzione hash per ogni k-gram e selezioniamo tra essi qualche sottoinsieme, che sarà la fingerprint.

Generare le fingerprint:

- Divide un documento in k-grams, cioè in sottostringhe di lunghezza k

## 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

- Utilizza una funzione hash per ogni k-gram e selezioniamo tra essi qualche sottoinsieme, che sarà la fingerprint.

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:

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



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.

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.

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.

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

Tue Jul 3 06:02:57 PDT 2018

Options -l c -d -m 10

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

File 1	File 2	Lines Matched
<a href="#">C:/Users/██████/ProgrammazioneI-UniPG/prova-finale-di-programmazione-secondo-appello/██████c98/ (96%)</a>	<a href="#">C:/Users/██████/ProgrammazioneI-UniPG/prova-finale-di-programmazione-secondo-appello/██████82/ (97%)</a>	1035
<a href="#">C:/Users/██████/ProgrammazioneI-UniPG/prova-finale-di-programmazione-secondo-appello/██████ni4/ (74%)</a>	<a href="#">C:/Users/██████/ProgrammazioneI-UniPG/prova-finale-di-programmazione-secondo-appello/██████elli/ (78%)</a>	776
<a href="#">C:/Users/██████/ProgrammazioneI-UniPG/prova-finale-di-programmazione-secondo-appello/██████lice/ (96%)</a>	<a href="#">C:/Users/██████/ProgrammazioneI-UniPG/prova-finale-di-programmazione-secondo-appello/██████ardo/ (97%)</a>	669
<a href="#">C:/Users/██████/ProgrammazioneI-UniPG/prova-finale-di-programmazione-secondo-appello/██████fa/ (85%)</a>	<a href="#">C:/Users/██████/ProgrammazioneI-UniPG/prova-finale-di-programmazione-secondo-appello/██████p0/ (92%)</a>	831

C:/Users/██████/ProgrammazioneI-UniPG/prova-finale-di-programmazione-secondo-appello/██████etta/ (60%)		C:/Users/██████/ProgrammazioneI-UniPG/prova-finale-di-programmazione-secondo-appello/██████c0/ (64%)	
<a href="#">370-494</a>		<a href="#">377-504</a>	
<a href="#">569-707</a>		<a href="#">570-701</a>	
<a href="#">248-317</a>		<a href="#">243-320</a>	
<a href="#">323-364</a>		<a href="#">327-370</a>	

```

if(prima_zona == NULL) //Controlla se nella lista c'è almeno
{
    return -2;
}
else if(conta_zone() < 8) //Controlla se il numero delle zone
{
    return -1;
}
else if(ultima_zona -> nome_zona == 5) //Se ultima zona è usc
{
    return -3;
}
else //Tutto OK!
{
    return 1;
}
    
```

```

if(prima_zona == NULL)//mappa vuota
{
    return -2;
}
else if(count_zona() < 8)//caso in cui non ci sono abbastanz
{
    return -1;
}
else if (ultima_zona -> nome_zona == 5)//caso in cui la mapp
{
    return -3;
}
else
{
    return 1;
}
    
```

Obiettivi raggiunti:

- 1 Scaricare i progetti assegnati su GitHub Classroom.
- 2 Gestire ed analizzare centinaia di progetti, analizzandoli tramite MOSS.

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.



Obiettivi raggiunti:

- 1 Scaricare i progetti assegnati su GitHub Classroom.
- 2 Gestire ed analizzare centinaia di progetti, analizzandoli tramite MOSS.

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.

# Grazie per l'attenzione