

Università degli Studi di Padova

DIPARTIMENTO DI MATEMATICA “TULLIO LEVI-CIVITA”

CORSO DI LAUREA TRIENNALE IN INFORMATICA



Sviluppo di un modulo web per la gestione
delle campagne marketing in ambiente
bancario/fintech

Tesi di laurea

Relatore

Prof. Ombretta Gaggi

Laureando

Michael Amista'
2009116

ANNO ACCADEMICO 2022-2023

Sommario

Il presente documento descrive il lavoro svolto durante il periodo di stage, della durata di circa trecento ore, dal laureando Michael Amista' presso l'azienda CWBI (Codice Web Banking Innovation).

L'obiettivo dello stage riguarda lo sviluppo di un modulo web per una webapp aziendale preesistente. In particolare ciò che è stato sviluppato è un sistema di gestione delle campagne a supporto dell'area marketing in un contesto bancario/fintech. Vincolo di sviluppo sono le metodologie di analisi, progettazione e sviluppo aziendale, oltre allo stack tecnologico utilizzato dall'azienda. Viene inoltre riportata un'analisi finale che pone a confronto le aspettative iniziali personali con quelli che sono stati i risultati finali e gli obiettivi raggiunti.

Ringraziamenti

Innanzitutto, vorrei ringraziare la Prof. Ombretta Gaggi per la disponibilità e l'aiuto fornitomi nel redarre questo documento, oltre al supporto durante il periodo di stage.

Ringrazio con affetto la mia famiglia per il sostegno, il supporto e gli sforzi fatti durante questi tre anni di studio.

Ringrazio i miei amici per i bellissimi momenti passati insieme durante questi anni universitari.

Desidero infine ringraziare il mio tutor aziendale Roberto Martina per il tempo dedicatomi durante lo stage e per la conoscenza trasmessami con tanta passione e metodo.

Padova, Settembre 2023

Michael Amista'

Indice

1	Introduzione	1
1.1	L'azienda	1
1.2	Aspettative personali	2
1.3	Organizzazione del testo	3
2	Descrizione dello stage	4
2.1	Il problema	4
2.2	Il progetto	4
2.3	Vincoli del progetto	5
3	Analisi dei requisiti	6
3.1	Casi d'uso	6
3.1.1	UC0: Menu	7
3.1.2	UC1: Crea-Modifica progetto	8
3.1.3	UC2: Tabella progetti	9
3.1.4	UC2.2: Visualizzazione lista progetti	11
3.1.5	UC2.3: Visualizzazione progetto in dettaglio	12
3.1.6	UC3: Tabella progetti aperti	13
3.1.7	UC4: Tabella clienti	14
3.1.8	UC4.2: Visualizzazione lista clienti	15
3.1.9	UC4.3: Visualizzazione cliente in dettaglio	15
3.1.10	UC4.3.3: Visualizzazione offerte correlate	16
3.1.11	UC4.3.3.1: Crea-Modifica offerta	17
3.1.12	UC5: Tabella offerte	18
3.1.13	UC5.2: Visualizzazione lista offerte	19
3.1.14	UC5.3: Visualizzazione offerta in dettaglio	20
3.2	Definizione e tracciamento dei requisiti	21
3.2.1	Requisiti funzionali	22
3.2.2	Requisiti di vincolo	25
4	Strumenti utilizzati	26
4.1	Ambiente di lavoro	26
4.2	Framework	27
4.2.1	Spring (core)	27
4.2.2	Hibernate	28
4.2.3	Bootstrap	28
4.2.4	Apache Struts	29
4.3	Linguaggi utilizzati	29

4.3.1	Java	29
4.3.2	HTML5, CSS3 e JavaScript	30
4.4	Librerie a supporto della codifica	30
4.4.1	Apache Commons	30
4.4.2	JSTL	31
4.5	IDE	31
5	Progettazione	32
5.1	Approccio MDA	32
5.2	Architettura MVC	35
6	Prodotto finale	38
6.1	Menu	38
6.2	Creazione di un progetto	39
6.3	Tabella progetti	40
6.4	Dettaglio di un progetto	41
6.5	Tabella progetti aperti	42
6.6	Tabella clienti	42
6.7	Dettaglio di un cliente	43
6.8	Tabella offerte	45
6.9	Dettaglio di un'offerta	46
7	Conclusioni	47
7.1	Test e validazione	47
7.1.1	Test di sistema	48
7.2	Resoconto dello stage	52
	Sitografia	53
	Glossario	54

Elenco delle figure

1.1	Logo dell'azienda	1
3.1	UC0: Menu	7
3.2	UC1: Crea-Modifica progetto	8
3.3	UC2: Tabella progetti	9
3.4	UC2.2: Visualizzazione lista progetti	11
3.5	UC2.3: Visualizzazione progetto in dettaglio	12
3.6	UC3: Tabella progetti aperti	13
3.7	UC4: Tabella clienti	14
3.8	UC4.2: Visualizzazione lista clienti	15
3.9	UC4.3: Visualizzazione cliente in dettaglio	15
3.10	UC4.3.3: Visualizzazione offerte correlate	16
3.11	UC4.3.3.1: Crea-Modifica offerta	17
3.12	UC5: Tabella offerte	18
3.13	UC5.2: Visualizzazione lista offerte	19
3.14	UC5.3: Visualizzazione offerta in dettaglio	20
4.1	Logo di Spring	27
4.2	Spring - core container	27
4.3	Logo di Hibernate	28
4.4	Logo di Bootstrap	28
4.5	Logo di Apache Struts	29
4.6	Logo di Java	29
4.7	Logo dello stack tecnologico del front-end	30
4.8	Logo di Apache Commons	30
4.9	Logo di Eclipse	31
5.1	Architettura MVC di Apache Struts	35
6.1	Modulo Campaign: Menu	38
6.2	Modulo Campaign: Creazione di un progetto	39
6.3	Modulo Campaign: Tabella progetti	40
6.4	Modulo Campaign: Dettaglio di un progetto	41
6.5	Modulo Campaign: Dettaglio di un progetto - Gestione clienti	41
6.6	Modulo Campaign: Tabella progetti aperti	42
6.7	Modulo Campaign: Tabella clienti	42
6.8	Modulo Campaign: Dettaglio di un cliente - Scheda progetti associati	43
6.9	Modulo Campaign: Dettaglio di un cliente - Scheda offerte associate	43
6.10	Modulo Campaign: Dettaglio di un cliente - Aggiungi offerta	44

6.11 Modulo Campaign: Tabella offerte	45
6.12 Modulo Campaign: Dettaglio di un'offerta	46

Elenco delle tabelle

3.1 Requisiti funzionali	25
3.2 Requisiti di vincolo	25
7.1 Test di sistema	51

Capitolo 1

Introduzione

Il seguente capitolo ha la funzione di introdurre l'azienda ospitante presso la quale è stato svolto lo stage. Vengono inoltre elencate quelle che sono le aspettative personali principalmente riguardo la crescita tecnica e professionale.

1.1 L'azienda



Figura 1.1: Logo dell'azienda

CWBI (logo in [figura 1.1](#)), acronimo di "Codice Web Banking Innovation", è un'azienda italiana che opera nel mercato dell'Information Communication Technology e 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.

Fondata a Padova nell'anno 2013, CWBI ha fidelizzato rapporti di collaborazione con aziende nazionali di primaria importanza attraverso la sua struttura interna costituita da professionisti con skills elevate, che negli anni hanno maturato un forte know-how in diversi settori di business quali: Banking, Media and Publishing, Insurance, Industry.

Anni di esperienza permettono all'azienda di affrontare con successo ogni singolo aspetto del ciclo di vita dei progetti nei quali è coinvolta; entusiasmo e visione strategica, accompagnati da un forte orientamento al risultato, sono il motore della sua capacità innovativa.

CWBI offre una vasta gamma di servizi, tra cui:

- sviluppo di applicazioni e portali web-based;
- sviluppo di 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 Aspettative personali

L'obiettivo dell'attività di stage, oltre lo sviluppo del progetto commissionato dall'azienda ospitante, è quello di crescere personalmente e professionalmente tramite un'impronta aziendale caratterizzata da competenza tecnica e professionale.

Di seguito vengono riportati gli obiettivi personali a titolo professionale e formativo da raggiungere:

- apprendimento di Java;
- apprendimento dei framework Spring e Hibernate;
- apprendimento del framework Bootstrap;
- apprendimento del sistema di controllo di versione aziendale;
- apprendimento dei processi aziendali;
- apprendimento degli strumenti per la gestione di progetto;
- apprendimento dell'ambiente di sviluppo;
- studio di fattibilità del progetto e realizzazione dello stesso mediante l'utilizzo delle tecnologie aziendali;
- capacità di trovare soluzioni alternative da quelle proposte.

1.3 Organizzazione del testo

Il [secondo capitolo](#) descrive l'attività di stage definendo il problema da affrontare e i vincoli da rispettare lungo lo sviluppo.

Il [terzo capitolo](#) approfondisce l'analisi dei requisiti effettuata, elencando i casi d'uso raccolti e definendo i rispettivi requisiti.

Il [quarto capitolo](#) approfondisce gli strumenti e le tecnologie utilizzate nello sviluppo del prodotto commissionato e che si pongono come vincolo di progetto.

Il [quinto capitolo](#) approfondisce la progettazione, caratterizzata dall'approccio MDA e dall'architettura del sistema.

Il [sesto capitolo](#) illustra il prodotto finale in tutte le sue componenti.

Nel [settimo capitolo](#) si possono trovare le conclusioni sul lavoro svolto, formate dai test effettuati e da un resoconto finale che pone a confronto le aspettative iniziali con quelli che sono stati i traguardi raggiunti al termine dell'esperienza 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];
- per la sitografia riguardante un certo argomento viene utilizzata la seguente nomenclatura: [argomento](#)^[b].

Capitolo 2

Descrizione dello stage

Il seguente capitolo ha la funzione di introdurre l'attività di stage definendo quella che è la richiesta aziendale e i vincoli di progetto da rispettare nello sviluppo dell'applicativo commissionato.

2.1 Il problema

L'azienda CWBI realizza prodotti software su commissione diretta dei clienti, oltre ad applicativi pensati per raggiungere nuovi acquirenti. L'azienda necessita di una metodologia efficiente per tenere traccia dei contatti con i clienti e delle offerte formulate per uno o più progetti di interesse. Attualmente la soluzione aziendale si basa su una cartella in rete dove sono contenuti tutti i documenti e i contatti con i clienti negli ultimi 10 anni di attività.

Ciò che l'azienda desidera è un applicativo web che consenta di creare, organizzare e tenere traccia di queste informazioni in modo da creare un ambiente di lavoro/tracciamento più efficiente ed efficace.

2.2 Il progetto

L'obiettivo dello stage riguarda quindi lo sviluppo di un modulo web per la webapp aziendale "CW GEST" preesistente. CW GEST è una webapp utilizzata dai dipendenti dell'area commerciale di CWBI e composta da diversi moduli, ciascuno dei quali assolve differenti compiti riguardo la gestione interna aziendale. In particolare ciò che deve essere integrato in CW GEST è un sistema di gestione delle campagne a supporto dell'area marketing nel contesto aziendale, ovvero l'ambiente bancario/fintech.

In particolare, il flusso principale risulta essere il seguente:

1. contatto con il cliente e raccolta preliminare delle informazioni (contatti, idee, budget, ecc.);
2. definizione del progetto da realizzare;
3. creazione dell'offerta;
4. proposta dell'offerta formulata e eventuale contrattazione;
5. approvazione o rifiuto dell'offerta.

La webapp sarà utilizzata dai dipendenti dell'area commerciale di CWBI e dovrà consentire di aggiungere, visualizzare e aggiornare la lista completa dei propri clienti, con annessi i progetti richiesti e le offerte formulate per essi.

2.3 Vincoli del progetto

In questa sezione vengono riportati tutti i vincoli che l'azienda ha imposto per lo sviluppo del prodotto commissionato, suddivisi in vincoli metodologici e tecnologici.

- **Vincoli metodologici:**

1. per il versionamento del codice l'azienda ha imposto l'utilizzo di TortoiseSVN, un servizio di repository da loro utilizzato.

- **Vincoli tecnologici:**

1. utilizzo di Java e dei framework Spring, Hibernate e Struts per lo sviluppo lato server;
2. utilizzo del framework Bootstrap e della libreria JSTL per lo sviluppo lato client;
3. utilizzo di HTML5, CSS3 e JavaScript come supporto allo sviluppo lato client.

Non sono stati imposti limiti sull'utilizzo di eventuali tecnologie aggiuntive.

Capitolo 3

Analisi dei requisiti

Il seguente capitolo assume la funzione di illustrare i casi d'uso e i requisiti raccolti nella fase di analisi del progetto commissionato.

3.1 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](#)^[g]. Tali diagrammi sono risultati fondamentali per individuare correttamente tutti i requisiti del sistema in questione.

Ciascun caso d'uso è costituito da:

- nome del caso d'uso;
- attore primario;
- precondizione;
- postcondizione;
- scenario principale;
- generalizzazioni (qualora fossero presenti);
- estensioni (qualora fossero presenti).

I casi d'uso identificati dalla sigla "UCE" rappresentano un caso d'uso d'errore. L'unico attore coinvolto e identificato come *Utente autenticato* corrisponde ad un qualunque dipendente dell'area commerciale di CWBI, al quale sono stati assegnati i permessi necessari a svolgere le operazioni di seguito riportate.

