

ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA

---

CAMPUS DI CESENA

DIPARTIMENTO DI INFORMATICA – SCIENZA E INGEGNERIA

CORSO DI LAUREA TRIENNALE IN INGEGNERIA E SCIENZE  
INFORMATICHE

**PROGETTAZIONE E SVILUPPO DI UN SISTEMA  
PER LA VALUTAZIONE DI ELABORATI D'ESAME**

Tesi in

PROGRAMMAZIONE

**Relatore:**  
**Dott.ssa Alessandra Lumini**

**Presentata da:**  
**Carlo Cariello**

**Sessione I**  
**Anno Accademico 2019/2020**



# Indice

<b>Introduzione</b>	<b>11</b>
<b>1 Problema affrontato</b>	<b>15</b>
1.1 Requisiti . . . . .	15
1.2 Applicazioni esistenti . . . . .	16
1.3 Tabella di confronto tra le applicazioni esistenti . . . . .	20
<b>2 Analisi dei requisiti</b>	<b>23</b>
2.1 Requisiti funzionali . . . . .	23
2.1.1 Registrazione e login . . . . .	23
2.1.2 Specifiche relative ai progetti da svolgere . . . . .	24
2.1.3 Gestione consegne . . . . .	24
2.1.4 Compilazione . . . . .	24
2.1.5 Testing . . . . .	24
2.1.6 Originalità . . . . .	25
2.2 Requisiti non funzionali . . . . .	25
2.2.1 Interfaccia user-friendly . . . . .	25
2.2.2 Architettura client-server . . . . .	26
2.3 Caratteristiche del sistema . . . . .	26
2.3.1 Gestione invio automatico delle e-mail . . . . .	26
2.3.2 Persistenza delle informazioni . . . . .	26
2.3.3 Estensibilità . . . . .	27
2.3.4 Facilità di utilizzo . . . . .	27
2.4 Modalità di utilizzo . . . . .	27
2.5 Qualità del software . . . . .	28
<b>3 Progettazione</b>	<b>31</b>
3.1 Architettura generale . . . . .	31
3.2 Progettazione della base dati . . . . .	33
3.2.1 Schema Entity-Relationship . . . . .	33
3.2.2 Schema Logico . . . . .	34

3.2.3	Struttura delle tabelle . . . . .	35
3.3	Progettazione delle funzioni . . . . .	37
3.3.1	Diagrammi casi d'uso . . . . .	37
3.3.2	Diagrammi di attività . . . . .	38
3.4	Progettazione dell'interfaccia . . . . .	41
3.4.1	Mockup . . . . .	41
3.4.2	Screenshot . . . . .	47
<b>4</b>	<b>Implementazione</b>	<b>55</b>
4.1	Tecnologie usate . . . . .	55
4.1.1	Tecnologie di progettazione . . . . .	55
4.1.2	DB-Main . . . . .	55
4.1.3	DBMS . . . . .	56
4.1.4	Lighthouse . . . . .	56
4.1.5	Tecnologie di progetto . . . . .	57
4.1.6	Bootstrap . . . . .	57
4.1.7	JavaScript . . . . .	58
4.1.8	XAMPP/LAMPP . . . . .	58
4.1.9	Aruba . . . . .	58
4.1.10	Amministrazione del server tramite SSH . . . . .	59
4.1.11	Linguaggi utilizzati . . . . .	59
4.2	Dettagli implementativi . . . . .	59
4.2.1	Invio e-mail . . . . .	59
4.2.2	Recupero password . . . . .	60
4.2.3	Compilatore C . . . . .	60
4.2.4	Testing . . . . .	60
<b>5</b>	<b>Indagine</b>	<b>61</b>
5.1	Soggetti . . . . .	61
5.2	Domande del questionario . . . . .	62
5.2.1	Sezione dei dati personali . . . . .	62
5.2.2	Sezione dedicata al progetto . . . . .	63
5.2.3	Sezione conclusiva . . . . .	64
5.3	Risultati . . . . .	64
5.3.1	Risultati dell'indagine . . . . .	65
5.3.2	Grafici riassuntivi . . . . .	67
	<b>Conclusioni e sviluppi futuri</b>	<b>75</b>
	<b>Bibliografia</b>	<b>77</b>

# Elenco delle figure

1.1	Esempio d'utilizzo piattaforma Miur . . . . .	17
1.2	Esempio d'utilizzo piattaforma Boss . . . . .	18
1.3	Esempio d'utilizzo piattaforma Ideone . . . . .	18
1.4	Esempio d'utilizzo piattaforma Submitty . . . . .	19
1.5	Esempio d'utilizzo piattaforma OpenSubmit . . . . .	19
2.1	Panoramica caratteristiche qualità del software . . . . .	30
3.1	Esempio funzionamento server . . . . .	32
3.2	Diagramma casi d'uso con user non registrato . . . . .	39
3.3	Diagramma casi d'uso con user registrato . . . . .	39
3.4	Diagramma di attività: registrazione e login user . . . . .	40
3.5	Diagramma di attività: sottomissione elaborato . . . . .	40
3.6	Diagramma E-R . . . . .	41
3.7	Mockup homepage desktop . . . . .	42
3.8	Mockup homepage smartphone . . . . .	42
3.9	Mockup registrazione da desktop . . . . .	43
3.10	Mockup registrazione smartphone . . . . .	43
3.11	Mockup login desktop . . . . .	44
3.12	Mockup login smartphone . . . . .	44
3.13	Mockup homepage user desktop . . . . .	45
3.14	Mockup homepage user smartphone . . . . .	45
3.15	Mockup conferma desktop . . . . .	46
3.16	Mockup conferma smartphone . . . . .	46
3.17	Screenshot homepage . . . . .	47
3.18	Screenshot che mostra la fase di login . . . . .	48
3.19	Screenshot relativo alla registrazione di un singolo studente . . . . .	48
3.20	Screenshot relativo alla registrazione di due studenti . . . . .	49
3.21	Screenshot relativo alla ricezione della e-mail di registrazione . . . . .	49
3.22	Screenshot che mostra l'area riservata . . . . .	50
3.23	Screenshot che mostra il caricamento dell'elaborato . . . . .	50
3.24	Screenshot che illustra la dashboard . . . . .	51

3.25	Screenshot relativo alla visualizzazione dell'errore di compilazione . . . . .	51
3.26	Screenshot relativo alla ricezione delle e-mail per l'esecuzione . . . . .	52
3.27	Screenshot homepage docente . . . . .	52
3.28	Screenshot relativo all'inserimento di un nuovo elaborato . . . . .	53
3.29	Esempio che mostra l'applicazione della funzione di similarity su due elaborati uguali, consegnati da due gruppi diversi . . . . .	53
3.30	Screenshot che illustra la ricerca degli elaborati . . . . .	54
4.1	Esempio di utilizzo di Lighthouse . . . . .	57
5.1	Grafico che mostra l'anno di appartenenza . . . . .	67
5.2	Grafico che illustra l'anno di nascita dei soggetti . . . . .	67
5.3	Grafico che illustra il sesso dei soggetti . . . . .	68
5.4	Grafico che illustra la prima impressione dei soggetti . . . . .	68
5.5	Grafico che illustra le difficoltà riscontrate dai soggetti . . . . .	69
5.6	Grafico che illustra le possibili soluzioni ai problemi trovati dai soggetti . . . . .	69
5.7	Grafico che illustra su una scala di valori da 0 a 3 se i contenuti sono semplici ed intuitivi . . . . .	70
5.8	Grafico che illustra su una scala di valori da 0 a 3 se il testo è facilmente leggibile . . . . .	70
5.9	Grafico che illustra su una scala di valori da 0 a 3 se il titolo delle pagine è esplicativo . . . . .	70
5.10	Grafico che illustra su una scala di valori da 0 a 3 se le pagine presentano opzioni di navigazione . . . . .	71
5.11	Grafico che illustra su una scala di valori da 0 a 3 se sono presenti funzionalità di annullamento delle operazioni . . . . .	71
5.12	Grafico che illustra su una scala di valori da 0 a 3 se sono presenti funzionalità per ritornare alla homepage . . . . .	71
5.13	Grafico che illustra su una scala di valori da 0 a 3 se il menù di navigazione è efficace . . . . .	72
5.14	Grafico che illustra su una scala di valori da 0 a 3 se l'applicazione ha un aspetto gradevole . . . . .	72
5.15	Grafico che illustra su una scala di valori da 0 a 3 se lo stile è adeguato alle varie schermate della piattaforma . . . . .	72
5.16	Grafico che illustra su una scala di valori da 0 a 3 se ci sono situazioni in cui la navigazione è facilitata dalla grafica . . . . .	73
5.17	Grafico che illustra su una scala di valori da 0 a 3 se ci sono situazioni in cui la navigazione è compromessa dalla grafica . . . . .	73

5.18	Grafico che illustra su una scala di valori da 0 a 3 se i tempi della compilazione dell'elaborato sono ragionevoli . . . . .	73
5.19	Grafico che illustra su una scala di valori da 0 a 3 se i tempi dell'esecuzione dell'elaborato sono ragionevoli . . . . .	74
5.20	Grafico che illustra su una scala di valori da 0 a 3 se i materiali messi a disposizione sono semplici da visionare . . . . .	74
5.21	Grafico che illustra su una scala di valori da 0 a 3 se la consegna di un elaborato è stata semplice . . . . .	74





# Elenco delle tabelle

1.1	Tabella esplicativa dei valori di ogni applicazione esistente . . .	21
3.1	Struttura della tabella ELABORATO . . . . .	35
3.2	Struttura della tabella UTENTE - entità padre . . . . .	35
3.3	Struttura della tabella GRUPPO - entità figlia . . . . .	35
3.4	Struttura della tabella CONSEGNA . . . . .	36
3.5	Struttura della tabella LOGS . . . . .	36



# Introduzione

Durante il percorso formativo è importante per gli studenti disporre di uno strumento che consenta rapidamente di ottenere un feedback sugli esercizi svolti dagli stessi.

A tale fine in questo lavoro di tesi è stata realizzata una piattaforma che dal lato studente fornisca la possibilità di effettuare la consegna periodica di esercizi in linguaggio C, potendo, in tempo reale, ottenere la compilazione per ogni consegna e la notifica del relativo esito e voto; dal lato amministratore, consentire le funzionalità per effettuare la pubblicazione con relativa notifica degli esercizi agli studenti, l'attribuzione del voto per ciascuna consegna, il rilevamento della similarità tra gli elaborati consegnati rispetto ad un dato esercizio, nonché la funzione di download per scaricare i vari elaborati.

Questo lavoro è importante perchè l'esistenza di una piattaforma che offra i servizi descritti potenzia l'attività didattica del docente e quella di apprendimento degli studenti in quanto ottimizza i tempi di gestione degli elaborati grazie all'automatizzazione dei processi durante le fasi della programmazione, compilazione ed esecuzione di un programma in C archiviandone gli esiti e le informazioni degli studenti.

Attraverso lo studio di questa tesi è stato possibile portare le conoscenze affrontate durante il percorso universitario all'interno di un sistema integrato di metodologie, processi e strumenti di sviluppo impiegati quotidianamente nel settore dell'IT.

Nella fase iniziale del presente lavoro si è affrontata l'analisi dei requisiti che la piattaforma deve offrire. Successivamente si è passati alla verifica dell'esistenza in rete strumenti analoghi o che potessero risultare modello di riferimento idoneo. Una volta definiti gli elementi essenziali del lavoro da svolgere è stata avviata la fase di progettazione del sistema.

L'attività di implementazione ha comportato lo sviluppo ex novo di tutte le funzionalità del sito, richiedendo oltre alle competenze acquisite durante il corso di studio, approfondimenti riguardo a specifiche tematiche per la messa online della piattaforma.

I risultati ottenuti sono conformi a quanto analizzato durante le fasi precedenti, poichè l'implementazione non ha comportato modifiche o rinunce dei requisiti iniziali.

Il lavoro di tesi è organizzato nei seguenti capitoli:

1. *Problema affrontato*: il lavoro degli studenti può essere facilitato da piattaforme che consentono rapidamente di verificarne gli esiti degli esercizi svolti; per questa ragione prima di avviare il presente lavoro sono stati valutati i software esistenti;
2. *Analisi dei requisiti*: la comparazione dei software valutata in precedenza non è risultata soddisfacente, per cui sono stati introdotti i requisiti necessari alla realizzazione;
3. *Progettazione*: descrive l'architettura generale del sistema, del relativo database e dell'interfaccia grafica dando spazio alla usabilità da dispositivo mobile;
4. *Implementazione*: illustra le tecnologie impiegate spiegando gli strumenti software per la produzione del codice client, degli stili delle pagine e del codice server;

5. *Indagine*: descrive il questionario somministrato agli utenti, illustra i risultati ottenuti tramite grafici, visualizza commenti e suggerimenti aggiunti dagli intervistati.



# Capitolo 1

## Problema affrontato

Lo scopo di questa tesi è la realizzazione di un sistema per la valutazione di elaborati d'esame del corso di Programmazione presso l'Università di Bologna nel campus di Cesena. Quest'idea nasce dall'esigenza di poter automatizzare il processo di consegna degli elaborati in un tempo circoscritto, ad esempio una settimana. Essendo, il primo anno, frequentato da molti studenti, è necessario avere uno strumento efficace e di semplice utilizzo per tutti gli utenti.

La piattaforma necessita di una pagina di registrazione, una di login ed una sezione per gestire gli elaborati consegnati da ciascuno studente o gruppo di studenti.

### 1.1 Requisiti

Il principale requisito architetturale è la necessità di avere una piattaforma con un'interfaccia responsive, che permetta agli studenti di utilizzare questa anche su device mobile, tablet o smartphone. Fornire agli utenti un'interfaccia intuitiva permette loro di sottomettere gli elaborati in maniera semplice e poter ricevere subito una valutazione del progetto.

Un'attività importante è quella di mostrare tutti gli step di consegna allo studente, cosicché possa comprendere appieno i principi della programmazione.

Dopo aver caricato sulla piattaforma l'elaborato, lo studente dovrà compilare il proprio programma. Se la compilazione andrà a buon fine passa allo step successivo, quello dell'esecuzione, altrimenti dovrà effettuare una nuova sottomissione dell'elaborato. Per alcuni elaborati il numero di sottomissioni potrà essere limitato, a discrezione del docente.

Se anche la fase di esecuzione è andata a buon fine lo studente riceverà un punteggio relativo all'elaborato, in caso contrario se ha avuto qualche errore lo studente dovrà risolvere e riprovare con una nuova consegna.

## 1.2 Applicazioni esistenti

Navigando su internet sono presenti diversi siti che offrono il servizio di compilazione ed esecuzione di programmi implementati in diversi linguaggi di programmazione. A seguire un elenco di alcuni sistemi di cui saranno elencate le principali caratteristiche:

- **digit.olinfo.it**: questo applicativo [12] del Ministero dell'Istruzione dell'Università e della Ricerca (MIUR) accessibile a docenti, permette loro, previa registrazione, di compilare online programmi in tre differenti linguaggi (C, C++, Pascal). L'interfaccia grafica è molto intuitiva, presenta pochi pulsanti ed un'area dedicata alla scrittura del codice da sottomettere, in alternativa si può caricare il file da compilare, esempio visibile alla figura 1.1; il sistema dispone inoltre di una fase di valutazione dei progetti inseriti;
- **BOSS Online Submission System 2**: è uno strumento [22] online di gestione dei corsi sviluppato dal Dipartimento di Informatica dell'Università di Warwick (Regno Unito). BOSS consente agli studenti di inviare i propri compiti (scritti in java) online in modo sicuro permettendo al personale amministrativo di contrassegnare le consegne online e verificare che l'esercizio proposto sia stato svolto correttamente. Il sistema viene gestito tramite un database in cui vengono archiviati tutti



i file, ed è presente una panoramica generale riguardo la sottomissione, ed una parte front-end per gli studenti, visionabile nella figura 1.2;

- **ideone.com**: questo [9] è un sito totalmente gratuito che permette la compilazione ed esecuzione online in più di sessanta linguaggi di programmazione differenti, riferimento alla figura 1.3; il sistema non prevede sottomissione e valutazione di progetti d'esame o esercizi;
- **Submitty**: sistema [10] open source di invio di esercizi lanciato dal Dipartimento di Informatica presso il Rensselaer Polytechnic Institute, New York. Permette allo studente di caricare il proprio file in diversi linguaggi come Python, C / C ++, Java, Scheme, Prolog, SQL ed altri ricevendo un feedback immediato sull'esito in base a test precedentemente caricati, figura 1.4;
- **OpenSubmit**: sistema [20] che aiuta gli studenti nella validazione del proprio programma. Dopo averlo caricato, il sistema valuta se è corretto per poi lasciare il compito di assegnare un punteggio all'admin, visionabile alla figura 1.5;

