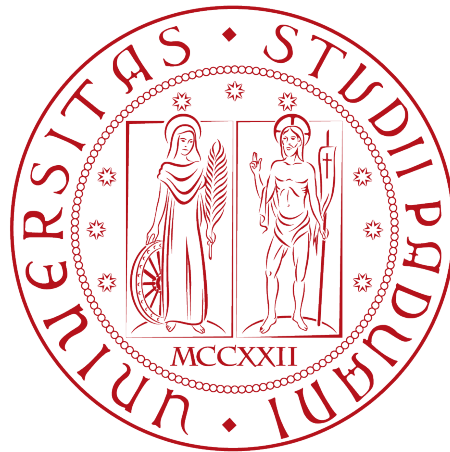


Università degli Studi di Padova

DIPARTIMENTO DI MATEMATICA “TULLIO LEVI-CIVITA”

CORSO DI LAUREA IN INFORMATICA



**Sviluppo di un'applicazione web per la
realizzazione e compilazione di questionari
valutativi**

Tesi di laurea

Relatore

Prof.essa Ombretta Gaggi

Laureanda

Elena Marchioro

ANNO ACCADEMICO 2022-2023

Elena Marchioro: *Sviluppo di un'applicazione web per la realizzazione e compilazione di questionari valutativi*, Tesi di laurea, © Dicembre 2023.

*“ These years pass by and we’re growin’ older
And I think of you, all we’ve made it through
Some have passed away, some have moved on
But I’m still here today. ”*

— My Life, Imagine Dragons

Dedicato alle persone che sono sempre state al mio fianco.

Dedicato a mia nonna Annamaria che avrebbe sempre voluto vedere questo giorno.

Sommario

Il presente documento descrive l'attività di stage svolta presso l'azienda CWBT (Codice Web Banking Technology). Lo stage è stato svolto alla conclusione del percorso di studi della laurea triennale in Informatica ed ha avuto la durata di circa trecento ore.

L'obiettivo dello stage consiste nello sviluppo di un'applicazione web dedicata alla creazione e compilazione di questionari valutativi a risposte chiuse. I vincoli imposti includono l'adeguamento alle metodologie di analisi, progettazione e sviluppo aziendale, nonché l'uso dello *stack* tecnologico specifico dell'azienda. Al termine, viene inoltre presentata un'analisi conclusiva che confronta le aspettative iniziali personali con i risultati ottenuti e gli obiettivi raggiunti durante lo stage.

“When you’re in pain, when you think you’ve had enough
don’t ever give up”

— Up&Up, Coldplay

Ringraziamenti

Innanzitutto, desidero esprimere la mia sincera gratitudine alla Professoressa Ombretta Gaggi per la sua guida preziosa, il supporto e il tempo messo a disposizione durante il percorso di realizzazione della mia tesi.

Vorrei inoltre ringraziare il mio tutor aziendale, Roberto Martina, per avermi trasmesso la passione per il suo lavoro e per avermi dato l’opportunità di crescere non solo professionalmente, ma soprattutto personalmente.

A mia madre, mio padre e mia nonna desidero dedicare questi ringraziamenti per aver creduto in me in ogni fase di questo percorso accademico. Grazie per avermi incoraggiata a continuare e insegnato a persistere nonostante tutto. Vi sono grata per avermi permesso di raggiungere questo traguardo.

Infine vorrei ringraziare i miei amici che hanno passato con me questi anni; grazie a Marco per avermi sopportata, soprattutto durante le interminabili sessioni, e grazie ad Alessia, Allegra, Andrea, Gabriele, Gaia, Giorgia, Gloria, Niccolò, Riccardo e Sofia per avermi accompagnata in questi anni. Vorrei inoltre ringraziare la persona che ha saputo capirmi prima che ci riuscissi io, che mi ha aiutata e supportata quando dubitavo e mi ha convinta a non mollare.

Padova, Dicembre 2023

Elena Marchioro

Indice

1	Introduzione	1
1.1	L'azienda	1
1.2	L'offerta di stage	2
1.2.1	Introduzione al progetto	2
1.2.2	Vincoli del progetto	2
1.2.3	Suddivisione delle ore	3
1.3	Aspettative personali	3
1.4	Organizzazione del testo	3
2	Analisi dei requisiti	5
2.1	Introduzione all'analisi	5
2.1.1	Caratteristiche degli utenti	5
2.2	Casi d'Uso	6
2.2.1	Diagrammi dei casi d'uso	6
2.2.2	UC01	7
2.2.3	UC01.1	7
2.2.4	UC01.2	8
2.2.5	UC01.3	8
2.2.6	UC01.3.1	9
2.2.7	UC02	9
2.2.8	UC03	9
2.2.9	UC03.1	10
2.2.10	UC03.2	11
2.2.11	UC03.2.1	11
2.2.12	UC04	11
2.2.13	UC05	12
2.2.14	UC05.1	13
2.2.15	UC05.2	13
2.2.16	UC06	14
2.2.17	UC06.1	15
2.2.18	UC06.2	15
2.2.19	UC06.3	15
2.2.20	UC06.4	16
2.2.21	UC07	16
2.2.22	UC07.1	17
2.2.23	UC07.2	17
2.3	Requisiti	18
2.3.1	Funzionali	18

2.3.2	Vincolo	21
3	Tecnologie e strumenti	22
3.1	Ambiente di sviluppo e IDE	22
3.2	Linguaggi	23
3.2.1	Java	23
3.2.2	HTML5, CSS3, JavaScript	23
3.2.3	Python	24
3.3	Framework	24
3.3.1	Spring	24
3.3.2	Apache Struts	25
3.3.3	Hibernate	25
3.3.4	Bootstrap	26
3.4	Librerie	26
3.4.1	JSTL	26
3.4.2	Apache Commons	27
3.4.3	JQuery	27
4	Progettazione e codifica	28
4.1	Base di dati	28
4.2	Architettura	30
4.2.1	Model	30
4.2.2	View	32
4.2.3	Controller	33
5	Prodotto finale	35
5.1	Parte dell'amministratore	35
5.1.1	Dashboard	35
5.1.2	Ricerca questionario	36
5.1.3	Creazione e modifica di un questionario	37
5.1.4	Dettaglio di un questionario	37
5.1.5	Dettaglio di una domanda	39
5.1.6	Creazione e modifica di una domanda	40
5.1.7	Ricerca di una prova	40
5.1.8	Creazione e modifica di una prova	41
5.1.9	Dettaglio di una prova	42
5.1.10	Dettaglio di una risposta	43
5.1.11	Creazione e modifica di una risposta	44
5.2	Parte del candidato	45
5.2.1	Pagina iniziale	45
5.2.2	Esecuzione della prova	45
5.2.3	Pagina dei risultati	46
5.2.4	Prova scaduta	47
6	Conclusioni	48
6.1	Verifica e validazione	48
6.1.1	Requisiti	48
6.1.2	Accessibilità	51
6.2	Consuntivo finale	51
6.3	Resoconto finale	52

<i>INDICE</i>	vii
6.3.1 Conoscenze acquisite	52
6.3.2 Valutazione personale	52
Acronimi e abbreviazioni	53
Glossario	54
Bibliografia	56

Elenco delle figure

1.1	Logo dell'azienda	1
2.1	<i>Dashboard</i> dell'amministratore	6
2.2	Casi d'uso del candidato	6
2.3	Use Case 01	7
2.4	Use Case 03	10
2.5	Use Case 05	13
2.6	Use Case 06	14
2.7	Use Case 07	17
3.1	Loghi Eclipse e VS Code	22
3.2	Logo Java	23
3.3	Loghi HTML, CSS, JavaScript	23
3.4	Logo Python	24
3.5	Logo Spring	24
3.6	Diagramma delle componenti di Spring	25
3.7	Logo Apache Struts	25
3.8	Logo Hibernate	25
3.9	Logo Bootstrap	26
3.10	Logo Apache Commons	27
3.11	Logo JQuery	27
4.1	Diagramma Apache Struts MVC	30
5.1	Dashboard dell'amministratore, pagina iniziale	35
5.2	Ricerca questionari	36
5.3	Creazione di un nuovo questionario	37
5.4	Dettaglio di un questionario	37
5.5	Domande di un questionario	37
5.6	Prove di un questionario	38
5.7	Dettaglio di una domanda	39
5.8	Risposte a una domanda	39
5.9	Creazione di una domanda	40
5.10	Ricerca prove	40
5.11	Creazione di una nuova prova	41
5.12	Dettaglio di una prova completata	42
5.13	Risposte di una prova	42
5.14	Dettaglio di una risposta	43
5.15	Creazione di una risposta	44

5.16	Pagina di inizio del candidato	45
5.17	Esempio domanda durante il completamento di una prova	46
5.18	Risultati di una prova	46
5.19	Pagina prova scaduta	47

Elenco delle tabelle

1.1	Tabella della suddivisione delle ore di lavoro	3
2.1	Tabella del tracciamento dei requisiti funzionali	18
2.2	Tabella del tracciamento dei requisiti di vincolo	21
4.1	La tabella rappresenta la struttura di <code>QuestionarioStruttura</code>	29
4.2	La tabella rappresenta la struttura di <code>QuestionarioStrutturaItem</code>	29
4.3	La tabella rappresenta la struttura di <code>QuestionarioUtente</code>	29
4.4	La tabella rappresenta la struttura di <code>QuestionarioUtenteItem</code>	30
6.1	Requisiti soddisfatti	48
6.2	Tabella finale della suddivisione delle ore di lavoro	51

Capitolo 1

Introduzione

1.1 L'azienda



Figura 1.1: Logo dell'azienda

CWBT (Codice Web Banking Technology), logo in figura 1.1, nasce nel 2013 a Padova come software house specializzata nello sviluppo di applicazioni e servizi in particolare per il settore finanziario e bancario, con focus sui pagamenti. Vanta un'esperienza decennale nell'ambito della programmazione informatica e della consulenza a vari livelli per aziende di medie-grandi dimensioni.

Grazie alle competenze acquisite negli anni in diversi settori quali Banking, Media and Publishing, Insurance e Industry riesce ad affrontare con successo ogni singolo aspetto del ciclo di vita dei progetti nei quali è coinvolta; inoltre supporta i propri clienti nello studio dei modelli di business, nella definizione dei processi organizzativi e nella progettazione e realizzazione di software con un forte orientamento alle nuove tecnologie.

L'azienda offre una vasta gamma di servizi tra i quali:

- Sviluppo Applicazioni e Portali web-based
- Sviluppo Applicazioni Mobile
- Studio di fattibilità e sostenibilità dei modelli di business
- Analisi e definizione dei processi organizzativi
- Studi di navigabilità e usabilità
- Studi di ergonomia del software

1.2 L'offerta di stage

Attualmente l'azienda, durante i colloqui di lavoro, pone dei questionari valutativi cartacei a risposte chiuse. Lo scopo dell'attività di stage è quindi informatizzare la procedura sopra descritta tramite la realizzazione di una [Web Application](#) apposita attraverso l'inserimento nell'area di sviluppo web come tecnico programmatore in linguaggio *Java*. Durante questo periodo verranno quindi sviluppate le conoscenze e le competenze necessarie ad effettuare analisi e sviluppo nel campo delle applicazioni web.

1.2.1 Introduzione al progetto

Viste le necessità dell'azienda, l'obiettivo dell'applicazione richiesta è quello di fornire un portale attraverso il quale ricavare un link da inviare tramite un'email per dare ai candidati la possibilità di compilare un questionario valutativo online. Verrà reso disponibile l'accesso al questionario per un certo periodo di tempo indicato e la compilazione sarà limitata a un solo utente attraverso il salvataggio dell'indirizzo IP usato durante il primo accesso. Viene inoltre richiesta la possibilità di inserire da parte dell'azienda nuovi questionari da sottoporre ai candidati e aggiornare quelli già presenti a sistema. Infine attraverso l'applicazione web si potranno vedere le risposte inserite e la loro correttezza.