L'applicativo messo a disposizione dell'utente consente di:

- **creare nuovi progetti**, corrispondenti a nuove opportunità di vendita per l'azienda;
- **visionare la lista completa dei progetti**. Per ogni progetto è inoltre possibile visualizzarne i dettagli e modificare o eliminare il progetto stesso;
- **visionare la lista completa dei progetti ancora aperti**, ovvero quelli ancora in sviluppo o in evoluzione;
- **visionare la lista completa dei clienti** e le loro informazioni specifiche;
- **visionare la lista completa delle offerte** formulate ai diversi clienti di CWBI per i progetti di loro interesse. Per ogni offerta è inoltre possibile visualizzarne i dettagli e modificare o eliminare l'offerta stessa.

3.1.1 UC0: Menu

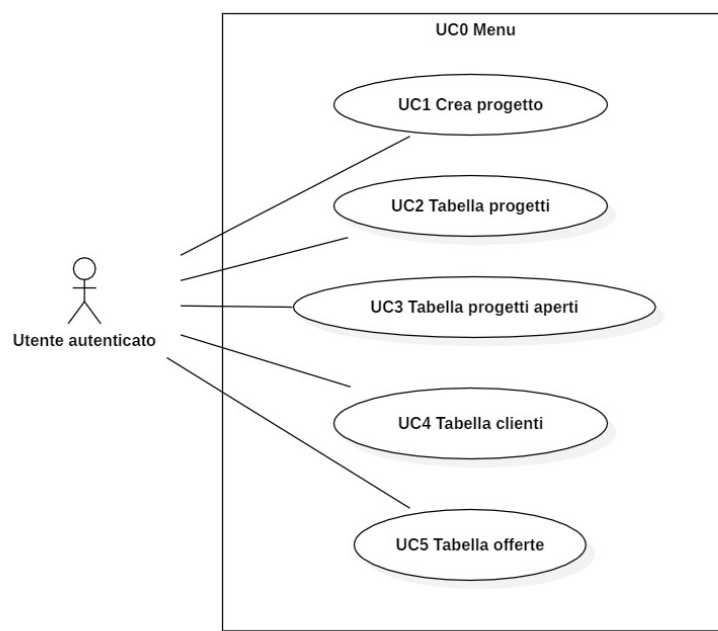


Figura 3.1: UC0: Menu

- **UC0: Menu** (figura 3.1)
- **Attore primario:** Utente autenticato.
- **Precondizione:** L'attore ha accesso al modulo *Campaign* della webapp CWGEST.

- **Postcondizione:** Il sistema è pronto all'uso.
- **Scenario principale:**
 1. L'attore crea un nuovo progetto [UC1].
 2. L'attore accede alla tabella dei progetti [UC2].
 3. L'attore accede alla tabella dei progetti aperti [UC3].
 4. L'attore accede alla tabella dei clienti [UC4].
 5. L'attore accede alla tabella delle offerte [UC5].

3.1.2 UC1: Crea-Modifica progetto

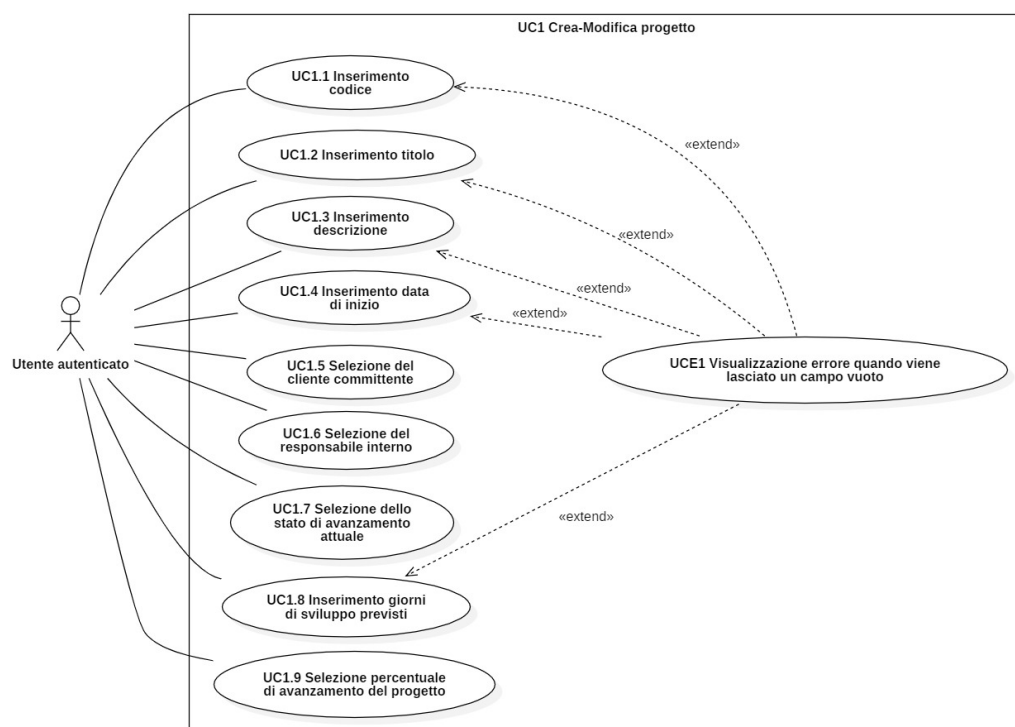


Figura 3.2: UC1: Crea-Modifica progetto

- **UC1: Crea-Modifica progetto** (figura 3.2)
- **Attore primario:** Utente autenticato.
- **Precondizione:** L'attore ha selezionato:
 - l'opzione *nuovo progetto* dal menu;
 - l'opzione *modifica* dal dettaglio di un progetto
- **Postcondizione:** L'attore ha creato/modificato un progetto.

- **Scenario principale:**

1. L'attore inserisce/modifica il codice del progetto [UC1.1].
2. L'attore inserisce/modifica il titolo del progetto [UC1.2].
3. L'attore inserisce/modifica la descrizione del progetto [UC1.3].
4. L'attore inserisce/modifica la data di inizio del progetto [UC1.4].
5. L'attore seleziona/modifica il cliente committente [UC1.5].
6. L'attore seleziona/modifica il responsabile interno assegnato al progetto [UC1.6].
7. L'attore seleziona/modifica lo stato di avanzamento attuale delle attività [UC1.7].
8. L'attore inserisce/modifica i giorni di sviluppo previsti [UC1.8].
9. L'attore seleziona/modifica la percentuale di avanzamento delle attività di progetto [UC1.9].

- **Estensioni:**

- L'attore visualizza un messaggio di errore quando viene lasciato un campo vuoto [UCE1].

3.1.3 UC2: Tabella progetti

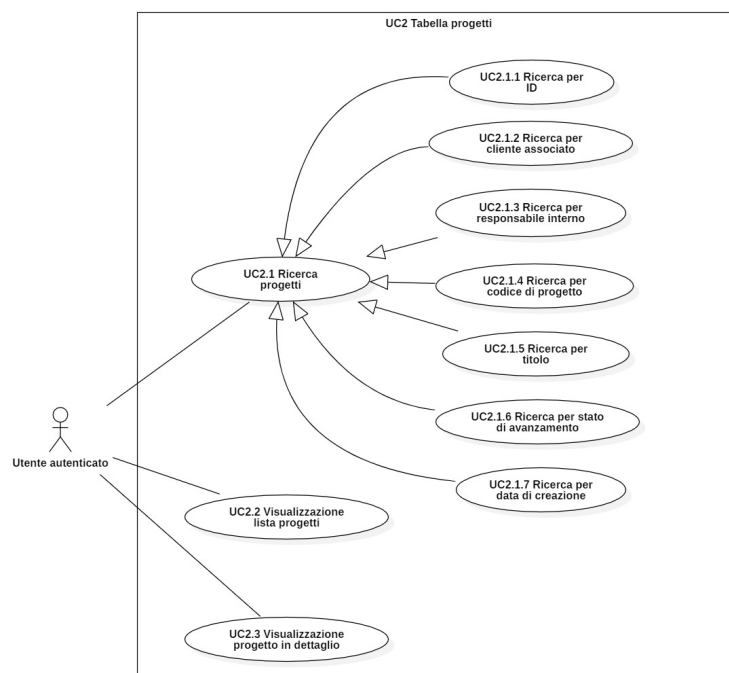


Figura 3.3: UC2: Tabella progetti

- **UC2: Tabella progetti** (figura 3.3)
- **Attore primario:** Utente autenticato.
- **Precondizione:** L'attore ha selezionato l'opzione *tabella progetti* dal menu.
- **Postcondizione:** L'attore ha accesso alla tabella dei progetti.
- **Scenario principale:**
 1. L'attore cerca uno o più progetti tramite la funzionalità di ricerca [UC2.1].
 2. L'attore visualizza la lista completa dei progetti presenti nel sistema [UC2.2].
 3. L'attore visualizza un progetto nel dettaglio [UC2.3].
- **Generalizzazioni:**
 1. L'attore ricerca un progetto per ID [UC2.1.1].
 2. L'attore ricerca un progetto per uno specifico cliente associato [UC2.1.2].
 3. L'attore ricerca un progetto per responsabile interno [UC2.1.3].
 4. L'attore ricerca un progetto per codice [UC2.1.4].
 5. L'attore ricerca un progetto per titolo [UC2.1.5].
 6. L'attore ricerca un progetto per stato di avanzamento [UC2.1.6].
 7. L'attore ricerca un progetto per data di creazione [UC2.1.7].

3.1.4 UC2.2: Visualizzazione lista progetti

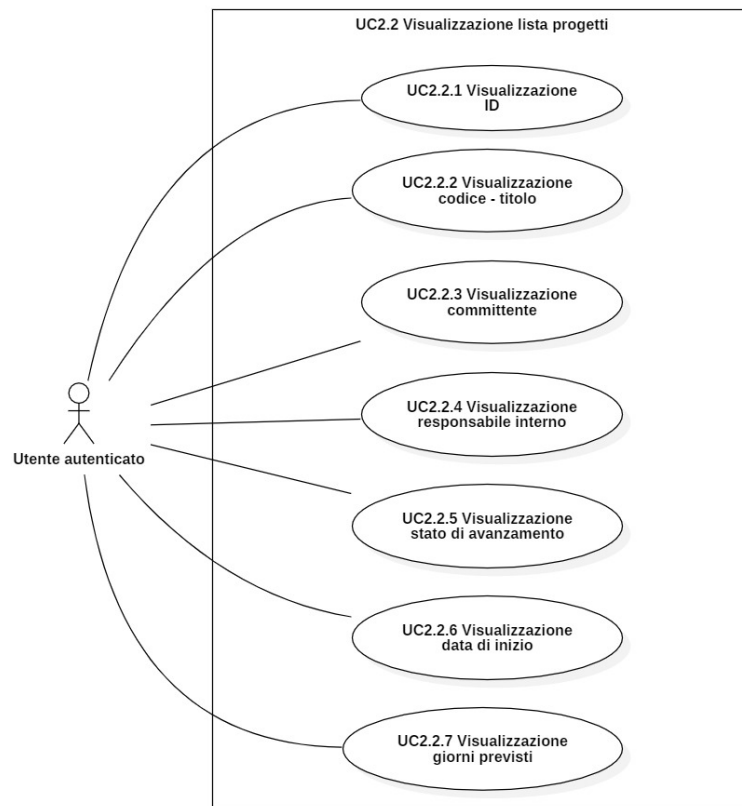


Figura 3.4: UC2.2: Visualizzazione lista progetti

- **UC2.2: Visualizzazione lista progetti** (figura 3.4)
- **Attore primario:** Utente autenticato.
- **Precondizione:** L'attore ha accesso alla tabella dei progetti.
- **Postcondizione:** L'attore ha visualizzato la lista dei progetti.
- **Scenario principale:**
 1. L'attore visualizza l'ID del progetto [UC2.2.1].
 2. L'attore visualizza la stringa "codice - titolo" del progetto [UC2.2.2].
 3. L'attore visualizza il committente del progetto [UC2.2.3].
 4. L'attore visualizza il responsabile interno del progetto [UC2.2.4].
 5. L'attore visualizza lo stato di avanzamento del progetto [UC2.2.5].
 6. L'attore visualizza la data di inizio del progetto [UC2.2.6].
 7. L'attore visualizza i giorni di sviluppo previsti del progetto [UC2.2.7].