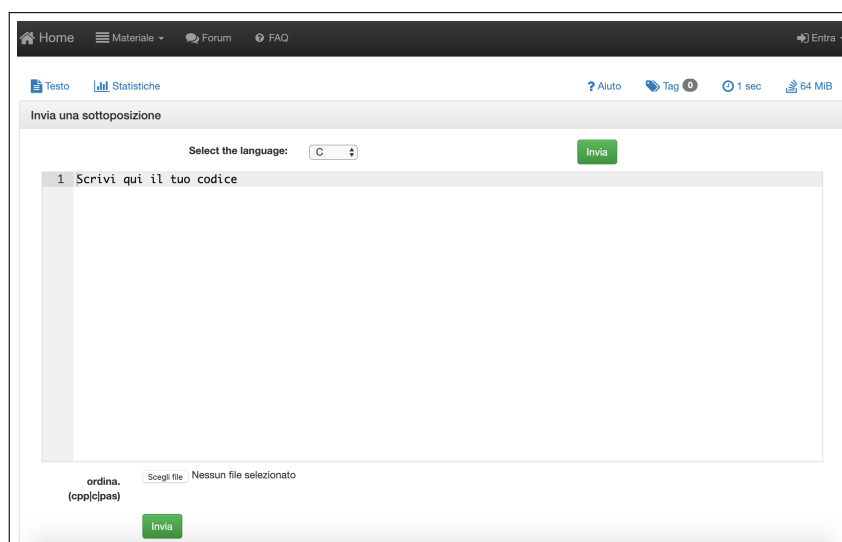


Figura 1.1: Esempio d'utilizzo piattaforma Miur

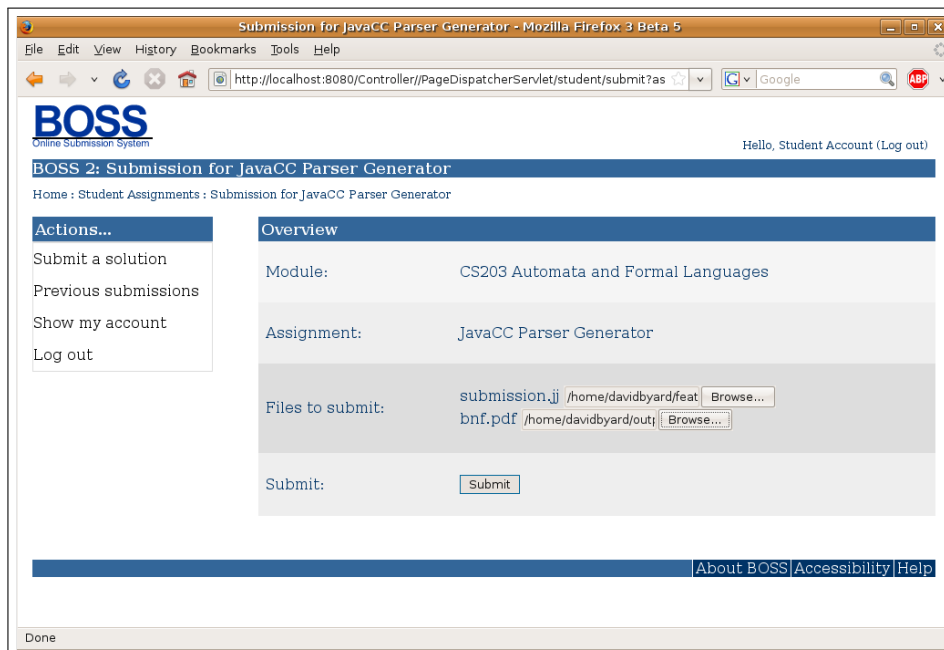


Figura 1.2: Esempio d'utilizzo piattaforma Boss

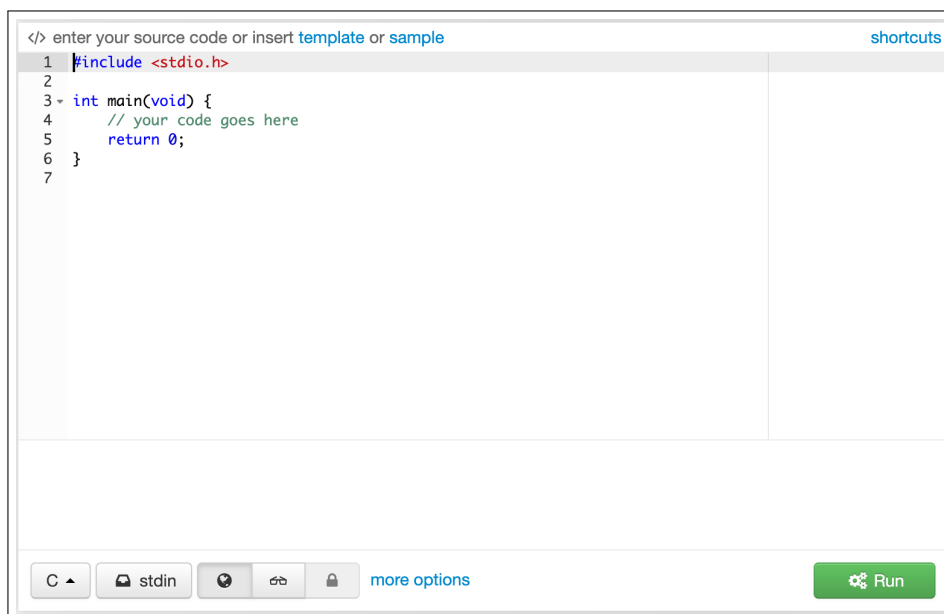


Figura 1.3: Esempio d'utilizzo piattaforma Ideone

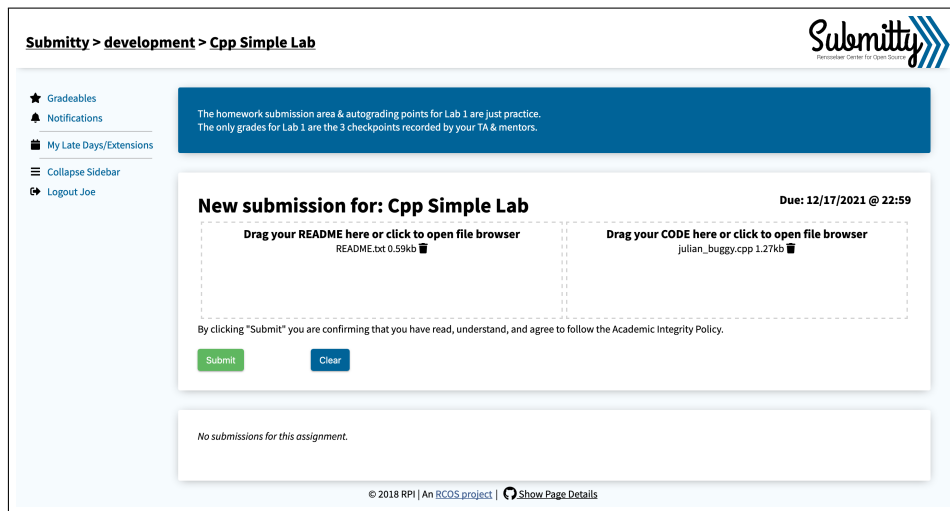


Figura 1.4: Esempio d'utilizzo piattaforma Submitty

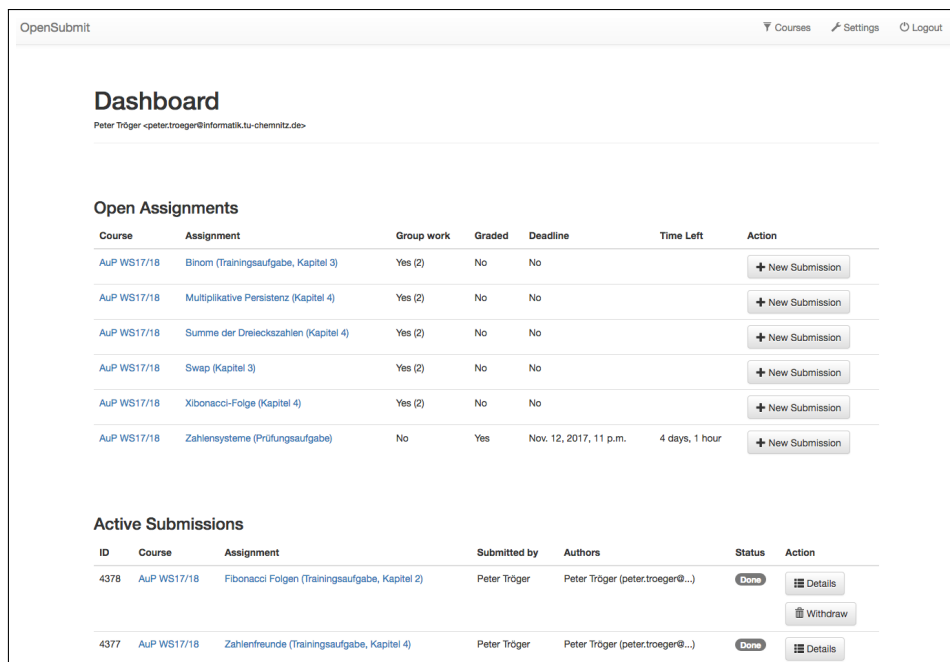


Figura 1.5: Esempio d'utilizzo piattaforma OpenSubmit

## 1.3 Tabella di confronto tra le applicazioni esistenti

La tabella seguente 1.1, analizza ciascun applicativo assegnando loro un punteggio (insufficiente, sufficiente, discreto, buono ed ottimo) su una determinata caratteristica. Le caratteristiche prese in considerazione sono:

- **Esecuzione:** questa caratteristica comprende compilazione ed esecuzione. Tutti gli applicativi presi in considerazione permettono di effettuare entrambe le fasi caricando il file.c o scrivendo nell'apposita area testuale.  
Dopo pochi secondi è possibile verificare l'output del programma inserito in modo semplice;
- **Similarità:** caratteristica che permette di effettuare un confronto tra tutti gli elaborati precedentemente consegnati. Genera all'utente una percentuale di uguaglianza tra i vari elaborati. Caratteristica non presente in nessuno degli applicativi testati;
- **Storico:** caratteristica presente in soli due applicativi tra quelli testati. Consente all'utente, previa registrazione, di poter visionare i programmi caricati in passato.
- **Voto:** caratteristica che permette all'utente di ricevere un voto riguardo la consegna di un certo elaborato secondo alcuni criteri (qualità del codice, strutture dati utilizzate, ecc...). Non presente in nessuno degli applicativi testati;
- **Interfaccia:** caratteristica importante affinché l'utente possa districarsi in maniera semplice nello svolgere il proprio programma. Come visibile dalla tabella 1.1 non in tutti gli applicativi hanno un punteggio quantomeno discreto in relazione alla propria interfaccia grafica.

**Caratteristiche comuni a tutti gli applicativi:** possibilità di scelta del linguaggio per compilare/eseguire molto vasta per tutti gli applicativi analizzati tranne che per il sistema boss [22].

	<b>Esecuzione</b>	<b>Similarità</b>	<b>Storico</b>	<b>Voto</b>	<b>Interfaccia</b>
<b>digit.olinfo</b>	buono	assente	discreto	assente	ottimo
<b>boss</b>	discreto	assente	sufficiente	sufficiente	discreto
<b>ideone</b>	buono	assente	assente	assente	discreto
<b>submitty</b>	buono	ottimo	sufficiente	sufficiente	buono
<b>opensubmit</b>	ottimo	buono	buono	discreto	ottimo

Tabella 1.1: Tabella esplicativa dei valori di ogni applicazione esistente

In conclusione, dopo aver analizzato le applicazioni esistenti, si è notato che nessuna di queste soddisfa i requisiti ricercati per il corso di Programmazione. È stato deciso di sviluppare un nuovo sistema automatico che permetta di gestire gli elaborati degli studenti come necessario.



# Capitolo 2

## Analisi dei requisiti

In questo capitolo saranno descritti i requisiti che il sistema dovrà possedere per poter soddisfare le esigenze del corso. La fase dell'analisi dei requisiti è una delle prime che vengono effettuate durante lo sviluppo di un progetto. Verranno analizzate le caratteristiche architettoniche dell'applicazione, i requisiti del suo funzionamento, le diverse modalità di utilizzo, ad esempio il lato amministratore ed il lato degli utenti, e la qualità del software.

### 2.1 Requisiti funzionali

L'applicazione si pone l'obiettivo di realizzare un sistema automatico per la sottomissione e la valutazione di esercizi in linguaggio C. Gli utenti del sistema saranno il docente/docenti del corso e gli studenti.

#### 2.1.1 Registrazione e login

Gli utenti devono necessariamente essere registrati per poter usufruire del servizio. Al momento della registrazione uno studente può scegliere se registrarsi in maniera individuale o in gruppo. Ad ogni individuo o gruppo saranno assegnati uno username (scelto in modo automatico dal sistema) e una password. Sarà presente una sezione in cui poter recuperare la propria password se smarrita. Una volta effettuata la registrazione l'utente riesce ad

accedere alla propria dashboard in cui vengono visualizzati gli esercizi che dovrà consegnare durante lo svolgimento del corso.

### **2.1.2 Specifiche relative ai progetti da svolgere**

Il docente (amministratore di sistema) si occupa di pubblicare gli esercizi da assegnare agli studenti, scegliendo una deadline per la consegna di ciascuno di questi, un testo descrittivo con le specifiche dell'elaborato ed eventualmente del materiale scaricabile dallo studente utile per lo svolgimento dell'elaborato. Inoltre si può impostare un numero massimo di consegne per elaborato, dipendente dalla difficoltà, per imporre un limite agli studenti.

### **2.1.3 Gestione consegne**

Ogni studente può consegnare un progetto, verificarne la correttezza e l'ammissibilità. Lo studente avrà a disposizione un numero limitato di tentativi, prefissato dal docente. Ogni tentativo di consegna prevede due fasi: compilazione ed esecuzione.

Se l'esito di queste due fasi risulta essere positivo allora riceverà un punteggio riguardo la qualità del codice scritto (*correttezza*).

### **2.1.4 Compilazione**

Nella fase di consegna lo studente tramite apposita interfaccia carica l'elaborato da correggere. Al termine del caricamento la piattaforma, dopo aver verificato provvede ad effettuare la compilazione in automatico. Questa fase può produrre due esiti che saranno visualizzati all'utente mediante un apposito output.

### **2.1.5 Testing**

E' stata predisposta una pagina che successivamente alla compilazione consente all'utente di poter lanciare la versione eseguibile del proprio esercizio



ed ottenere un feedback sulla validità dell'esecuzione in base a valori di test predisposti dall'admin.

### **2.1.6 Originalità**

Questa fase provvede alla comparazione degli elaborati consegnati dai vari gruppi per individuare eventuali copiatore e casi di plagiarismo. L'algoritmo su cui si basa la ricerca delle corrispondenze è quello di MOSS [18] (Measure Of Software Similarity). A tal proposito la piattaforma esegue una specifica classe messa a disposizione in grado di generare direttamente sul server della Stanford University un documento html contenente i dati risultanti dalla comparazione che rimangono a disposizione dell'amministratore per due settimane. Trascorso questo periodo è comunque possibile sottomettere nuovamente i dati per ottenere una nuova analisi.

## **2.2 Requisiti non funzionali**

I requisiti non funzionali sono quelle proprietà misurabili dall'utente e non correlate al comportamento funzionale del sistema. Per questo progetto, i requisiti funzionali individuati sono esposti qui di seguito.

### **2.2.1 Interfaccia user-friendly**

L'interfaccia del software da realizzare deve essere intuitiva per permettere agli studenti di consegnare nel modo più semplice possibile il proprio elaborato. Infatti sono stati utilizzati dei componenti grafici elementari come: tabelle, radio button, bottoni, legende e testi che ne permettono una notevole usabilità. L'interfaccia web, inoltre deve essere responsive per permettere allo studente di accedere in qualsiasi momento attraverso dispositivi mobili.

## **2.2.2 Architettura client-server**

In questo caso è richiesto lo sviluppo di un'applicazione che permette agli studenti di consegnare e correggere i propri elaborati in autonomia.

L'applicazione si basa su un'architettura di rete di tipo client-server, questa consiste in uno o più computer, detti client, che si connettono ad un server per la fruizione di un certo servizio. In questo caso tutti gli studenti registrati, attraverso il proprio dispositivo si connettono alla piattaforma per la consegna degli elaborati. Analogamente il docente si connette al server per verificare la situazione degli elaborati consegnati.

## **2.3 Caratteristiche del sistema**

Questa piattaforma nasce dall'esigenza di automatizzare il processo della consegna degli elaborati da parte degli studenti. Di seguito verranno approfondite le caratteristiche principali.

### **2.3.1 Gestione invio automatico delle e-mail**

Una volta che l'utente si è registrato gli verrà inviata in modo automatico una e-mail riepilogativa contenente username e password che dovrà utilizzare per accedere. Se si registra un gruppo l'e-mail arriverà a tutti i suoi componenti. Tutti i componenti del gruppo riceveranno una ulteriore email automatica per la notifica della avvenuta corretta esecuzione ed anche quando l'admin inserisce un nuovo elaborato da far sottomettere agli studenti.

### **2.3.2 Persistenza delle informazioni**

In informatica la persistenza delle informazioni è una caratteristica da non sottovalutare. Con questa ci si riferisce alla caratteristica dei dati di un programma di sopravvivere all'esecuzione del programma stesso che li ha creati. Non avendo informazioni particolarmente sensibili per l'utilizzo della

piattaforma, è stato concordato di salvare i dati in chiaro in un database relazionale.

### **2.3.3 Estensibilità**

La piattaforma è tesa ad implementare un'attività didattica di grado universitario. Essa è pensata per poter durare nel tempo senza dover apportare grosse modifiche. Anche se tra qualche anno il numero di matricola sarà composto da sette cifre invece delle attuali sei, la piattaforma è in grado di supportare questi cambiamenti. Questo perché ogni studente inserirà la propria matricola completa degli zeri. Ad esempio 0000658818 invece di 658818. Attualmente la piattaforma consente la creazione di gruppi di lavoro composti da un singolo studente o da due studenti. Nel caso, tra qualche anno, si renda necessario ampliare il numero dei componenti del gruppo non risulterà problematica la variazione, in quanto sarà necessario apportare semplici modifiche.

### **2.3.4 Facilità di utilizzo**

La piattaforma ha un'interfaccia grafica facile da comprendere e minimale in modo da permettere di trovare tutti i contenuti con pochi click.