1.2.2 Vincoli del progetto

In questa sezione vengono riportati tutti i vincoli che l'azienda ha imposto per lo sviluppo del prodotto; oltre a quelli elencati di seguito non sono posti ulteriori vincoli riguardo l'uso di tecnologie aggiuntive:

- Per lo sviluppo lato server è richiesto l'utilizzo dei framework: *Java*, *Hibernate*, *Apache Struts* e *Spring*
- Per il lato client invece *Bootstrap*, [HTML5](#), [CSS3](#), *JavaScript* e la libreria [JSTL](#)
- Il sistema di versionamento in uso dall'azienda è [TortoiseSVN](#)^[g]
- La *webapp* deve rispettare le regole per l'accessibilità poste dal [W3C](#)

1.2.3 Suddivisione delle ore

La durata complessiva dello stage è stata di trecento ore, suddivise nel seguente modo secondo il piano di lavoro definito con l'azienda:

Tabella 1.1: Tabella della suddivisione delle ore di lavoro

Durata in ore	Descrizione	Settimana
40	Formazione iniziale e raccolta dei requisiti	W1
40	Realizzazione fase 1/2 del prototipo <i>WebApp</i> con i primi use case	W2
40	Realizzazione fase 2/2 del prototipo <i>WebApp</i> con i primi use case	W3
80	Realizzazione con connessioni ai servizi <i>REST</i> per la fase di <i>System Integration Test</i> ^[g]	W4/W5
40	Test e supporto <i>User Acceptance Testing</i> ^[g]	W6
40	Documentazione progetto	W7
20	Chiusura progetto e supporto ai primi ricicli	W8

1.3 Aspettative personali

Lo scopo dello stage non consiste solo nello sviluppo di un prodotto, ma anche nell'integrazione del percorso di studi con il mondo del lavoro. Di conseguenza gli obiettivi personali posti inizialmente, oltre all'apprendimento di competenze tecniche, sono:

- Apprendimento dei processi per realizzare un prodotto conforme alle aspettative
- Comprensione del lavoro di gruppo all'interno di un'azienda
- Apprendere come rapportarsi con i committenti e altre persone appartenenti al settore

1.4 Organizzazione del testo

Il testo sarà composto dai seguenti capitoli:

Capitolo 1: Introduzione, il capitolo corrente, dove viene presentata l'azienda, introdotta l'attività di stage ed espone le aspettative personali;

Capitolo 2: Analisi dei requisiti in cui vengono espone i casi d'uso e i corrispondenti requisiti;

Capitolo 3: Tecnologie e strumenti nel quale sono elencate le varie tecnologie e gli strumenti utilizzati;

Capitolo 4: Progettazione e codifica in cui è illustrato l'approccio alla progettazione del sistema;

Capitolo 5: Prodotto finale il quale illustra la *WebApp* nelle sue componenti;

Capitolo 6: Conclusioni in cui viene riportata l'attività di test e un resoconto sull'attività di stage.

Riguardo la stesura del testo, relativamente al documento sono state adottate le seguenti convenzioni tipografiche:

- gli acronimi, le abbreviazioni e i termini ambigui o di uso non comune menzionati vengono definiti nel glossario, situato alla fine del presente documento;
- per la prima occorrenza dei termini riportati nel glossario viene utilizzata la seguente nomenclatura: *parola*^[g];
- i termini in lingua straniera o facenti parti del gergo tecnico sono evidenziati con il carattere *corsivo*.

Capitolo 2

Analisi dei requisiti

2.1 Introduzione all'analisi

Lo scopo di questo capitolo è quello di raccogliere i risultati dell'attività di analisi dei requisiti, includendo la descrizione dei casi d'uso del software e i requisiti necessari per la sua realizzazione. Per l'analisi del prodotto sono stati creati dei diagrammi dei casi d'uso (in inglese *Use Case Diagram*); sono diagrammi di tipo [UML](#)^[8] dedicati alla descrizione delle funzioni o dei servizi offerti dal sistema, così come sono percepiti e utilizzati dagli attori che interagiscono con esso stesso.

2.1.1 Caratteristiche degli utenti

Per il progetto è prevista la presenza di due categorie distinte di utenti, ovvero l'amministratore di sistema e il candidato.

L'utente che ricoprirà il ruolo di amministratore può eseguire le seguenti operazioni:

- Accedere alla [dashboard](#)^[8]
- Creare un questionario e aggiungere delle domande
- Creare una nuova prova per un candidato
- Visualizzare le risposte dei candidati

Il candidato potrà invece eseguire le seguenti azioni:

- Visualizzare le domande di un questionario
- Rispondere alle domande
- Visualizzare i dettagli del questionario da lui compilato

2.2 Casi d'Uso

Per poter capire e studiare a fondo tutte le funzionalità che devono essere messe a disposizione dell'utente che utilizza l'applicativo da sviluppare, sono stati realizzati i relativi diagrammi dei casi d'uso di tipo *UML*. Tali diagrammi sono risultati fondamentali per individuare correttamente tutti i requisiti del sistema in questione.

2.2.1 Diagrammi dei casi d'uso

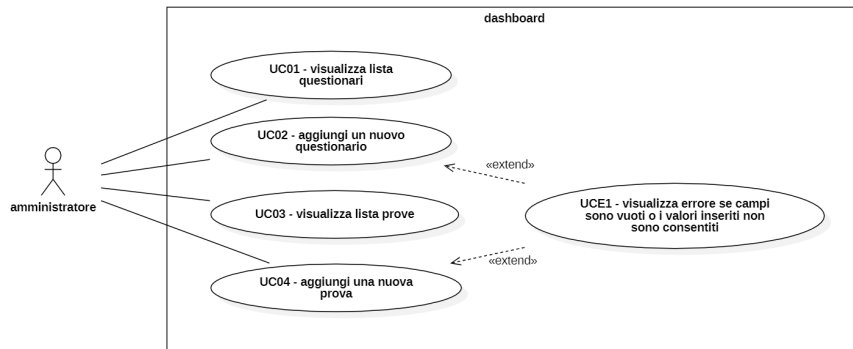


Figura 2.1: *Dashboard* dell'amministratore

Nella figura 2.1 vengono illustrate le azioni che l'amministratore può compiere nella *dashboard* del sistema.

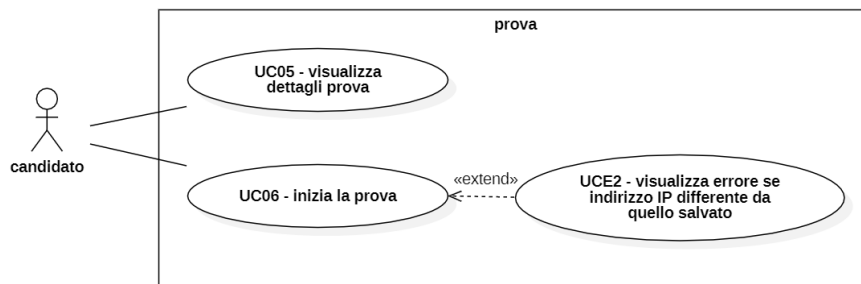


Figura 2.2: Casi d'uso del candidato

Nella figura 2.2 sono mostrate le azioni che può compiere un candidato dopo aver aperto il link per l'accesso alla prova.

2.2.2 UC01

UC01: Visualizza lista dei questionari - Figura 2.3

Attori Principali: Amministratore.

Precondizioni: L'attore vuole visualizzare la lista dei questionari inseriti a sistema.

Postcondizioni: La *webapp* fornisce le informazioni richieste.

Scenario Principale

- L'amministratore accede alla *dashboard*
- L'amministratore visualizza la lista dei questionari
- L'amministratore ricerca un questionario [UC01.1]
- L'amministratore modifica un questionario [UC01.2]
- L'amministratore elimina il questionario [UC01.3]
- L'amministratore visualizza i dettagli di un questionario [UC07]

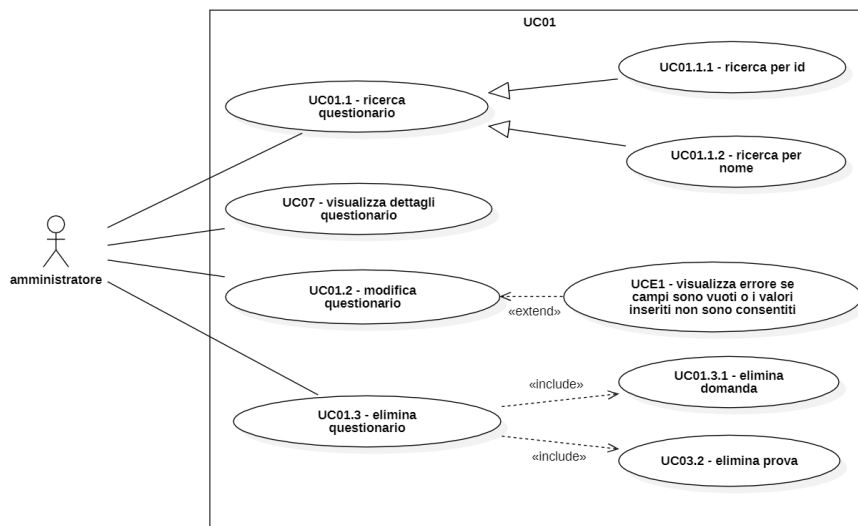


Figura 2.3: Use Case 01

2.2.3 UC01.1

UC01.1: Ricerca di un questionario

Attori Principali: Amministratore.

Precondizioni: L'attore vuole cercare un questionario.

Postcondizioni: La *webapp* fornisce le informazioni richieste.

Scenario Principale

- L'amministratore accede alla lista dei questionari
- L'amministratore inserisce l'informazione da cercare
- La *webapp* mostra i questionari corrispondenti alla ricerca

Generalizzazioni

- L'amministratore ricerca un questionario per ID [UC01.1.1]
- L'amministratore ricerca un questionario per nome [UC01.1.2]

2.2.4 UC01.2

UC01.2: Modifica questionario

Attori Principali: Amministratore.

Precondizioni: L'attore vuole modificare un questionario.

Postcondizioni: Il questionario viene modificato.

Scenario Principale

- L'amministratore accede alla lista dei questionari
- L'amministratore modifica il questionario tramite l'apposito tasto
- Il questionario viene modificato

Estensioni I valori inseriti non sono validi UCE1.

2.2.5 UC01.3

UC01.3: Elimina questionario

Attori Principali: Amministratore.

Precondizioni: L'attore vuole eliminare un questionario.

Postcondizioni: Il questionario viene eliminato.

Scenario Principale

- L'amministratore accede alla lista dei questionari
- L'amministratore elimina il questionario tramite l'apposito tasto
- Il questionario viene eliminato

2.2.6 UC01.3.1

UC01.3.1: Elimina domanda

Attori Principali: Amministratore.

Precondizioni: L'attore vuole eliminare una domanda.

Postcondizioni: La domanda viene eliminata.

Scenario Principale

- L'amministratore accede alle domande di un questionario
- L'amministratore elimina la domanda tramite l'apposito tasto
- La domanda viene eliminata

2.2.7 UC02

UC02: Aggiunta di un nuovo questionario

Attori Principali: Amministratore.

Precondizioni: L'attore vuole aggiungere un nuovo questionario.

Postcondizioni: La *webapp* aggiunge il questionario a sistema.

Scenario Principale

- L'amministratore accede alla *dashboard*
- L'amministratore visualizza la pagina di inserimento di un nuovo questionario
- L'amministratore inserisce il nome del questionario
- L'amministratore salva il questionario attraverso l'apposito tasto

Estensioni I campi sono lasciati vuoti o i valori inseriti non sono validi UCE1.

2.2.8 UC03

UC03: Visualizza la lista delle prove - Figura 2.4

Attori Principali: Amministratore.

Precondizioni: L'attore vuole visualizzare la lista delle prove inserite a sistema.