3.1.5 UC2.3: Visualizzazione progetto in dettaglio

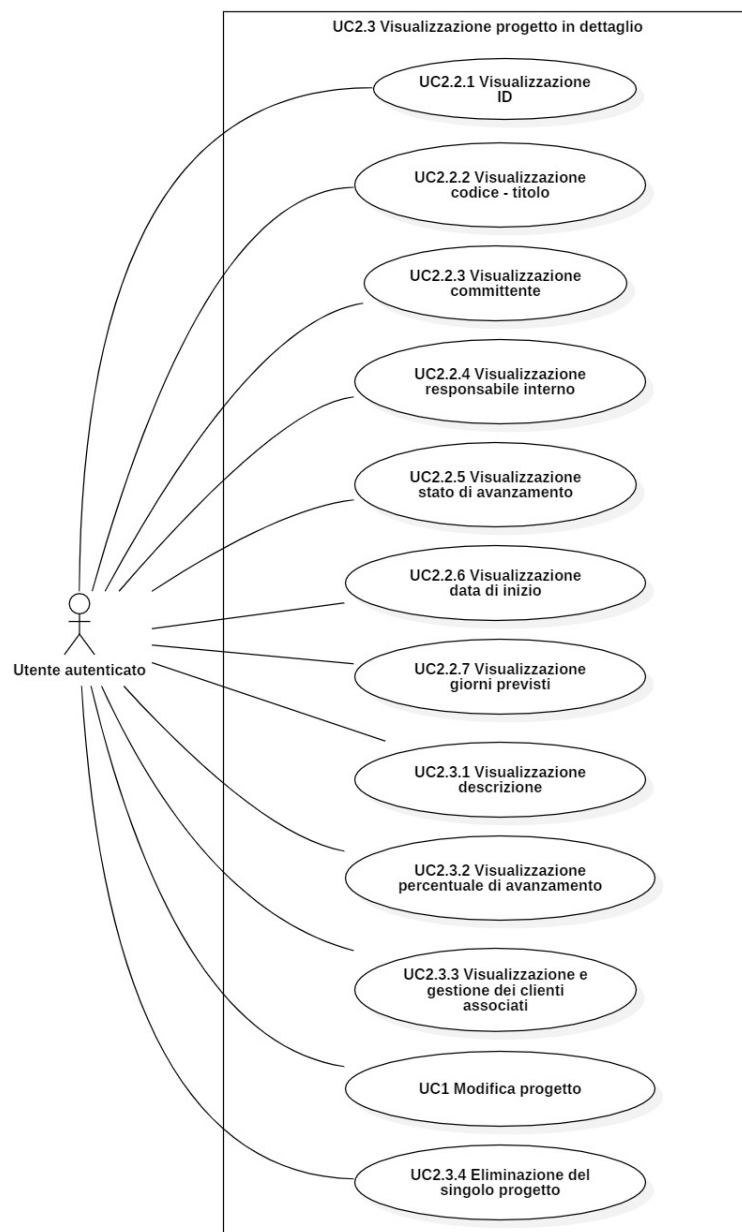


Figura 3.5: UC2.3: Visualizzazione progetto in dettaglio

- **UC2.3: Visualizzazione progetto in dettaglio** (figura 3.5)
- **Attore primario:** Utente autenticato.
- **Precondizione:** L'attore ha accesso alla tabella dei progetti.

- **Postcondizione:** L'attore ha visualizzato il dettaglio di un progetto.
- **Scenario principale:**
 1. L'attore visualizza l'ID del progetto [UC2.2.1].
 2. L'attore visualizza la stringa "codice - titolo" del progetto [UC2.2.2].
 3. L'attore visualizza il committente del progetto [UC2.2.3].
 4. L'attore visualizza il responsabile interno del progetto [UC2.2.4].
 5. L'attore visualizza lo stato di avanzamento del progetto [UC2.2.5].
 6. L'attore visualizza la data di inizio del progetto [UC2.2.6].
 7. L'attore visualizza i giorni di sviluppo previsti del progetto [UC2.2.7].
 8. L'attore visualizza la descrizione del progetto [UC2.3.1].
 9. L'attore visualizza la percentuale di avanzamento del progetto [UC2.3.2].
 10. L'attore visualizza e gestisce i clienti associati al progetto [UC2.3.3].
 11. L'attore può modificare il progetto [UC1].
 12. L'attore può eliminare il progetto [UC2.3.4].

3.1.6 UC3: Tabella progetti aperti

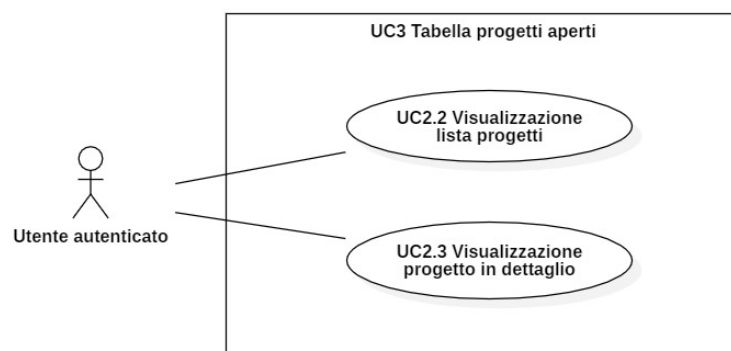


Figura 3.6: UC3: Tabella progetti aperti

- **UC3: Tabella progetti aperti** (figura 3.6)
- **Attore primario:** Utente autenticato.
- **Precondizione:** L'attore ha selezionato l'opzione *tabella progetti aperti* dal menu.
- **Postcondizione:** L'attore ha accesso alla tabella dei progetti aperti.
- **Scenario principale:**
 1. L'attore visualizza la lista dei progetti aperti presenti nel sistema [UC2.2].
 2. L'attore visualizza un progetto aperto nel dettaglio [UC2.3].

3.1.7 UC4: Tabella clienti

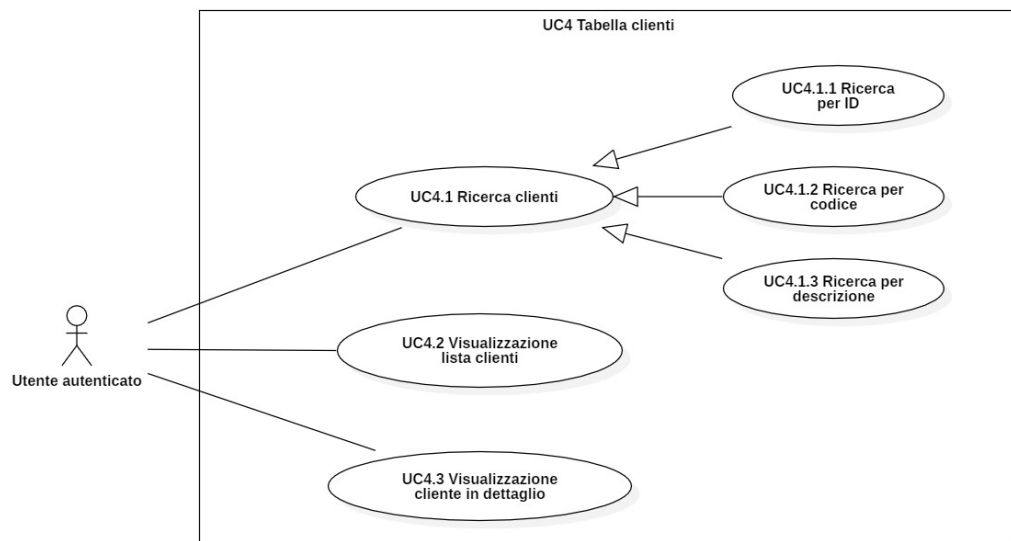


Figura 3.7: UC4: Tabella clienti

- **UC4: Tabella clienti** (figura 3.7)
- **Attore primario:** Utente autenticato.
- **Precondizione:** L'attore ha selezionato l'opzione *tabella clienti* dal menu.
- **Postcondizione:** L'attore ha accesso alla tabella dei clienti.
- **Scenario principale:**
 1. L'attore cerca uno o più clienti tramite la funzionalità di ricerca [UC4.1].
 2. L'attore visualizza la lista completa dei clienti presenti nel sistema [UC4.2].
 3. L'attore visualizza un cliente nel dettaglio [UC4.3].
- **Generalizzazioni:**
 1. L'attore ricerca un cliente per ID [UC4.1.1].
 2. L'attore ricerca un cliente per codice (nominativo aziendale) [UC4.1.2].
 3. L'attore ricerca un cliente per descrizione [UC4.1.3].

3.1.8 UC4.2: Visualizzazione lista clienti

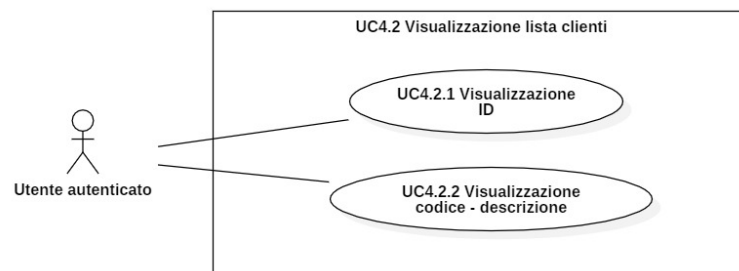


Figura 3.8: UC4.2: Visualizzazione lista clienti

- **UC4.2: Visualizzazione lista clienti** ([figura 3.8](#))
- **Attore primario:** Utente autenticato.
- **Precondizione:** L'attore ha accesso alla tabella dei clienti.
- **Postcondizione:** L'attore ha visualizzato la lista dei clienti.
- **Scenario principale:**
 1. L'attore visualizza l'ID del cliente [UC4.2.1].
 2. L'attore visualizza la stringa "codice - descrizione" del cliente [UC4.2.2].

3.1.9 UC4.3: Visualizzazione cliente in dettaglio

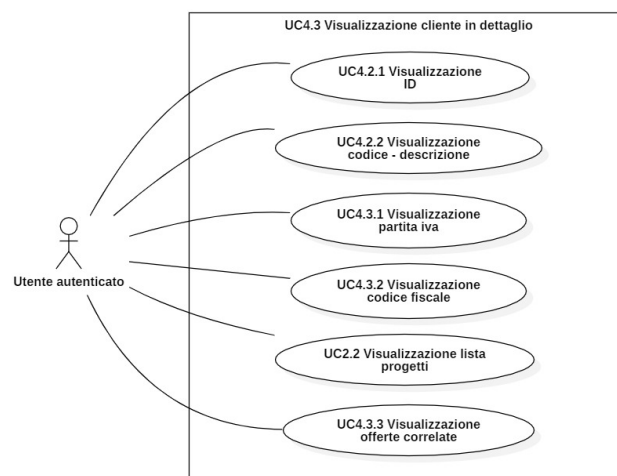


Figura 3.9: UC4.3: Visualizzazione cliente in dettaglio

- **UC4.3: Visualizzazione cliente in dettaglio** (figura 3.9)
- **Attore primario:** Utente autenticato.
- **Precondizione:** L'attore ha accesso alla tabella dei clienti.
- **Postcondizione:** L'attore ha visualizzato il dettaglio di un cliente.
- **Scenario principale:**
 1. L'attore visualizza l'ID del cliente [UC4.2.1].
 2. L'attore visualizza la stringa "codice - descrizione" del cliente [UC4.2.2].
 3. L'attore visualizza la partita iva del cliente [UC4.3.1].
 4. L'attore visualizza il codice fiscale del cliente [UC4.3.2].
 5. L'attore visualizza la lista dei progetti associati al cliente [UC2.2].
 6. L'attore visualizza le offerte correlate al cliente [UC4.3.3].

3.1.10 UC4.3.3: Visualizzazione offerte correlate

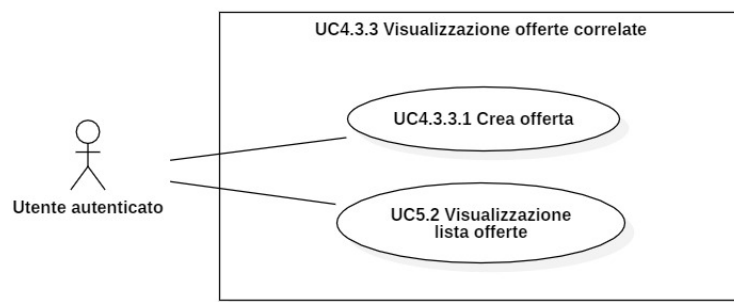


Figura 3.10: UC4.3.3: Visualizzazione offerte correlate

- **UC4.3.3: Visualizzazione offerte correlate** (figura 3.10)
- **Attore primario:** Utente autenticato.
- **Precondizione:** L'attore ha visualizzato il dettaglio di un cliente.
- **Postcondizione:** L'attore ha visualizzato le offerte correlate al cliente.
- **Scenario principale:**
 1. L'attore può creare una nuova offerta correlata al cliente [UC4.3.3.1].
 2. L'attore visualizza la lista delle offerte del cliente [UC5.2].

3.1.11 UC4.3.3.1: Crea-Modifica offerta

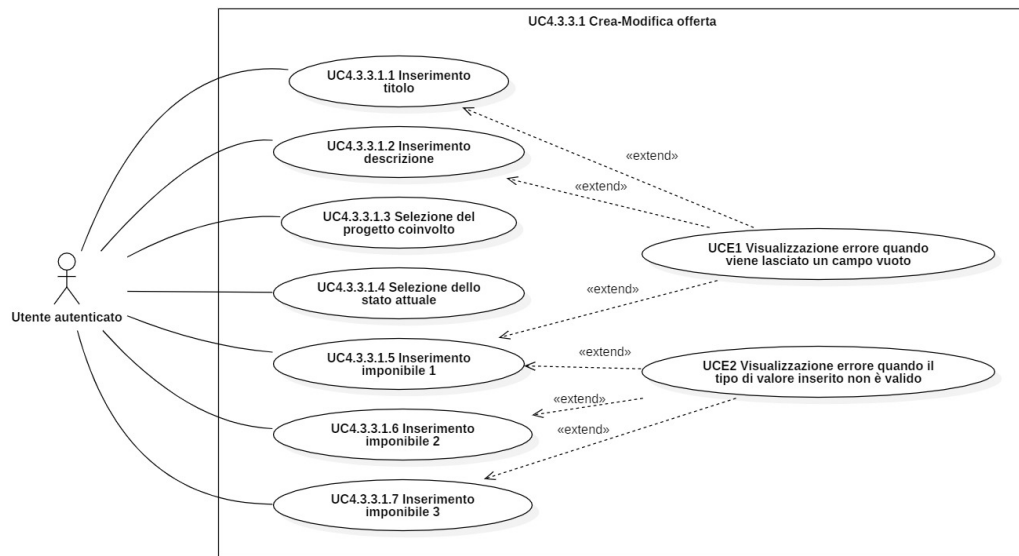


Figura 3.11: UC4.3.3.1: Crea-Modifica offerta

- **UC4.3.3.1: Crea-Modifica offerta** (figura 3.11)
- **Attore primario:** Utente autenticato.
- **Precondizione:** L'attore ha selezionato:
 - l'opzione *nuova offerta* dal dettaglio di un cliente;
 - l'opzione *modifica* dal dettaglio di un'offerta.
- **Postcondizione:** L'attore ha creato/modificato un'offerta.
- **Scenario principale:**
 1. L'attore inserisce/modifica il titolo di un'offerta [UC4.3.3.1.1].
 2. L'attore inserisce/modifica la descrizione di un'offerta [UC4.3.3.1.2].
 3. L'attore seleziona/modifica il progetto di riferimento di un'offerta [UC4.3.3.1.3].
 4. L'attore seleziona/modifica lo stato attuale di un'offerta [UC4.3.3.1.4].
 5. L'attore inserisce/modifica l'imponibile 1 di un'offerta [UC4.3.3.1.5].
 6. L'attore inserisce/modifica l'imponibile 2 di un'offerta [UC4.3.3.1.6].
 7. L'attore inserisce/modifica l'imponibile 3 di un'offerta [UC4.3.3.1.7].
- **Estensioni:**
 - L'attore visualizza un messaggio di errore quando viene lasciato un campo vuoto [UCE1].
 - L'attore visualizza un messaggio di errore quando viene inserito un tipo di valore non compatibile con quello del campo di interesse [UCE2].

