

Università degli Studi di Padova

DIPARTIMENTO DI MATEMATICA "TULLIO LEVI-CIVITA"

CORSO DI LAUREA IN INFORMATICA



SmartFarm Technologies: Sistema Integrato
per l'ottimizzazione e l'efficientamento della
gestione agricola sostenibile

Tesi di laurea

Relatore

Prof. Marco Zanella

Laureando

Valerio Occhinegro

Matricola 2011069

ANNO ACCADEMICO 2023-2024

Sommario

Il presente documento descrive il lavoro svolto durante il periodo di stage dal laureando Valerio Occhinegro presso l'azienda Spazio Dev Srl di Tombolo (PD). Lo stage, svoltosi al termine del percorso di studi della Laurea Triennale in "Scienze Informatiche", ha avuto una durata complessiva di 306 ore.

Il lavoro in questione è stato suddiviso in 4 diverse fasi, ognuna delle quali caratterizzata da obiettivi specifici e scadenze precise, e si basa su un'analisi intelligente di siti web, improntata alla vendita dei servizi aziendali a potenziali clienti. Il ruolo del laureando è quello di gestire e archiviare i siti web target e successivamente classificarli per fornire a Spazio Dev input utili per acquisire l'abilità di ampliare la quantità dei propri clienti in maniera rapida e automatizzata. Nello specifico, il progetto di stage ha l'obiettivo di sviluppare competenze avanzate nell'intelligenza artificiale, con un focus particolare sulla classificazione e organizzazione dei dati; uno studio che si inserisce nel contesto di "SalesCRM: Customer Relationship Manager", un sistema integrato che mira a ottimizzare la gestione delle relazioni con i clienti e a migliorare l'efficienza dei venditori. INSERIRE VERO TITOLO Lo scopo ultimo del progetto è la creazione di un sistema intelligente che sia in grado di analizzare dataset complessi, classificare la qualità dei siti web (siti che potrebbero essere migliorati e siti che non necessitano di modifiche) e proporre i propri servizi alle aziende che ne necessitano in maniera automatica. Il laureando è responsabile dello sviluppo di funzionalità chiave del sistema che consentiranno di fornire soluzioni strategiche agli utilizzatori della piattaforma.

“Non importa quanto piano vai, finché non ti fermi”

— Confucio

Ringraziamenti

Padova, Dicembre 2024

Valerio Occhinegro

Indice

1	Introduzione	1
1.1	L'azienda	1
1.2	Servizi offerti	2
1.3	Clientela	2
1.4	Prodotto di punta	3
1.5	Organizzazione del testo	3
2	Processi e metodologie	4
2.1	Ciclo di vita del software	4
2.1.1	Filosofia Agile	4
2.1.2	Tecnologie di supporto	5
2.2	Gestione della configurazione	5
2.2.1	Tecnologie di supporto	6
3	Il progetto	8
3.1	Analisi del progetto	8
3.1.1	SalesCRM	8
3.1.2	Integrazione	10
3.2	Analisi e gestione dei rischi	10
3.3	Obiettivi	12
3.4	Pianificazione	12
4	Analisi dei requisiti	13
4.1	Casi d'uso	13
4.2	Tracciamento dei requisiti	16
5	Strumenti e tecnologie	18
5.1	Strumenti	18
5.2	Tecnologie	19
5.2.1	Python	19
5.2.2	CUDA	20
5.2.3	Docker	20
5.2.4	DDEV	20
5.2.5	WSL	20
5.2.6	Laravel	20
5.2.7	Filament	20
5.2.8	MySQL	20
5.3	Ciclo di vita del software	20
5.4	Progettazione	20

<i>INDICE</i>	v
5.5 Design Pattern utilizzati	21
5.6 Codifica	21
6 Progettazione e codifica	22
7 Conclusioni	23
7.1 Consuntivo finale	23
7.2 Raggiungimento degli obiettivi	23
7.3 Conoscenze acquisite	23
7.4 Valutazione personale	23
A Appendice A	24
Acronimi e abbreviazioni	25
Glossario	26
Bibliografia	27