Postcondizioni: La *webapp* fornisce le informazioni richieste.

Scenario Principale

- L'amministratore visualizza la lista delle prove
- L'amministratore esegue una ricerca di una prova [UC03.1]
- L'amministratore elimina una prova [UC03.2]
- L'amministratore visualizza i dettagli di una prova [UC05]

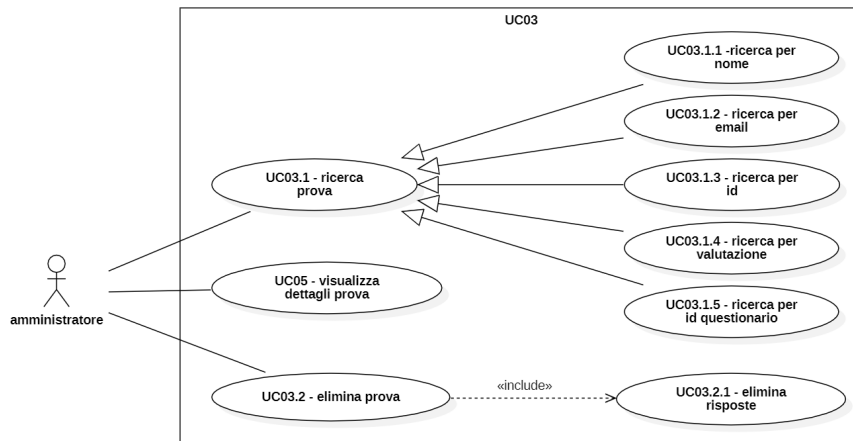


Figura 2.4: Use Case 03

2.2.9 UC03.1

UC03.1: Ricerca di una prova

Attori Principali: Amministratore.

Precondizioni: L'attore vuole cercare una prova.

Postcondizioni: La *webapp* fornisce le informazioni richieste.

Scenario Principale

- L'amministratore accede alla lista delle prove
- L'amministratore inserisce l'informazione da cercare
- La *webapp* mostra le prove corrispondenti alla ricerca

Generalizzazioni

- L'amministratore ricerca un tentativo per nome [UC03.1.1]
- L'amministratore ricerca un tentativo per email [UC03.1.2]
- L'amministratore ricerca un tentativo per ID [UC03.1.3]

- L'amministratore ricerca un tentativo per valutazione [UC03.1.4]
- L'amministratore ricerca un tentativo per ID questionario [UC03.1.5]

2.2.10 UC03.2

UC03.2: Elimina prova

Attori Principali: Amministratore.

Precondizioni: L'attore vuole eliminare una prova.

Postcondizioni: La prova viene eliminata.

Scenario Principale

- L'amministratore accede alla lista delle prove
- L'amministratore elimina la prova tramite l'apposito tasto
- La prova viene eliminata

2.2.11 UC03.2.1

UC03.2.1: Elimina risposta

Attori Principali: Amministratore.

Precondizioni: L'attore vuole eliminare una risposta.

Postcondizioni: La risposta viene eliminata.

Scenario Principale

- L'amministratore accede alla lista delle risposte di una prova
- L'amministratore elimina la risposta tramite l'apposito tasto
- La risposta viene eliminata

2.2.12 UC04

UC04: Aggiunta di una nuova prova

Attori Principali: Amministratore.

Precondizioni: L'attore vuole aggiungere una nuova prova.

Postcondizioni: La *webapp* aggiunge la prova a sistema.

Scenario Principale

- L'amministratore accede alla *dashboard*
- L'amministratore visualizza la pagina di inserimento di una nuova prova
- L'amministratore inserisce l'ID del questionario afferente
- L'amministratore inserisce l'email del candidato
- L'amministratore inserisce il nome del candidato
- L'amministratore inserisce la data di scadenza per la compilazione della prova
- L'amministratore salva la prova attraverso l'apposito tasto

Estensioni I campi sono lasciati vuoti o i valori inseriti non sono validi UCE1.

2.2.13 UC05**UC05: Visualizza i dettagli di una prova - Figura 2.5**

Attori Principali: Amministratore, Candidato.

Precondizioni: L'attore vuole visualizzare i dettagli di una prova.

Postcondizioni: La *webapp* fornisce le informazioni richieste.

Scenario Principale

- L'attore visualizza l'ID della prova
- L'attore visualizza l'email della prova
- L'attore visualizza il nome del candidato
- L'attore visualizza la data di scadenza della prova
- L'attore visualizza la valutazione della prova
- L'amministratore visualizza le risposte [UC05.1]
- L'amministratore aggiunge una risposta [UC05.2]

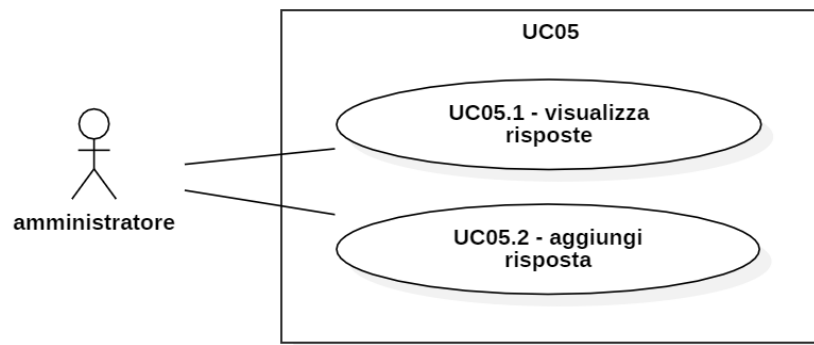


Figura 2.5: Use Case 05

2.2.14 UC05.1

UC05.1: Visualizza le risposte di una prova

Attori Principali: Amministratore.

Precondizioni: L'attore vuole visualizzare le risposte di una prova.

Postcondizioni: La *webapp* fornisce le informazioni richieste.

Scenario Principale

- L'amministratore si trova nella pagina di dettaglio di una prova
- L'amministratore visualizza i dettagli delle risposte

2.2.15 UC05.2

UC05.2: Aggiunta di una nuova risposta

Attori Principali: Amministratore.

Precondizioni: L'attore vuole aggiungere una nuova risposta a una prova.

Postcondizioni: La *webapp* aggiunge la nuova risposta.

Scenario Principale

- L'amministratore si trova nella pagina di aggiunta di una nuova prova
- L'amministratore aggiunge la risposta
- L'amministratore salva la risposta attraverso l'apposito tasto

2.2.16 UC06

UC06: Inizio di una prova - Figura 2.6

Attori Principali: Candidato.

Precondizioni: L'attore vuole iniziare una prova.

Postcondizioni: La prova viene iniziata e aggiornata.

Scenario Principale

- Il candidato visualizza una domanda
- Il candidato inserisce una risposta [UC06.1]
- Il candidato passa alla domanda successiva [UC06.2]
- Il candidato torna alla domanda precedente [UC06.3]
- Il candidato termina la prova e la invia [UC06.4]

Estensioni L'indirizzo IP con cui il candidato accede è diverso da quello salvato UCE2.

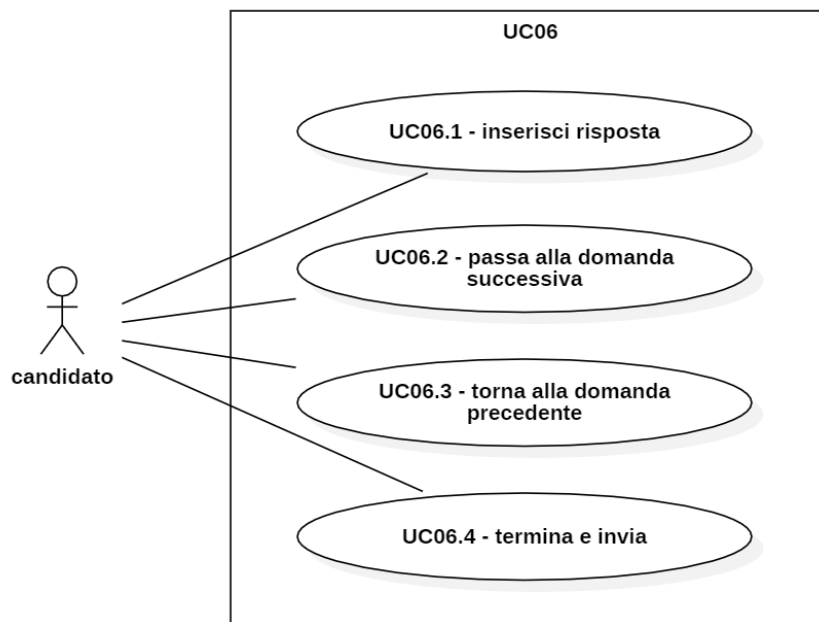


Figura 2.6: Use Case 06

2.2.17 UC06.1

UC06.1: Inserimento di una risposta

Attori Principali: Candidato.

Precondizioni: L'attore vuole inserire una risposta a una domanda.

Postcondizioni: La risposta viene inserita.

Scenario Principale

- Il candidato visualizza una domanda
- Il candidato seleziona la risposta scelta
- La *webapp* salva la risposta

2.2.18 UC06.2

UC06.2: Visualizza la domanda successiva

Attori Principali: Candidato.

Precondizioni: L'attore vuole visualizzare la domanda successiva.

Postcondizioni: La domanda successiva viene visualizzata.

Scenario Principale

- Il candidato preme l'apposito tasto per passare alla domanda successiva
- La *webapp* mostra la domanda successiva

2.2.19 UC06.3

UC06.3: Visualizza la domanda precedente

Attori Principali: Candidato.

Precondizioni: L'attore vuole visualizzare la domanda precedente.

Postcondizioni: La domanda precedente viene visualizzata.

Scenario Principale

- Il candidato preme l'apposito tasto per tornare alla domanda precedente
- La *webapp* mostra la domanda precedente

2.2.20 UC06.4

UC06.4: Termina la prova e invia

Attori Principali: Candidato.

Precondizioni: L'attore vuole terminare la prova.

Postcondizioni: La *webapp* salva la prova e aggiorna la valutazione.

Scenario Principale

- Il candidato preme l'apposito tasto per terminare la prova
- La *webapp* salva le risposte e calcola la valutazione

2.2.21 UC07

UC07: Visualizza i dettagli del questionario - Figura 2.7

Attori Principali: Amministratore.

Precondizioni: L'attore vuole visualizzare i dettagli del questionario.

Postcondizioni: La *webapp* fornisce le informazioni richieste.

Scenario Principale

- L'amministratore visualizza l'ID del questionario
- L'amministratore visualizza il nome del questionario
- L'amministratore visualizza le domande del questionario [UC07.1]
- L'amministratore aggiunge una domanda al questionario [UC07.2]
- L'amministratore visualizza le prove del questionario [UC03]
- L'amministratore aggiunge una nuova prova al questionario [UC04]

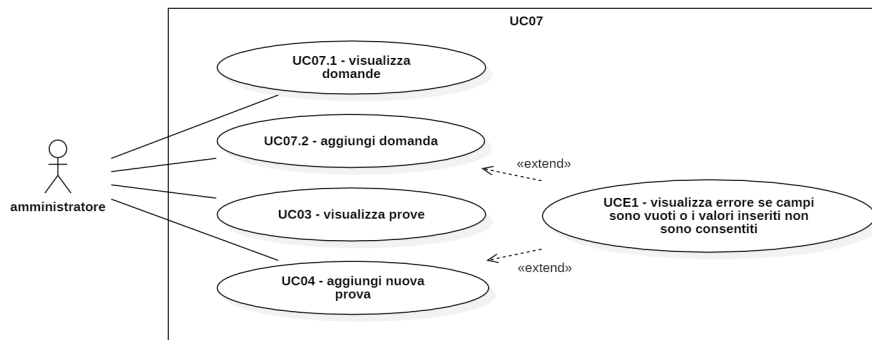


Figura 2.7: Use Case 07

2.2.22 UC07.1

UC07.1: Visualizza le domande

Attori Principali: Amministratore.