E' anche possibile utilizzare la piattaforma da smartphone e/o tablet, essendo un'applicazione responsive. Una buona applicazione responsive è in grado di adattarsi graficamente in modo automatico al dispositivo coi quali vengono visualizzati (computer con diverse risoluzioni, tablet, smartphone, cellulari), riducendo al minimo la necessità dell'utente di ridimensionare e scorrere i contenuti.

## **2.4 Modalità di utilizzo**

L'applicazione in questione è progettata per essere utilizzata in due diverse modalità:

- **Lato admin:** permette all'admin di avere un login specifico con cui gestire tutte le funzionalità della piattaforma;
- **Lato user:** ogni studente o gruppo ha le proprie credenziali scelte nella fase di registrazione; può caricare e visualizzare i propri elaborati.

## 2.5 Qualità del software

La qualità del software [4] valuta tutto il comportamento di un software durante la realizzazione dei compiti per i quali viene usato. A seconda di quali siano questi compiti la qualità si misura dando importanza diversa a diversi fattori:

- **Stabilità:** è la capacità di un programma di gestire gli imprevisti e i malfunzionamenti senza bloccarsi o compromettere i dati. La stabilità del software si garantisce creando programmi predisposti ad affrontare situazioni anomale.
- **Sicurezza:** evita accessi indesiderati ai dati e alle funzioni del programma. Questa caratteristica assume molta importanza nella gestione di dati sensibili, ad esempio dati contabili o personali. La sicurezza si garantisce utilizzando utenti, permessi e opportune procedure di verifica.
- **Usabilità:** si riferisce alla facilità, velocità ed intuitività d'uso del programma. L'usabilità è fondamentale se il software deve essere usato da utenti senza apposita formazione, ad esempio operatori manuali o il pubblico di un sito web. Per ottenere un programma usabile esistono opportuni criteri e verifiche da applicare alle interfacce utente.
- **Efficienza:** misura la velocità con cui il programma esegue i suoi calcoli producendo i risultati attesi. L'efficienza di un software dipende da come si codificano i dati, da come si salvano e da come si manipola-

no, quindi per garantire alta l'efficienza del software è essenziale poter contare su un alto profilo tecnico degli sviluppatori.

- **Portabilità:** un sistema è portabile se è in grado di funzionare in ambienti diversi. È diventato un aspetto fondamentale perché consente di avere vantaggi economici, in quanto si possono ammortizzare i costi trasportando l'applicazione in diversi ambienti. Si usano strumenti e tecniche appositamente pensate per dare luogo ad oggetti portabili.
- **Correttezza:** un sistema software si dice corretto se si comporta esattamente secondo quanto previsto dalla sua analisi dei requisiti. E' una qualità assoluta (un sistema è corretto o non lo è) e sostanzialmente impossibile da misurare (verificare); non è possibile stabilire con certezza che un sistema sia corretto.
- **Flessibilità:** è la facilità con cui è possibile modificare il codice per adattarlo a nuove richieste. Oltre alle richieste analoghe a quelle per la manutenibilità, le specifiche possono prevedere la possibilità di configurare il software tramite meccanismi utilizzabili da una utenza non professionale (file di configurazione, opzioni, ecc...).
- **Robustezza:** è la misura in cui il sistema si comporta in modo ragionevole in situazioni impreviste, non contemplate dalle specifiche. Situazioni di questo tipo in genere riguardano errori ed eccezioni di varia natura (dati di input scorretti, fallimenti di componenti software o hardware esterni al sistema e interagenti con esso, e così via).

L'immagine 2.1 seguente, schematizza le caratteristiche da conferire alla piattaforma web.

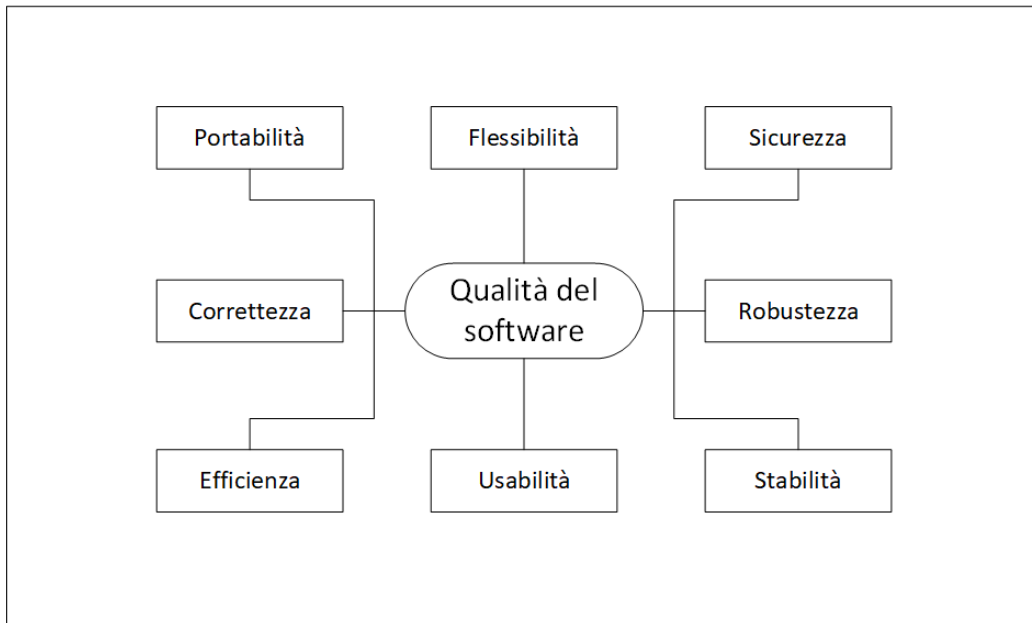


Figura 2.1: Panoramica caratteristiche qualità del software

# Capitolo 3

## Progettazione

La progettazione di un sistema informatico non deve mirare a soddisfare solo le specifiche attuali ma deve prevedere anche quelle future, poiché la capacità di prevedere i cambiamenti a cui il software sarà sottoposto durante il suo ciclo di vita determina la sua semplicità di manutenzione e la sua riusabilità. A tale proposito la piattaforma gestisce il cambio dell'anno solare effettuando la lettura dell'anno corrente dalla data di sistema ed il suo controllo durante la registrazione di ogni gruppo. Solo durante la registrazione del primo gruppo creerà la cartella relativa all'anno corrente che conterrà i materiali consegnati dai gruppi presenti. Nel seguente capitolo verranno descritte le scelte progettuali, trasformando i requisiti analizzati precedentemente in logiche applicative e procedure di utilizzo.

Il seguente sistema informativo verrà descritto tramite diversi diagrammi UML, quali diagrammi dei casi d'uso (Figure 3.2 e 3.3) e di attività (Figure 3.4 e 3.5).

### 3.1 Architettura generale

L'applicazione realizzata si basa su tre fondamentali elementi:

- **Il database:** strumento di fondamentale importanza in grado di raccogliere tutte le informazioni necessarie inserite in input e di estrapolarle

in base a quelle più richieste.

- **Interfaccia Web:** indispensabile per visualizzare una rappresentazione dei dati salvati nel db. E' lo strumento grazie al quale l'utente può interfacciarsi con l'applicazione, ottimizzando la visualizzazione su vari device.
- **Server Web:** è il componente che si occupa della gestione logica del sistema, fornendo servizi al client e svolgendo operazioni sul database.

Nella seguente figura 3.1 si può vedere un esempio esplicativo di utilizzo tra server e client.

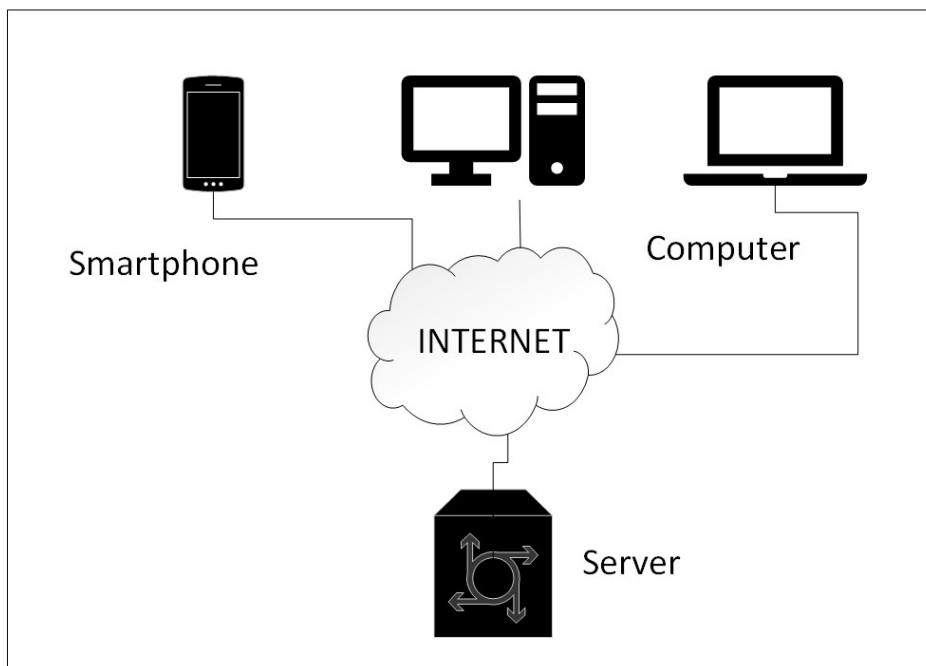


Figura 3.1: Esempio funzionamento server



## 3.2 Progettazione della base dati

Per la memorizzazione dei dati inseriti dagli utenti è stato utilizzato un database relazionale, ossia un archivio per la registrazione delle informazioni che si fonda sul modello relazionale. Il principio base del modello relazionale è che tutte le informazioni siano rappresentate da valori inseriti in relazioni (tabelle); dunque un database relazionale è un insieme di tabelle contenenti valori, ed il risultato di qualunque interrogazione sui dati può essere rappresentato anch'esso da ulteriori tabelle (relazioni). La fase di progettazione di un database comprende la stesura di un diagramma denominato E-R, da "entità-relazione". Il DBMS [8] (Database Management System) che si è scelto di utilizzare è MySQL [13] di Oracle. Questo è molto popolare tra gli sviluppatori web, poichè risulta ideale per la realizzazione di siti web, avendo velocità elevate. E' un sistema open-source, quindi la pubblicazione del codice sorgente è aperta a tutti.

### 3.2.1 Schema Entity-Relationship

La figura 3.6 è lo schema E-R che rappresenta l'applicativo sviluppato.

L'entità **ELABORATO** registra le informazioni relative ad un certo esercizio da consegnare, ad esempio nome, data di consegna, anno, testo, ecc... . Un elaborato è consegnabile 0-N volte da ciascun gruppo o studente (il numero massimo di consegne è scelto dal docente). Chiave primaria dell'entità **ELABORATO** è l'**id**.

L'entità **CONSEGNA** rappresenta la sottomissione del singolo elaborato. Ha una cardinalità di 1-1 verso **ELABORATO** perchè la consegna è relativa ad uno specifico elaborato così come è 1-1 verso **GRUPPO** poichè è relativo ad un certo gruppo o studente. Ha come chiave primaria la chiave composta tra l'attributo **id** di **ELABORATO**, lo **username** di **GRUPPO** e un numero progressivo che indica l'ordine di consegna.

E' presente la gerarchia **UTENTE** (entità padre) che racchiude studenti e

docente. E' di tipo totale-esclusiva, avente come id **username** e password come attributo semplice. La gerarchia si divide in due entità figlie:

- **GRUPPO**: raccoglie le informazioni di tutti gli studenti registrati alla piattaforma come dati anagrafici (attributo composto) e matricola, username e password ereditati da UTENTE. Questa entità ha una cardinalità di 0-N verso CONSEGNA poichè il gruppo o lo studente singolo ha la possibilità di effettuare più consegne;
- **DOCENTE**: rappresenta la figura del docente avente come attributi l'id **username** e password ereditati da UTENTE.

L'entità **LOGS** raccoglie le informazioni di ogni gruppo riguardo l'ultimo accesso. Ha una chiave primaria incrementale e come attributi semplici l'informazione relativa all'accesso di un certo gruppo, la data e l'ora.

### 3.2.2 Schema Logico

Una volta costruito lo schema concettuale (modello E-R) occorre tradurre lo schema ottenuto in uno schema logico ad esso equivalente. Di seguito è possibile visionare la ristrutturazione derivato dallo schema E-R.

**ELABORATO** (Id, Nome, Data\_Cons, Anno, Materiale, Testo, Note, Punti, Max\_Cons, Attivo);

**CONSEGNA** (Num, Data, File, Similarity, Voto, Output\_Compilazione, Output\_Test, Launched, Id: ELABORATO, Username: GRUPPO);

**GRUPPO** (Username, Password, Anno, Matricola1, Nome1, Cognome1, E-Mail1, Matricola2, Nome2, Cognome2, E-Mail2);

**DOCENTE** (Username, Password);

**LOGS** (Idlog, Gruppo, Data, Ora).

### 3.2.3 Struttura delle tabelle

Le tabelle visionabili 3.1, 3.2, 3.3, 3.4 e 3.5 descrivono il modo in cui sono composte le tabelle con ciascun attributo di ciascuna entità ed il relativo tipo con la lunghezza, quando necessaria.

Tabella 3.1: Struttura della tabella ELABORATO

<b>Attributo</b>	<b>Tipo</b>
<i>idelaborato</i>	int(6)
nomefile	char(20)
titolo	text
dataConsegna	date
anno	int(4)
materiale	char(50)
note	char(50)
attivo	tinyint(1)
max_cons	tinyint(1)

Tabella 3.2: Struttura della tabella UTENTE - entità padre

<b>Attributo</b>	<b>Tipo</b>
<i>username</i>	varchar(15)
password	varchar(15)

Tabella 3.3: Struttura della tabella GRUPPO - entità figlia

<b>Attributo</b>	<b>Tipo</b>
anno	int(4)
cogn1	text
nome1	text

Attributo	Tipo
email1	varchar(45)
matr1	varchar(10)
cogn2	text
nome2	text
email2	varchar(45)
matr2	varchar(10)

La tabella **DOCENTE**, essendo figlia dell'entità **UTENTE**, non ha nessun attributo se non quelli che eredita dall'entità padre.

Tabella 3.4: Struttura della tabella CONSEGNA

Attributo	Tipo	Collegamenti a
<i>idcons</i>	int(6)	
cod_gruppo	varchar(15)	gruppo (idgruppo)
cod_elaborato	int(6)	elaborato (idelaborato)
dataConsegna	date	
nomeFile	varchar(15)	
voto	int(2)	
out_compilazione	text	
out_test	text	
launched	int(1)	
similarity	text	

Tabella 3.5: Struttura della tabella LOGS

Attributo	Tipo	Null
<i>idlog</i>	int(6)	No
gruppo	varchar(9)	Sì
data	date	No
ora	time	No

## 3.3 Progettazione delle funzioni

Nella realizzazione del progetto si è cercato di creare dei semplici collegamenti tra le varie pagine, con un numero minimo di passaggi affinché l'utente possa completare l'inserimento dell'elaborato nella maniera più chiara possibile, anche per non dover sovraccaricare il sistema di risorse superflue.

Per specificare nella maniera corretta le funzioni ed il loro collegamento verranno illustrate nelle prossime pagine i relativi diagrammi attraverso il formalismo UML [21].

### 3.3.1 Diagrammi casi d'uso

Questa tipologia di diagrammi UML analizza i casi d'uso del sistema realizzato. In questo caso verranno analizzati i principali casi d'uso che il software dovrà gestire. Gli attori analizzati sono: l'utente e l'admin.

#### Diagramma d'uso con user non registrato

Nel diagramma 3.2, viene illustrato il caso in cui l'utente non è registrato. Infatti senza questa operazione l'utente può solamente visionare la homepage del sito senza ovviamente poter sottomettere elaborati. L'admin potrà gestire il sistema, approfondito nel diagramma 3.3.

#### Diagramma d'uso con user registrato

Nel diagramma 3.3, viene analizzato il caso d'uso in cui l'utente è registrato. Come è possibile vedere l'utente, una volta registrato e loggato, ha la possibilità di visualizzare, sottomettere i propri esercizi, verificare che soddisfino le specifiche dell'elaborato attraverso una corretta compilazione ed esecuzione, per poi vedersi assegnato il relativo punteggio. Se ci sono errori anche solamente in una delle due fasi, lo studente dovrà sottoporre una nuova versione dell'elaborato.

L'admin, in questo caso il docente, ha la possibilità di visualizzare tutti gli esercizi, gestire il sistema, partendo dalla gestione dei gruppi, caricare gli

esercizi e consentire allo studente di poter consegnare esercizi, impostando per ognuno di essi una deadline entro la quale caricare l'elaborato ed un limite massimo di sottomissioni.

### **3.3.2 Diagrammi di attività**

Il diagramma di attività è un tipo di diagramma che permette di descrivere un processo attraverso dei grafi in cui i nodi rappresentano le attività e gli archi l'ordine con cui vengono eseguite.

#### **Diagramma attività: registrazione e login dell'utente**

Come visionabile nel diagramma 3.4, l'utente deve essere necessariamente registrato. Questo avviene in un apposito form specificando prima il numero di componenti e successivamente valorizzando i vari campi richiesti come: nome, cognome, matricola, e-mail, mentre la password è unica indipendentemente dal numero dei componenti. Nella casella postale indicata durante la fase di registrazione verrà inviata in automatico una e-mail riepilogativa contenente username e password. Una volta completata la fase di registrazione potrà effettuare login attraverso l'apposita pagina.