Elenco delle figure

1.1	Logo dell'azienda	2
1.2	Logo dell'applicazione RelAI	3
2.1	Schema del modello scrum	4
2.2	Logo di Plane	5
2.3	Logo di Telegram	5
2.4	Logo di Git	6
2.5	Logo di Gitea	6
2.6	Esempio di Git Flow	7
3.1	Home page di SalesCRM	9
3.2	Calendario contenuto all'interno di SalesCRM	9
3.3	Mappa contenuta all'interno di SalesCRM	9
4.1	Use Case - UC0: Scenario principale	13
4.2	Use Case - UC1: Scenario principale	14
4.3	Use Case - UC2: Clusterizzazione degli screenshot	15
4.4	Use Case - UC3: IA classificativa	15
4.5	Use Case - UC4: invio e-mail	16
5.1	Logo di Visual Studio Code	18
5.2	Logo di phpMyAdmin	19
5.3	Logo di Remote Ripple	19

Elenco delle tabelle

3.1	Tabella degli obiettivi	12
-----	-----------------------------------	----

4.1	Tabella del tracciamento dei requisiti funzionali	17
4.2	Tabella del tracciamento dei requisiti qualitativi	17
4.3	Tabella del tracciamento dei requisiti di vincolo	17

Capitolo 1

Introduzione

L'esperienza dello stage è l'elemento che porta alla conclusione del percorso accademico dello studente; pertanto è da considerare una delle prove più formanti e impegnative di tutto l'arco di studi.

Durante questo periodo lo studente ha l'occasione di mettere alla prova le conoscenze acquisite negli anni, ma soprattutto ha la possibilità di valutare le proprie abilità di autoapprendimento, scontrandosi con tecnologie e problemi mai visti in precedenza. L'introduzione al mondo del lavoro è fondamentale per rendere il più lineare possibile la transizione da studio ad applicazione pratica delle conoscenze, fornisce inoltre un assaggio delle responsabilità che gravano su un programmatore durante la permanenza nella propria azienda. In questo capitolo viene presentata l'azienda Spazio Dev presso la quale lo studente ha svolto lo stage nel periodo che va dal giorno 23/09/2024 al 14/11/2024.

1.1 L'azienda

Spazio Dev è una software house italiana nata a Tombolo (PD) nel 2023; nonostante la giovane età è riuscita ad affermarsi nel suo campo dando vita a molteplici progetti ambiziosi. L'obiettivo primario dell'azienda è quello di supportare diverse imprese tramite soluzioni digitali e tecnologiche sviluppate su misura. L'azienda conta 16 membri che ricoprono i diversi ruoli di sviluppatori, grafici, responsabili marketing e sistemisti. L'organico ha un'età media inferiore ai trenta anni, questo fattore contribuisce a rendere l'ambiente lavorativo dinamico e conviviale perfetto per coltivare nuovi talenti. Spazio Dev affianca e aggiorna il cliente tramite una comunicazione costante, finalizzata alla personalizzazione del prodotto in base alle diverse esigenze. Il focus principale dell'azienda è infatti quello di fornire prodotti specifici per ciascun cliente e non la vendita di prodotti standard.

Il logo dell'azienda è rappresentato in Fig. [1.1](#)



Figura 1.1: Logo dell'azienda

1.2 Servizi offerti

L'offerta di Spazio Dev è molto ampia e consente allo staff di aiutare i clienti nei seguenti ambiti:

- **Sviluppo di siti web:** La produzione di siti web è uno dei servizi più richiesti all'azienda e rappresenta uno dei motori trainanti. Gli sviluppatori hanno un occhio di riguardo per le nuove tecnologie, che sfruttano per offrire al cliente un prodotto accattivante e ottimizzato in maniera tale da garantire numerose visite.
- **Sviluppo software e integrazione IA:** I software prodotti sono solitamente indirizzati a una clientela variegata e possono spaziare in vari campi, in base alle esigenze di ciascun acquirente. Gli applicativi più venduti hanno come scopo principale la gestione dell'azienda e delle relazioni con i clienti. Per differenziarsi dalla concorrenza Spazio Dev ha introdotto nuove funzionalità impiegando modelli di IA capaci di predire trend futuri di acquisto e di anticipare i bisogni dei clienti.
- **Marketing:** Gli esperti di marketing creano contenuti di alta qualità, pertinenti e ottimizzati, atti a migliorare la presenza on-line del brand. Per aumentare l'engagement vengono studiate apposite campagne social e di posta elettronica.

1.3 Clientela

La clientela di Spazio Dev è molto variegata ed è suddivisibile nelle seguenti categorie:

- imprese di varie dimensioni specializzate in commercio e manifattura che necessitano di migliorare la propria presenza nel web, la gestione della contabilità, la logistica e le risorse umane.
- start-up in cerca di visibilità e di strategie di marketing.
- liberi professionisti come avvocati, medici, eccetera.