Precondizioni: L'attore vuole visualizzare la lista di domande di un questionario.

Postcondizioni: La *webapp* fornisce le informazioni richieste.

Scenario Principale

- L'amministratore si trova nella pagina di dettaglio di un questionario
- L'amministratore visualizza i dettagli delle domande

2.2.23 UC07.2

UC07.2: Aggiunta di una domanda

Attori Principali: Amministratore.

Precondizioni: L'attore vuole aggiungere una nuova domanda.

Postcondizioni: Viene aggiunta una nuova domanda a sistema.

Scenario Principale

- L'amministratore inserisce l'ID dell'immagine
- L'amministratore inserisce il testo
- L'amministratore inserisce la soluzione
- L'amministratore inserisce la tipologia
- L'amministratore inserisce il progressivo
- L'amministratore aggiunge la nuova domanda tramite l'apposito tasto
- La domanda viene aggiunta a sistema

Estensioni I campi sono lasciati vuoti o i valori inseriti non sono validi UCE1.

2.3 Requisiti

Da un'attenta analisi dei requisiti e dei casi d'uso effettuata sul progetto è stata stilata la tabella che traccia i requisiti in rapporto agli use case.

Ne sono stati individuati di diversi tipi e si è quindi fatto utilizzo di un codice identificativo per distinguerli così strutturato R(F/V)(N/D/O), dove:

R = requisito

F = funzionale

V = di vincolo

N = obbligatorio (necessario)

D = desiderabile

O = opzionale

Nelle tabelle 2.1 e 2.2 sono riassunti i requisiti, il loro tracciamento con gli use case delineati in fase di analisi e la loro classificazione.

2.3.1 Funzionali

Tabella 2.1: Tabella del tracciamento dei requisiti funzionali

Requisito	Descrizione	Use Case
RFN1	L'amministratore deve poter visualizzare la lista dei questionari	UC01
RFN2	L'amministratore deve poter ricercare un questionario	UC01.1
RFN3	L'amministratore deve poter modificare un questionario	UC01.2
RFN4	L'amministratore deve poter eliminare un questionario	UC01.3
RFN5	L'amministratore deve poter aggiungere un nuovo questionario	UC02
RFN6	L'amministratore deve poter visualizzare i dettagli di un questionario	UC07
RFN7	L'amministratore deve poter visualizzare le domande di un questionario	UC07.1
RFN8	L'amministratore deve poter aggiungere una nuova domanda a un questionario	UC07.2
RFN10	Il sistema deve visualizzare un messaggio d'errore nel caso manchi il nome nell'aggiunta di un nuovo questionario	UCE1
RFN11	Il sistema deve visualizzare un messaggio d'errore nel caso manchi l'ID dell'immagine nell'aggiunta di una nuova domanda	UCE1

RFN12	Il sistema deve visualizzare un messaggio d'errore nel caso manchi il testo nell'aggiunta di una nuova domanda	UCE1
RFN13	Il sistema deve visualizzare un messaggio d'errore nel caso manchi il progressivo nell'aggiunta di una nuova domanda	UCE1
RFN14	Il sistema deve visualizzare un messaggio d'errore nel caso manchi la soluzione nell'aggiunta di una nuova domanda	UCE1
RFN15	Il sistema deve visualizzare un messaggio d'errore nel caso manchi l'ID del questionario nell'aggiunta di una nuova domanda	UCE1
RFN16	L'amministratore deve poter visualizzare la lista di prove	UC03
RFN17	L'amministratore deve poter ricercare una prova per nome	UC03.1.1
RFN18	L'amministratore deve poter ricercare una prova per email	UC03.1.2
RFN19	L'amministratore deve poter ricercare una prova per ID	UC03.1.3
RFN20	L'amministratore deve poter ricercare una prova per valutazione	UC03.1.4
RFN21	L'amministratore deve poter ricercare una prova per ID del questionario	UC03.1.5
RFN22	L'amministratore deve poter eliminare una prova	UC03.2
RFN23	L'amministratore deve poter aggiungere una prova	UC04
RFN24	Il sistema deve visualizzare un messaggio d'errore nel caso manchi l'email nell'aggiunta di una nuova prova	UCE1
RFN25	Il sistema deve visualizzare un messaggio d'errore nel caso manchi il nome del candidato nell'aggiunta di una nuova prova	UCE1
RFN26	Il sistema deve visualizzare un messaggio d'errore nel caso manchi la data di scadenza nell'aggiunta di una nuova prova	UCE1
RFN27	L'amministratore deve poter visualizzare i dettagli di una prova	UC05
RFN28	L'amministratore deve poter visualizzare le risposte di una prova	UC05.1
RFN29	L'amministratore deve poter visualizzare i dettagli delle risposte di una prova	UC05.1
RFN30	L'amministratore deve poter aggiungere una nuova risposta a una prova	UC05.2
RFN31	Il candidato deve poter visualizzare i dettagli della sua prova	UC05
RFN32	Il candidato deve poter iniziare la sua prova	UC06
RFN33	Il candidato deve poter visualizzare una domanda una volta iniziata la prova	UC06
RFN34	Il candidato deve poter inserire una risposta a una domanda	UC06.1
RFN35	Il candidato deve poter passare alla domanda successiva	UC06.2

RFN36	Il candidato deve poter tornare alla domanda precedente	UC06.3
RFN37	Il candidato deve poter concludere la sua prova	UC06.4
RFD38	Il sistema deve segnare il tempo usato per completare la prova	UC06
RFO39	Il sistema deve segnare il tempo usato per rispondere a una domanda	UC06

2.3.2 Vincolo

Tabella 2.2: Tabella del tracciamento dei requisiti di vincolo

Requisito	Descrizione	Classificazione
RV1	L'applicativo lato back-end è realizzato in <i>Java</i> .	Obbligatorio
RV2	L'applicativo lato back-end è realizzato mediante il <i>framework</i> ^[g] <i>Spring</i> .	Obbligatorio
RV3	L'applicativo lato back-end è realizzato mediante il <i>framework</i> <i>Hibernate</i> .	Obbligatorio
RV4	L'applicativo lato front-end è realizzato tramite <i>HTML5</i> , <i>CSS3</i> e <i>JavaScript</i> .	Obbligatorio
RV5	L'applicativo lato front-end è realizzato mediante il <i>framework</i> <i>Bootstrap</i> .	Obbligatorio
RV6	L'applicativo deve essere funzionante in tutte le sue componenti.	Obbligatorio
RV7	L'applicativo deve essere conforme alle regole per l'accessibilità del <i>WCAG</i> .	Obbligatorio

Capitolo 3

Tecnologie e strumenti

In questo capitolo viene data una panoramica delle tecnologie e degli strumenti utilizzati per lo sviluppo.

3.1 Ambiente di sviluppo e IDE

Per lo sviluppo della *webapp* è stata resa disponibile, attraverso il software *VMware Workstation*, una macchina virtuale con sistema operativo *Windows 7*.

Il software citato semplifica il lavoro collaborativo in quanto rende possibile la visualizzazione e l'utilizzo di una macchina virtuale da più macchine fisiche differenti.

Gli IDE utilizzati per lo sviluppo del progetto sono:

- Eclipse (logo in figura 3.1) consigliato dall'azienda per la sua comodità nel [refactoring](#)^[8] e l'integrazione con *Apache Tomcat*, un [webserver](#)^[8] per l'esecuzione di applicazioni web basate su Java, e *Apache Maven*, uno strumento per gestire le dipendenze tra un progetto e le sue librerie;
- Visual Studio Code (logo in figura 3.1) scelto per preferenza personale e usato per il popolamento del [database](#)^[8].



Figura 3.1: Loghi Eclipse e VS Code

3.2 Linguaggi

Per lo sviluppo è stato richiesto dall'azienda di utilizzare i seguenti linguaggi:

3.2.1 Java



Figura 3.2: Logo Java

Java (logo in figura 3.2) è un linguaggio di programmazione orientato agli oggetti ampiamente usato nelle applicazioni web. Rispetta il principio WORA (Write once, Run anywhere) grazie alla JVM (Java Virtual Machine); è quindi indipendente dall'hardware dove viene eseguito e il codice non necessita di modifiche in caso di portabilità su più dispositivi.

È stata usata anche JSP (JavaServer Pages), una tecnologia che permette la creazione di pagine web con componenti sia statici sia dinamici, che offre la possibilità di includere codice Java all'interno delle pagine HTML.

3.2.2 HTML5, CSS3, JavaScript



Figura 3.3: Loghi HTML, CSS, JavaScript

Nelle applicazioni web i linguaggi maggiormente in uso per la parte di front-end sono:

- HTML (HyperText Markup Language) (logo in figura 3.3) un linguaggio di markup per la creazione di siti web
- CSS (Cascading Style Sheets) (logo in figura 3.3) un linguaggio per definire la formattazione dei documenti HTML
- JS (JavaScript) (logo in figura 3.3) un linguaggio di programmazione orientato agli eventi



Figura 3.4: Logo Python

3.2.3 Python

Python (logo in figura 3.4) è un linguaggio di programmazione ad alto livello. La scelta del suo utilizzo è stata dettata dall'ampia gamma di librerie a disposizione che hanno permesso di velocizzare il lavoro di popolazione del database.

3.3 Framework

3.3.1 Spring



Figura 3.5: Logo Spring

Spring (logo in figura 3.5) è un framework Java che permette di gestire le dipendenze tra i diversi componenti di un'applicazione, facilitando l'organizzazione e il riutilizzo del codice. Una delle caratteristiche principali di Spring è il fatto di essere modulare, offrendo quindi la possibilità di utilizzare solo le funzionalità necessarie senza il bisogno di importare l'intero framework.

I moduli utilizzati maggiormente per lo sviluppo appartengono al Core Container, in figura 3.6; tra questi ci sono:

- Beans, che insieme al modulo Context fa parte dell'IoC (Inversion of Control) Container, il responsabile della gestione degli oggetti dell'applicazione. Beans fornisce le classi per l'istanziamento degli oggetti come BeansFactory e le funzionalità di base;
- Context è invece definito come una sotto interfaccia di Beans e aggiunge delle funzionalità più specifiche;
- Expression Language, il quale permette di eseguire espressioni a runtime.

Inoltre è stato usato anche Spring Security per concedere le autorizzazioni necessarie all'uso della *webapp*.

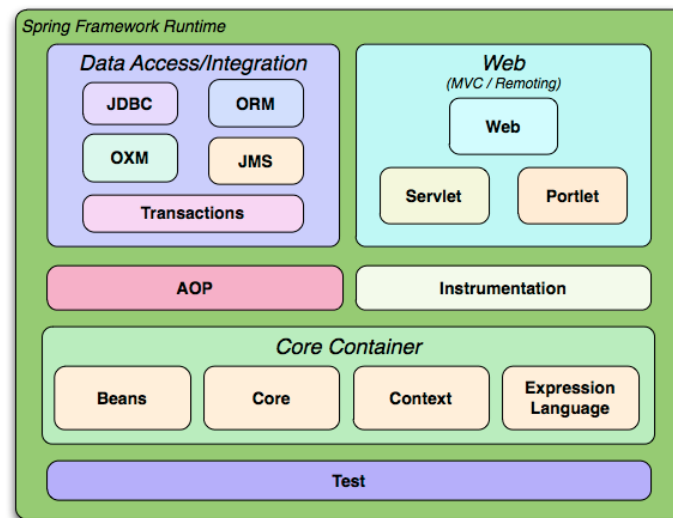


Figura 3.6: Diagramma delle componenti di Spring

3.3.2 Apache Struts



Figura 3.7: Logo Apache Struts

Struts (logo in figura 3.7) è un framework [open source](#)^[g] per lo sviluppo di applicazioni web in Java che si integra con Spring. Fornisce un'infrastruttura per creare webapp basate su un modello architetturale MVC (Model-View-Controller), portando grandi vantaggi nello sviluppo, nel test e nella manutenzione di ciascuna parte indipendentemente dall'altra.