#### **Diagramma attività: sottomissione elaborato**

Come visionabile del diagramma in figura 3.5, lo studente una volta effettuato il login, può scegliere gli esercizi tra quelli da consegnare. Dopo aver caricato il proprio esercizio avverranno le due fasi di controllo: compilazione ed esecuzione. La compilazione verrà fatta qualche secondo dopo aver sottomesso l'esercizio. Se la compilazione è andata a buon fine passa allo step successivo, l'esecuzione. Se anche questa fase è riuscita correttamente l'utente potrà avere la sua valutazione secondo una serie di test pre-caricati.

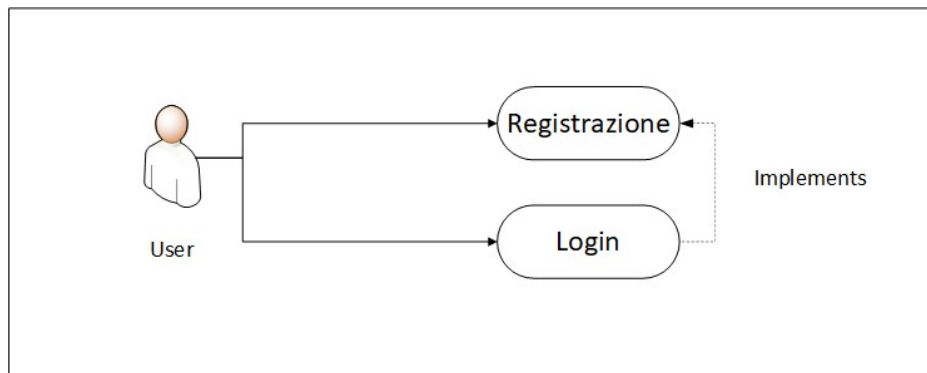


Figura 3.2: Diagramma casi d'uso con user non registrato

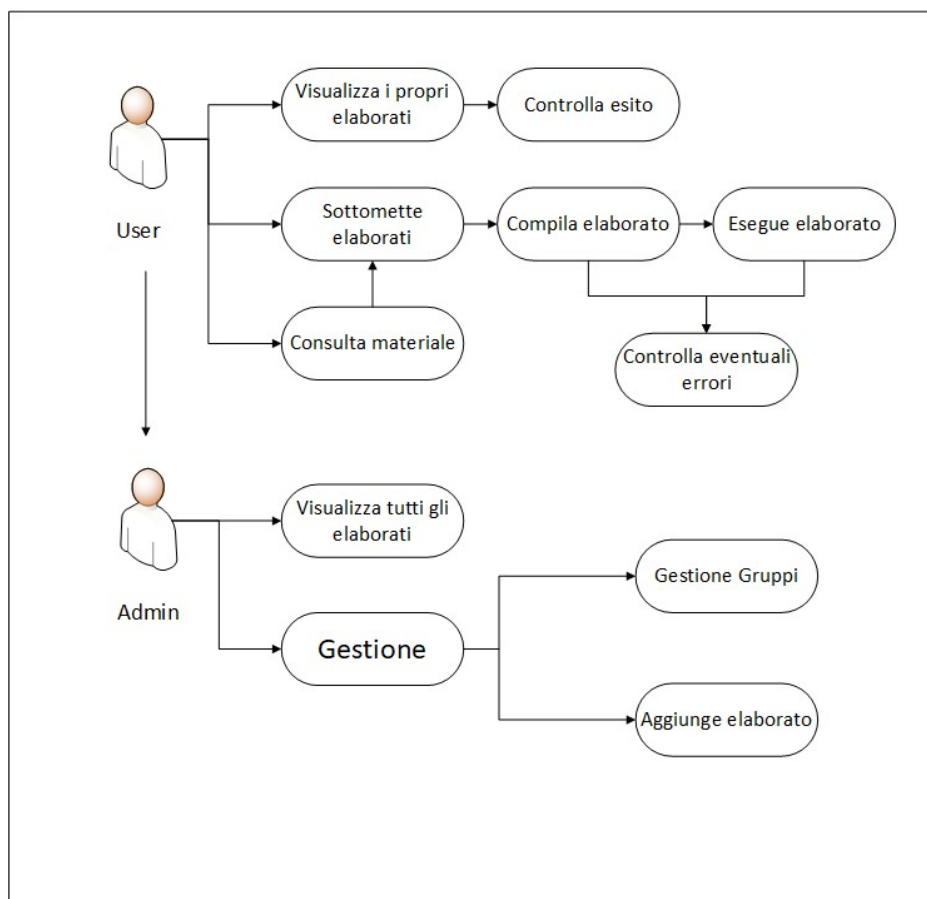


Figura 3.3: Diagramma casi d'uso con user registrato

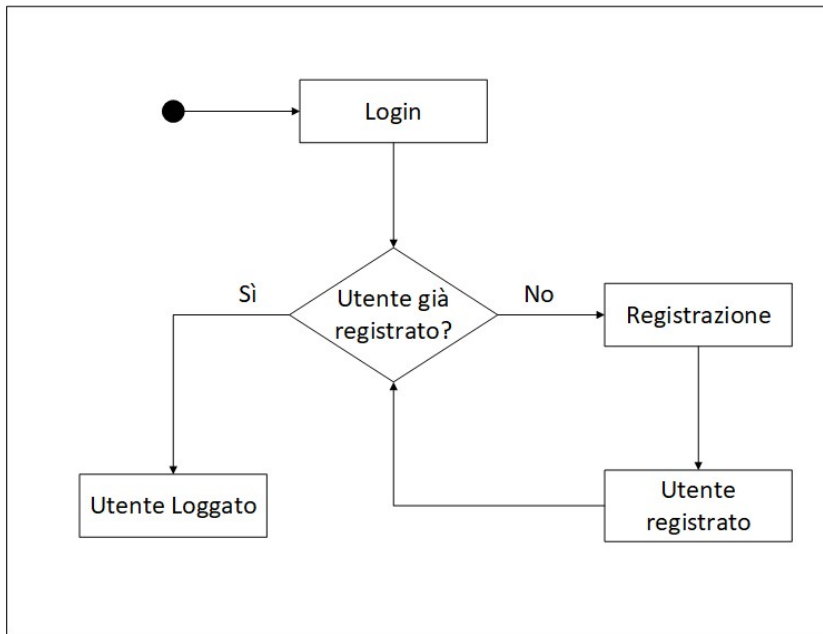


Figura 3.4: Diagramma di attività: registrazione e login user

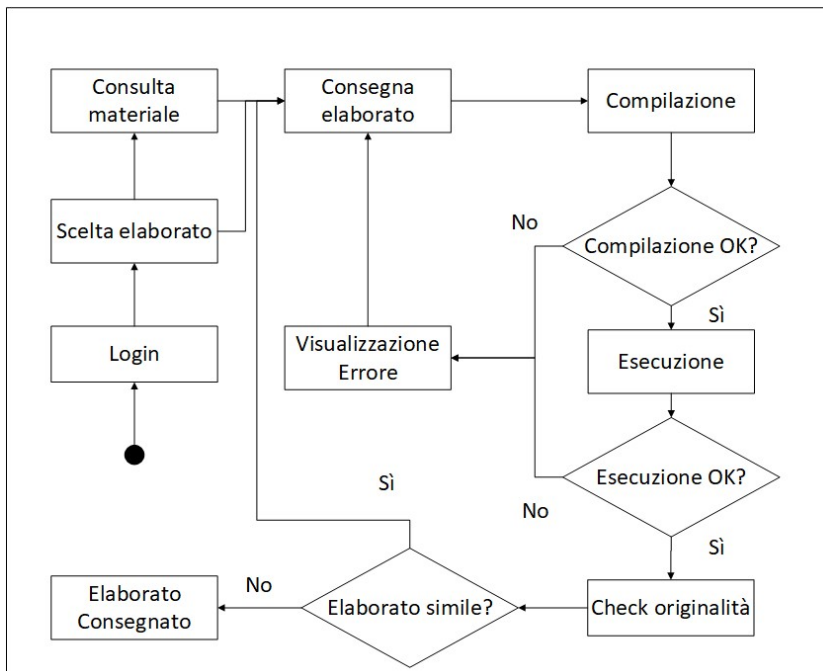


Figura 3.5: Diagramma di attività: sottomissione elaborato



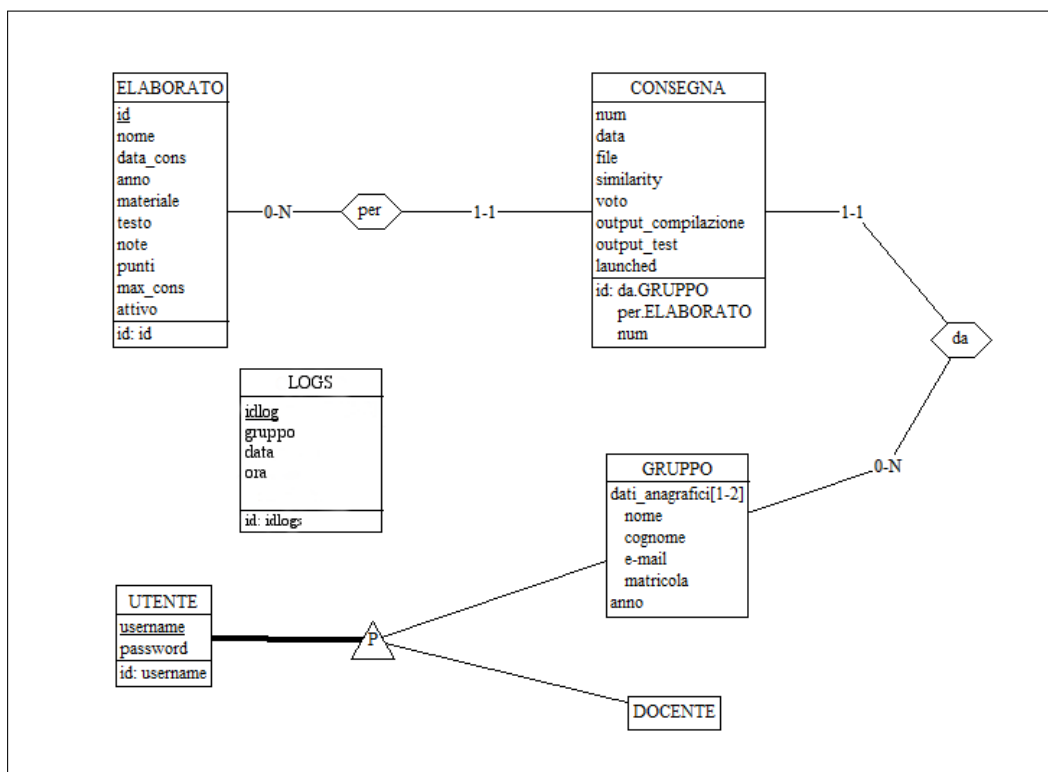


Figura 3.6: Diagramma E-R

## 3.4 Progettazione dell'interfaccia

L'interfaccia web è composta da varie schermate con componenti minimali. Ogni pagina del sito web permette di effettuare poche operazioni proprio per rendere semplice l'esperienza all'utente. Realizzarla in questo modo però non è stato immediato e semplice. Si è partito dalla realizzazione di "bozze" di mockup di ogni pagina. Le pagine sono state raffinate fino a raggiungere lo stile desiderato.

### 3.4.1 Mockup

I Mockup sono una realizzazione a scopo illustrativo o meramente espositivo di un oggetto o un sistema, senza le complete funzioni dell'originale; un moc-

kup può rappresentare la totalità o solo una parte dell'originale di riferimento essere in scala reale oppure variata. Per la realizzazione di questi si è deciso di utilizzare il software Balsamiq Mockup 3 [2]. Di seguito verranno esposti vari mockup da desktop e mobile delle pagine principali del sito web.

### Mockup homepage da desktop

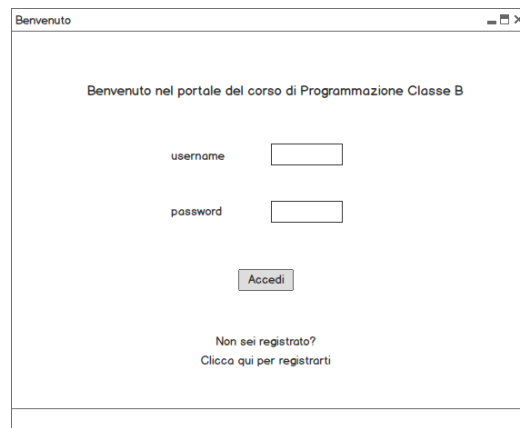


Figura 3.7: Mockup homepage desktop

### Mockup homepage da mobile

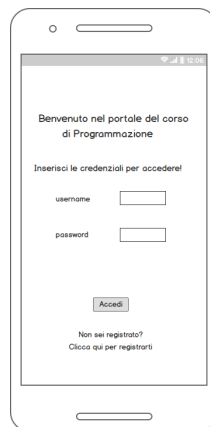


Figura 3.8: Mockup homepage smartphone

## Mockup registrazione da desktop

The image shows two desktop registration form mockups side-by-side, both titled "Registrazione".

The left mockup is titled "Registrazione" and has the subtitle "Registri per la consegna di esercizi". It features two radio buttons: "1 studente" (selected) and "2 studenti". Below are five input fields: "Nome studente", "Cognome studente", "Mail studente", "Matricola studente", and "Password". A "Registral" button is at the bottom.

The right mockup is also titled "Registrazione" with the same subtitle. It features two radio buttons: "1 studente" and "2 studenti" (selected). Below are ten input fields: "Nome studente 1", "Cognome studente 1", "Mail studente 1", "Matricola studente 1", "Nome studente 2", "Cognome studente 2", "Mail studente 2", "Matricola studente 2", and "Password". A "Registral" button is at the bottom.

Figura 3.9: Mockup registrazione da desktop

## Mockup registrazione da mobile

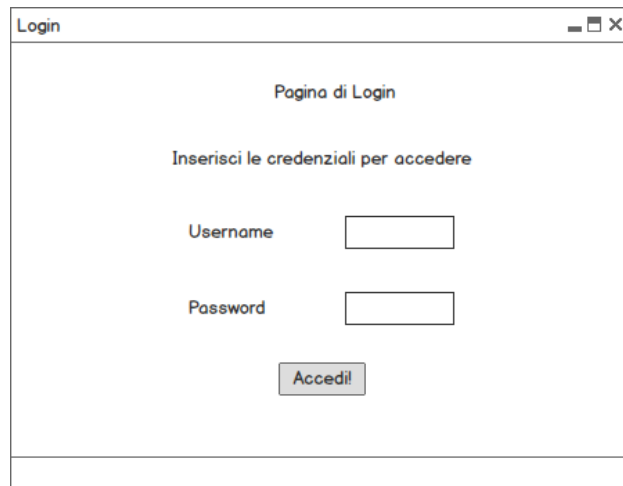
The image shows two smartphone registration form mockups side-by-side, both titled "Pagina di Registrazione".

The left mockup is titled "Pagina di Registrazione" and has the subtitle "Pagina di Registrazione". It features two radio buttons: "Uno studente" (selected) and "Due studenti". Below are five input fields: "Nome studente", "Cognome studente", "Mail studente", "Matricola studente", and "Password". A "Registral" button is at the bottom. Below the button, it says "Tutti i campi sono obbligatori".

The right mockup is also titled "Pagina di Registrazione" with the same subtitle. It features two radio buttons: "Uno studente" and "Due studenti" (selected). Below are ten input fields: "Nome studente 1", "Cognome studente 1", "Mail studente 1", "Matricola studente 1", "Nome studente 2", "Cognome studente 2", "Mail studente 2", "Matricola studente 2", and "Password". A "Registral" button is at the bottom. Below the button, it says "Tutti i campi sono obbligatori".

Figura 3.10: Mockup registrazione smartphone

## Mockup login da desktop



A desktop window titled "Login" with standard window controls (minimize, maximize, close) in the top right corner. The content area is titled "Pagina di Login" and contains the instruction "Inserisci le credenziali per accedere". Below this, there are two input fields: "Username" and "Password". At the bottom center is a button labeled "Accedi".

Figura 3.11: Mockup login desktop

## Mockup login da mobile



A mobile phone mockup showing the login page. The phone has a notch at the top and a home indicator at the bottom. The screen displays the same "Pagina di Login" content as the desktop version, including the title, instruction, "Username" and "Password" input fields, and the "Accedi" button. The status bar at the top shows signal strength, Wi-Fi, and the time 12:15.