3.1.12 UC5: Tabella offerte

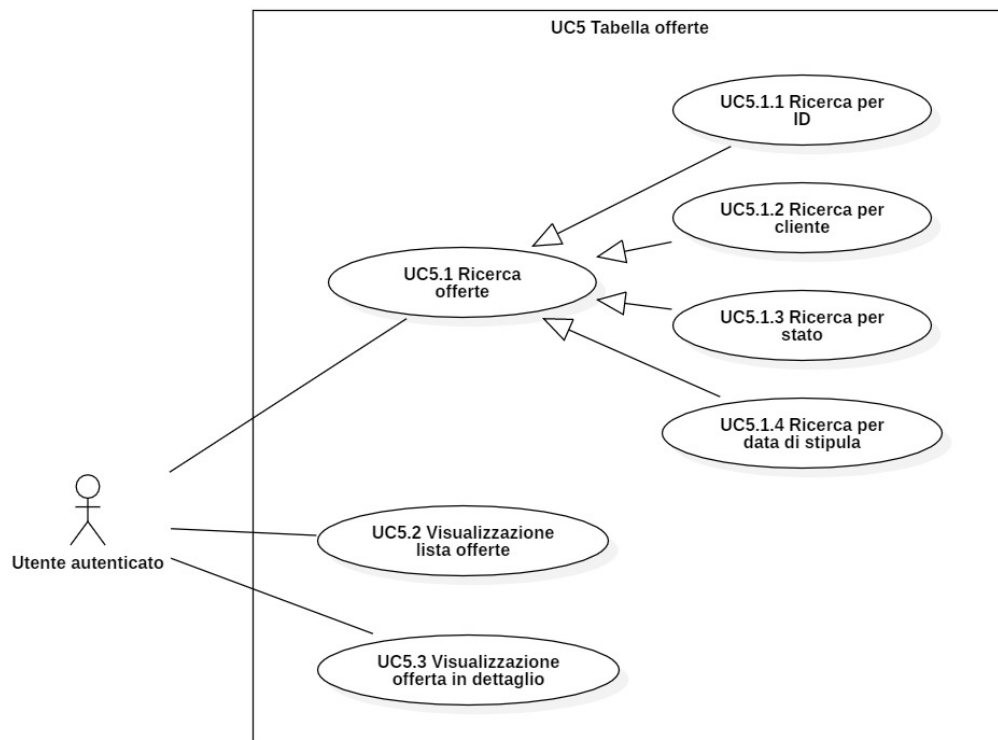


Figura 3.12: UC5: Tabella offerte

- **UC5: Tabella offerte** (figura 3.12)
- **Attore primario:** Utente autenticato.
- **Precondizione:** L'attore ha selezionato l'opzione *tabella offerte* dal menu.
- **Postcondizione:** L'attore ha accesso alla tabella delle offerte.
- **Scenario principale:**
 1. L'attore cerca una o più offerte tramite la funzionalità di ricerca [UC5.1].
 2. L'attore visualizza la lista completa delle offerte presenti nel sistema [UC5.2].
 3. L'attore visualizza un'offerta nel dettaglio [UC5.3].
- **Generalizzazioni:**
 1. L'attore ricerca un'offerta per ID [UC5.1.1].
 2. L'attore ricerca un'offerta per cliente associato [UC5.1.2].
 3. L'attore ricerca un'offerta per stato attuale [UC5.1.3].
 4. L'attore ricerca un'offerta per data di stipula [UC5.1.4].

3.1.13 UC5.2: Visualizzazione lista offerte

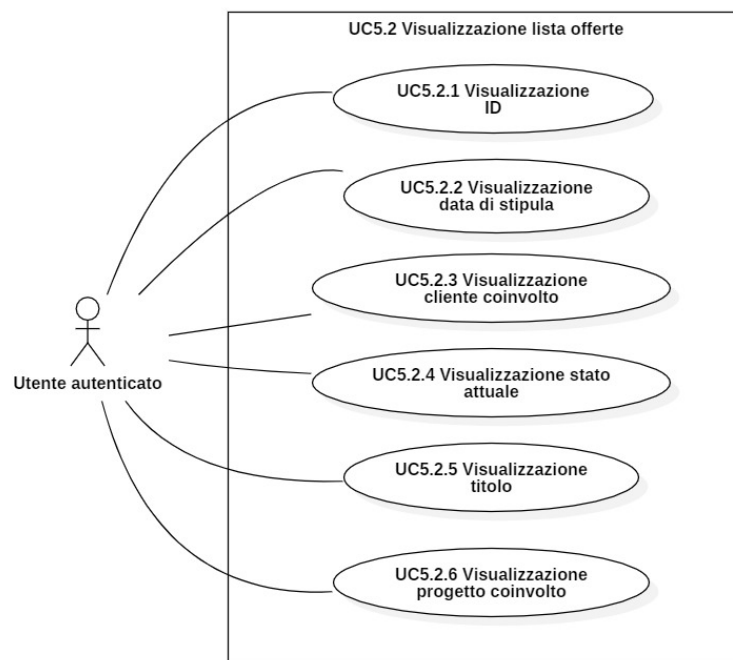
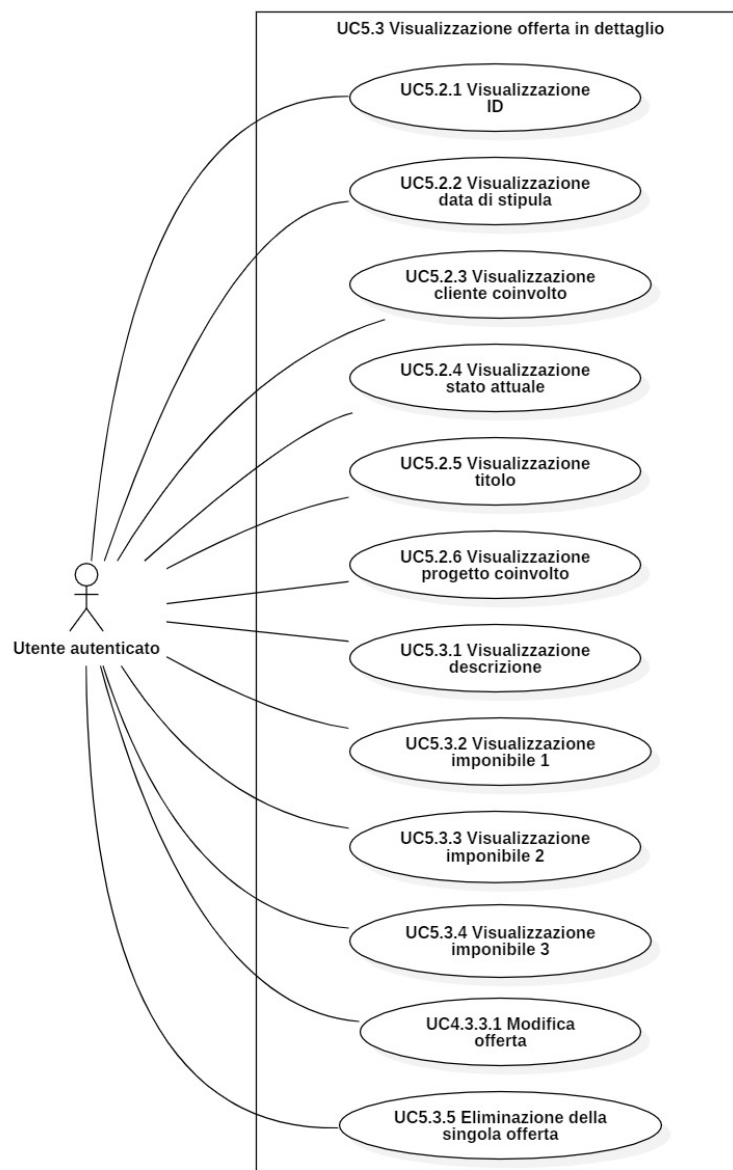


Figura 3.13: UC5.2: Visualizzazione lista offerte

- **UC5.2: Visualizzazione lista offerte** (figura 3.13)
- **Attore primario:** Utente autenticato.
- **Precondizione:** L'attore ha accesso alla tabella delle offerte.
- **Postcondizione:** L'attore ha visualizzato la lista delle offerte.
- **Scenario principale:**
 1. L'attore visualizza l'ID dell'offerta [UC5.2.1].
 2. L'attore visualizza la data di stipula dell'offerta [UC5.2.2].
 3. L'attore visualizza il cliente coinvolto nell'offerta [UC5.2.3].
 4. L'attore visualizza lo stato attuale dell'offerta [UC5.2.4].
 5. L'attore visualizza il titolo dell'offerta [UC5.2.5].
 6. L'attore visualizza il progetto coinvolto nell'offerta [UC5.2.6].

3.1.14 UC5.3: Visualizzazione offerta in dettaglio**Figura 3.14:** UC5.3: Visualizzazione offerta in dettaglio

- **UC5.3: Visualizzazione offerta in dettaglio** ([figura 3.14](#))
- **Attore primario:** Utente autenticato.
- **Precondizione:** L'attore ha accesso alla tabella delle offerte.
- **Postcondizione:** L'attore ha visualizzato il dettaglio di un'offerta.

- **Scenario principale:**

1. L'attore visualizza l'ID dell'offerta [UC5.2.1].
2. L'attore visualizza la data di stipula dell'offerta [UC5.2.2].
3. L'attore visualizza il cliente coinvolto nell'offerta [UC5.2.3].
4. L'attore visualizza lo stato attuale dell'offerta [UC5.2.4].
5. L'attore visualizza il titolo dell'offerta [UC5.2.5].
6. L'attore visualizza il progetto coinvolto nell'offerta [UC5.2.6].
7. L'attore visualizza la descrizione dell'offerta [UC5.3.1].
8. L'attore visualizza l'imponibile 1 dell'offerta [UC5.3.2].
9. L'attore visualizza l'imponibile 2 dell'offerta [UC5.3.3].
10. L'attore visualizza l'imponibile 3 dell'offerta [UC5.3.4].
11. L'attore può modificare l'offerta [UC4.3.3.1].
12. L'attore può eliminare l'offerta [UC5.3.5].

3.2 Definizione e tracciamento dei requisiti

Per concordare gli obiettivi da raggiungere e i problemi da risolvere sono stati trovati una serie di requisiti. Per identificare ciascun [requisito](#)^[8] si è ricorso ad un codice identificativo composto da **R** [**F/V**] **O** **N** dove:

- **R** = requisito;
- **F** = funzionale;
- **V** = di vincolo;
- **O** = obbligatorio;
- **N** = numero del requisito.

Di seguito si riporta una tabella contenente tutti i requisiti individuati con relativa descrizione e riferimento al relativo use case.