3.3.3 Hibernate



Figura 3.8: Logo Hibernate

Hibernate (logo in figura 3.8) è un framework open source per la gestione dell'accesso ai dati in applicazioni Java. Usa la tecnica ORM (Object-Relational Mapping), la quale consente agli sviluppatori di mappare oggetti Java a tabelle in un database relazionale, semplificando così l'interazione tra il codice e i dati memorizzati nel database. Inoltre gestisce automaticamente le operazioni di persistenza, come l'inserimento, l'aggiornamento e la rimozione dei dati, occupandosi della sincronizzazione tra gli oggetti e il database.

Hibernate evita la scrittura di codice [SQL](#), favorisce la separazione tra la logica di business e di accesso ai dati, e migliora la manutenibilità e la flessibilità delle applicazioni.

3.3.4 Bootstrap



Figura 3.9: Logo Bootstrap

Bootstrap (logo in figura 3.9) è un framework open source per il front-end usato nella creazione di *webapp responsive*. Definisce delle classi il cui inserimento all'interno dell'attributo "*class*" degli elementi HTML permette di gestire l'interfaccia grafica senza dover ricorrere a fogli di stile personalizzati. Inoltre mette a disposizione anche una serie di plugin JavaScript con lo scopo di aggiungere componenti interattivi come navbar, form, card e caroselli senza la necessità di creare del codice personalizzato.

3.4 Librerie

3.4.1 JSTL

JSTL (JavaServer Pages Standard Tag Library) è una libreria di tag personalizzati per applicazioni web Java. Si tratta di un'estensione di JSP creata per semplificare e migliorare la separazione tra la logica di presentazione e la logica di business. JSTL fornisce un insieme di tag personalizzati utilizzati, ad esempio durante lo sviluppo del progetto, per eseguire operazioni come cicli, condizioni e formattazione dei dati. Il codice JSP, in questo modo, diventa più leggibile facilitando di conseguenza la manutenzione e la gestione delle pagine web.

3.4.2 Apache Commons



Figura 3.10: Logo Apache Commons

Apache Commons (logo in figura 3.10) offre una raccolta di librerie frequentemente utilizzate nello sviluppo di applicazioni Java; è suddiviso in diverse componenti, ognuna delle quali offre numerose funzionalità. All'interno del progetto sono state usate principalmente due librerie:

- Apache Commons Lang che fornisce delle funzionalità per la gestione delle stringhe, delle date, delle eccezioni e altre operazioni comuni;
- Apache Commons Collections che offre una vasta gamma di collezioni, liste e mappe e le funzionalità per operare su di esse.

3.4.3 JQuery



Figura 3.11: Logo JQuery

JQuery (logo in figura 3.11) è una libreria di JavaScript che nasce con l'obiettivo di semplificare la creazione di pagine web, permettendo di agevolare la manipolazione del DOM (Document Object Model) e l'interazione con i vari elementi. La scelta del suo utilizzo nel progetto, rispetto a JavaScript nativo, è dovuta alla riduzione della complessità del codice.

Capitolo 4

Progettazione e codifica

4.1 Base di dati

L'azienda usa un sistema di gestione dei database ispirato alla tecnica NoSQL (Not Only SQL); in questo modo ottiene maggiore scalabilità e flessibilità nei modelli, permettendo di inserire e combinare facilmente le tabelle senza dover modificare lo schema nella sua interezza. Infatti le tabelle che utilizza non fanno uso di chiavi composte e non dipendono dai campi inseriti.

Per popolare il database si è scelto di utilizzare Python. I dati erano presenti in file Word ed Excel e sono stati poi importati grazie a DBeaver, uno strumento per la gestione di database.

Tablelle

Per la creazione delle tabelle è stata fatta un'attenta analisi iniziale, che poi ha comunque subito delle modifiche con l'evolversi dello sviluppo, fino ad arrivare al risultato finale composto da quattro tabelle:

- QuestionarioStruttura, composta come illustrato dalla tabella [4.1](#), forma la struttura di un questionario;
- QuestionarioStrutturaItem, tabella [4.2](#), forma la struttura di una domanda del questionario;
- QuestionarioUtente, rappresentata dalla tabella [4.3](#), forma la struttura della prova di un candidato;
- QuestionarioUtenteItem, strutturata come in tabella [4.4](#), corrisponde alla risposta a una domanda della prova di un candidato.

QuestionarioStruttura	
Id	Identificativo univoco di ogni QuestionarioStruttura
Nome	Nome del questionario

Tabella 4.1: La tabella rappresenta la struttura di QuestionarioStruttura

QuestionarioStrutturaItem	
Id	Identificativo univoco di ogni QuestionarioStrutturaItem
idQuestionarioStruttura	Identificativo del QuestionarioStruttura a cui si riferisce
Progressivo	Numero della domanda
ItemID	Id dell'immagine della domanda a cui si riferisce
Descrizione	Testo dell'immagine della domanda
Tipologia	Tipologia della domanda
Soluzione	Risposta corretta alla domanda

Tabella 4.2: La tabella rappresenta la struttura di QuestionarioStrutturaItem

QuestionarioUtente	
Id	Identificativo univoco di ogni QuestionarioUtente
IdQuestionarioStruttura	Identificativo del QuestionarioStruttura a cui si riferisce
DescrizioneUtente	Nome e cognome del candidato
Email	Indirizzo email del candidato
IpAddress	Indirizzo Ip del candidato
ExpirationDate	Data limite per la compilazione del questionario
Stato	Stato della prova (PENDING, EXPIRED, COMPLETED)
Valutazione1	Valutazione della parte di logica
Valutazione2	Valutazione della parte di comprensione del testo
Valutazione3	Valutazione della parte di competenze digitali
Valutazione4	Valutazione della parte di inglese
ValutazioneTotale	Valutazione totale della prova
ExecutionTime	Tempo usato dal candidato per completare la prova

Tabella 4.3: La tabella rappresenta la struttura di QuestionarioUtente

QuestionarioUtenteItem	
Id	Identificativo univoco di ogni QuestionarioUtenteItem
IdQuestionarioStrutturaItem	Identificativo del QuestionarioStrutturaItem a cui si riferisce
IdQuestionarioUtente	Identificativo del QuestionarioUtente a cui si riferisce
RispostaUtente	Risposta dell'utente alla domanda
Stato	Stato della risposta (PENDING, ESATTA, ERRATA)

Tabella 4.4: La tabella rappresenta la struttura di QuestionarioUtenteItem

4.2 Architettura

Per la parte di *backend* l'azienda utilizza un'architettura di tipo MDA (Model-Driven Architecture); in questo modo il modello diventa la parte fondamentale dell'applicazione, indipendentemente dalla tecnologia sottostante. Questo approccio promuove la riusabilità, l'efficienza e la manutenibilità del software.

Insieme all'architettura MDA viene usato anche il pattern architetturale MVC (Model-View-Controller), messo a disposizione da Apache Struts, descritto in figura 4.1.

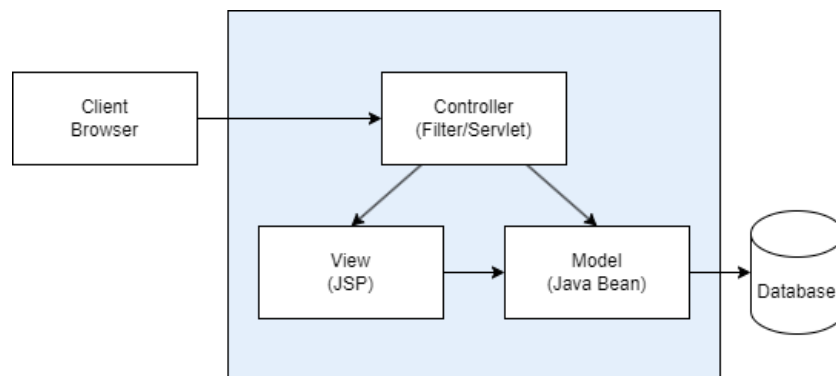


Figura 4.1: Diagramma Apache Struts MVC

4.2.1 Model

In Struts, il Model rappresenta la parte dell'applicazione che gestisce i dati e la logica di business, come ad esempio oggetti JavaBean o dati dal database. Nell'azienda, il Model è strutturato nel seguente modo:

- **XML**, file utilizzato per mappare gli attributi di una classe e inserirli nel database;
- **DB**, esempio di codice 4.1, classe dove viene definito l'oggetto i cui attributi devono corrispondere a quelli definiti nel file *xml*;
- **DAO**, che rappresenta l'interfaccia in cui vengono inserite le firme delle funzioni per manipolare i dati del database. L'implementazione delle funzioni avviene poi in una classe denominata, per standard aziendale, *DaoHibernate*;

- **Service**, classe al cui interno sono definite le funzioni, poi utilizzate dal Controller, per interagire con il database. La classe Service dopo aver manipolato i dati invoca le funzioni della classe Dao necessarie, come si può vedere ad esempio nel listato 4.2;
- **Manager**, è un supporto alle classi Service, non utilizzato nel progetto.

Listing 4.1: QuestionarioUtenteItemDB.java

```
1 public class QuestionarioUtenteItemDB extends BaseBeanInstituteDB
2 {
3     private Long      idQuestionarioStrutturaItem;
4     private Long      idQuestionarioUtente;
5     private String    rispostaUtente;
6     private String    stato;
7
8     public QuestionarioUtenteItemDB(
9
10        //campi del BaseBeanInstituteDB
11        Long          id,
12        Long          versionBean,
13        GregorianCalendar created,
14        GregorianCalendar lastModified,
15        String        institute,
16        //campi della risposta
17        Long          idQuestionarioStrutturaItem,
18        Long          idQuestionarioUtente,
19        String        rispostaUtente,
20        String        stato
21    ) {
22        super(id, versionBean, null, created, lastModified,
23            institute);
24        this.idQuestionarioStrutturaItem =
25            idQuestionarioStrutturaItem;
26        this.idQuestionarioUtente       = idQuestionarioUtente;
27        this.itemID                     = itemID;
28        this.rispostaUtente             = rispostaUtente;
29        this.tipologia                  = tipologia;
30        this.stato                      = stato;
31    }
32    //getters e setters dei campi
33    ...
34 }
```

Listing 4.2: Esempio della funzione details di QuestionarioUtenteService.java

```
1 public interface QuestionarioUtenteService extends Service {
2     @Autowired QuestionarioUtenteDao questionarioUtenteDao;
3     @Autowired QuestionarioUtenteItemDao
4         questionarioUtenteItemDao;
5
6     public QuestionarioUtente details(Long id) throws Exception {
7         //recupero i dettagli dell'oggetto QuestionarioUtenteDB
8         QuestionarioUtenteDB questionarioUtenteDB =
9             questionarioUtenteDao.find(QuestionarioUtenteDB.class,
10                id);
11
12         if(questionarioUtenteDB == null) return null;
13
14         //creo un oggetto QuestionarioUtenteItemDB e imposto il suo
15         //attributo IdQuestionarioUtente con l'id del QuestionarioUtente
16         //del quale voglio ricavare le risposte
17         QuestionarioUtenteItemDB questionarioUtenteItemDBFinder =
18             new QuestionarioUtenteItemDB;
19         questionarioUtenteItemDBFinder.setIdQuestionarioUtente(id
20            );
21
22         //recupero tutti gli oggetti che coincidono alla ricerca
23         List<QuestionarioUtenteItemDB> questionarioUtenteItemDBs
24             = questionarioUtenteItemDao.findByExample(
25                questionarioUtenteItemDBFinder);
26
27         return new QuestionarioUtente(questionarioUtenteDB,
28            questionarioUtenteItemDBs);
29     }
30 }
```

4.2.2 View

La View in Struts è rappresentata dalle pagine JSP, in cui è possibile visualizzare dinamicamente i dati dal Model. L'utente in questo modo ha la possibilità di interagire con i dati richiamando delle Action, descritte in seguito, associate tramite la configurazione del file *struts.xml*.