1.4 Prodotto di punta

RelAI (Fig. 1.2) rappresenta in tutto e per tutto l'azienda poiché nasce dall'unione di tutti i suoi core business: sviluppo web, integrazione IA e marketing. Il software è un CRM (Customer Relationship Manager) in cloud che consente di personalizzare le relazioni con i clienti e di incrementare le vendite grazie a campagne di comunicazione su misura. La sua architettura modulare e le numerose configurazioni si adattano perfettamente in molti settori. L'IA implementata è utile per predire futuri trend di acquisto fondamentali per la creazione di strategie di marketing efficaci.



Figura 1.2: Logo dell'applicazione RelAI

1.5 Organizzazione del testo

[Il secondo capitolo](#) descrive ...

[Il terzo capitolo](#) approfondisce ...

[Il quarto capitolo](#) approfondisce ...

[Il quinto capitolo](#) approfondisce ...

[Il sesto capitolo](#) approfondisce ...

[Nel settimo capitolo](#) descrive ...

Per quanto riguarda la stesura del 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

Processi e metodologie

In questo capitolo sono riassunti i processi e le metodologie che lo studente ha applicato durante lo sviluppo del progetto di stage.

2.1 Ciclo di vita del software

Il ciclo di vita del software è l'insieme delle attività e dei processi che tutti i membri coinvolti devono seguire per la buona riuscita del prodotto. L'adozione di un modello di ciclo di vita è fondamentale per rispettare le scadenze, rimanere all'interno del budget ed evitare conflitti all'interno e all'esterno del team.

2.1.1 Filosofia Agile

Per favorire la collaborazione l'azienda si affida alla filosofia Agile e in particolare al modello scrum (Fig. 2.1) che ha come unità fondamentale lo "sprint" ossia un periodo di tempo breve (1 settimana) che viene utilizzato per sviluppare e testare nuove funzionalità; grazie a questi avanzamenti rapidi è possibile fornire codice flessibile e rapidamente adattabile alle nuove esigenze.

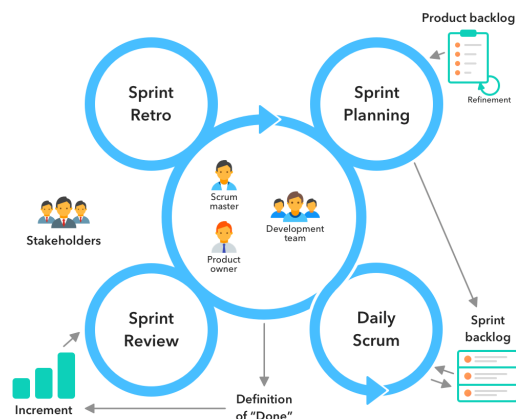


Figura 2.1: Schema del modello scrum

2.1.2 Tecnologie di supporto

- Plane (Fig. 2.2): è un ITS (Issue Tracking System) utilizzato per monitorare l'avanzamento del progetto e per assegnare a ciascun membro i compiti da effettuare; consente a tutte le parti coinvolte di avere una visione di insieme su quello che bisogna realizzare per portare a compimento lo sviluppo. In particolare è stato scelto perché altamente personalizzabile e per la sua natura open-source.



Figura 2.2: Logo di Plane

- Telegram (Fig. 2.3): è una piattaforma di messaggistica completa con molte funzionalità, viene sfruttata dall'azienda per coordinare i vari team di lavoro tramite l'ausilio di chat di gruppo. Fornisce inoltre molte funzionalità avanzate come bot utili per i più svariati casi d'uso.



Figura 2.3: Logo di Telegram

2.2 Gestione della configurazione

La gestione della configurazione è il processo tramite il quale è possibile identificare, controllare e coordinare i vari componenti del software e le risorse a esso associate durante l'intero ciclo di vita del prodotto. Grazie a questo processo gli sviluppatori possono tenere traccia delle modifiche e gestire le varie versioni garantendo il funzionamento del prodotto nel tempo.

2.2.1 Tecnologie di supporto

Le tecnologie utilizzate per il versionamento del prodotto sono:

- Git (Fig. 2.4): è un DVCS (Distributed Version Control System) ampiamente diffuso che consente di gestire e monitorare i cambiamenti del codice e della documentazione. Questo strumento è fondamentale per avere una storia completa dello sviluppo da poter sfruttare in caso di necessità. Il codice viene inserito all'interno di un repository, ossia una cartella che contiene tutti i file relativi al progetto, dove è possibile salvare ogni cambiamento tramite commit. Il commit segnala ogni modifica e la data in cui viene attuata, è dunque sufficiente spostarsi tra i vari commit per tornare ad una versione più o meno aggiornata.