Figura 3.12: Mockup login smartphone

## Mockup homepage user da desktop

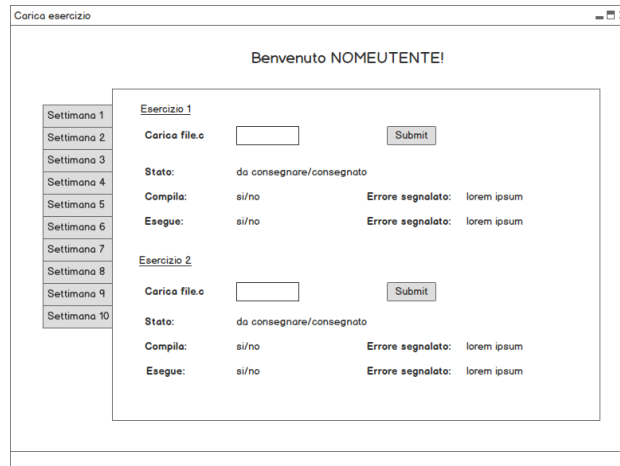


Figura 3.13: Mockup homepage user desktop

## Mockup homepage user da mobile

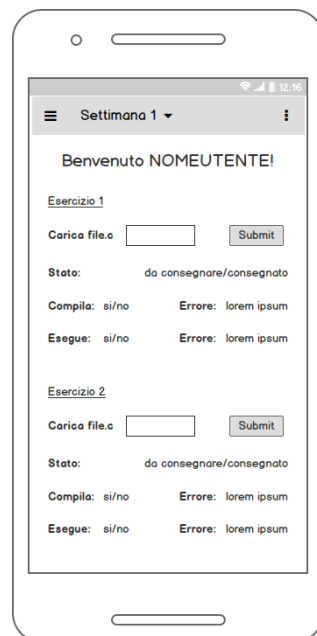


Figura 3.14: Mockup homepage user smartphone

## Mockup conferma da desktop



Figura 3.15: Mockup conferma desktop

## Mockup conferma da mobile

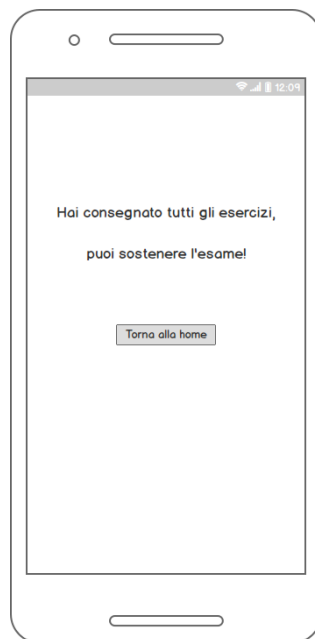


Figura 3.16: Mockup conferma smartphone

### 3.4.2 Screenshot

Dopo aver analizzato le varie pagine effettuando gli opportuni mockup si è giunti alla realizzazione dell'interfaccia finale del sistema, visionabili nelle figure qui sotto riportate. Verranno suddivisi in due sezioni, la prima che descrive le funzioni per lo studente e la seconda le funzioni per il docente.

#### Screenshot lato studente



Figura 3.17: Screenshot homepage

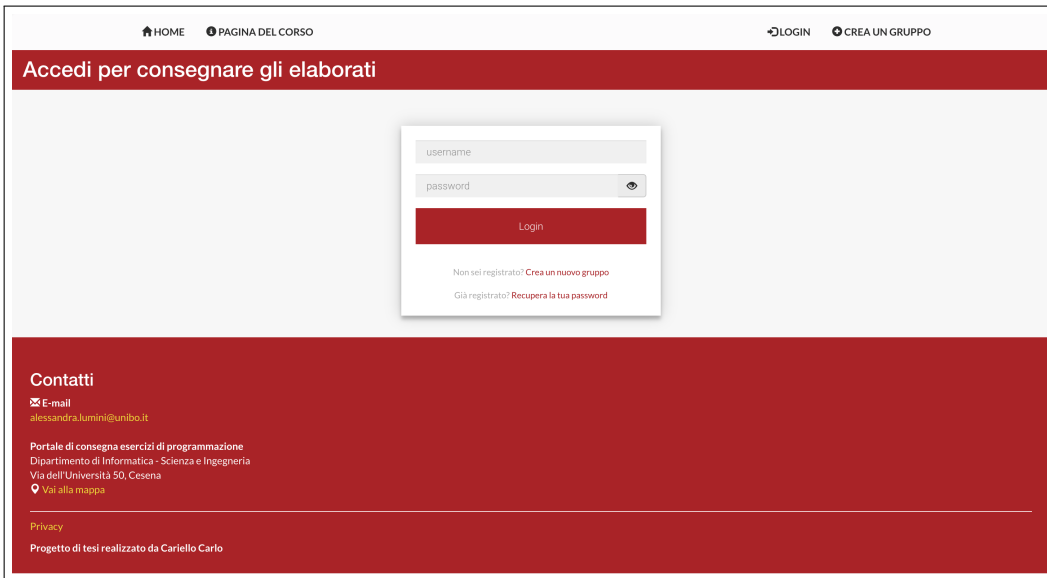


Figura 3.18: Screenshot che mostra la fase di login

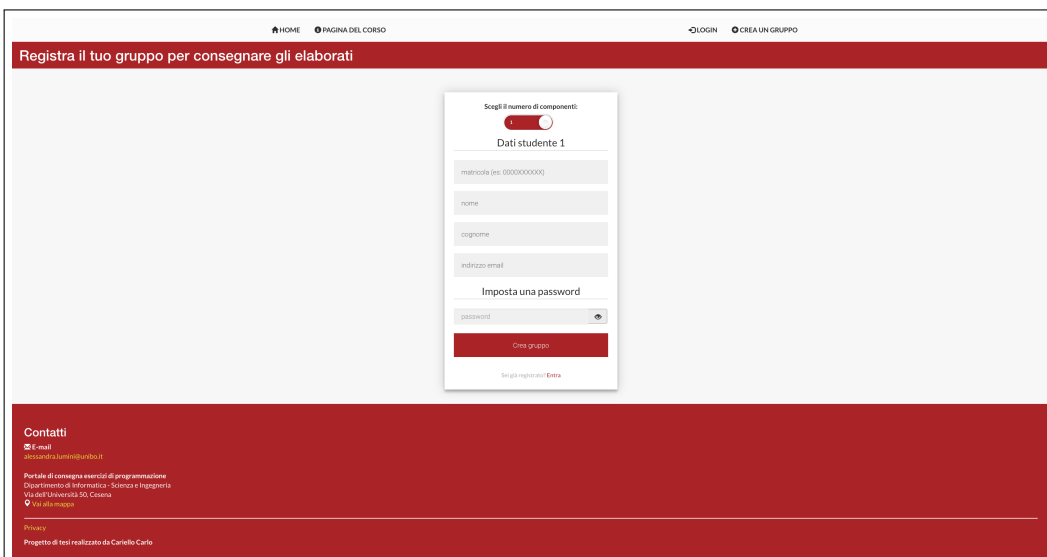


Figura 3.19: Screenshot relativo alla registrazione di un singolo studente



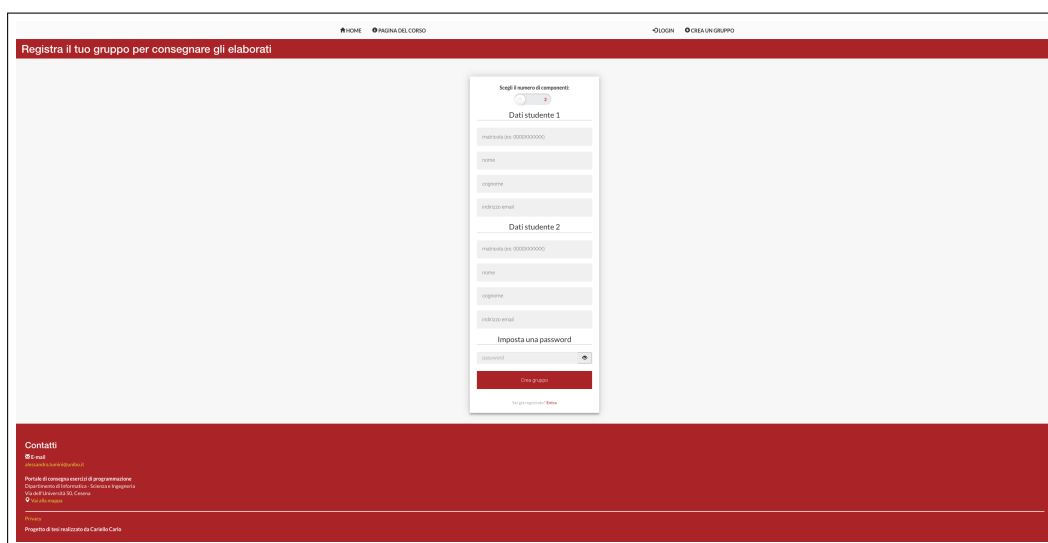


Figura 3.20: Screenshot relativo alla registrazione di due studenti

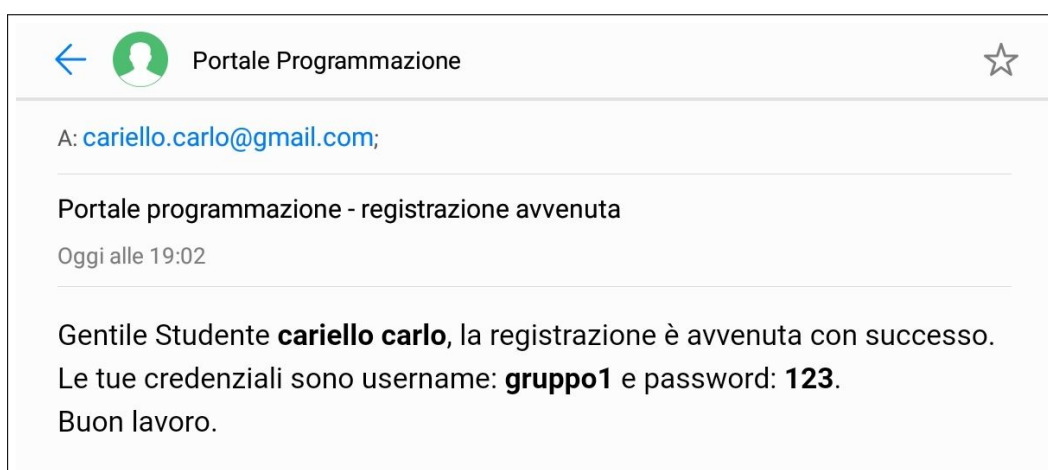


Figura 3.21: Screenshot relativo alla ricezione della e-mail di registrazione

HOME PAGINA DEL CORSO DASHBOARD LOGOUT

Area Riservata → gruppo1 ultimo accesso: 22-06-2020 ore 15:11

### Elaborati consegnabili

Id	Titolo	Testo	Deadline	Anno	Materiale	Note	Test	Libreria	Max consegne	
1	Hello World!	ElaboratoTesting.pdf	16-07-2020	2020	materiale.txt	Note.txt			15	Consegna

---

**Contatti**

E-mail  
alessandra.lumini@unibo.it

Portale di consegna esercizi di programmazione  
Dipartimento di Informatica - Scienza e Ingegneria  
Via dell'Università 50, Cesena  
[Vai alla mappa](#)

---

Privacy  
Progetto di tesi realizzato da Carliello Carlo

Figura 3.22: Screenshot che mostra l'area riservata

HOME PAGINA DEL CORSO DASHBOARD LOGOUT

Pagina di consegna del gruppo1

### Consegna dell'elaborato 1

Id	Titolo	Anno	Inserisci file	
1	Hello World!	2020	<input type="button" value="Scegli file"/> Nessun file selezionato	<input type="button" value="Carica"/>

---

**Contatti**

E-mail  
alessandra.lumini@unibo.it

Portale di consegna esercizi di programmazione  
Dipartimento di Informatica - Scienza e Ingegneria  
Via dell'Università 50, Cesena  
[Vai alla mappa](#)

---

Privacy  
Progetto di tesi realizzato da Carliello Carlo

Figura 3.23: Screenshot che mostra il caricamento dell'elaborato

HOME PAGINA DEL CORSO CONSEGNA LOGOUT

Area Riservata → gruppo1 → Situazione elaborati

### Riepilogo degli elaborati consegnati

Id	Titolo	File consegnato	Data consegna	Anno	Voto	Completazione
1	Hello World!	esercizio1.c	14-06-2020	2020	0	●
1	Hello World!	esercizio1.c	16-06-2020	2020	0	●

**Legenda**

- Fase di compilazione conclusa correttamente - Fare clic per lanciare la versione eseguibile
- Fase della compilazione fallita - Fare clic per visualizzare gli errori rilevati dal compilatore C

**Contatti**

E-mail  
alessandra.lumini@unibo.it

Portale di consegna esercizi di programmazione  
Dipartimento di Informatica - Scienza e Ingegneria  
Via dell'Università 50, Cesena  
Vai alla mappa

Privacy  
Progetto di tesi realizzato da Carliello Carlo

Figura 3.24: Screenshot che illustra la dashboard

```
In function 'main':
11:18: error: expected ';' before 'char'
 11 | float costo_iniz
    |           ^
    |           ;
 12 | char categoria;
    |     ~~~~
16:15: error: 'costo_iniz' undeclared (first use in this function)
 16 | scanf("%f", &costo_iniz);
    |           ^~~~~~
16:15: note: each undeclared identifier is reported only once for each function it appears in
```

Figura 3.25: Screenshot relativo alla visualizzazione dell'errore di compilazione

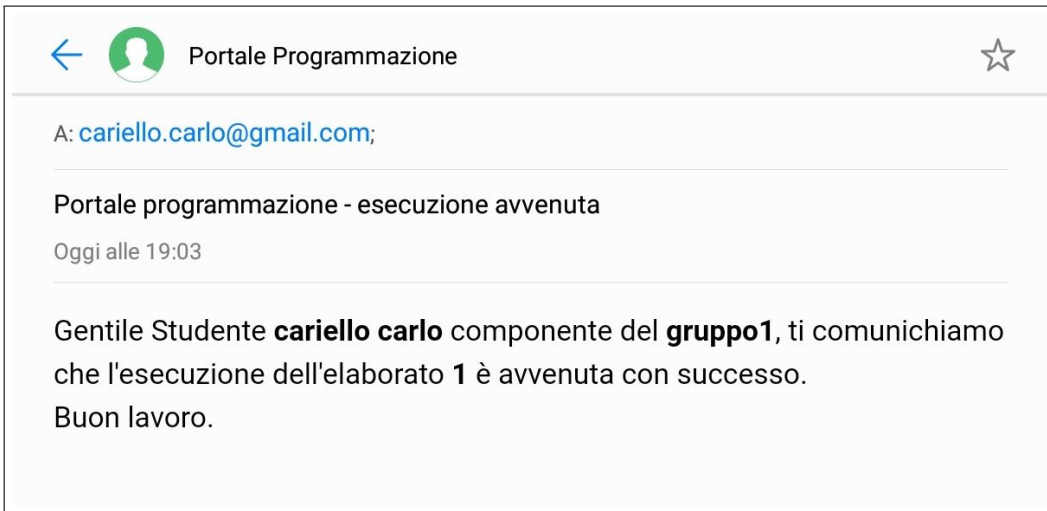


Figura 3.26: Screenshot relativo alla ricezione delle e-mail per l'esecuzione

## Screenshot lato docente



Figura 3.27: Screenshot homepage docente

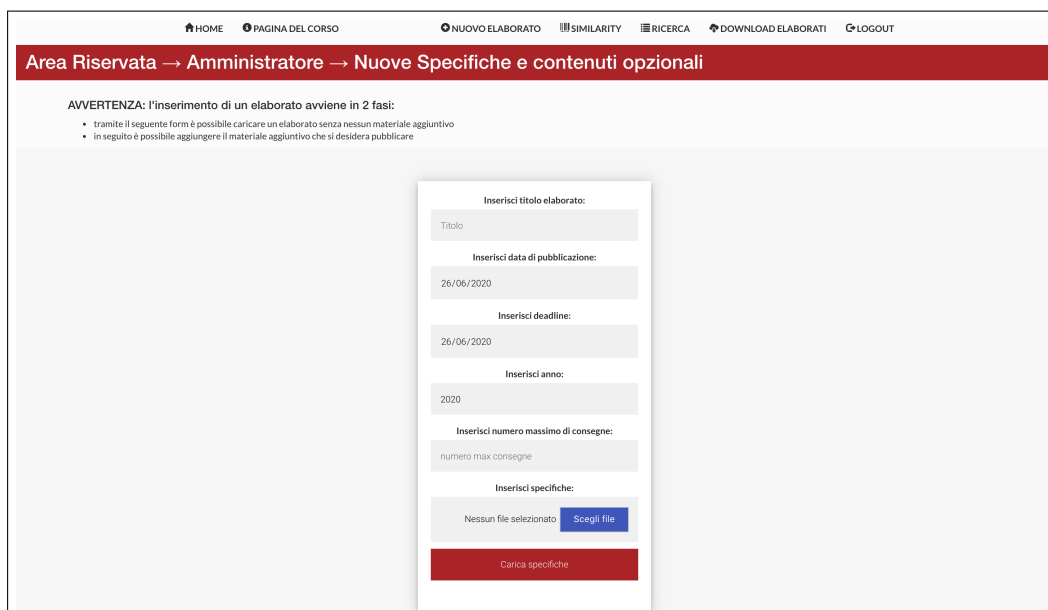


Figura 3.28: Screenshot relativo all'inserimento di un nuovo elaborato

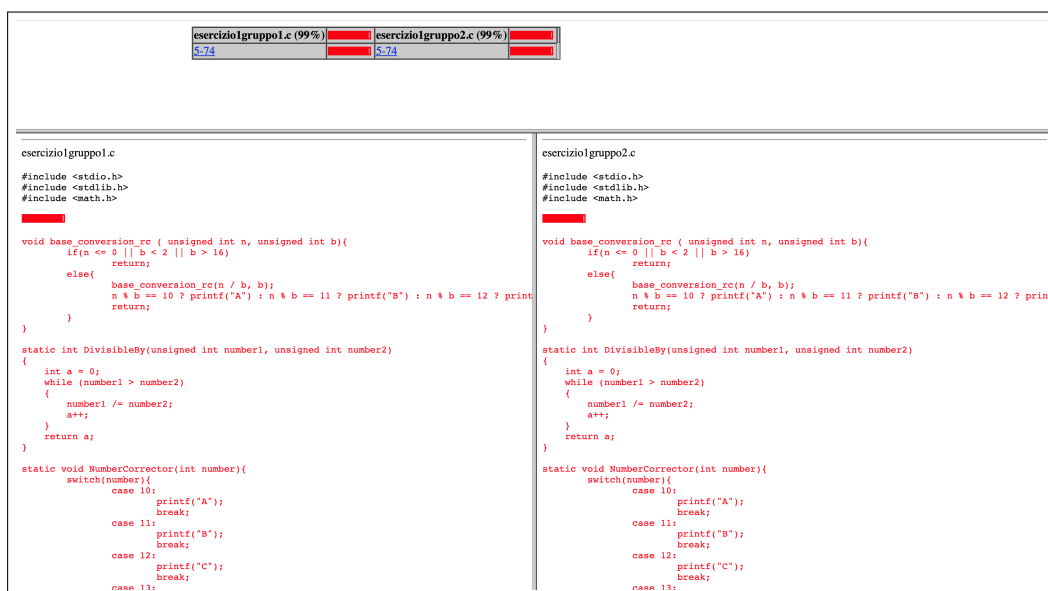


Figura 3.29: Esempio che mostra l'applicazione della funzione di similarity su due elaborati uguali, consegnati da due gruppi diversi