4.2.3 Controller

Il Controller in Struts è gestito dall'ActionServlet, il cui scopo è instradare le richieste degli utenti alle azioni appropriate. Un'azione è una classe Java che interagisce con il Model per ottenere dati o aggiornarli.

All'interno delle Action si trovano degli oggetti Service, per ciascun modello di interesse, dichiarati con *@Autowired*, annotazione di Spring per indicare le dipendenze dei *bean*^[5]. Esse sono necessarie per recuperare e aggiornare i dati del Model dal database.

Il Controller dell'azienda è stato studiato per coprire il maggior numero di casi con meno Action possibili ed è strutturato come descritto di seguito.

Cartella Form è la cartella nella quale possiamo trovare le classi Form e SearchForm. Se prendiamo come esempio QuestionarioUtente si ha dunque:

- QuestionarioUtenteForm, classe in cui sono definiti i campi di input per la creazione o la modifica di un oggetto QuestionarioUtente. Viene usata quando l'utente, nella pagina *questionarioUtente_form*, visualizza il form in cui ha la possibilità di inserire i dati che poi andranno a valorizzare gli attributi di questa classe;
- QuestionarioUtenteSearchForm, classe in cui sono definiti i campi di input per la ricerca di un oggetto QuestionarioUtente.

Cartella Action è la cartella in cui sono definite le classi Action, ossia i Controller, indicate con l'annotazione di Spring *@Controller*. Prendendo sempre in esempio QuestionarioUtente, le classi sono suddivise nel seguente modo:

- QuestionarioUtenteAction, descritto nel listato 4.3, usata per le pagine non contenenti un form; implementa funzioni come *details*, che grazie al metodo *details* di QuestionarioUtenteService ritorna i dettagli di un oggetto tramite il suo id, e *delete*, che utilizza anch'essa il Service per eliminare l'oggetto dal database;
- QuestionarioUtenteFormAction, quest'azione è utilizzata per l'inserimento e la modifica di un oggetto. Le funzioni chiamate una volta entrati in pagina, *input* per un nuovo oggetto, *edit* per la modifica, valorizzano gli attributi di QuestionarioUtenteForm e una volta modificati dall'utente il loro valore viene salvato nel database attraverso il corrispondente Service;
- QuestionarioUtenteSearchFormAction, è un'azione usata nelle pagine di ricerca. Definisce le funzioni *input* e *search*: la prima, all'entrata in pagina, valorizza i campi di QuestionarioUtenteSearchForm a *null*; la seconda invece crea un oggetto di esempio, i cui attributi sono valorizzati nel form dall'utente, e con il quale viene effettuata una ricerca a database attraverso la funzione del Service *findByExample* di tutti i QuestionarioUtente che corrispondono ai filtri inseriti.

Listing 4.3: Esempio della funzione details di QuestionarioUtenteAction.java

```
1 @Controller(value = recruiting.questionarioUtenteAction)
2 public class QuestionarioUtenteAction extends
   RecruitingActionSupport {
3     @Autowired QuestionarioUtenteService
       questionarioUtenteService;
4     Long id;
5     QuestionarioUtente questionarioUtente;
6
7     public String details() throws Exception {
8         if (id == null) throw new Exception("id is null");
9
10        questionarioUtente = questionarioUtenteService.details(id
11        );
12        if (questionarioUtente == null) throw new Exception("
13        QuestionarioUtente is null");
14
15        return DETAILS;
16    }
17 }
```

Capitolo 5

Prodotto finale

In questo capitolo è illustrato il prodotto finale in tutte le sue componenti.

5.1 Parte dell'amministratore

Le seguenti sezioni illustrano le varie pagine a cui l'amministratore ha accesso per gestire i questionari e le prove dei candidati.

5.1.1 Dashboard

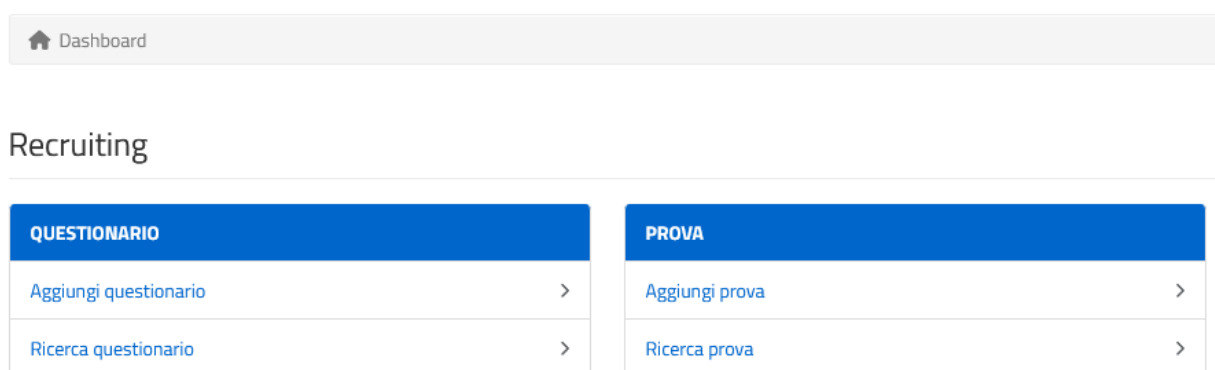


Figura 5.1: Dashboard dell'amministratore, pagina iniziale

La pagina principale è la dashboard, in figura 5.1, dalla quale è possibile svolgere le seguenti azioni:

- ricercare un questionario tra quelli inseriti a sistema;
- creare un nuovo questionario;
- ricercare una prova;
- creare una nuova prova per un candidato.

5.1.2 Ricerca questionario



Figura 5.2: Ricerca questionari

Dalla dashboard si può accedere alla pagina di ricerca di un questionario, in figura 5.2. In questa pagina viene visualizzata la lista di tutti i questionari inseriti a sistema, data la possibilità di impostare dei filtri per la ricerca ed è presente un pulsante che reindirizza alla pagina di aggiunta di un nuovo questionario.

Nella parte sinistra della pagina si trova il menu di ricerca, nel quale possono essere inseriti i filtri desiderati e successivamente, attraverso l'apposito pulsante "cerca", vengono applicati. I filtri messi a disposizione sono:

- id;
- nome;
- data di creazione.

Al centro troviamo la tabella con la lista dei questionari corrispondenti ai filtri di ricerca inseriti, se non sono stati impostati saranno mostrati tutti i questionari nel sistema. Nella tabella per ogni questionario vengono visualizzati:

- id;
- data di creazione;
- nome;
- un pulsante per modificarlo;
- un pulsante per visualizzarne i dettagli.

5.1.3 Creazione e modifica di un questionario

Figura 5.3: Creazione di un nuovo questionario

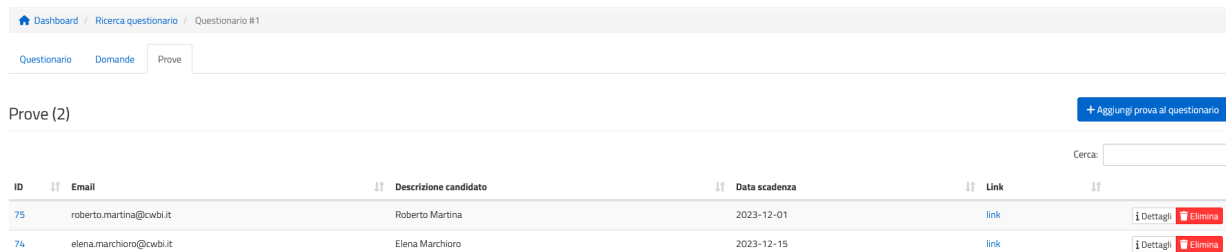
Nella pagina illustrata in figura 5.3, a cui si può accedere dalla dashboard o dalla lista dei questionari, viene data all'amministratore la possibilità di creare un nuovo questionario. Il nome è l'unico campo da compilare presente nel *form* ed è necessario inserirlo per poter salvare il questionario.

5.1.4 Dettaglio di un questionario

Figura 5.4: Dettaglio di un questionario

Progressivo	ID immagine	Soluzione	Tipologia	
1	4053	C	LOGICA	i Dettagli Elimina
2	4404	A	LOGICA	i Dettagli Elimina
3	3568	C	LOGICA	i Dettagli Elimina
4	4654	C	LOGICA	i Dettagli Elimina
5	6581	D	LOGICA	i Dettagli Elimina
6	6464	A	LOGICA	i Dettagli Elimina

Figura 5.5: Domande di un questionario



Dashboard / Ricerca questionario / Questionario #1

Questionario Domande Prove

Prove (2) + Aggiungi prova al questionario

Cerca:

ID	Email	Descrizione candidato	Data scadenza	Link	
75	roberto.martina@cwbi.it	Roberto Martina	2023-12-01	link	i Dettagli Elimina
74	elena.marchioro@cwbi.it	Elena Marchioro	2023-12-15	link	i Dettagli Elimina

Figura 5.6: Prove di un questionario

La pagina del dettaglio di un questionario, raggiungibile dalla ricerca questionario e dal dettaglio di una prova, è composta da tre schede:

1. dettaglio, in figura 5.4, nella quale sono visualizzati il nome del questionario e la data di creazione; si ha inoltre la possibilità di modificare o eliminare il questionario. Conseguentemente all'eliminazione vengono rimosse tutte le domande e le prove che si riferiscono a esso;
2. domande, in figura 5.5, in cui sono visualizzati una tabella con tutte le domande afferenti al questionario e un pulsante per aggiungerne di nuove. Per ogni domanda sono disponibili le seguenti azioni e informazioni:
 - o progressivo;
 - o ID dell'immagine corrispondente;
 - o risposta corretta;
 - o tipologia;
 - o dettaglio della domanda;
 - o eliminazione della domanda.
3. prove, in figura 5.6, nella quale viene mostrata la lista delle prove relative al questionario con le seguenti informazioni:
 - o id;
 - o email del candidato;
 - o descrizione del candidato;
 - o data di scadenza;
 - o link per accedere alla prova.

Viene data anche la possibilità di accedere al dettaglio, eliminare o aggiungere una prova.

5.1.5 Dettaglio di una domanda

Figura 5.7: Dettaglio di una domanda

Id	Risposta Esatta	Risposta Candidato	
3360	C		Dettaglio Elimina
3310	C	C	Dettaglio Elimina

Figura 5.8: Risposte a una domanda

Nella pagina del dettaglio di una domanda, in figura 5.7, a cui si accede dal dettaglio di un questionario, è possibile eliminare o modificare la domanda. Le informazioni che vengono mostrate sono:

- la data di creazione;
- il progressivo;
- la tipologia;
- l'itemID;
- la soluzione;
- l'immagine associata all'itemID;
- una scheda con una lista delle risposte che i vari candidati hanno dato alla domanda, in figura 5.8.

5.1.6 Creazione e modifica di una domanda

Figura 5.9: Creazione di una domanda

Dal dettaglio di un questionario e dal dettaglio di una domanda si può arrivare alla pagina di aggiunta o modifica, in figura 5.9. Il *form* usato per inserire o modificare i vari attributi di una domanda è composto da:

- progressivo;
- tipologia;
- ID immagine, l'identificativo dell'immagine della domanda;
- soluzione;
- descrizione, il testo della domanda trascritto.

Una volta inseriti i vari campi è possibile salvarli attraverso l'apposito pulsante.