Figura 2.4: Logo di Git

- Gitea (Fig. 2.5): è un servizio che supporta le repository di Git, è simile a GitHub, Bitbucket e GitLab, ma ha il vantaggio di essere self-hosted. Essendo tutti i repository aziendali mantenuti sul server interno, le comunicazioni atte al versionamento hanno risposta molto rapida.



Figura 2.5: Logo di Gitea

Regole di branch e commit

Per standardizzare l'utilizzo delle tecnologie di versionamento l'azienda adotta due convenzioni:

- Git Flow (Fig. 2.6): è un modello di branching ideato per avere un miglior controllo delle release.
- Conventional Commits: è una specifica basata su regole, che consentono una facile lettura dei commit sia da parte di utenti che da parte di strumenti automatici.

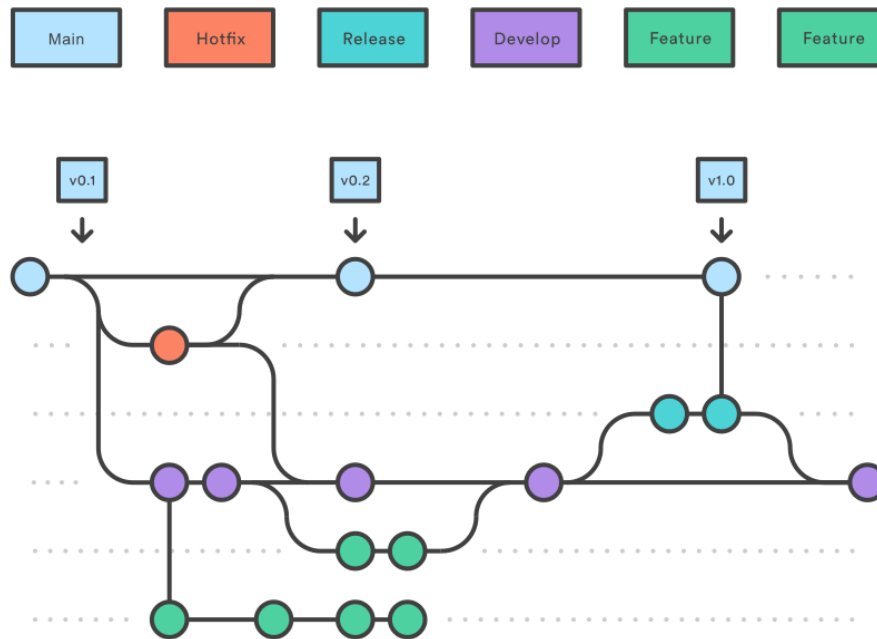


Figura 2.6: Esempio di Git Flow

Capitolo 3

Il progetto

Questo capitolo contiene le informazioni preliminari che sono state fornite allo studente e le misure preventive adottate per iniziare la produzione.

3.1 Analisi del progetto

Il progetto si basa sull'integrazione di un nuovo workflow all'interno di un applicativo preesistente; lo studente deve occuparsi dello sviluppo di una soluzione in grado di migliorare la portata dell'azienda per quanto riguarda il raggiungimento di nuovi clienti.

3.1.1 SalesCRM

SalesCRM è il CRM che viene utilizzato dai commerciali interni all'azienda; contiene tutte le informazioni relative ai contatti che sono stati raggiunti dal reparto vendite. Gli utenti sono in grado di immagazzinare i dati raccolti su clienti e potenziali clienti tramite un interfaccia intuitiva sviluppata utilizzando [Laravel](#) e [Filament](#).

Componenti principali

- Home page (Fig. [3.1](#)) contenente grafici informativi.
- Il modulo contenente il calendario (Fig. [3.2](#)) è utile per pianificare appuntamenti e visite.
- La sezione della mappa (Fig. [3.3](#)) viene sfruttato per conoscere il posizionamento dei clienti e organizzare incontri locali.