3.2.1 Requisiti funzionali

Identificativo	Descrizione	Use case
RFO1	L'utente deve poter utilizzare il menu.	UC0
RFO2	L'utente deve poter creare un nuovo progetto.	UC1
RFO3	L'utente deve poter accedere alla tabella dei progetti.	UC2
RFO4	L'utente deve poter accedere alla tabella dei progetti aperti.	UC3
RFO5	L'utente deve poter accedere alla tabella dei clienti.	UC4
RFO6	L'utente deve poter accedere alla tabella delle offerte.	UC5
RFO7	L'utente inserisce il codice del progetto.	UC1.1
RFO8	L'utente inserisce il titolo del progetto.	UC1.2
RFO9	L'utente inserisce la descrizione del progetto.	UC1.3
RFO10	L'utente inserisce la data di inizio del progetto.	UC1.4
RFO11	L'utente seleziona il cliente committente del progetto.	UC1.5
RFO12	L'utente seleziona il responsabile interno del progetto.	UC1.6
RFO13	L'utente seleziona lo stato di avanzamento del progetto.	UC1.7
RFO14	L'utente inserisce i giorni di sviluppo previsti.	UC1.8
RFO15	L'utente seleziona la percentuale di avanzamento del progetto.	UC1.9
RFO16	L'utente visualizza un errore quando lascia vuoto il campo codice del progetto.	UCE1
RFO17	L'utente visualizza un errore quando lascia vuoto il campo titolo del progetto.	UCE1
RFO18	L'utente visualizza un errore quando lascia vuoto il campo descrizione del progetto.	UCE1
RFO19	L'utente visualizza un errore quando lascia vuoto il campo data di inizio del progetto.	UCE1
RFO20	L'utente visualizza un errore quando lascia vuoto il campo giorni di sviluppo del progetto.	UCE1
RFO21	L'utente può ricercare uno o più progetti.	UC2.1
RFO22	L'utente ricerca un progetto per ID.	UC2.1.1
RFO23	L'utente ricerca un progetto per cliente associato.	UC2.1.2
RFO24	L'utente ricerca un progetto per responsabile interno.	UC2.1.3
RFO25	L'utente ricerca un progetto per codice di progetto.	UC2.1.4
RFO26	L'utente ricerca un progetto per titolo.	UC2.1.5

Identificativo	Descrizione	Use case
RFO27	L'utente ricerca un progetto per stato di avanzamento.	UC2.1.6
RFO28	L'utente ricerca un progetto per data di creazione.	UC2.1.7
RFO29	L'utente visualizza la lista dei progetti.	UC2.2
RFO30	L'utente visualizza il dettaglio di un progetto.	UC2.3
RFO31	L'utente visualizza l'ID di un progetto.	UC2.2.1
RFO32	L'utente visualizza la stringa "codice - titolo" di un progetto.	UC2.2.2
RFO33	L'utente visualizza il committente di un progetto.	UC2.2.3
RFO34	L'utente visualizza il responsabile interno di un progetto.	UC2.2.4
RFO35	L'utente visualizza lo stato di avanzamento di un progetto.	UC2.2.5
RFO36	L'utente visualizza la data di inizio di un progetto.	UC2.2.6
RFO37	L'utente visualizza i giorni di sviluppo previsti di un progetto.	UC2.2.7
RFO38	L'utente visualizza la descrizione del progetto.	UC2.3.1
RFO39	L'utente visualizza la percentuale di avanzamento del progetto.	UC2.3.2
RFO40	L'utente visualizza e gestisce i clienti del progetto.	UC2.3.3
RFO41	L'utente modifica il progetto.	UC1
RFO42	L'utente elimina il progetto.	UC2.3.4
RFO43	L'utente ricerca uno o più clienti.	UC4.1
RFO44	L'utente ricerca un cliente per ID.	UC4.1.1
RFO45	L'utente ricerca un cliente per codice.	UC4.1.2
RFO46	L'utente ricerca un cliente per descrizione.	UC4.1.3
RFO47	L'utente visualizza la lista dei clienti.	UC4.2
RFO48	L'utente visualizza il dettaglio di un cliente.	UC4.3
RFO49	L'utente visualizza l'ID di un cliente.	UC4.2.1
RFO50	L'utente visualizza la stringa "codice - descrizione" di un cliente.	UC4.2.2
RFO51	L'utente visualizza partita iva del cliente.	UC4.3.1
RFO52	L'utente visualizza il codice fiscale del cliente.	UC4.3.2
RFO53	L'utente visualizza la lista dei progetti associati al cliente.	UC2.2

Identificativo	Descrizione	Use case
RFO54	L'utente visualizza la lista delle offerte associate al cliente.	UC4.3.3
RFO55	L'utente crea una nuova offerta.	UC4.3.3.1
RFO56	L'utente visualizza la lista delle offerte.	UC5.2
RFO57	L'utente inserisce il titolo dell'offerta.	UC4.3.3.1.1
RFO58	L'utente inserisce la descrizione dell'offerta.	UC4.3.3.1.2
RFO59	L'utente seleziona il progetto coinvolto nell'offerta.	UC4.3.3.1.3
RFO60	L'utente seleziona lo stato attuale dell'offerta.	UC4.3.3.1.4
RFO61	L'utente inserisce l'imponibile 1 dell'offerta.	UC4.3.3.1.5
RFO62	L'utente inserisce l'imponibile 2 dell'offerta.	UC4.3.3.1.6
RFO63	L'utente inserisce l'imponibile 3 dell'offerta.	UC4.3.3.1.7
RFO64	L'utente visualizza un errore quando lascia vuoto il campo titolo dell'offerta.	UCE1
RFO65	L'utente visualizza un errore quando lascia vuoto il campo descrizione dell'offerta.	UCE1
RFO66	L'utente visualizza un errore quando lascia vuoto il campo imponibile 1 dell'offerta.	UCE1
RFO67	L'utente visualizza un errore quando il tipo di valore inserito del campo imponibile 1 non è numerico.	UCE2
RFO68	L'utente visualizza un errore quando il tipo di valore inserito del campo imponibile 2 non è numerico.	UCE2
RFO69	L'utente visualizza un errore quando il tipo di valore inserito del campo imponibile 3 non è numerico.	UCE2
RFO70	L'utente ricerca una o più offerte.	UC5.1
RFO71	L'utente ricerca un'offerta per ID.	UC5.1.1
RFO72	L'utente ricerca un'offerta per cliente associato.	UC5.1.2
RFO73	L'utente ricerca un'offerta per stato attuale.	UC5.1.3
RFO74	L'utente ricerca un'offerta per data di stipula.	UC5.1.4
RFO75	L'utente visualizza il dettaglio di un'offerta.	UC5.3
RFO76	L'utente visualizza l'ID di un'offerta.	UC5.2.1
RFO77	L'utente visualizza la data di stipula di un'offerta.	UC5.2.2
RFO78	L'utente visualizza il cliente coinvolto di un'offerta.	UC5.2.3
RFO79	L'utente visualizza lo stato di un'offerta.	UC5.2.4

Identificativo	Descrizione	Use case
RFO80	L'utente visualizza il titolo di un'offerta.	UC5.2.5
RFO81	L'utente visualizza il progetto di un'offerta.	UC5.2.6
RFO82	L'utente visualizza la descrizione dell'offerta.	UC5.3.1
RFO83	L'utente visualizza l'imponibile 1 dell'offerta.	UC5.3.2
RFO84	L'utente visualizza l'imponibile 2 dell'offerta.	UC5.3.3
RFO85	L'utente visualizza l'imponibile 3 dell'offerta.	UC5.3.4
RFO86	L'utente modifica l'offerta.	UC4.3.3.1
RFO87	L'utente elimina l'offerta.	UC5.3.5

Tabella 3.1: Requisiti funzionali

3.2.2 Requisiti di vincolo

Identificativo	Descrizione
RVO1	L'applicativo lato back-end è realizzato in Java.
RVO2	L'applicativo lato back-end è realizzato mediante il framework Spring.
RVO3	L'applicativo lato back-end è realizzato mediante il framework Hibernate.
RVO4	L'applicativo lato fron-end è realizzato tramite HTML5, CSS3 e JavaScript.
RVO5	L'applicativo lato front-end è realizzato mediante il framework Bootstrap.
RVO6	L'applicativo deve essere funzionante in tutte le sue componenti.

Tabella 3.2: Requisiti di vincolo

Capitolo 4

Strumenti utilizzati

Il seguente capitolo ha la funzione di introdurre tutti gli strumenti utilizzati come supporto alle attività di sviluppo dell'applicativo, tra cui: ambiente di lavoro, framework, linguaggi, librerie a supporto della codifica e IDE. Tutte le tecnologie elencate rappresentano vincolo di sviluppo del progetto.

4.1 Ambiente di lavoro

CWBI ha a disposizione circa 30 macchine fisiche con sistema operativo Debian, ciascuna delle quali viene utilizzata come supporto per delle macchine virtuali in ambiente Windows 7. Il lavoro quotidiano viene svolto in ambiente virtuale e non fisico; per la connessione alla propria macchina virtuale viene utilizzato il software *VMware Workstation*. Il vantaggio nell'utilizzare macchine virtuali sta nel fatto che tutti i dipendenti presentano la medesima configurazione dell'ambiente di sviluppo e, essendo il lavoro in azienda molto collaborativo, al presentarsi di un problema o di un dubbio sul "cosa fare" i colleghi possono connettersi alla macchina virtuale dove si è presentato il problema per aiutare e velocizzare lo scambio di idee sul come fare.

4.2 Framework

Nella seguente sezione sono riportati i [framework](#)^[g] utilizzati nelle attività di sviluppo del progetto commissionato.

4.2.1 Spring (core)



Figura 4.1: Logo di Spring

[Spring](#)^[b] (logo in [figura 4.1](#)) è un framework open source nato per lo sviluppo di applicazioni Java EE. Una delle sue maggiori peculiarità risiede nell'essere **modulare**, il che consente di utilizzare anche solo una parte delle funzionalità che il framework mette a disposizione. Altro punto di forza è la facile integrazione con altri framework esistenti, come: Hibernate, Apache Struts, ecc.

Per lo sviluppo back-end viene utilizzato solamente il *core container* (struttura in [figura 4.2](#)) offerto dal framework Spring e così composto:

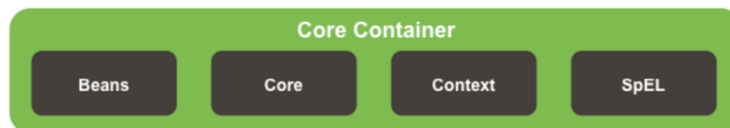


Figura 4.2: Spring - core container

1. il modulo bean fornisce le classi factory per l'istanziamento dei bean;
2. il modulo core è la componente fondamentale del framework ed implementa la Dependency Injection;
3. il modulo context fornisce l'accesso a tutti i bean definiti in funzione dei relativi contesti;
4. il modulo SpEL fornisce un expression language per la manipolazione dei bean a runtime.

4.2.2 Hibernate



Figura 4.3: Logo di Hibernate

Hibernate (logo in [figura 4.3](#)) è un framework open source che permette di rendere persistenti i dati dall'ambiente Java al database mappando gli oggetti in apposite tabelle di un database relazionale.

[Hibernate](#)^[b] gestisce inoltre l'accesso e la manipolazione dei dati stessi generando automaticamente le query in linguaggio SQL; questo agevola il lavoro dello sviluppatore, che non si deve occupare della scrittura manuale delle query di accesso ai dati del database ma solamente di definire gli oggetti che andranno memorizzati e gestiti nel database.

Il vantaggio principale nell'utilizzare Hibernate risiede nel fatto che è il framework stesso a costruire le query e questo rende portabili le applicazioni sui diversi servizi di database esistenti (MySQL, PostgreSQL, MariaDB, ecc.).

4.2.3 Bootstrap



Figura 4.4: Logo di Bootstrap

[Bootstrap](#)^[b] (logo in [figura 4.4](#)) è un framework open source per lo sviluppo di interfacce web basato su CSS. Bootstrap mette a disposizione una serie di file CSS contenenti stili e regole applicabili ai vari componenti HTML.

Gli stili che bootstrap propone sono ottimizzati per la visualizzazione smartphone, tablet e desktop. Inoltre, dato che la maggior parte degli accessi ad applicativi web avviene tramite cellulare, il framework mette al primo posto il layout per smartphone. Bootstrap è inoltre compatibile con i browser più moderni.

4.2.4 Apache Struts



Figura 4.5: Logo di Apache Struts

[Apache Struts](#)^[b] (logo in [figura 4.5](#)) è un framework open source Java che aiuta gli sviluppatori a creare applicazioni web flessibili, gestibili e sicure in Java. Il framework è caratterizzato da un architettura MVC che consente di raccogliere le richieste provenienti dalla vista e richiamare attraverso il controller le operazioni del modello necessarie a rendere persistenti i dati nel database. Tale architettura si integra con il framework Spring sopra citato.

4.3 Linguaggi utilizzati

Nella seguente sezione sono riportati i linguaggi utilizzati nella codifica dell'applicativo commissionato.

4.3.1 Java



Figura 4.6: Logo di Java

Per lo sviluppo dell'applicativo lato back-end si è utilizzato Java (logo in [figura 4.6](#)), un linguaggio di programmazione orientato agli oggetti. La caratteristica principale di questo linguaggio risiede nel rendere indipendente la scrittura di codice dall'ambiente di esecuzione fisico istanziando un ambiente di esecuzione virtuale del codice noto come JVM.

4.3.2 HTML5, CSS3 e JavaScript



Figura 4.7: Logo dello stack tecnologico del front-end

Per lo sviluppo dell'applicativo lato front-end si è utilizzato lo stack tecnologico (logo in [figura 4.7](#)) costituito da:

- HTML5: linguaggio di markup per la creazione di siti web;
- CSS3: linguaggio utilizzato per definire la formattazione dei documenti HTML attraverso apposite regole di stile;
- JavaScript: linguaggio di programmazione multi paradigma orientato agli eventi.

4.4 Librerie a supporto della codifica

Nella seguente sezione sono riportate le librerie utilizzate nelle attività di sviluppo del progetto commissionato.

4.4.1 Apache Commons



Figura 4.8: Logo di Apache Commons

[Apache Commons](#)^[b] (logo in [figura 4.8](#)) è una libreria Java molto utilizzata nello sviluppo di applicazioni basate su Java. Questa libreria offre molte classi caratterizzate da diversi metodi che possono essere richiamati dallo sviluppatore semplicemente importando la relativa classe di interesse. Alcune tra le classi offerte sono le seguenti:

- StringUtils: implementa una serie di operazioni sulle stringhe che completano/e-
stendono quelle offerte dalla classe standard `java.lang.String`;
- ArrayUtils: implementa dei metodi che consentono di processare e controllare gli
array;

- NumberUtils: implementa una serie di metodi a supporto dei tipi numerici offerti da Java.

4.4.2 JSTL

JavaServer Pages Standard Tag Library (JSTL) è una libreria utilizzata nello sviluppo di applicazioni web Java EE. È un'estensione di Java Server Page (JSP) ed incorpora un insieme di tag HTML definiti tramite file XML e programmati in linguaggio Java. Oltre ai tag standard proposti dalla libreria, JSTL permette allo sviluppatore di definire dei tag custom.

Il vantaggio della creazione di tag risiede nel definire dei comportamenti univoci dello stesso tag in contesti diversi eliminando la necessità di definire funzioni che si ripetono all'interno delle pagine.