5.1.7 Ricerca di una prova

ID	Stato	Descrizione candidato	Data scadenza	Valutazione	
76	EXPIRED	Mario Rossi	22/11/2023	La prova è scaduta.	Link i Dettagli
75	PENDING	Roberto Martina	01/12/2023	Prova in attesa di completamento.	Link i Dettagli
74	COMPLETED	Elena Marchioro	15/12/2023	46	Link i Dettagli

Figura 5.10: Ricerca prove

Come per la ricerca di un questionario, la pagina di ricerca di una prova, in figura 5.10, è composta da un menu sulla sinistra, la lista delle varie prove inserite a sistema

al centro e, in alto a destra, un pulsante per l'aggiunta di una nuova prova. Il menu di ricerca ha come filtri:

- id;
- email del candidato;
- descrizione del candidato;
- id del questionario afferente alla prova;
- data di creazione.

La tabella è invece composta dalla lista delle prove che corrispondono ai filtri inseriti. Per ogni prova si hanno i seguenti elementi:

- id;
- stato;
- descrizione candidato;
- data di scadenza;
- valutazione;
- pulsante per copiare il link da inviare al candidato;
- pulsante per visualizzare il dettaglio della prova.

5.1.8 Creazione e modifica di una prova

Dashboard / Ricerca prova / Aggiungi prova

Aggiungi una nuova prova

Questionario e data di scadenza della prova*:

Questionario:
Modulo1 (ID-1)

Questionario dal quale verranno prese le domande da mostrare nella prova

Data scadenza*:
15/12/2023

Data di scadenza per la compilazione del questionario

Descrizione ed email del candidato*:

Descrizione candidato*:
Elena Marchioro

Nome e cognome del candidato

Email*:
elena.marchioro@cwbi.it

Email del candidato

(*) Richiesto

X Annulla ✓ Salva

Figura 5.11: Creazione di una nuova prova

Dalla ricerca di una prova, dalla dashboard o dal dettaglio di un questionario si può accedere alla pagina di aggiunta o modifica di una prova, in figura 5.11. Per aggiungere una prova di un candidato a sistema è necessario inserire i seguenti campi:

- questionario; se l'accesso alla pagina avviene attraverso la pagina di dettaglio di un questionario questo campo non verrà visualizzato;
- data di scadenza;

- descrizione del candidato;
- email del candidato.

Una volta inseriti i vari campi è necessario salvare le modifiche attraverso l'apposito pulsante in fondo alla pagina.

5.1.9 Dettaglio di una prova

Dashboard / Ricerca prova / Prova #74

Dettagli prova Risposte

Dettaglio prova (ID: 74) Modifica Elimina

PROVA	VALUTAZIONE	CANDIDATO
Questionario afferente: i Dettaglio	Tempo di completamento (ore:minuti): 1:06	Descrizione candidato: Elena Marchioro
Data creazione: 17/11/2023	Capacità logiche: 16	Email: elena.marchioro@cwbi.it
Data scadenza: 15/12/2023	Capacità di comprensione del testo: 16	
Stato: COMPLETED	Competenze digitali: 7	
Link (da inviare al candidato): /cwgtest-web-backoffice/recruitingPublic/questionari/utente/details.action?id=74	Inglese: 7	
	Totale: 46	

Figura 5.12: Dettaglio di una prova completata

Dashboard / Ricerca prova / Prova #74

Dettagli prova Risposte

Risposte (50) + Aggiungi risposta

Cerca:

Progressivo	Tipologia	Risposta Esatta	Risposta Candidato	Status	
1	LOGICA	C	C	ESATTA	i Dettaglio Elimina
2	LOGICA	A	A	ESATTA	i Dettaglio Elimina
3	LOGICA	C	C	ESATTA	i Dettaglio Elimina
4	LOGICA	C	C	ESATTA	i Dettaglio Elimina
5	LOGICA	D	D	ESATTA	i Dettaglio Elimina
6	LOGICA	A	A	ESATTA	i Dettaglio Elimina
7	LOGICA	D	C	ERRATA	i Dettaglio Elimina
8	LOGICA	B	B	ESATTA	i Dettaglio Elimina

Figura 5.13: Risposte di una prova

Al dettaglio di una prova, figura 5.12, si può accedere o dalla ricerca di una prova o dal dettaglio di un questionario. La pagina è composta da due schede:

1. dettaglio, che mostra informazioni diverse in base allo stato della prova. Se si trova negli stati "PENDING" o "EXPIRED" viene visualizzato:
 - data di creazione;
 - data di scadenza;

- stato;
- descrizione del candidato;
- email del candidato.

Quando la prova viene completata lo stato viene impostato su "COMPLETED" e oltre alle informazioni già elencate vengono mostrate anche le seguenti:

- tempo per il completamento della prova;
- valutazione parte di logica;
- valutazione parte di comprensione del testo;
- valutazione parte di competenze informatiche;
- valutazione parte di inglese;
- valutazione totale.

Da questa scheda è anche possibile eliminare la prova o modificarla.

- risposte, permette di visualizzare la lista delle risposte con le loro informazioni, dare la possibilità di aggiungere una nuova risposta, accedere al dettaglio o eliminare una risposta, come mostrato in figura 5.13.

5.1.10 Dettaglio di una risposta

DETTAGLIO
Stato: ESATTA
Progressivo: 1
Risposta corretta: C
Risposta del candidato: C

Figura 5.14: Dettaglio di una risposta

Nella pagina di dettaglio di una risposta, in figura 5.14, a cui si può accedere dal dettaglio di una prova o una domanda, oltre a poter eliminare o modificare la risposta, vengono visualizzati i seguenti dati:

- stato;
- progressivo;
- risposta corretta;
- risposta del candidato.

5.1.11 Creazione e modifica di una risposta

The screenshot shows a web interface for adding a new response. At the top, there is a breadcrumb trail: 'Dashboard / Aggiungi risposta'. Below this, the title 'Aggiungi una nuova risposta' is displayed. The form is divided into two main sections. The left section is titled 'Risposta*' and contains three input fields: 'Progressivo:' with a text input field containing 'Progressivo:', a small text label 'Numero della risposta nella prova, corrispondente alla domanda con lo stesso progressivo', and 'Risposta:' with a dropdown menu showing 'A' and a label 'Risposta del candidato'. The right section is titled 'idQuestionarioUtente*' and contains a dropdown menu showing 'roberto.martina@cwbi.it (ID:75)' and a label 'idQuestionarioUtente'. At the bottom left of the form, there is a note '(*) Richiesto'. At the bottom right, there are two buttons: 'X Annulla' and '✓ Salva'.

Figura 5.15: Creazione di una risposta

Alla pagina di creazione o modifica di una risposta si accede dal dettaglio di una risposta, di una domanda o di una prova. Per inserire la risposta a sistema è necessario inserire i seguenti campi, come mostrato in figura 5.15:

- progressivo, usato per collegare la risposta alla domanda corrispondente;
- risposta del candidato;
- prova a cui si riferisce.

5.2 Parte del candidato

In questa sezione vengono illustrate le pagine a cui un candidato, una volta ricevuto il *link* alla prova dall'amministratore, può accedere.

5.2.1 Pagina iniziale



Figura 5.16: Pagina di inizio del candidato

La pagina in figura 5.16 viene mostrata una volta che il candidato apre il *link* ricevuto. Vengono fornite delle informazioni per la compilazione della prova e attraverso il pulsante "Iniziamo!" è possibile iniziare la prova.

5.2.2 Esecuzione della prova

Nella pagina in figura 5.17 sono visualizzate le varie domande a cui il candidato deve rispondere. Si ha la possibilità di svolgere diverse azioni come:

- selezionare la risposta che si ritiene corretta attraverso i bottoni "Risposta 1", "Risposta 2", "Risposta 3" e "Risposta 4";
- navigare tra le varie domande grazie al menu di navigazione sulla destra;
- tornare alla domanda precedente;
- passare alla domanda successiva;
- tornare alla pagina iniziale;
- concludere la prova attraverso l'apposito pulsante, visibile solo per l'ultima domanda.

CWBI
Bank Innovation

Domanda 17/50

Domanda 17/50 (codice originale: 4586), non risposte 4/150. Capacità logiche

Quale delle seguenti alternative contiene una lettera ripetuta esattamente cinque volte?

- LWWLWLVWQWQW
- HHJKJJKHHKHK
- YYIUUYUUYIUU
- EEETRETTTRRR

Seleziona la risposta che ritieni corretta

Risposta 1
Risposta 2
Risposta 3
Risposta 4

← Precedente
→ Successiva

[X Torna alla pagina iniziale](#)

Navigazione domande

01	02	03	04	05	06	07	08
09	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50						

Figura 5.17: Esempio domanda durante il completamento di una prova


5.2.3 Pagina dei risultati

CWBI
Bank Innovation


Grazie Elena Marchioro!

Hai completato il questionario in **1 ora e 6 minuti** con una valutazione totale di:


46 / 50




Capacità logiche:
16 / 18



Capacità di comprensione del testo:
16 / 18



Competenze digitali:
7 / 7



Inglese:
7 / 7

Risposte

Numero domanda	Risposta candidato	Risposta corretta	Esito
1	C	C	ESATTA
2	A	A	ESATTA
3	C	C	ESATTA
4	C	C	ESATTA
5	D	D	ESATTA
6	A	A	ESATTA
7	C	D	ERRATA
8	B	B	ESATTA

Figura 5.18: Risultati di una prova

Una volta conclusa la prova il candidato viene portato nella pagina in figura 5.18, dove ha la possibilità di visualizzare i suoi risultati per ogni parte della prova e controllare la correttezza di ogni risposta. Se il candidato riapre il *link* inviatogli dall'amministratore dopo aver concluso la prova, egli verrà indirizzato su questa pagina.

5.2.4 Prova scaduta



Oh no! Il questionario è scaduto il 22/11/2023 !

Se credi ci sia stato un errore non esitare a contattarci!

Figura 5.19: Pagina prova scaduta

Se la data di scadenza della prova è precedente al giorno in cui si accede al *link* ricevuto e non è ancora stata completata si visualizzerà questa pagina, in figura 5.19, la quale informa il candidato che la prova non può più essere completata in quanto scaduta.

Capitolo 6

Conclusioni

In conclusione lo stage, della durata di trecento ore, ha visto lo sviluppo dell'applicazione web nella sua interezza. L'azienda ora ha la possibilità di proporre i questionari valutativi ai candidati senza la necessità di organizzare un incontro in sede e dando immediatamente i risultati, contrariamente a come si svolgevano i colloqui precedentemente, in cui c'era la necessità della presenza del candidato e di una correzione manuale da parte di colui che somministrava il questionario. Lo sviluppo ha quindi soddisfatto tutti i requisiti necessari poiché il prodotto sia pronto all'uso, rendendo disponibili ricerca, creazione e compilazione di questionari e prove.

Di seguito viene fatto un resoconto finale che comprende l'attività di verifica e validazione del prodotto e delle valutazioni personali sull'esperienza di stage.

6.1 Verifica e validazione

La verifica, ovvero il controllo che all'introduzione di nuovi elementi nel codice non vengano prodotti errori e che i requisiti siano rispettati, è stata effettuata durante l'intero sviluppo del progetto. L'attività di sviluppo è stata suddivisa in più parti per permettere un controllo efficiente e accurato, verificando che a ogni introduzione di nuovo codice l'insieme fosse funzionante.

La validazione, necessaria per ottenere l'approvazione del progetto, è stata eseguita insieme al tutor *Roberto Martina* che ha appurato la fedeltà ai canoni della struttura aziendale. I test eseguiti hanno riportato un esito positivo e i requisiti sono stati tutti rispettati, fatta eccezione del requisito funzionale opzionale *RFO39*, in tabella 2.1.

6.1.1 Requisiti

Di seguito viene riportata la tabella 6.1 che illustra i requisiti soddisfatti dai test di sistema eseguiti.