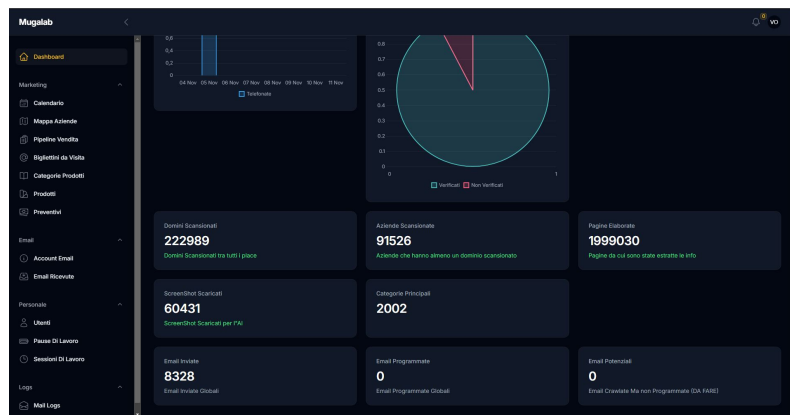


Figura 3.1: Home page di SalesCRM

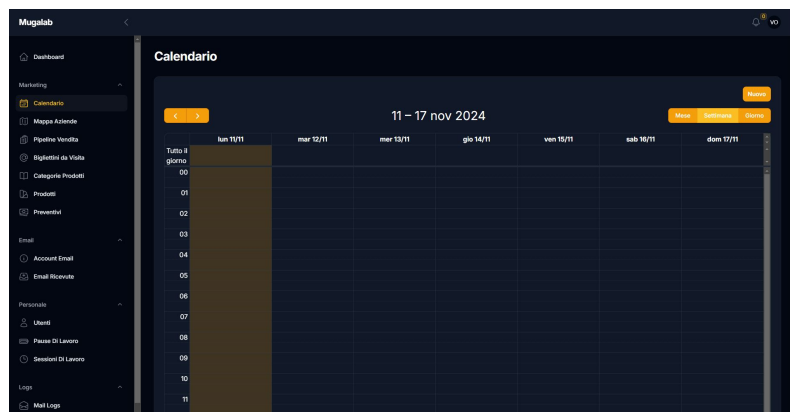


Figura 3.2: Calendario contenuto all'interno di SalesCRM

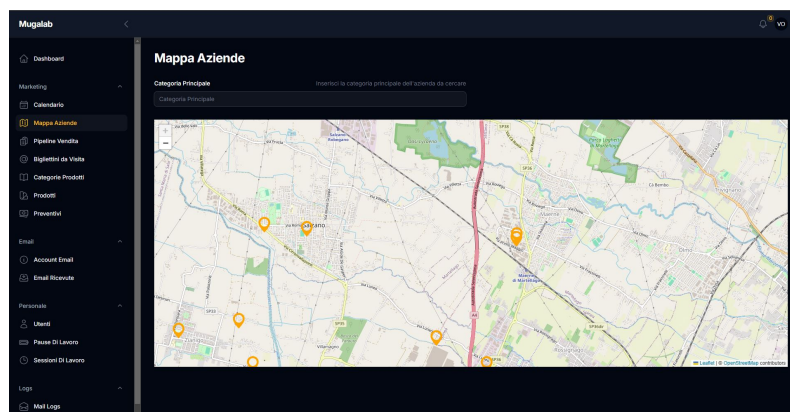


Figura 3.3: Mappa contenuta all'interno di SalesCRM

3.1.2 Integrazione

- Raccolta di dati: l'applicativo ha una funzionalità di web-scraping che raccoglie i link dei siti web di tutti i potenziali clienti, il compito del tirocinante è quello di creare un workflow automatico che si integri con il web-scraping per acquisire screenshot delle varie pagine del sito web.
- Clusterizzazione: la fase successiva alla raccolta dei dati consiste nello sviluppo di una IA addestrata in maniera non supervisionata che sia in grado di suddividere le varie immagini in clusters, differenziati in base alle caratteristiche riconosciute in ogni sito.
- IA classificativa: in seguito alla raccolta di un dataset di dimensioni congrue si procede alla suddivisione manuale degli screenshot precedentemente clusterizzati su una base qualitativa (siti migliorabili e siti ottimi); questo processo viene svolto con l'ottica dell'addestramento di una IA classificativa. L'IA addestrata in maniera supervisionata ha l'obiettivo di affidare un punteggio in valori centesimali a ogni sito.
- Invio di e-mail automatizzato: l'automazione della posta elettronica procede con l'invio di e-mail personalizzate ai proprietari dei siti web che hanno ricevuto una valutazione scarsa, per offrire loro un servizio di miglioramento.

3.2 Analisi e gestione dei rischi

Durante l'analisi del progetto lo stagista ha individuato alcuni rischi in cui potrà incorrere. Nella lista seguente sono elencati i rischi e le soluzioni ideate.