Area Riservata → Amministratore							
<i>Riepilogo degli elaborati consegnati</i>							
ID	Gruppo	Esercizio	Data consegna	Nome File	Voto	Test	Compilazione
1	gruppo1	1	14-06-2020	esercizio1.c	0	0	●
2	gruppo1	1	14-06-2020	esercizio1.c	0	1	●
3	gruppo2	1	19-06-2020	esercizio1.c	0	0	●
4	gruppo2	1	19-06-2020	esercizio1.c	0	1	●
5	gruppo3	1	19-06-2020	esercizio1.c	0	0	●
6	gruppo3	1	19-06-2020	esercizio1.c	0	1	●
7	gruppo4	1	19-06-2020	esercizio1.c	0	0	●
8	gruppo4	1	19-06-2020	esercizio1.c	0	1	●
9	gruppo5	1	20-06-2020	esercizio1.c	0	0	●
10	gruppo6	1	20-06-2020	esercizio1.c	0	1	●
11	gruppo6	1	20-06-2020	esercizio1.c	0	0	●
12	gruppo6	1	20-06-2020	esercizio1.c	0	0	●
13	gruppo7	1	20-06-2020	esercizio1.c	0	1	●
14	gruppo7	1	20-06-2020	esercizio1.c	0	1	●
15	gruppo7	1	20-06-2020	esercizio1.c	0	0	●
16	gruppo8	1	20-06-2020	esercizio1.c	0	1	●

Figura 3.30: Screenshot che illustra la ricerca degli elaborati

# Capitolo 4

## Implementazione

Le fasi di analisi e progettazione sono risultate fondamentali per l'implementazione del progetto che si è svolta nel rispetto delle scelte prese precedentemente.

### 4.1 Tecnologie usate

HTML5, CSS3 e JavaScript sono di fatto uno standard nel campo degli strumenti di sviluppo per la realizzazione di un sito web, pertanto l'adozione di questi è stata una scelta naturale. Oltre queste tre tecnologie si è utilizzato il linguaggio PHP e la tecnologia Bootstrap per quanto riguarda il lato client. Lato server, invece, si è utilizzato PHP, MySQL e per la gestione del VPS su Aruba oltre all'interfaccia grafica standard anche la modalità SSH tramite terminale. Di seguito sono descritte, suddivise per tecnologie di progettazione e di progetto, le tecnologie citate sopra.

#### 4.1.1 Tecnologie di progettazione

#### 4.1.2 DB-Main

DB-Main [17] è un software open-source per la modellazione dei dati, prodotto da Rever. Prevede diversi processi legati all'ingegnerizzazione dei da-

tabase, tra cui la modellazione dei diagrammi Entity-Relationship, utilizzata in questo progetto. Utile non solo per la creazione del diagramma E/R ma anche per la creazione delle tabelle, grazie alla possibilità di trasformare le entità in query di creazione.

### 4.1.3 DBMS

Il DBMS [8] (Database Management System) che si è scelto di utilizzare è MySQL [13] di Oracle. Di seguito un elenco dei suoi vantaggi:

- **Veloce:** l'obiettivo principale del team che ha sviluppato MySQL è stata la velocità. Così, il software sin dalla prima release ha garantito questa caratteristica;
- **Poco costoso:** è libero sotto la licenza open source e il canone per una licenza commerciale è ragionevole;
- **Facile da usare:** è possibile creare ed interagire con un database MySQL utilizzando alcune affermazioni semplici nel linguaggio SQL, che è il linguaggio standard per la comunicazione con RDBMS;
- **Cross-platform:** MySQL gira pressochè su tutti i sistemi operativi compresi Windows, Linux, Mac OS X e la maggior parte delle varietà di Unix.
- **Sicuro:** consente di fornire agli utenti o gruppi di utenti, alcuni o tutti i privilegi di accesso al database. Le password sono criptate attraverso moderni algoritmi;
- **Personalizzabile:** grazie alla licenza open source consente ai programmatori di sviluppare e migliorare il codice.

### 4.1.4 Lighthouse

Lighthouse [6] è un progetto open-source promosso da Google rivolto a migliorare la qualità delle pagine Web. Si tratta di uno strumento che aiuta il



web-developer a risolvere i problemi più comuni che danneggiano l'esperienza di navigazione dell'utente. Lighthouse si distingue per la sua facilità di utilizzo che lo rende adatto allo studio dell'ottimizzazione lato utente e può essere eseguito su qualsiasi pagina Web pubblica o che richiede autenticazione. Dalla figura 4.1 si possono vedere i valori assegnati da Lighthouse alla home della piattaforma.

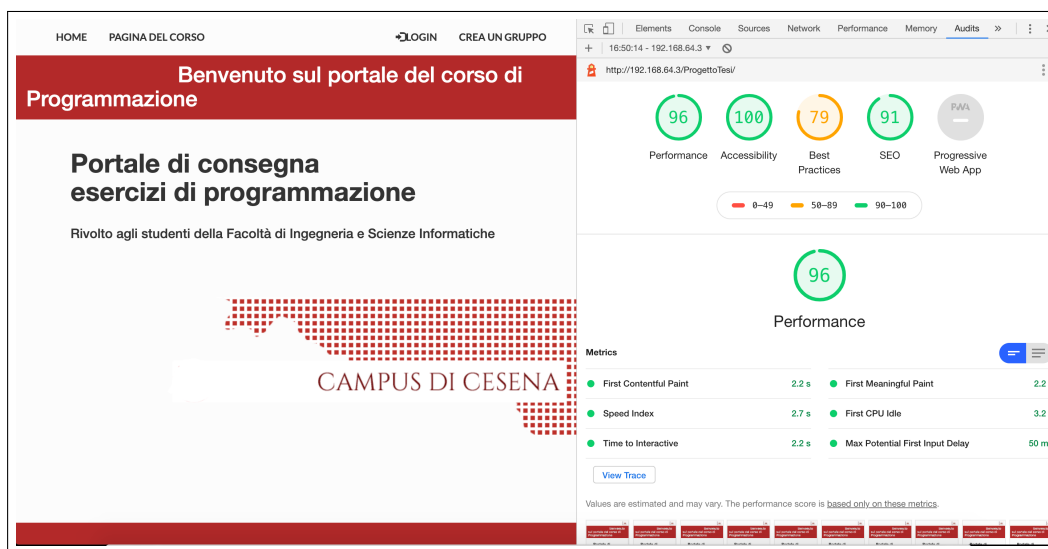


Figura 4.1: Esempio di utilizzo di Lighthouse

### 4.1.5 Tecnologie di progetto

### 4.1.6 Bootstrap

Bootstrap [3] è un toolkit open-source per lo sviluppo tramite HTML, CSS e JS. E' stato di fondamentale utilizzo in quanto ha consentito all'interfaccia di prendere subito forma. Permette di usufruire immediatamente di template utilizzabili sui propri lavori, modificandoli in base alle proprie esigenze. Una ulteriore peculiarità di Bootstrap è stata quella di favorire l'aspetto responsive dell'interfaccia anche dal punto di vista mobile-friendly.

### 4.1.7 JavaScript

JavaScript [16] è stato inizialmente creato per "rendere vive le pagine Web". Le sue linee di codice sono chiamate script. Possono essere scritti direttamente nella sezione HTML di una pagina Web ed eseguiti automaticamente al caricamento della stessa. Gli script vengono forniti ed eseguiti come testo normale. Non hanno bisogno di preparazioni o compilazioni speciali per essere eseguiti. Nello sviluppo del progetto è stato particolarmente utile durante le fasi di registrazione e login dell'utente.

### 4.1.8 XAMPP/LAMPP

XAMPP [5] è il più popolare ambiente di sviluppo PHP. E' una distribuzione di Apache completamente gratuita e semplice da installare, contenente MySQL, PHP e Perl. Questo software è stato di fondamentale aiuto perchè ha permesso di creare in locale il server web, Apache, ed il database basato su MySQL. Inizialmente è stato un buon appoggio per simulare il tutto, ma una volta che il progetto prendeva corpo si è deciso di caricarlo sulla piattaforma di hosting fornita da Aruba.

### 4.1.9 Aruba

In un primo momento si era individuato in Altrivista [19] il servizio di hosting gratuito per il primo prototipo funzionante del progetto. Durante la fase di richiamo del compilatore C è emersa però, l'impossibilità di eseguire sia le direttive necessarie ad ottenere la versione compilata del codice sorgente sia l'assenza del supporto per il lancio di importanti comandi del sistema operativo. Per tali ragioni si è optato per impiegare la piattaforma hosting fornita da Aruba. Tale servizio [1] al primo avvio consente la configurazione di un VPS (Virtual Private Server) che sarà il server dedicato al progetto. La fase di configurazione consente di scegliere tra vari layout corrispondenti a sistemi operativi commerciali ed open-source.

#### **4.1.10 Amministrazione del server tramite SSH**

SSH [23] (Secure SHell) è un protocollo che permette di stabilire una sessione remota cifrata tramite interfaccia a riga di comando con un altro host di una rete informatica. Ha sostituito Telnet, poichè basa la connessione su un protocollo sicuro viene sfruttato negli ambienti enterprise, oltre che dagli utenti più avanzati. SSH è un utile strumento di connessione che si è rivelato rapido ed efficiente rispetto all'interfaccia grafica poco performante disponibile su Aruba. Il funzionamento avviene tramite riga di comando accedendo con le proprie credenziali.

#### **4.1.11 Linguaggi utilizzati**

Sono stati utilizzati HTML5 e CSS3 per la realizzazione dell'interfaccia web, insieme a piccoli elementi di JavaScript. Per quanto riguarda il linguaggio per lo sviluppo del progetto è stato scelto PHP [7] sia lato front-end che back-end. Esso è un linguaggio di scripting, avente una struttura di gestione e di controllo che offre strumenti open-source per tutti i sistemi operativi.

## **4.2 Dettagli implementativi**

### **4.2.1 Invio e-mail**

La piattaforma prevede l'invio di e-mail in alcune sue fasi. Dopo aver effettuato la registrazione lo studente riceverà una e-mail contenente le credenziali per accedere. L'username è scelto in modo automatico dal sistema in base al numero di richieste ricevute, la password è scelta dall'utente.

Una seconda mail che l'utente riceverà è quella della corretta esecuzione.

Una terza mail sarà inviata nel momento in cui l'admin caricherà un nuovo elaborato.

In caso di gruppo l'e-mail verrà inviata a tutti i membri.

## **4.2.2 Recupero password**

Se richiesto nella pagina di login è presente un link dedicato al recupero della password.

## **4.2.3 Compilatore C**

La scelta del server con layout Ubuntu implicitamente offre il pacchetto GCC quale compilatore C. Attraverso gli script php è stato possibile eseguirne le direttive.

## **4.2.4 Testing**

La fase di verifica dei risultati di ogni elaborato avviene mediante l'elaborazione di dati appositamente idonei a consentire di discriminare i vari casi possibili durante il flusso di processo, mostrando all'utente l'esito della verifica.

# Capitolo 5

## Indagine

L'indagine è stata svolta utilizzando un documento di tipo Google Form per la sua praticità, in quanto ha la possibilità di avere domande compilabili e di varie tipologie. La somministrazione del questionario agli utenti è fondamentale per conoscere quali parti vorrebbero migliorate e individuare le sezioni meno accessibili. Una volta terminato il lavoro, la fase di testing risulta molto importante, in quanto, permette di trovare e/o correggere eventuali bug che durante la fase di sviluppo possono essere stati sottovalutati.

I vari test sono stati effettuati, completamente in forma anonima, su tutte le funzionalità che la piattaforma offre quali login, registrazione, consegna, compilazione, esecuzione dell'elaborato e visualizzazione della dashboard.

### 5.1 Soggetti

Sono stati scelti diversi tipi di utenti: studenti iscritti al primo anno, agli anni successivi ed anche lavoratori che hanno frequentato il CdL in Ingegneria e Scienze Informatiche.

Poter effettuare il test su più tipologie di persone ha permesso di creare una panoramica ampia e composta da diversi punti di vista sulla piattaforma, rilevando così, diversi tipi di errori.

Questa scelta è stata fondamentale in quanto poter raccogliere le impressioni

degli studenti che hanno già seguito il corso di Programmazione permette di capire se il sito è strutturato in maniera adeguata ai contenuti trattati durante il semestre e se è adatto allo scopo del corso.

## 5.2 Domande del questionario

Le domande inserite in questa indagine sono state costruite per poter effettuare un'indagine di tipo qualitativo che interessa la piattaforma in tutte le sue caratteristiche, come illustrato nel paragrafo 5.2.3.

Grazie alle domande iniziali è possibile raccogliere alcuni dati personali, così da poter categorizzare i soggetti che partecipano al questionario.

Le tipologie di domande utilizzate sono state le seguenti:

- *Domande Aperte*: per dare la possibilità all'utente di esprimersi come preferisce riguardo a suggerimenti o questioni più importanti da approfondire;
- *Domande Chiuse*: in cui rispondere in modo standardizzato e sintattico;
- *Domande Linear Scale*: per avere una valutazione quantitativa sull'usabilità e la semplicità della piattaforma. Il punteggio va da un minimo di 0 ad un massimo di 3.

### 5.2.1 Sezione dei dati personali

- Quale anno frequenti del CdL di Ingegneria e Scienze Informatiche?
- Quale è il tuo sesso?
- Quale è il tuo anno di nascita?

## 5.2.2 Sezione dedicata al progetto

### FeedBack

- Quale è stata la tua prima impressione?
- In quali parti hai riscontrato maggiore difficoltà?
- Quali sono delle possibili soluzioni ai problemi riscontrati?

**Generale:** nei suoi aspetti più generali sono da valutare i titoli esplicativi, e se i contenuti sono strutturati secondo una logica semplice, ordinata e di facile utilizzo.

- I contenuti sono strutturati in maniera semplice ed intuitiva?
- Il testo è facilmente leggibile?
- Le diverse sezioni sono dotate di un titolo significativo che illustra il tipo di informazione visualizzata?

**Navigazione:** dare un giudizio riguardo la navigazione tra le differenti pagine grazie all'utilizzo di componenti semplici ed intuitivi.

- Tutte le pagine presentano opzioni di navigazione?
- Sono presenti funzionalità di annullamento delle operazioni?
- Il menù di navigazione è efficace?
- Sono presenti funzionalità per ritornare alla homepage?

**Interfaccia:** dare un giudizio riguardo l'aspetto e lo stile grafico utilizzato.

- L'applicazione ha un aspetto gradevole?
- Lo stile grafico è adeguato a rappresentare le funzionalità delle diverse schermate?
- Ci sono situazioni in cui la navigazione è facilitata dalla grafica?

- Ci sono situazioni in cui la navigazione è compromessa dalla grafica?

**Funzionalità:** dare un giudizio riguardo le funzionalità implementate su questa piattaforma.

- I tempi della compilazione dell'elaborato sono ragionevoli?
- I tempi della esecuzione dell'elaborato sono ragionevoli?
- E' semplice usufruire del materiale messo a disposizione per ogni elaborato?
- E' stato semplice effettuare la consegna di un elaborato?

**Suggerimenti:** Segnalare quanto sia possibile o necessario migliorare le seguenti parti.

- Interfaccia grafica e layout;
- Menù di navigazione;
- Funzionalità dell'applicazione;
- Organizzazione dei contenuti;

### 5.2.3 Sezione conclusiva

- Scrivere eventuali considerazioni personali;
- Quali funzionalità/possibilità aggiungeresti?

## 5.3 Risultati

L'indagine è stata sottoposta ad undici soggetti, tutti appartenenti al contesto del CdL di Ingegneria e Scienze Informatiche. Secondo uno studio [14], facendo testare una piattaforma a cinque utenti si rileva l'85% dei problemi di usabilità del servizio. Questo perchè più utenti verranno intervistati, più



sarà alta la probabilità di trovare le stesse osservazioni. Si otterrà quindi una buona indagine qualitativa, secondo la letteratura, che permetterà di poter migliorare significativamente la piattaforma sotto molti aspetti.

Come spiegato in uno studio [15], per ottenere una significatività statistica, bisogna effettuare un tipo di indagine quantitativa e quindi considerare un numero più alto di soggetti, almeno venti.

Dato che le indagini qualitative sono più importanti di quelle quantitative, in questo caso, si è optato per effettuare la prima tipologia di studio e poter raccogliere le informazioni che interessano il miglioramento della piattaforma.

Di seguito verranno descritti i risultati raccolti durante l'indagine.