4.5 IDE



Figura 4.9: Logo di Eclipse

Eclipse (logo in [figura 4.9](#)) è un ambiente di sviluppo che supporta diversi linguaggi, tra cui Java. Questo [IDE](#)^[8] è stato utilizzato per lo sviluppo dell'intero progetto. Tra le diverse funzionalità offerte da questo ambiente troviamo:

- importare un progetto esistente;
- creare un nuovo progetto;
- attività di refactor;
- attività di debugging.

Capitolo 5

Progettazione

Il seguente capitolo ha la funzione di illustrare l'approccio alla progettazione del sistema commissionato. Viene inoltre descritta l'architettura dell'intero sistema.

5.1 Approccio MDA

La filosofia dell'azienda per fronteggiare nuove attività e progetti si basa su un approccio di tipo **MDA (Model Driven Architecture)**. L'idea alla base di tale approccio risiede nel fatto che il cuore di ogni applicazione si traduce in un modello solido e ben strutturato.

L'approccio MDA è stato utilizzato anche per affrontare la fase di progettazione del sistema commissionato. In particolare la prima attività di progettazione è stata quella di definire i modelli su cui si basa l'intero sistema. Per modello l'azienda intende un qualunque oggetto che deve essere mappato e gestito in un'opportuna tabella di un database relazionale.

I modelli vengono rappresentati dalla seguente stringa:

NomeModello (TipoAttributo1 nomeAttributo1, ..., TipoAttributoN nomeAttributoN).

In particolare i modelli identificati in base alle richieste del committente sono i seguenti:

- **Progetto** (Long idProponente, Long idReferente, String codice, String titolo, String descrizione, String statoAvanzamento, Integer giorni, Integer percentualeAvanzamento, GregorianCalendar dataInizio)

DESCRIZIONE: questo modello rappresenta un progetto che viene commissionato all'azienda da parte di un cliente oppure un progetto che viene ideato dall'azienda per raggiungere nuovi clienti.

ATTRIBUTI:

- *idProponente* corrisponde all'identificativo del proponente, ovvero l'azienda da cui nasce l'idea del progetto (CWBI o altro cliente);
- *idReferente* corrisponde all'identificativo di un dipendente di CWBI che assume il ruolo di responsabile del progetto;
- *codice* corrisponde all'identificativo del progetto (es. 2023.70);
- *titolo* corrisponde al titolo del progetto;
- *descrizione* corrisponde alla descrizione generale del progetto, composta da obiettivi e funzionalità del sistema;
- *statoAvanzamento* corrisponde allo stato attuale del progetto. Lo stato può assumere i seguenti valori: aperto, chiuso, approvato, rifiutato, sospeso;
- *giorni* corrisponde ai giorni di sviluppo previsti per portare a termine il progetto;
- *percentualeAvanzamento* corrisponde alla percentuale di avanzamento delle attività di progetto;
- *dataInizio* corrisponde alla data di inizio delle attività di progetto.

- **CorrelazioneProgettoCliente** (Long idCliente, Long idProgetto)

DESCRIZIONE: questo modello rappresenta la correlazione che persiste tra un progetto e un certo cliente. Si fa notare che uno stesso progetto può raggiungere più di un cliente e che un cliente può essere coinvolto in più progetti.

ATTRIBUTI:

- *idCliente* corrisponde all'identificativo di un cliente;
- *idProgetto* corrisponde all'identificativo di un progetto esistente.

- **Offerta** (Long idCliente, Long idProgetto, String titolo, String descrizione, String stato, GregorianCalendar dataStipula, BigDecimal imponibile1, BigDecimal imponibile2, BigDecimal imponibile3)

DESCRIZIONE: questo modello rappresenta l'offerta che viene stipulata per un certo cliente in merito ad uno dei progetti a disposizione.

ATTRIBUTI:

- *idCliente* corrisponde all'identificativo di un cliente;
- *idProgetto* corrisponde all'identificativo di un progetto esistente;
- *titolo* corrisponde al titolo dell'offerta (es. Contratto n.7/2023);
- *descrizione* corrisponde alla descrizione dell'offerta emessa (note, informazioni utili, ecc.);
- *stato* corrisponde allo stato attuale dell'offerta. Lo stato può assumere i seguenti valori: in preparazione, in accettazione, accettata;
- *dataStipula* corrisponde alla data in cui l'offerta è stata emessa;
- *imponibile1* corrisponde al primo prezzo proposto per firmare l'offerta;
- *imponibile2* corrisponde al secondo prezzo proposto per firmare l'offerta;
- *imponibile3* corrisponde al terzo prezzo proposto per firmare l'offerta.

I modelli **Cliente** e **Utente**, necessari per realizzare il progetto commissionato, erano già stati definiti per un altro modulo dell'applicazione CW GEST. In particolare questi modelli rappresentano rispettivamente un cliente e un dipendente di CWBI che assume il ruolo di responsabile per un certo progetto. Dato che tali modelli presentano le medesime caratteristiche pensate per il progetto in questione non si è reso necessario alterarli per l'utilizzo.

Ai fini del progetto in questione non si rende necessario esporre la struttura del modello Utente; per tale modello infatti l'unico campo utile nell'applicativo risulta essere il nominativo, in modo da conoscere il dipendente responsabile di un certo progetto. Data invece la centralità del modello Cliente nel contesto dell'applicativo commissionato, si riporta comunque una sua caratterizzazione:

- **Cliente** (String codice, String descrizione, String partitaIva, String codiceFiscale)

DESCRIZIONE: questo modello rappresenta un cliente di CWBI.

ATTRIBUTI:

- *codice* corrisponde al nominativo del cliente (es. azienda x);
- *descrizione* corrisponde al nominativo esteso del cliente (es. azienda x SPA);
- *partitaIva* corrisponde alla partita IVA del cliente;
- *codiceFiscale* corrisponde al codice fiscale del cliente.

5.2 Architettura MVC

Come citato in precedenza, il framework Apache Struts mette a disposizione un'architettura MVC (Model View Controller) per costruire applicazioni web.

L'architettura di Struts non si differenzia particolarmente dalla filosofia della tradizionale MVC: il controller gestisce tutte le operazioni della sistema e ha un riferimento sia al modello sia alla vista.

L'architettura MVC messa a disposizione da Apache Struts può essere rappresentata dal seguente schema:

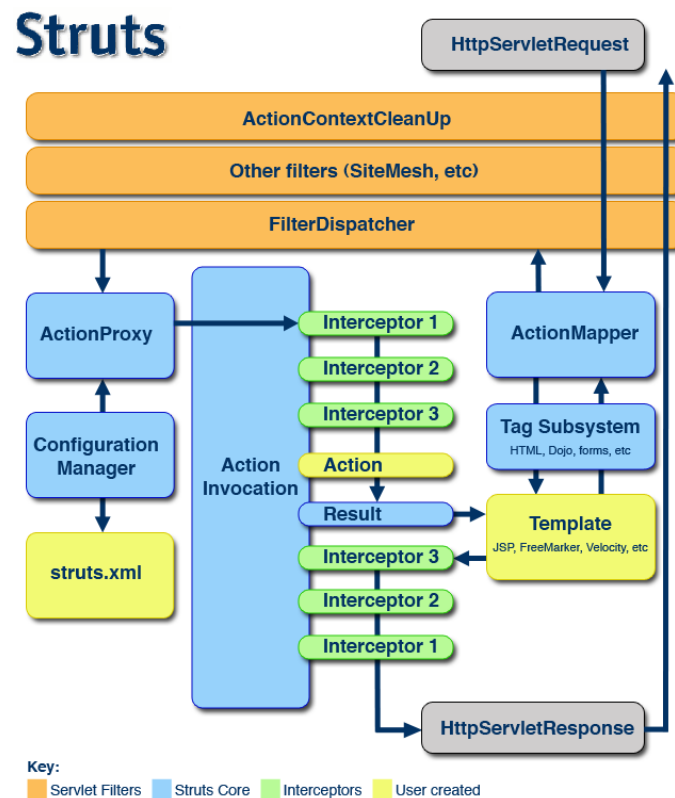


Figura 5.1: Architettura MVC di Apache Struts

I tre componenti principali della filosofia MVC corrispondono a:

- **FilterDispatcher** (Controller);
- **Action** (Model);
- **Template** (View).

Come si può notare dalla [figura 5.1](#), l'architettura offerta dal framework è caratterizzata da diversi componenti che risulta necessario menzionare per apprendere a fondo il

pieno funzionamento dell'architettura MVC di Apache Struts. Questi componenti sono i seguenti:

- **FilterDispatcher**: rappresenta il controller del framework ed è il primo componente ad agire nel ciclo di elaborazione della richiesta. Il controller ispeziona ogni richiesta in arrivo per determinare quale action dovrebbe gestire la richiesta.
- **Action**: rappresenta una funzione dell'applicativo ed è qui che viene definita la logica applicativa del sistema.

In particolare ogni modello è caratterizzato da n action che gestiscono le n funzionalità che devono essere garantite all'utente per quel modello (es. una action per recuperare il dettaglio di un progetto, una per la creazione/modifica di un progetto, una per la ricerca di un progetto attraverso dei filtri, ecc.).

Nel dettaglio, ciascuna action ha visibilità delle classi *service*, una per ciascun modello di interesse, necessarie per recuperare i dati del modello dal database. Questo avviene invocando le opportune query SQL costruite mediante il framework Hibernate e esposte dalla classe *service* in modo che queste possano essere utilizzate dalle opportune action per recuperare i dati di interesse dal database.

- **ActionMapper**: viene utilizzata da FilterDispatcher per determinare se la richiesta deve richiamare una action o meno. Quando viene fornito un `HttpServletRequest`, ActionMapper può restituire null se nessuna action corrisponde alla richiesta oppure può restituire un `ActionMapping` che descrive la chiamata.
- **ActionProxy**: se ActionMapper determina che deve essere richiamata una action per tale richiesta, FilterDispatcher delega il controllo ad ActionProxy. L'ActionProxy si consulta con ConfigurationManager, un'interfaccia che descrive la configurazione del framework, e crea una `ActionInvocation` che si occupa di richiamare l'opportuna action che deve gestire la richiesta entrante.
- **Interceptor**: corrispondono a dei componenti che vengono richiamati prima di eseguire la action e hanno il compito di preparare e inizializzare la action stessa.
- **Result**: corrisponde al risultato costruito e ritornato dalla action. Il risultato viene rappresentato da una pagina JSP.
- **Template**: corrispondono alle pagine di visualizzazione che effettivamente restituiscono la risposta all'utente.

Il flusso dell'architettura, che parte da una richiesta lato client, risulta essere il seguente:

1. la richiesta viene raccolta da `ActionMapper`;
2. viene chiamato il filtro `FilterDispatcher` che consulta `ActionMapper` per determinare se un'azione deve essere richiamata;
3. se `ActionMapper` trova che un'azione deve essere richiamata, `FilterDispatcher` delega il controllo ad `ActionProxy`;
4. `ActionProxy` legge il file di configurazione (`struts.xml`). `ActionProxy` crea un'istanza della classe `ActionInvocation` e delega il controllo.
5. `ActionInvocation` richiama gli `Interceptor` uno per uno (se richiesto) e quindi invoca la `action`;
6. la `action` ritorna il risultato, gli `Interceptor` vengono eseguiti nuovamente in ordine inverso e la risposta viene restituita a `FilterDispatcher`;
7. il risultato viene ritornato al client.

Capitolo 6

Prodotto finale

Il seguente capitolo ha la funzione di illustrare il prodotto finale realizzato in tutte le sue componenti.

6.1 Menu

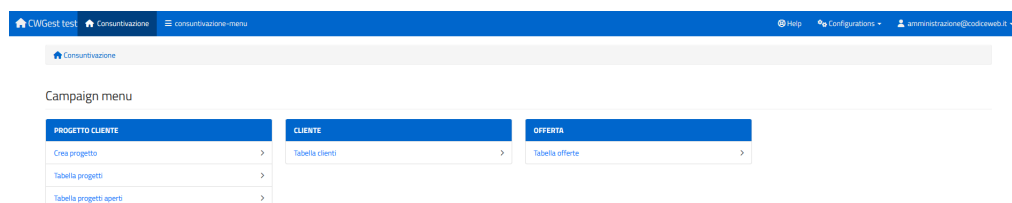


Figura 6.1: Modulo Campaign: Menu

Il menu ([figura 6.1](#)) mette a disposizione dell'utente tutte le funzionalità del modulo **Campaign** della webapp CW GEST. Tra le operazioni che l'utente può svolgere troviamo:

- creazione di un nuovo progetto;
- accesso alla tabella dei progetti;
- accesso alla tabella dei progetti aperti;
- accesso alla tabella dei clienti;
- accesso alla tabella delle offerte.

6.2 Creazione di un progetto

The screenshot shows a web application interface for creating a new project. The header bar is blue with the text 'CWGest test' and navigation links. The main content area is titled 'Nuovo progetto'. The form is organized into three main sections. The first section, 'Informazioni generali*', contains four input fields: 'Codice (es. 2020.20, 2019.01)', 'Titolo', 'Descrizione', and 'Data di inizio' which is pre-filled with '20/07/2023 - 14:17'. The second section, 'Referenti*', has two dropdown menus: 'Cliente' (selected: Heliadote - Heliadote S.p.A.) and 'Responsabile' (selected: ADMIN, Admin). The third section, 'Avanzamento delle attività*', includes three dropdowns: 'Stato' (selected: OPEN), 'Giorni (es. 200)', and 'Avanzamento di progetto' (selected: 0%). At the bottom right of the form, there are two buttons: 'Annulla' and 'Salva'. A small note at the bottom left indicates that fields marked with an asterisk are required.