1. Rimodulazione dell'attività

Descrizione: dopo un mese dall'inizio dello stage lo studente è stato riposizionato sul progetto attuale scartando il progetto precedente e trovandosi dunque con meno settimane a disposizione per la produzione.

Soluzione: ridimensionamento delle attività e richieste di supporto più frequenti.

2. Approccio sperimentale

Descrizione: il progetto prevede dei contributi originali e sperimentali per cui non sono disponibili soluzioni già pronte.

Soluzione: auto apprendimento tramite tutorial on-line e richiesta di coinvolgimento di colleghi più esperti nell'ambito.

3. Costo dell'Addestramento

Descrizione: l'addestramento dell'intelligenza artificiale richiede l'utilizzo di potenti GPU di cui spesso l'hardware a disposizione è sprovvisto.

Soluzione: utilizzo del server aziendale per l'addestramento su CPU, ottenendo un compromesso tra costi e tempo di addestramento.

4. Quantità di dati di training insufficiente

Descrizione: l'IA necessita una grande quantità di dati in input per effettuare un training efficace.

Soluzione: aumento manuale del dataset e ricerca di dataset già pronti on-line.

5. Risultati della clusterizzazione non soddisfacenti

Descrizione: il clustering potrebbe risultare non conforme alle aspettative.

Soluzione: valutare la quantità di cluster da creare e sperimentare con altri metodi di clustering.

6. Overfitting del modello

Descrizione: il modello IA fornisce valutazioni accurate solo per le immagini utilizzate per il training.

Soluzione: valutare la quantità di cluster da creare e sperimentare con altri metodi di clustering.

3.3 Obiettivi

Gli obiettivi hanno lo scopo di delineare il percorso che lo studente deve affrontare per portare a termine il progetto nella maniera desiderata dall'azienda. Sono suddivisibili in:

- **O**: obbligatori
- **D**: desiderabili
- **F**: facoltativi

Codice	Descrizione
O01	Implementare un sistema robusto per la cattura degli screenshot, assicurando l'integrazione con il database per l'archiviazione e l'analisi.
O02	Garantire la creazione di una documentazione tecnica completa che supporti sia l'uso che la manutenzione del sistema sviluppato.
O03	Creazione di un IA in grado di suddividere in cluster i siti web.
O04	Creazione di un IA classificativa in grado di assegnare un punteggio ai siti-web analizzati.
D01	Aggiungere la valutazione creata dall'IA nel database utilizzato dal CRM.
D02	Automatizzare l'invio di e-mail alle aziende che hanno ottenuto una valutazione scarsa.
F01	Migliorare la raccolta dei dati delle aziende dal web
F02	Aggiornare automaticamente la valutazione dei siti web

Tabella 3.1: Tabella degli obiettivi

3.4 Pianificazione

Capitolo 4

Analisi dei requisiti

L'analisi dei requisiti è alla base della comprensione del prodotto da sviluppare, è utile per chiarire eventuali dubbi e per procedere in maniera spedita evitando perdite di tempo future

4.1 Casi d'uso

Per rendere più chiari i casi d'uso sono stati utilizzati dei diagrammi di tipo [Unified Modeling Language \(UML\)](#). Questo tipo di diagrammi descrivono funzioni e servizi forniti dal sistema agli attori che lo utilizzano. Essendo il progetto l'implementazione di un workflow automatico, le interazioni dell'utente devono essere minime. Per questo i casi d'uso sono pochi e molto sintetici.

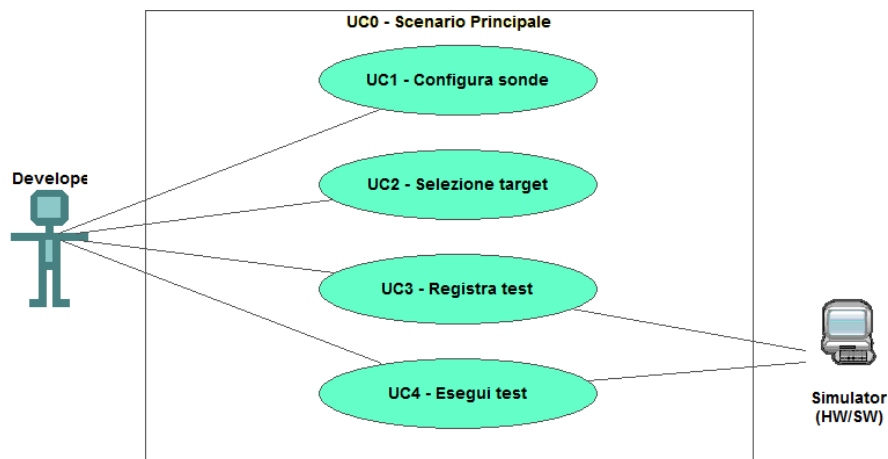


Figura 4.1: Use Case - UC0: Scenario principale

UC0: Scenario principale

Attori Principali: Programmatore.