### **5.3.1 Risultati dell'indagine**

Nei grafici, 5.1, 5.2, 5.3 si possono visualizzare i dati personali raccolti dai differenti utenti. Per quanto riguarda la parte generale, dell'interfaccia e delle funzionalità sono stati raggiunti dei punteggi medio-alti come visionabili nei seguenti grafici: 5.7, 5.8, 5.9, 5.14, 5.15, 5.18, 5.19, 5.21.

La parte di navigazione, invece, ha riscontrato feedback meno positivi segnalando in alcune parti della piattaforma l'assenza di funzionalità per l'annullamento delle operazioni, come visibile nel grafico 5.11. Inoltre la prima impressione degli utenti è stata nettamente positiva in quanto la maggior parte di loro si è mostrato molto soddisfatta (grafico 5.4).

Di seguito vengono elencati alcuni commenti lasciati dai soggetti:

- Utente 2: "Visivamente é molto gradevole e anche il menù é intuitivo e capibile";
- Utente 3: "Piattaforma veloce e funzionante";
- Utente 6: "Sito fatto molto bene, solo poco intuitivo il pulsante per consegnare un secondo elaborato e non mi fa impazzire il modo di indicare il numero di componenti del gruppo";

- Utente 10: "Colori in linea con quelli dei siti UNIBO. Sito di semplicissima navigazione";
- Utente 11: "Per quanto alcune interazioni non abbiano interfaccia grafica ma solo l'alert box tutto funziona in maniera rapida ed intuitiva.".

Come riassunto nel grafico 5.5, la parte in cui è stata riscontrata maggior difficoltà è quella del login/registrazione e gli utenti hanno segnalato questi commenti:

- Utente 4: "Non era ben specificato che la password avesse dei requisiti minimi, e ogni volta che inserivo una password sbagliata mi azzerava l'intera form di registrazione";
- Utente 7: "Nella pagina iniziale bisognerebbe mettere più in rilievo il login";

Per poter risolvere queste incomprensioni i soggetti hanno riassunto le possibili soluzioni nel grafico 5.6.

Altrettante persone non hanno trovato alcuna difficoltà.

In conclusione, i soggetti hanno lasciato alcune considerazioni personali:

- Utente 1: "Bella, molto, averla avuta ai tempi della mia triennale avrei avuto molti meno problemi.";
- Utente 7: "Utilizzando l'applicazione sia mobile sia web, funziona correttamente, ma l'utilizzo del telefono rende la visualizzazione delle tabelle ambigua e confusionaria. Nonostante questo "piccolo problema" mobile, l'applicazione mi sembra ottima e davvero giusta per il goal principale.";
- Utente 9: "Trovo molto utile la funzionalità, in particolare avere un feedback subito può aiutare nel primo anno quando si ha poca esperienza.".

Inoltre hanno consigliato alcune eventuali funzionalità da aggiungere:

- Utente 3: "La possibilità di aggiungere note alla consegna dell'elaborato e la possibilità di una chat/forum per le domande con il professore.";
- Utente 5: "La possibilità di cancellarsi dalle mail".

### 5.3.2 Grafici riassuntivi

Di seguito sono riportati i grafici più rilevanti che riassumono i dati raccolti durante l'indagine.

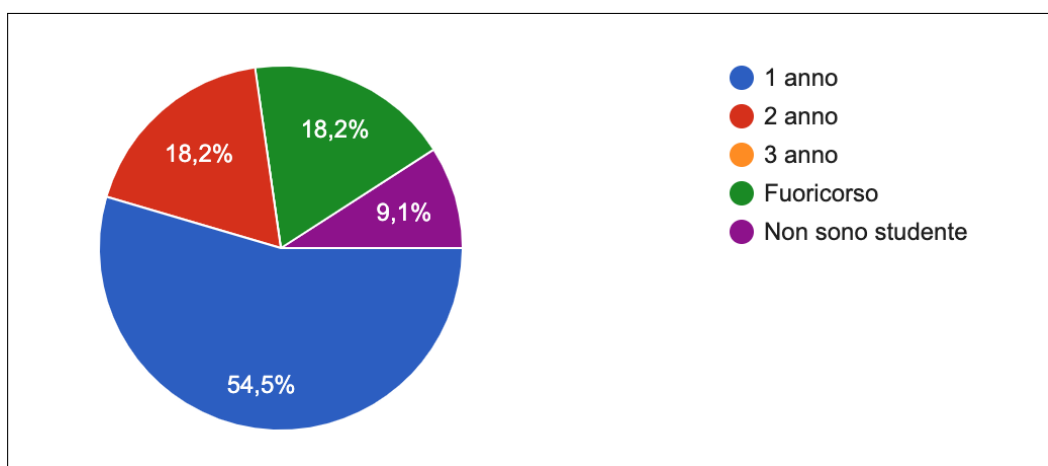


Figura 5.1: Grafico che mostra l'anno di appartenenza

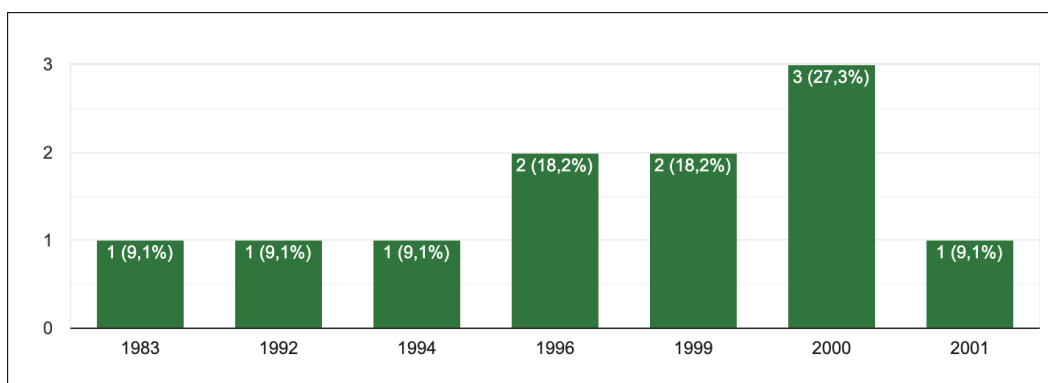


Figura 5.2: Grafico che illustra l'anno di nascita dei soggetti

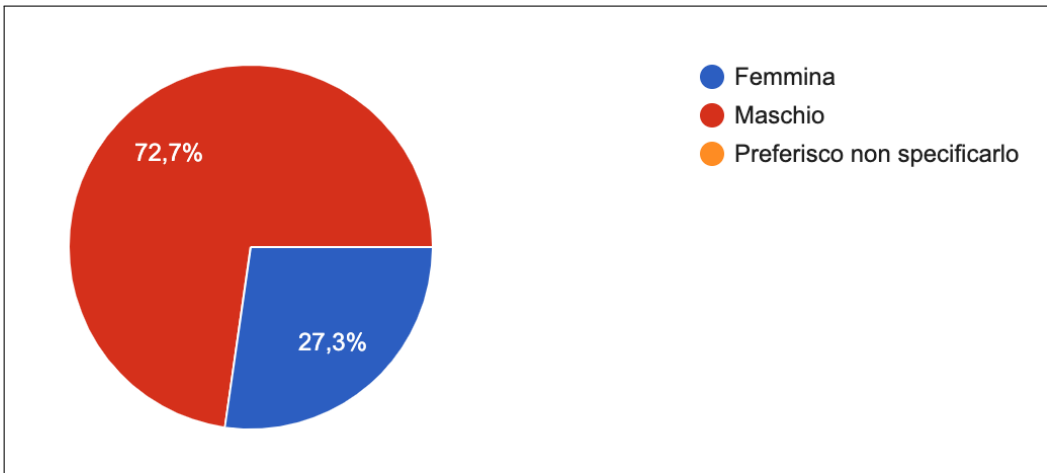


Figura 5.3: Grafico che illustra il sesso dei soggetti

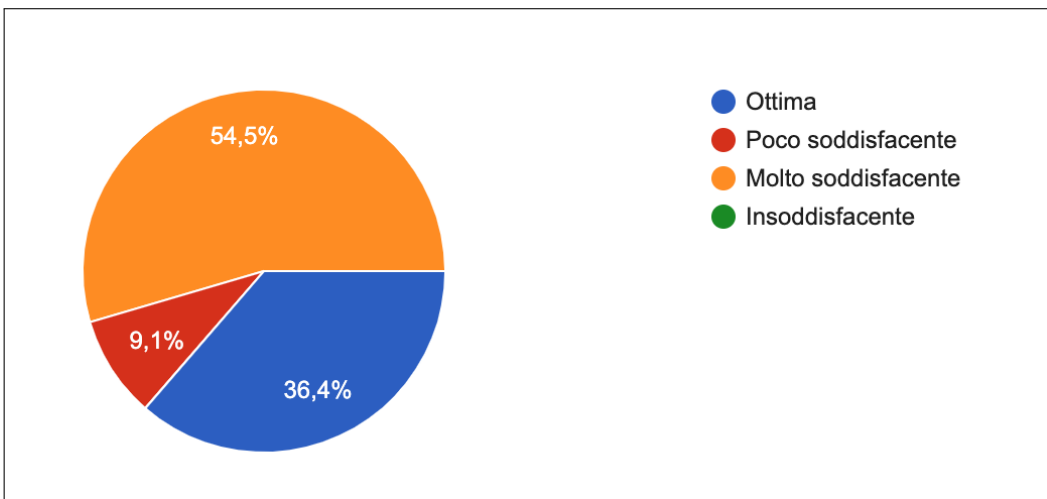


Figura 5.4: Grafico che illustra la prima impressione dei soggetti

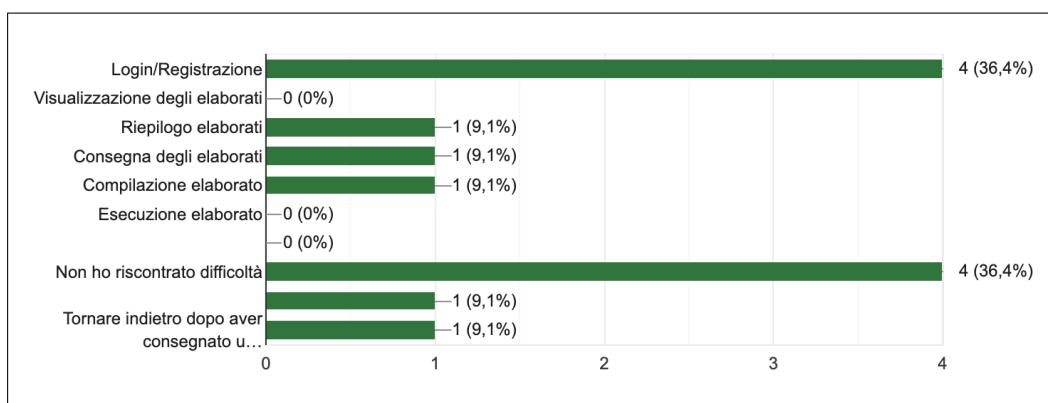


Figura 5.5: Grafico che illustra le difficoltà riscontrate dai soggetti

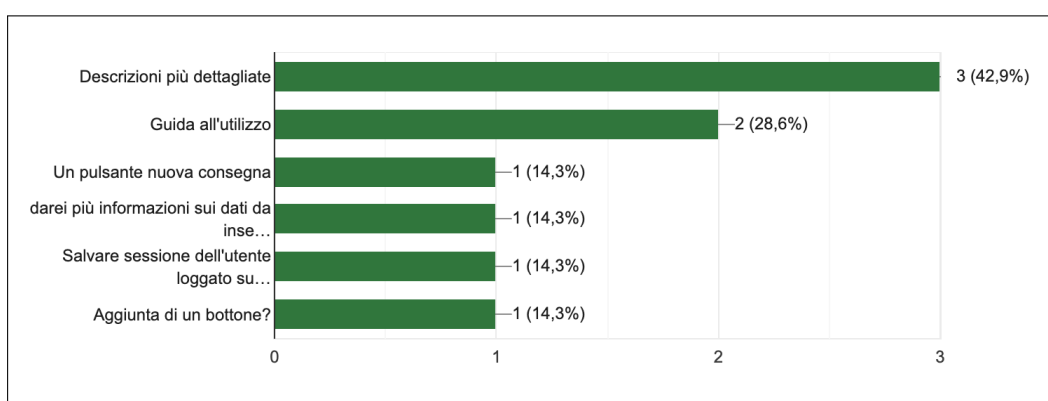


Figura 5.6: Grafico che illustra le possibili soluzioni ai problemi trovati dai soggetti

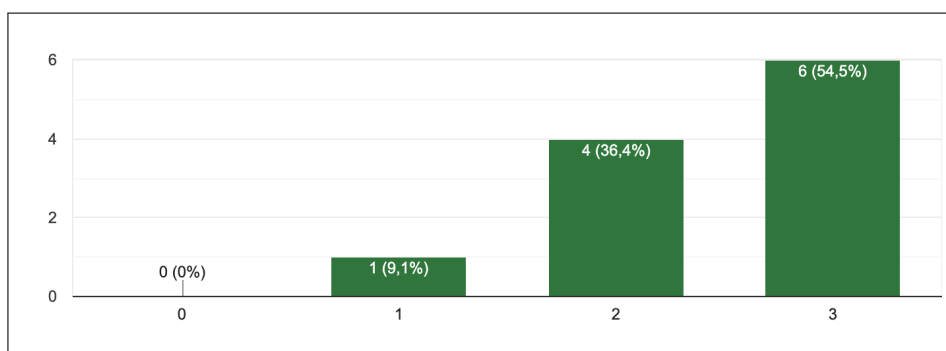


Figura 5.7: Grafico che illustra su una scala di valori da 0 a 3 se i contenuti sono semplici ed intuitivi

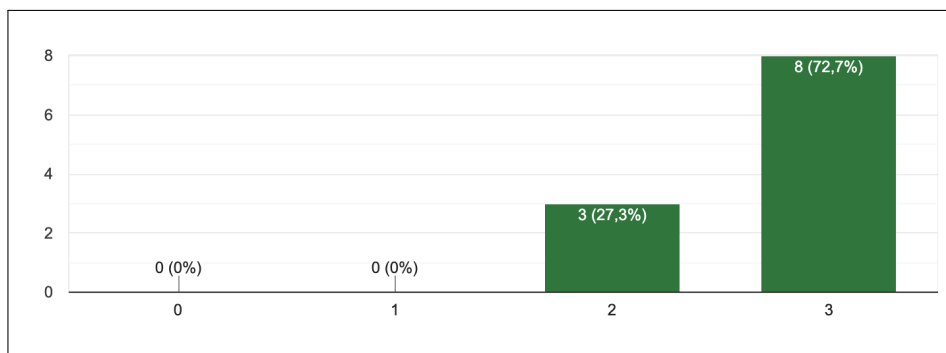


Figura 5.8: Grafico che illustra su una scala di valori da 0 a 3 se il testo è facilmente leggibile

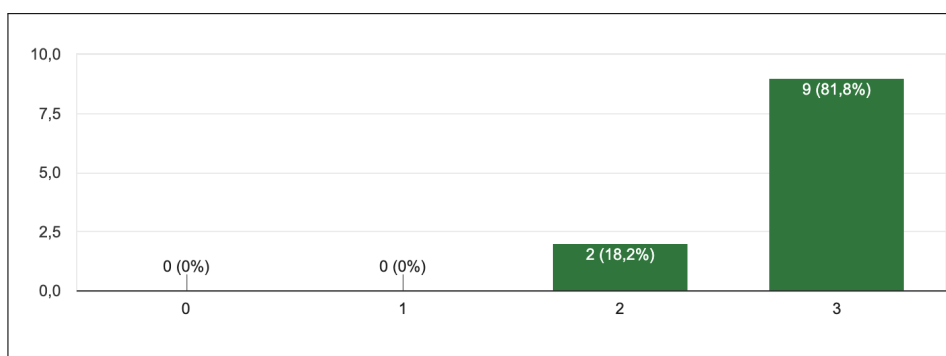


Figura 5.9: Grafico che illustra su una scala di valori da 0 a 3 se il titolo delle pagine è esplicativo

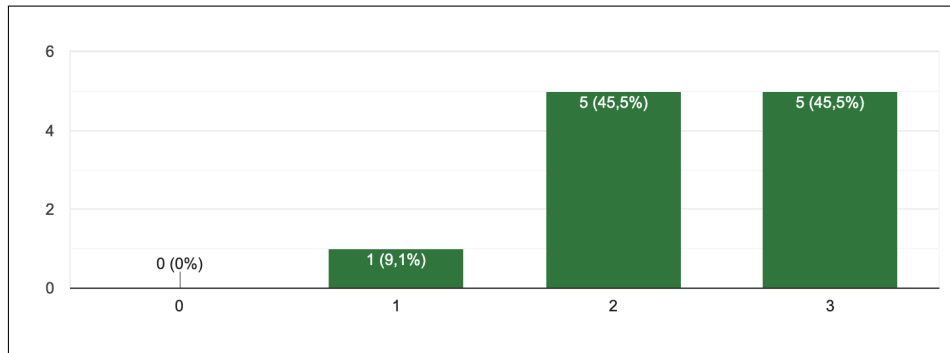


Figura 5.10: Grafico che illustra su una scala di valori da 0 a 3 se le pagine presentano opzioni di navigazione