Figura 6.2: Modulo Campaign: Creazione di un progetto

La pagina di creazione di un progetto (figura 6.2), raggiungibile dal menu (figura 6.1) o dalla tabella dei progetti (figura 6.3), mette a disposizione dell'utente un form composto da una serie di campi, tutti obbligatori, che devono essere compilati per creare il progetto. Tali campi sono i seguenti:

- codice identificativo (es. 2023/70);
- titolo;
- descrizione, composta dalle caratteristiche del sistema e dalle sue funzionalità;
- data di inizio dell'attività di sviluppo;
- cliente proponente (CWBI o altro);
- responsabile interno di CWBI assegnato al progetto;
- stato attuale (aperto, chiuso, approvato, rifiutato, sospeso);
- giorni di sviluppo previsti per portare a termine il progetto;
- percentuale di avanzamento del progetto.

6.3 Tabella progetti

ID	Progetto	Proponente	Referente	Stato	Data inizio	Giorni previsti	
18	2023.20 - Gameometry	CWBI - CWBI Banking Innovation	AMISTIA Michael (amistiamichael@gmail.com)	OPEN	27/11/2022	200	Modifica Dettaglio
14	2023.51 - Taxforce	CWBI - CWBI Banking Innovation	ADMIN Admin (amministrazione@codiceweb.it) *YOU*	SOSPESO	12/07/2023	400	Modifica Dettaglio
20	2023.75 - MyLife	Firmamavia - Firmamavia S.R.L.	ADMIN Admin (amministrazione@codiceweb.it) *YOU*	OPEN	04/06/2023	150	Modifica Dettaglio
17	2023.72 - Ticket	Heliodote - Heliodote S.P.A.	AMISTIA Michael (amistiamichael@gmail.com)	OPEN	02/07/2023	150	Modifica Dettaglio
13	2023.50 - CRM	Heliodote - Heliodote S.P.A.	AMISTIA Michael (amistiamichael@gmail.com)	OPEN	19/06/2023	300	Modifica Dettaglio

Figura 6.3: Modulo Campaign: Tabella progetti

La tabella dei progetti (figura 6.3) consente all'utente di:

- visualizzare la lista completa dei progetti presenti nel sistema. Per ogni progetto visualizzato è possibile inoltre:
 - visionare l'anteprima del progetto;
 - modificare il progetto in linea aprendo la pagina in figura 6.2 con l'unica differenza che i campi saranno precompilati con i valori del progetto memorizzato;
 - visualizzare il dettaglio del progetto in linea (figura 6.4).
- creare un nuovo progetto;
- ricercare uno o più progetti tramite il form di ricerca composto dai seguenti campi tutti opzionali:
 - istituto, campo standard presente in tutti i form di ricerca dell'azienda. Consente di ricercare progetti correlati ad una delle sedi di CWBI;
 - ID del progetto, assegnato dal database;
 - cliente;
 - codice;
 - titolo;
 - stato attuale;
 - data di creazione (da gg/mm/aaaa a gg/mm/aaaa).

L'utente può inoltre cancellare i dati inseriti nel form di ricerca tramite l'opzione *Reset* e effettuare la ricerca tramite l'opzione *Cerca*.

6.4 Dettaglio di un progetto

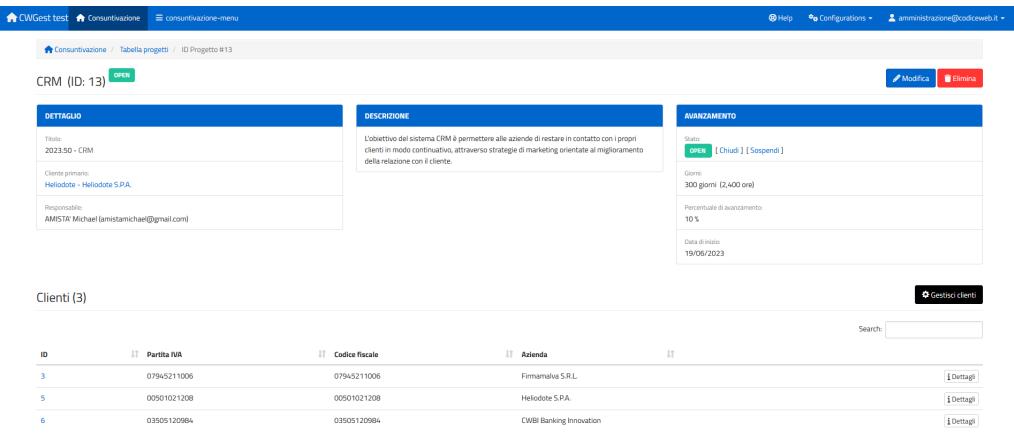


Figura 6.4: Modulo Campaign: Dettaglio di un progetto

La pagina di dettaglio di un progetto (figura 6.4) consente all’utente di visualizzare i dettagli di uno specifico progetto. Nella pagina sono presenti i medesimi dati inseriti in fase di creazione del progetto (figura 6.2). Nella pagina sono inoltre presenti le funzionalità di *modifica*, solo per i progetti non ancora chiusi, ed *eliminazione* del progetto. È inoltre presente una tabella che mostra l’elenco di tutti i clienti interessati al progetto.

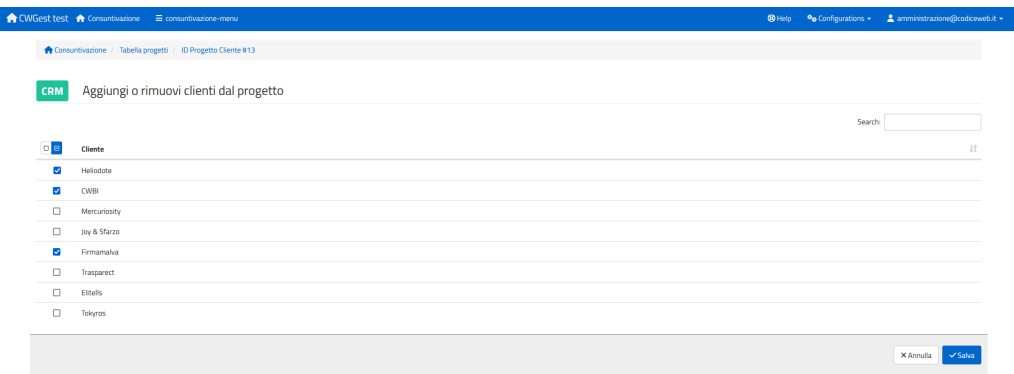


Figura 6.5: Modulo Campaign: Dettaglio di un progetto - Gestione clienti

Tramite la funzionalità *Gestione clienti* presente nella pagina di dettaglio di un progetto (figura 6.4) si arriva alla pagina (figura 6.5) che consente di aggiungere e rimuovere i clienti interessati al progetto.

6.5 Tabella progetti aperti

ID	Progetto	PropONENTE	Referente	Percentuale di avanzamento	Data Inizio	Giorni previsti	
18	2023.20 - Gameometry	CWBI - CWBI Banking Innovation	AMISTAR Michael (amistarmichael@gmail.com)	100%	27/11/2022	200	Modifica Dettaglio
13	2023.50 - CRM	Heliodote - Heliodote S.P.A.	AMISTAR Michael (amistarmichael@gmail.com)	0%	19/06/2023	300	Modifica Dettaglio
17	2023.72 - Ticket	Heliodote - Heliodote S.P.A.	AMISTAR Michael (amistarmichael@gmail.com)	50%	02/07/2023	150	Modifica Dettaglio
20	2023.75 - MyLife	Firmamulva - Firmamulva S.R.L.	ADMIN Admin (amministrazione@codiceweb.it) *YOU*	0%	04/06/2023	150	Modifica Dettaglio

Figura 6.6: Modulo Campaign: Tabella progetti aperti

La tabella dei progetti aperti (figura 6.6) consente all'utente di visualizzare la lista di tutti i progetti ancora aperti. Oltre a visualizzare l'anteprima dei diversi progetti in elenco, la pagina consente, per ciascun progetto in linea, di accedere alla pagina di modifica e alla pagina di dettaglio (figura 6.4).

6.6 Tabella clienti

ID	Codice - Descrizione	
5	Heliodote - Heliodote S.P.A.	Dettaglio
6	CWBI - CWBI Banking Innovation	Dettaglio
13	Mercuriosidy - Mercuriosidy & Co	Dettaglio
4	Joy & Sferzo - Joy & Sferzo S.P.A.	Dettaglio
3	Firmamulva - Firmamulva S.R.L.	Dettaglio
14	Transparent - Transparent & Co	Dettaglio
1	Elitello - Elitello & Co	Dettaglio
10	Tokiyros - Tokiyros S.R.L.	Dettaglio

Figura 6.7: Modulo Campaign: Tabella clienti

La tabella dei clienti (figura 6.7) consente all'utente di:

- visualizzare la lista completa dei clienti presenti nel sistema. Per ogni cliente visualizzato è possibile accedere alla sua pagina di dettaglio (figura 6.8);
- ricercare uno o più clienti tramite il form di ricerca composto dai seguenti campi tutti opzionali:
 - istituto, campo standard presente in tutti i form di ricerca dell'azienda. Consente di ricerca progetti correlati ad una delle sedi di CWBI;

- ID del cliente, assegnato dal database;
- codice, nome dell'azienda;
- descrizione, nome esteso dell'azienda.

L'utente può inoltre cancellare i dati inseriti nel form di ricerca tramite l'opzione *Reset* e effettuare la ricerca tramite l'opzione *Cerca*.

6.7 Dettaglio di un cliente

La pagina di dettaglio di un cliente consente di visualizzare, oltre ai dati di un cliente (codice, descrizione, partita iva e codice fiscale), due schede:

1. **scheda dei progetti associati** (figura 6.8), contenente la lista di tutti i progetti correlati al cliente e suddivisi in base allo stato del progetto stesso;
2. **scheda delle offerte associate** (figura 6.9), contenente la lista di tutte le offerte formulate al cliente e relative ad uno dei progetti a cui è associato.

Heliodote (ID: 5)

Progetti Offerte

DESCRIZIONE	PARTITA IVA	CODICE FISCALE
Heliodote S.p.A.	00001021208	00001021208

OPEN (3) SUSPENDED (1)

ID	Stato	Progetto	Giorni previsti	Percentuale di avanzamento	Responsabile	Data inizio	
18	OPEN [Chiudi] [Sospendi]	2023.20 - Gasmometry	200	81%	AMSTIA Michael (amstiamichael@gmail.com)	27/11/2022	[Modifica] [Dettaglio]
17	OPEN [Chiudi] [Sospendi]	2023.72 - Ticket	150	50%	AMSTIA Michael (amstiamichael@gmail.com)	02/07/2023	[Modifica] [Dettaglio]
13	OPEN [Chiudi] [Sospendi]	2023.50 - CRM	300	10%	AMSTIA Michael (amstiamichael@gmail.com)	19/06/2023	[Modifica] [Dettaglio]

Figura 6.8: Modulo Campaign: Dettaglio di un cliente - Scheda progetti associati

Heliodote (ID: 5)

Progetti Offerte

DESCRIZIONE	PARTITA IVA	CODICE FISCALE
Heliodote S.p.A.	00001021208	00001021208

ACCETTATA (2) PREPARAZIONE (1) + Aggiungi offerta

ID	Titolo	Stato	Progetto di riferimento	Data di stipula	
22	Contratto n.10/2023	ACCETTATA	18	25/07/2023	[Dettaglio]
19	Contratto n.7/2023	ACCETTATA	13	25/07/2023	[Dettaglio]

Figura 6.9: Modulo Campaign: Dettaglio di un cliente - Scheda offerte associate

Nella scheda delle offerte ([figura 6.9](#)), oltre a visualizzare tutte le offerte formulate per il cliente selezionato, suddivise in base al loro stato attuale, è possibile creare una nuova offerta per il cliente.

Nuova Offerta

Dettagli*:

Titolo *

Descrizione *

Progetto: 2023.20 - Gameometry

Progetto di riferimento

Stato: PREPARAZIONE*

Stato attuale

Imponibili*:

Imponibile 1 *

Imponibile 2 *

Imponibile 3 *

(*) required fields

Annulla Salva

Figura 6.10: Modulo Campaign: Dettaglio di un cliente - Aggiungi offerta

Nella pagina della nuova offerta ([figura 6.10](#)) l'utente ha la possibilità di aggiungere una nuova offerta per il cliente selezionato in precedenza nella pagina in [figura 6.9](#). I campi che compongono la nuova offerta sono i seguenti:

- titolo;
- descrizione;
- progetto su cui formulare l'offerta. Alla lista dei progetti selezionabili corrispondono i progetti associati al cliente in questione e visibili nella scheda dei progetti associati della pagina di dettaglio del cliente ([figura 6.8](#)).
- stato (in preparazione, in accettazione, accettata);
- imponibile 1, corrispondente al primo prezzo formulato al cliente per il progetto selezionato;
- imponibile 2, corrispondente al secondo prezzo formulato al cliente per il progetto selezionato;
- imponibile 3, corrispondente al terzo prezzo formulato al cliente per il progetto selezionato.

6.8 Tabella offerte

ID	Titolo	Cliente coinvolto	Stato attuale	Data
22	Contratto n. 10/2023	Heliodote	ACCETTATA	25/07/2023
21	Contratto n. 9/2023	Firmamalba	ACCETTATA	25/07/2023
20	Contratto n. 8/2023	Firmamalba	ACCETTATA	25/07/2023
19	Contratto n. 7/2023	Heliodote	ACCETTATA	25/07/2023
17	Proposta di contratto	Heliodote	PREPARAZIONE	21/07/2023

Figura 6.11: Modulo Campaign: Tabella offerte

La tabella delle offerte (figura 6.11) consente all'utente di:

- visualizzare la lista completa delle offerte presenti nel sistema. Per ogni offerta visualizzata è possibile modificare l'offerta stessa tramite la pagina in figura 6.10, con l'unica differenza che i campi saranno precompilati con i valori dell'offerta memorizzata, e/o accedere alla pagina di dettaglio dell'offerta di interesse (figura 6.12);
- ricercare una o più offerte tramite il form di ricerca composto dai seguenti campi tutti opzionali:
 - istituto, campo standard presente in tutti i form di ricerca dell'azienda. Consente di ricerca progetti correlati ad una delle sedi di CWBI;
 - ID del cliente, assegnato dal database;
 - cliente correlato;
 - progetto correlato;
 - stato attuale;
 - data di creazione (da gg/mm/aaaa a gg/mm/aaaa).