Precondizioni: Il programmatore ha avviato l'ambiente di programmazione integrato.

Descrizione: Avvio delle componenti base dell'applicativo.

Postcondizioni: Il sistema è pronto per eseguire il workflow.

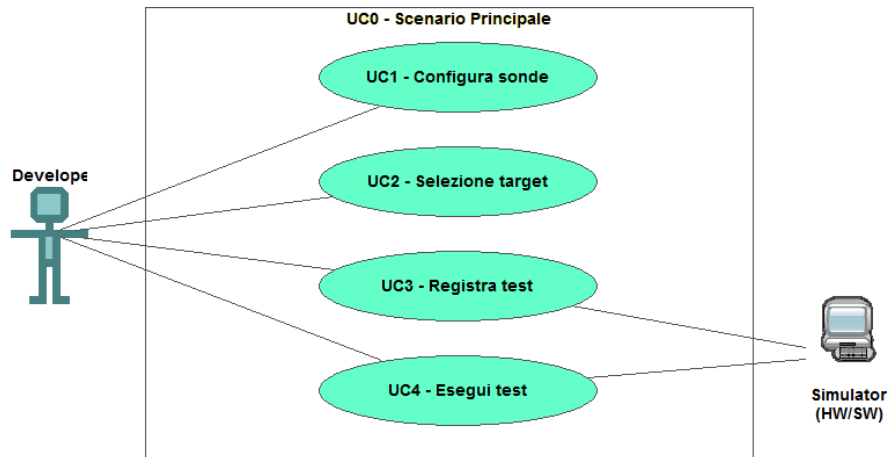


Figura 4.2: Use Case - UC1: Scenario principale

UC1: Acquisizione screenshot

Attori Principali: Programmatore, Web-scraper.

Precondizioni: Il web-scraper ha già raccolto i link alle pagine dei siti web delle aziende.

Descrizione: Il programmatore prepara il workflow per acquisire in maniera automatica gli screenshot delle pagine raccolte in precedenza.

Postcondizioni: Gli screenshot vengono salvati nel database, pronti per la fase di analisi e classificazione.

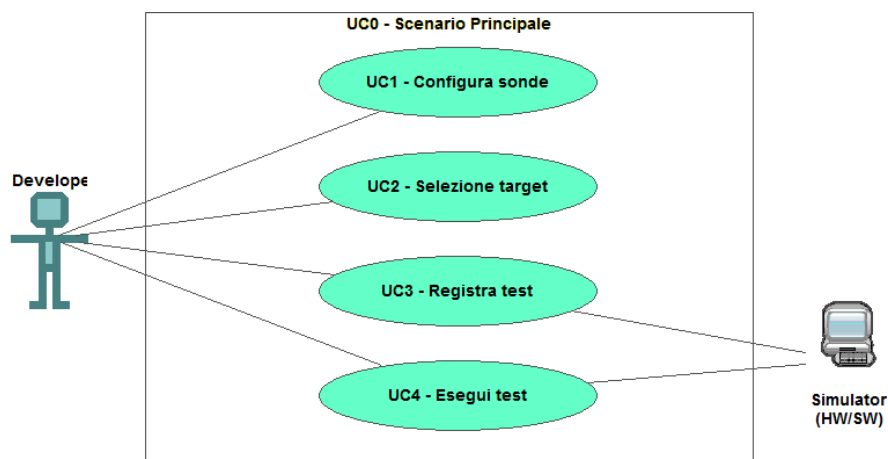


Figura 4.3: Use Case - UC2: Clusterizzazione degli screenshot

UC2: Clusterizzazione degli screenshot

Attori Principali: Programmatore, IA non supervisionata.

Precondizioni: Gli screenshot delle pagine sono stati acquisiti e salvati nel database.

Descrizione: Il programmatore avvia il processo di clustering, che organizza gli screenshot in clusters in base alle feature estratte.

Postcondizioni: Gli screenshot vengono clusterizzati e inseriti in apposite cartelle.