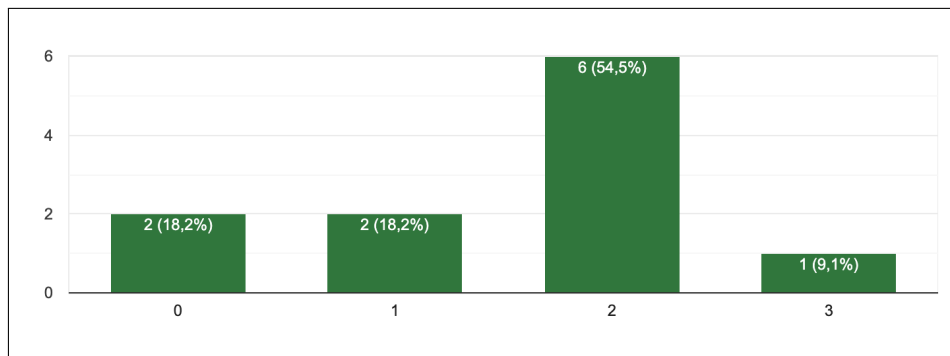


Figura 5.11: Grafico che illustra su una scala di valori da 0 a 3 se sono presenti funzionalità di annullamento delle operazioni

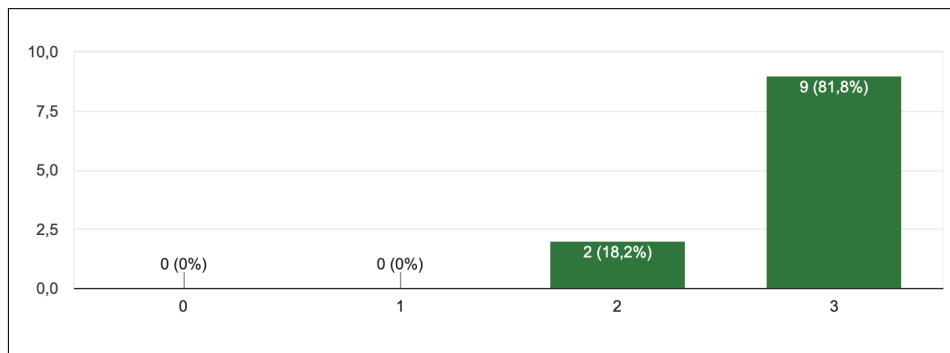


Figura 5.12: Grafico che illustra su una scala di valori da 0 a 3 se sono presenti funzionalità per ritornare alla homepage

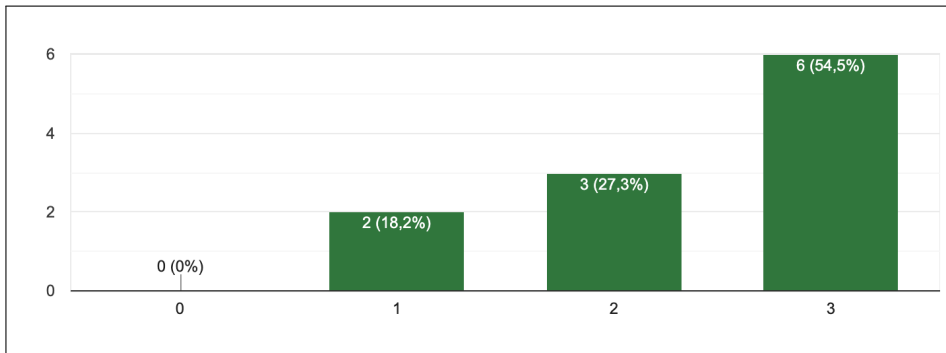


Figura 5.13: Grafico che illustra su una scala di valori da 0 a 3 se il menù di navigazione è efficace

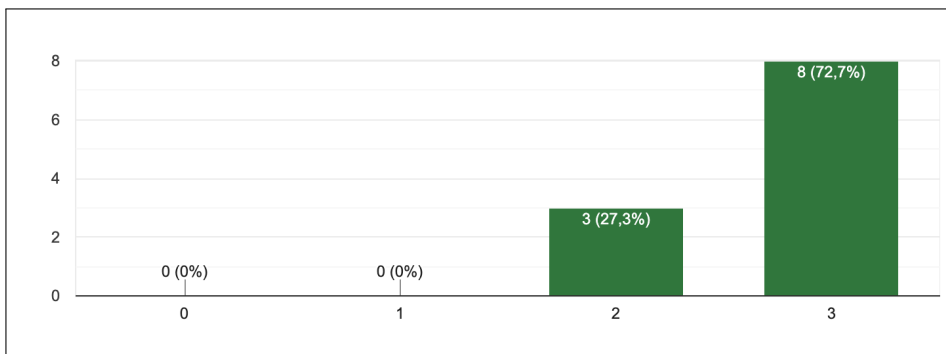


Figura 5.14: Grafico che illustra su una scala di valori da 0 a 3 se l'applicazione ha un aspetto gradevole

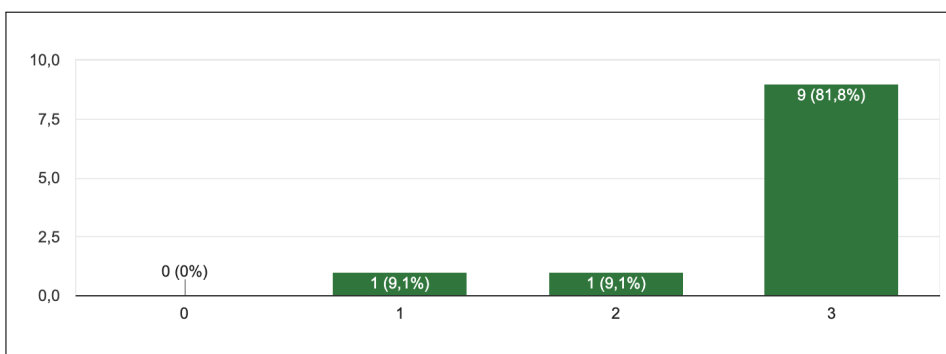


Figura 5.15: Grafico che illustra su una scala di valori da 0 a 3 se lo stile è adeguato alle varie schermate della piattaforma



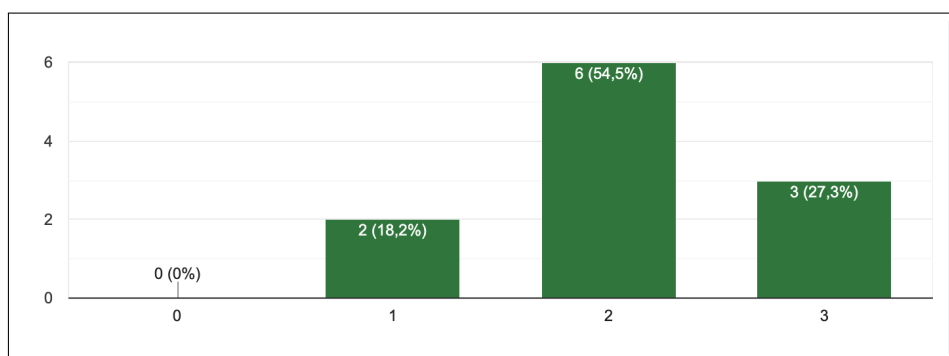


Figura 5.16: Grafico che illustra su una scala di valori da 0 a 3 se ci sono situazioni in cui la navigazione è facilitata dalla grafica

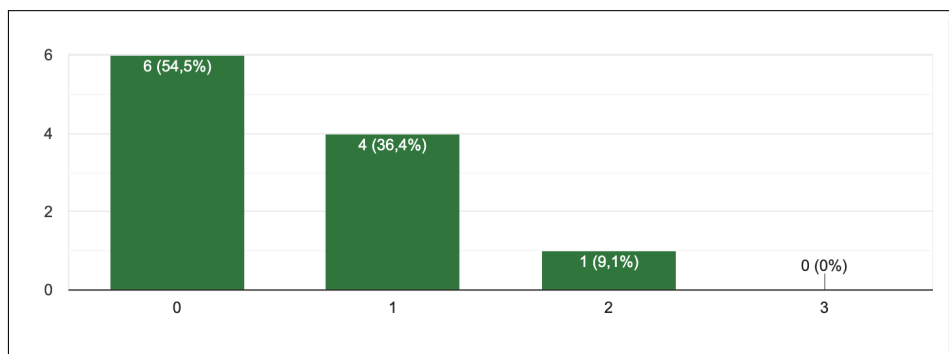


Figura 5.17: Grafico che illustra su una scala di valori da 0 a 3 se ci sono situazioni in cui la navigazione è compromessa dalla grafica

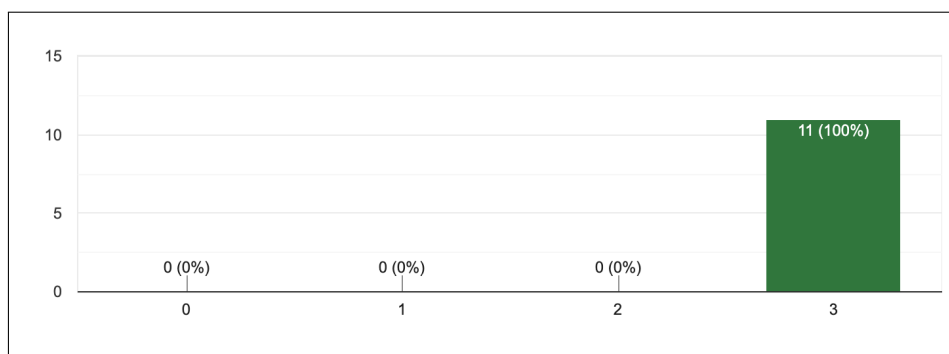


Figura 5.18: Grafico che illustra su una scala di valori da 0 a 3 se i tempi della compilazione dell'elaborato sono ragionevoli

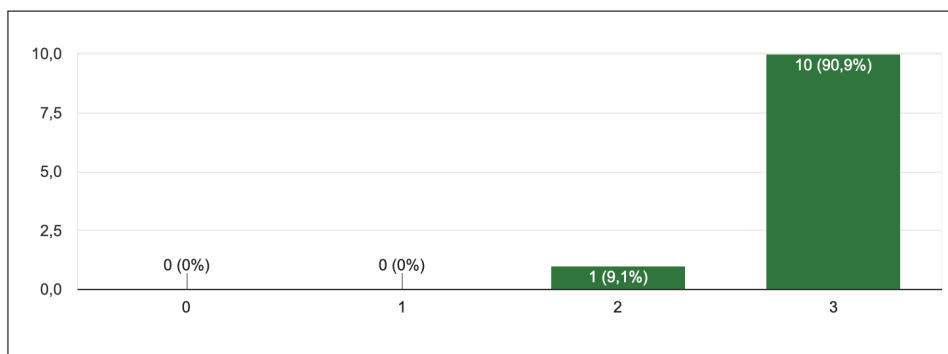


Figura 5.19: Grafico che illustra su una scala di valori da 0 a 3 se i tempi dell'esecuzione dell'elaborato sono ragionevoli

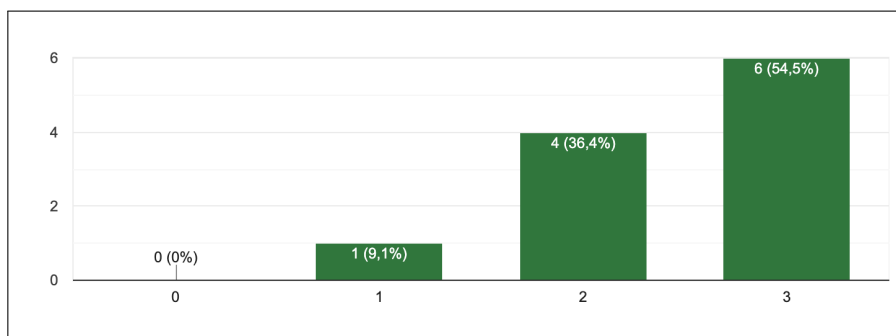


Figura 5.20: Grafico che illustra su una scala di valori da 0 a 3 se i materiali messi a disposizione sono semplici da visionare

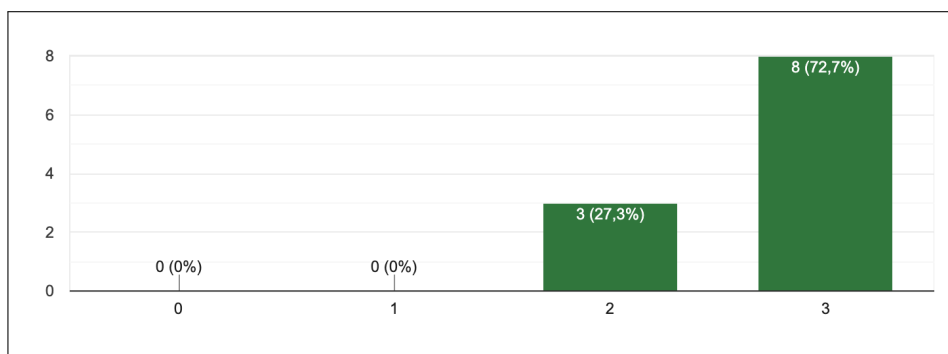


Figura 5.21: Grafico che illustra su una scala di valori da 0 a 3 se la consegna di un elaborato è stata semplice

# Conclusioni e sviluppi futuri

Scopo di questo lavoro di tesi è stato quello di affrontare l'analisi, la progettazione e la realizzazione di una piattaforma per studenti e docenti per la correzione di esercizi di programmazione in C. Permettere agli studenti di correggere in totale autonomia non solo riduce il tempo di attesa tra un elaborato e l'altro ma permette anche allo studente di avere una maggiore responsabilità nell'affrontare un corso basilare, come questo, all'interno del CdL di Ingegneria e Scienze Informatiche. D'altro canto, il docente, risparmia tempo prezioso da dedicare alla didattica e ai suoi studenti, in prima persona.

Per il perseguimento degli obiettivi è stato necessario valutare i software esistenti e, considerando i limiti di questi, si sono definiti i requisiti di funzionamento. Dopo questa prima fase di ricerche si è giunti alla conclusione che non c'era nessuna piattaforma che potesse soddisfare le esigenze del corso. L'implementazione è stata eseguita adottando le tecnologie più idonee ed approfondendo le tematiche studiate nel corso di laurea. La scelta di utilizzare un design responsive e facilmente intuitivo è stata naturale poichè gli studenti del XXI secolo utilizzano sempre di più dispositivi mobili. Ovviamente, in questa scelta, non è stata messa da parte la priorità dell'utilizzo di un computer per poter progettare gli elaborati.

Una volta conclusa tutta la fase di implementazione il sistema è stato sottoposto alla valutazione di alcuni studenti e sono state raccolte le loro impressioni sulle parti di: usabilità, accessibilità e contenuti.

I risultati raccolti durante l'indagine mostrano che, in linea generale, la piattaforma ha ricevuto molti feedback positivi grazie al design semplice ed intuitivo e all'interfaccia responsive. Inoltre, le indicazioni degli intervistati hanno attribuito un valore elevato quale elemento facilitatore per la frequenza al meglio del corso di studi. Tuttavia, gli utenti hanno segnalato dei piccoli aspetti negativi che in fase di progettazione non erano stati presi in considerazione.

I possibili sviluppi futuri potranno riguardare, senza sostanziali variazioni, la composizione dei gruppi che attualmente, per scelta, si è fissato ad un massimo di due studenti. Attualmente, il sito, non presenta limiti importanti poichè sono state adottate tecnologie avanzate sia riguardo i software per la realizzazione del codice che per l'amministrazione del DBMS.

Questa piattaforma è predisposta ad essere estesa anche ad altri linguaggi di programmazione, con le dovute modifiche ed analisi.

Grazie all'indagine sono stati raccolti dei suggerimenti preziosi da parte degli utenti che verranno presi in considerazione per un'eventuale evoluzione della piattaforma: come, ad esempio, permettere allo studente di cancellarsi dalle mail automatiche o la possibilità di aggiungere note alla consegna dell'elaborato e l'opportunità di una chat/forum per le domande con il docente.

In accordo con la relatrice, questo progetto di tesi verrà, nei prossimi mesi, inserito nel server dell'università per permettere agli studenti del prossimo anno accademico di usufruirne.

# Bibliografia

- [1] Aruba. Aruba, 2020.
- [2] Balsamiq. mockup, 2020.
- [3] Bootstrap. bootstrap, 2020.
- [4] ethicsoft. La qualità del software, 2019.
- [5] Apache Friends. Xampp, 2020.
- [6] Google. Lighthouse. <https://developers.google.com/web/tools/lighthouse/>, 2013.
- [7] The PHP Group. Php, 2001-2020.
- [8] Guru99. dbms, 2020.
- [9] ideone. ideone. <https://ideone.com/>, 2019. Visitato il 01/03/2020.
- [10] Rensselaer Polytechnic Institute. Submittty. <https://submittty.org/>, 2020. Visitato il 07/03/2020.
- [11] inupi. requisiti, 2019.
- [12] Miur. Corso Online per Potenziare le Competenze Digitali. <https://digit.olino.it/#/task/ordina/submissions>, 2019. Visitato il 29/02/2020.
- [13] MySQL. mysql, 2020.
- [14] Jakob Nielsen. Why you only need to test with 5 users, 2000.

- [15] Jakob Nielsen. Quantitative studies: How many users to test?, 2006.
- [16] Pluralsight. Javascript, 2016-2020.
- [17] Rever. dbmain, 2020.
- [18] Mondadori Media Spa. Moss, 1994-2018.
- [19] Mondadori Media Spa. Altvista, 2020.
- [20] Peter Tröger. OpenSubmit. <http://docs.open-submit.org/en/latest/>, 2018. Visitato il 07/03/2020.
- [21] Uml. uml, 2020.
- [22] universityWarwick. boss. <https://www.dcs.warwick.ac.uk/boss/index.php>, 2003. Visitato il 07/03/2020.
- [23] Tatu Ylonen. Ssh, 1995.