Tabella 6.1: Requisiti soddisfatti

Requisito	Descrizione	Esito
RFN1	L'amministratore deve poter visualizzare la lista dei questionari	Soddisfatto
RFN2	L'amministratore deve poter ricercare un questionario	Soddisfatto

RFN3	L'amministratore deve poter modificare un questionario	Soddisfatto
RFN4	L'amministratore deve poter eliminare un questionario	Soddisfatto
RFN5	L'amministratore deve poter aggiungere un nuovo questionario	Soddisfatto
RFN6	L'amministratore deve poter visualizzare i dettagli di un questionario	Soddisfatto
RFN7	L'amministratore deve poter visualizzare le domande di un questionario	Soddisfatto
RFN8	L'amministratore deve poter aggiungere una nuova domanda a un questionario	Soddisfatto
RFN10	Il sistema deve visualizzare un messaggio d'errore nel caso manchi il nome nell'aggiunta di un nuovo questionario	Soddisfatto
RFN11	Il sistema deve visualizzare un messaggio d'errore nel caso manchi l'ID dell'immagine nell'aggiunta di una nuova domanda	Soddisfatto
RFN12	Il sistema deve visualizzare un messaggio d'errore nel caso manchi il testo nell'aggiunta di una nuova domanda	Soddisfatto
RFN13	Il sistema deve visualizzare un messaggio d'errore nel caso manchi il progressivo nell'aggiunta di una nuova domanda	Soddisfatto
RFN14	Il sistema deve visualizzare un messaggio d'errore nel caso manchi la soluzione nell'aggiunta di una nuova domanda	Soddisfatto
RFN15	Il sistema deve visualizzare un messaggio d'errore nel caso manchi l'ID del questionario nell'aggiunta di una nuova domanda	Soddisfatto
RFN16	L'amministratore deve poter visualizzare la lista di prove	Soddisfatto
RFN17	L'amministratore deve poter ricercare una prova per nome	Soddisfatto
RFN18	L'amministratore deve poter ricercare una prova per email	Soddisfatto
RFN19	L'amministratore deve poter ricercare una prova per ID	Soddisfatto
RFN20	L'amministratore deve poter ricercare una prova per valutazione	Soddisfatto
RFN21	L'amministratore deve poter ricercare una prova per ID del questionario	Soddisfatto
RFN22	L'amministratore deve poter eliminare una prova	Soddisfatto
RFN23	L'amministratore deve poter aggiungere una prova	Soddisfatto
RFN24	Il sistema deve visualizzare un messaggio d'errore nel caso manchi l'email nell'aggiunta di una nuova prova	Soddisfatto
RFN25	Il sistema deve visualizzare un messaggio d'errore nel caso manchi il nome del candidato nell'aggiunta di una nuova prova	Soddisfatto
RFN26	Il sistema deve visualizzare un messaggio d'errore nel caso manchi la data di scadenza nell'aggiunta di una nuova prova	Soddisfatto

RFN27	L'amministratore deve poter visualizzare i dettagli di una prova	Soddisfatto
RFN28	L'amministratore deve poter visualizzare le risposte di una prova	Soddisfatto
RFN29	L'amministratore deve poter visualizzare i dettagli delle risposte di una prova	Soddisfatto
RFN30	L'amministratore deve poter aggiungere una nuova risposta a una prova	Soddisfatto
RFN31	Il candidato deve poter visualizzare i dettagli della sua prova	Soddisfatto
RFN32	Il candidato deve poter iniziare la sua prova	Soddisfatto
RFN33	Il candidato deve poter visualizzare una domanda una volta iniziata la prova	Soddisfatto
RFN34	Il candidato deve poter inserire una risposta a una domanda	Soddisfatto
RFN35	Il candidato deve poter passare alla domanda successiva	Soddisfatto
RFN36	Il candidato deve poter tornare alla domanda precedente	Soddisfatto
RFN37	Il candidato deve poter concludere la sua prova	Soddisfatto
RFD38	Il sistema deve segnare il tempo usato per completare la prova	Soddisfatto
RFO39	Il sistema deve segnare il tempo usato per rispondere a una domanda	Non soddisfatto
RV1	L'applicativo lato back-end è realizzato in <i>Java</i> .	Soddisfatto
RV2	L'applicativo lato back-end è realizzato mediante il <i>framework</i> ^[6] <i>Spring</i> .	Soddisfatto
RV3	L'applicativo lato back-end è realizzato mediante il <i>framework Hibernate</i> .	Soddisfatto
RV4	L'applicativo lato front-end è realizzato tramite <i>HTML5</i> , <i>CSS3</i> e <i>JavaScript</i> .	Soddisfatto
RV5	L'applicativo lato front-end è realizzato mediante il <i>framework Bootstrap</i> .	Soddisfatto
RV6	L'applicativo deve essere funzionante in tutte le sue componenti.	Soddisfatto
RV7	L'applicativo deve essere conforme alle regole per l'accessibilità del <i>WCAG</i> .	Soddisfatto

6.1.2 Accessibilità

Per rispettare il requisito di vincolo RV7, in tabella 2.2, e quindi verificare che il prodotto sviluppato sia conforme alle regole di accessibilità definite nel WCAG, si è optato per:

- utilizzare W3C Validator e TotalValidator per verificare la conformità agli standard;
- grazie a NVDA e Chrome Screen reader, verificare che tutte le pagine fossero navigabili attraverso l'utilizzo di uno screen reader;
- usare l'estensione WAVE Evaluation Tool per controlli generali;
- verificare la compatibilità con i principali browser.

Ulteriori attenzioni nel campo dell'accessibilità sono risultate in:

- tabelle accessibili;
- navigazione da tastiera completa e coerente con la struttura grafica delle pagine;
- uso ridotto di colori e contrasto conforme allo standard WCAG AA;
- utilizzo dell'attributo *lang*;
- validazione dell'*input* nei *form* con messaggi di errore informativi e i relativi campi messi in evidenza.

6.2 Consuntivo finale

La pianificazione iniziale dello stage, in tabella 1.1, non è stata rispettata completamente risultando invece come mostrato di seguito, nella tabella 6.2.

Tabella 6.2: Tabella finale della suddivisione delle ore di lavoro

Durata in ore	Descrizione	Settimana
30	Formazione iniziale	W1
50	Raccolta dei requisiti	W1/W2
40	Realizzazione fase 1/2 del prototipo <i>WebApp</i> con i primi use case	W3
40	Realizzazione fase 2/2 del prototipo <i>WebApp</i> con i primi use case	W4
40	Realizzazione con connessioni ai servizi REST per la fase di <i>SIT</i> ^[g]	W5
40	Test e supporto <i>UAT</i> ^[g]	W6
40	Documentazione progetto	W7
20	Chiusura progetto e supporto ai primi ricicli	W8

Andando ad analizzare le varie attività, la raccolta dei requisiti ha occupato più tempo del previsto: si è deciso infatti di concentrarsi maggiormente su questa attività in quanto ritenuta fondamentale per capire tutti gli aspetti dello sviluppo seguente. Contrariamente alle aspettative, è poi stato recuperato del tempo durante la fase di codifica e di test.

In conclusione il lavoro svolto è stato più rapido di quanto supposto permettendo così, grazie al tempo aggiuntivo, un miglioramento qualitativo e un'analisi più approfondita del *Way of Working*^[8] dell'azienda.

6.3 Resoconto finale

6.3.1 Conoscenze acquisite

Le aspettative iniziali sul periodo di stage sono state tutte ampiamente rispettate. L'esperienza infatti ha approfondito temi non attesi inizialmente, in quanto mi è stato permesso di lavorare a contatto con figure appartenenti ad aree differenti. Il lavoro svolto ha visto un coinvolgimento iniziale con la parte tecnica dalla quale sono state apprese le conoscenze necessarie per lo sviluppo del progetto, oltre alla capacità di lavorare in gruppo. Successivamente, durante la fase di test sono stata affiancata a una persona dell'area commerciale con la quale ho assistito alla gestione dei rapporti con i committenti. Infine, essendo il progetto sviluppato relativo all'assunzione di personale, mi è stato insegnato anche il modo in cui valutare le predisposizioni di una persona ed essere più convincente nelle relazioni interpersonali.

6.3.2 Valutazione personale

In conclusione, da un punto di vista personale, ritengo di aver imparato molto durante questa esperienza; se inizialmente ho affrontato diverse difficoltà, in seguito proprio queste si sono sviluppate in opportunità di crescita. Infatti, rispetto al mondo accademico, l'ambiente lavorativo risulta più complesso, in quanto necessita di flessibilità, prontezza nella risoluzione di problemi e le aspettative sono maggiori, ma nonostante le sfide incontrate lungo il percorso, il risultato finale è stato gratificante.

Acronimi e abbreviazioni

- CSS3** Cascading Style Sheets. [2](#)
- CWBT** Codice Web Banking Technology. [1](#)
- HTML5** Hyper Text Markup Language. [2](#)
- JSTL** JSP Standard Tag Library. [2](#)
- REST** REpresentational State Transfer. [3](#)
- SQL** Structured Query Language. [26](#)
- UML** Unified Modeling Language. [6](#)
- W3C** World Wide Web Consortium. [2](#)
- WebApp** Web Application. [3](#), [4](#)

Glossario

Bean oggetti che costituiscono la struttura portante dell'applicazione. [33](#)

Dashboard rappresentazione visuale delle informazioni chiave, le cui caratteristiche principali includono la capacità di fornire una panoramica immediata delle informazioni cruciali e la facilità di lettura e interpretazione. [5](#)

Database sistema organizzato per la raccolta, l'archiviazione, la gestione e la ricerca di dati. [22](#)

Framework architettura software che fornisce una struttura di supporto per lo sviluppo di applicazioni. Offre un insieme di strumenti, librerie, linee guida e convenzioni organizzative che semplificano e agevolano il processo di sviluppo del software. [21](#)

Open source detto di un software il cui codice sorgente è reso disponibile al pubblico, consentendo a chiunque di visualizzarlo, modificare e distribuirlo. [25](#)

Refactoring pratica di sviluppo software che consiste nel ristrutturare il codice sorgente di un programma senza modificarne il comportamento esterno. [22](#)

SIT fase di test nel ciclo di sviluppo di un software, che si concentra sulla verifica della corretta integrazione tra le diverse componenti di un sistema. [3](#)

TortoiseSVN client di interfaccia grafica per il sistema di controllo versione. [2](#)

UAT fase di test che si svolge alla fine del processo di sviluppo del software. Durante questa fase, gli utenti finali testano l'applicazione per verificare se soddisfa i requisiti specificati e se è pronta per l'implementazione in un ambiente di produzione. [3](#)

UML in ingegneria del software *UML* (Unified Modeling Language) è un linguaggio di modellazione e specifica basato sul paradigma object-oriented. L'*UML* svolge un'importantissima funzione di "lingua franca" nella comunità della progettazione e programmazione a oggetti. [5](#)

Way of Working letteralmente "modo di lavorare" si riferisce alla metodologia, agli approcci, alle pratiche e ai comportamenti adottati nel contesto del lavoro. [52](#)

WebApp è un'applicazione software che viene eseguita su un server web e può essere accessibile attraverso un browser. Le web app sono accessibili su diverse piattaforme e dispositivi connessi a Internet, rendendo più flessibile l'accesso e l'uso da parte degli utenti. [2](#)

Webserver software o sistema informatico che gestisce le richieste provenienti attraverso Internet e fornisce di conseguenza le pagine web e altri contenuti al richiedente. [22](#)

Bibliografia

Siti web consultati

Apache Commons. URL: <https://commons.apache.org/>.

Apache Struts. URL: <https://struts.apache.org/>.

Bootstrap. URL: <https://getbootstrap.com/>.

Hibernate. URL: <https://hibernate.org/>.

Java. URL: <https://www.java.com/>.

jQuery. URL: <https://jquery.com/>.

JSP. URL: <https://www.oracle.com/java/technologies/javaserver-faq.html>.

JSTL. URL: <https://it.wikipedia.org/wiki/JSTL>;

Python. URL: <https://www.python.it/>.

Spring. URL: <https://docs.spring.io/springframework/docs/3.0.0.M4/reference/html/ch01s02.html>.