L'utente può inoltre cancellare i dati inseriti nel form di ricerca tramite l'opzione *Reset* e effettuare la ricerca tramite l'opzione *Cerca*.

6.9 Dettaglio di un'offerta

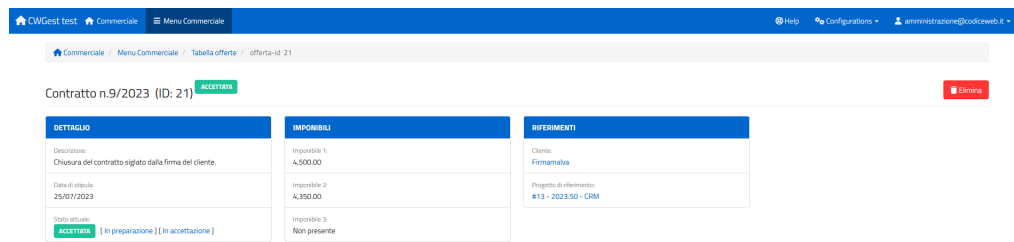


Figura 6.12: Modulo Campaign: Dettaglio di un'offerta

La pagina di dettaglio di un'offerta (figura 6.12) consente all'utente di visualizzare i dettagli di una specifica offerta. Nella pagina sono presenti i medesimi dati inseriti in fase di creazione dell'offerta (figura 6.10). Sono inoltre presenti le funzionalità di *modifica*, solo per le offerte non ancora accettate, ed *eliminazione* dell'offerta.

Capitolo 7

Conclusioni

Il seguente capitolo ha la funzione di illustrare l'attività di test eseguita sul prodotto finale. Viene proposto anche un resoconto complessivo sull'attività di stage.

7.1 Test e validazione

L'attività conclusiva del percorso di stage è stata quella di testare e verificare insieme al tutor aziendale il prodotto finale realizzato secondo i requisiti individuati e raccolti nell'attività di analisi.

Si riporta quindi una tabella riassuntiva dei [test di sistema](#)^[g] effettuati sul prodotto finale. Ciascun test è caratterizzato da:

- requisito di riferimento;
- test;
- esito del test.

7.1.1 Test di sistema

Requisito	Test	Esito
RFO1	Si verifica che l'utente possa utilizzare il menu.	Superato
RFO2	Si verifica che l'utente possa creare un nuovo progetto.	Superato
RFO3	Si verifica che l'utente possa accedere alla tabella dei progetti.	Superato
RFO4	Si verifica che l'utente possa accedere alla tabella dei progetti aperti.	Superato
RFO5	Si verifica che l'utente possa accedere alla tabella dei clienti.	Superato
RFO6	Si verifica che l'utente possa accedere alla tabella delle offerte.	Superato
RFO7	Si verifica che l'utente inserisca il codice del progetto.	Superato
RFO8	Si verifica che l'utente inserisca il titolo del progetto.	Superato
RFO9	Si verifica che l'utente inserisca la descrizione del progetto.	Superato
RFO10	Si verifica che l'utente inserisca la data di inizio del progetto.	Superato
RFO11	Si verifica che l'utente selezioni il cliente committente del progetto.	Superato
RFO12	Si verifica che l'utente selezioni il responsabile interno del progetto.	Superato
RFO13	Si verifica che l'utente selezioni lo stato di avanzamento del progetto.	Superato
RFO14	Si verifica che l'utente inserisca i giorni di sviluppo previsti.	Superato
RFO15	Si verifica che l'utente selezioni la percentuale di avanzamento del progetto.	Superato
RFO16	Si verifica che l'utente visualizzi un errore quando lascia vuoto il campo codice del progetto.	Superato
RFO17	Si verifica che l'utente visualizzi un errore quando lascia vuoto il campo titolo del progetto.	Superato
RFO18	Si verifica che l'utente visualizzi un errore quando lascia vuoto il campo descrizione del progetto.	Superato
RFO19	Si verifica che l'utente visualizzi un errore quando lascia vuoto il campo data di inizio del progetto.	Superato
RFO20	Si verifica che l'utente visualizzi un errore quando lascia vuoto il campo giorni di sviluppo del progetto.	Superato
RFO21	Si verifica che l'utente possa ricercare uno o più progetti.	Superato
RFO22	Si verifica che l'utente ricerchi un progetto per ID.	Superato
RFO23	Si verifica che l'utente ricerchi un progetto per cliente associato.	Superato

Requisito	Descrizione	Esito
RFO24	Si verifica che l'utente ricerchi un progetto per responsabile interno.	Superato
RFO25	Si verifica che l'utente ricerchi un progetto per codice di progetto.	Superato
RFO26	Si verifica che l'utente ricerchi un progetto per titolo.	Superato
RFO27	Si verifica che l'utente ricerchi un progetto per stato di avanzamento.	Superato
RFO28	Si verifica che l'utente ricerchi un progetto per data di creazione.	Superato
RFO29	Si verifica che l'utente visualizzi la lista dei progetti.	Superato
RFO30	Si verifica che l'utente visualizzi il dettaglio di un progetto.	Superato
RFO31	Si verifica che l'utente visualizzi l'ID di un progetto.	Superato
RFO32	Si verifica che l'utente visualizzi la stringa "codice - titolo" di un progetto.	Superato
RFO33	Si verifica che l'utente visualizzi il committente di un progetto.	Superato
RFO34	Si verifica che l'utente visualizzi il responsabile interno di un progetto.	Superato
RFO35	Si verifica che l'utente visualizzi lo stato di avanzamento di un progetto.	Superato
RFO36	Si verifica che l'utente visualizzi la data di inizio di un progetto.	Superato
RFO37	Si verifica che l'utente visualizzi i giorni di sviluppo previsti di un progetto.	Superato
RFO38	Si verifica che l'utente visualizzi la descrizione del progetto.	Superato
RFO39	Si verifica che l'utente visualizzi la percentuale di avanzamento del progetto.	Superato
RFO40	Si verifica che l'utente visualizzi e gestisce i clienti del progetto.	Superato
RFO41	Si verifica che l'utente modifichi il progetto.	Superato
RFO42	Si verifica che l'utente elimini il progetto.	Superato
RFO43	Si verifica che l'utente ricerchi uno o più clienti.	Superato
RFO44	Si verifica che l'utente ricerchi un cliente per ID.	Superato
RFO45	Si verifica che l'utente ricerchi un cliente per codice.	Superato
RFO46	Si verifica che l'utente ricerchi un cliente per descrizione.	Superato
RFO47	Si verifica che l'utente visualizzi la lista dei clienti.	Superato
RFO48	Si verifica che l'utente visualizzi il dettaglio di un cliente.	Superato
RFO49	Si verifica che l'utente visualizzi l'ID di un cliente.	Superato

Requisito	Descrizione	Esito
RFO50	Si verifica che l'utente visualizzi la stringa "codice – descrizione" di un cliente.	Superato
RFO51	Si verifica che l'utente visualizzi partita iva del cliente.	Superato
RFO52	Si verifica che l'utente visualizzi il codice fiscale del cliente.	Superato
RFO53	Si verifica che l'utente visualizzi la lista dei progetti associati al cliente.	Superato
RFO54	Si verifica che l'utente visualizzi la lista delle offerte associate al cliente.	Superato
RFO55	Si verifica che l'utente crei una nuova offerta.	Superato
RFO56	Si verifica che l'utente visualizzi la lista delle offerte.	Superato
RFO57	Si verifica che l'utente inserisca il titolo dell'offerta.	Superato
RFO58	Si verifica che l'utente inserisca la descrizione dell'offerta.	Superato
RFO59	Si verifica che l'utente selezioni il progetto coinvolto nell'offerta.	Superato
RFO60	Si verifica che l'utente selezioni lo stato attuale dell'offerta.	Superato
RFO61	Si verifica che l'utente inserisca l'imponibile 1 dell'offerta.	Superato
RFO62	Si verifica che l'utente inserisca l'imponibile 2 dell'offerta.	Superato
RFO63	Si verifica che l'utente inserisca l'imponibile 3 dell'offerta.	Superato
RFO64	Si verifica che l'utente visualizzi un errore quando lascia vuoto il campo titolo dell'offerta.	Superato
RFO65	Si verifica che l'utente visualizzi un errore quando lascia vuoto il campo descrizione dell'offerta.	Superato
RFO66	Si verifica che l'utente visualizzi un errore quando lascia vuoto il campo imponibile 1 dell'offerta.	Superato
RFO67	Si verifica che l'utente visualizzi un errore quando il tipo di valore inserito del campo imponibile 1 non è numerico.	Superato
RFO68	Si verifica che l'utente visualizzi un errore quando il tipo di valore inserito del campo imponibile 2 non è numerico.	Superato
RFO69	Si verifica che l'utente visualizzi un errore quando il tipo di valore inserito del campo imponibile 3 non è numerico.	Superato
RFO70	Si verifica che l'utente ricerchi una o più offerte.	Superato
RFO71	Si verifica che l'utente ricerchi un'offerta per ID.	Superato
RFO72	Si verifica che l'utente ricerchi un'offerta per cliente associato.	Superato
RFO73	Si verifica che l'utente ricerchi un'offerta per stato attuale.	Superato
RFO74	Si verifica che l'utente ricerchi un'offerta per data di stipula.	Superato
RFO75	Si verifica che l'utente visualizzi il dettaglio di un'offerta.	Superato

Requisito	Descrizione	Esito
RFO76	Si verifica che l'utente visualizzi l'ID di un'offerta.	Superato
RFO77	Si verifica che l'utente visualizzi la data di stipula di un'offerta.	Superato
RFO78	Si verifica che l'utente visualizzi il cliente coinvolto di un'offerta.	Superato
RFO79	Si verifica che l'utente visualizzi lo stato di un'offerta.	Superato
RFO80	Si verifica che l'utente visualizzi il titolo di un'offerta.	Superato
RFO81	Si verifica che l'utente visualizzi il progetto di un'offerta.	Superato
RFO82	Si verifica che l'utente visualizzi la descrizione dell'offerta.	Superato
RFO83	Si verifica che l'utente visualizzi l'imponibile 1 dell'offerta.	Superato
RFO84	Si verifica che l'utente visualizzi l'imponibile 2 dell'offerta.	Superato
RFO85	Si verifica che l'utente visualizzi l'imponibile 3 dell'offerta.	Superato
RFO86	Si verifica che l'utente modifichi l'offerta.	Superato
RFO87	Si verifica che l'utente elimini l'offerta.	Superato
RVO1	Si verifica che l'applicativo lato back-end sia realizzato in Java.	Superato
RVO2	Si verifica che l'applicativo lato back-end sia realizzato mediante il framework Spring.	Superato
RVO3	Si verifica che l'applicativo lato back-end sia realizzato mediante il framework Hibernate.	Superato
RVO4	Si verifica che l'applicativo lato fron-end sia realizzato tramite HTML5, CSS3 e JavaScript.	Superato
RVO5	Si verifica che l'applicativo lato front-end sia realizzato mediante il framework Bootstrap.	Superato
RVO6	Si verifica che l'applicativo sia funzionante in tutte le sue componenti.	Superato

Tabella 7.1: Test di sistema

Tutti i requisiti richiesti, obbligatori e di vincolo, sono stati soddisfatti con un riscontro positivo anche da parte del tutor aziendale.

7.2 Resoconto dello stage

L'attività di stage mi ha permesso di entrare a far parte di un nuovo lato della programmazione, ovvero lo sviluppo di applicazioni Java EE. Le metodologie di approccio aziendali volte all'affrontare nuovi progetti e attività sono state stimolanti e rappresentano un metodo di lavoro solido e duraturo che hanno sicuramente portato ad una mia crescita professionale.

Lo stage mi ha permesso di conoscere e apprendere nuove tecnologie molto utilizzate nella programmazione, tra cui: Spring, Hibernate, Bootstrap e Struts. Questa conoscenza appresa ha arricchito il mio bagaglio informatico.

Ho avuto inoltre la possibilità di lavorare con un team di esperti e appassionati del settore che mi hanno trasmesso la loro conoscenza e la passione con la quale operano ogni giorno nel mondo dello sviluppo software.

Sitografia

Spring <https://spring.io/>

Hibernate <https://hibernate.org/>

Bootstrap <https://getbootstrap.com/>

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

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

Glossario

UML (Unified Modeling Language) Linguaggio che permette, tramite l'utilizzo di modelli visuali, di analizzare, descrivere, specificare e documentare un sistema software anche complesso.

Requisito Descrizione dei servizi che un sistema software deve fornire, insieme ai vincoli da rispettare sia in fase di sviluppo che durante la fase di operatività del software.

Framework Architettura logica di supporto sulla quale un software può essere progettato e realizzato, spesso facilitandone lo sviluppo da parte del programmatore.

IDE (Integrated Development Environment) Software che offre agli sviluppatori un ambiente per lo sviluppo, il test e il debug di un'applicazione.

Test di sistema Una tipologia di test del software che viene sempre condotto su un intero sistema. Verifica se il sistema è conforme ai suoi requisiti, qualunque essi siano.