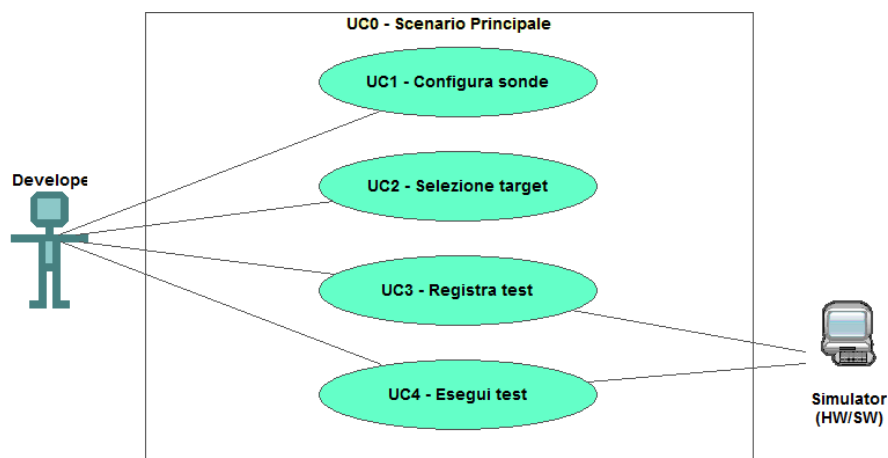


Figura 4.4: Use Case - UC3: IA classificativa

UC3: Valutazione qualitativa e classificazione

Attori Principali: Programmatore, IA classificativa.

Precondizioni: Gli screenshot sono stati inseriti in cartelle.

Descrizione: Lo sviluppatore addestra l'IA classificativa sfruttando i cluster precedentemente creati, e creando un dataset.

Postcondizioni: Ogni sito ottiene un punteggio qualitativo che viene salvato nel database per operazioni future.

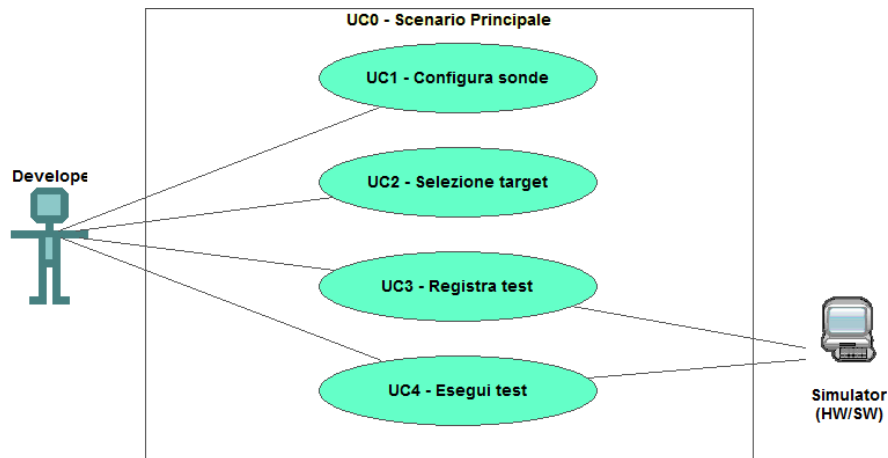


Figura 4.5: Use Case - UC4: invio e-mail

UC4: Automazione dell'invio di e-mail

Attori Principali: Programmatore, Sistema di posta elettronica.

Precondizioni: I siti web contenuti nel database hanno ricevuto una valutazione.

Descrizione: Il sistema invia automaticamente e-mail personalizzate ai proprietari dei siti con punteggi di qualità bassi, offrendo servizi di miglioramento.

Postcondizioni: Le e-mail vengono inviate con successo ai destinatari.

4.2 Tracciamento dei requisiti

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

Sono stati individuati diversi tipi di requisiti e si è quindi fatto utilizzo di un codice identificativo per distinguerli.

Il codice dei requisiti è così strutturato R(F/Q/V)(N/D/O) dove:

R = requisito

F = funzionale

Q = qualitativo

V = di vincolo

N = obbligatorio (necessario)

D = desiderabile

Z = opzionale

Nelle tabelle 4.1, 4.2 e 4.3 sono riassunti i requisiti e il loro tracciamento con gli use case delineati in fase di analisi.

Tabella 4.1: Tabella del tracciamento dei requisiti funzionali

Requisito	Descrizione	Use Case
RFN-1	L'interfaccia permette di configurare il tipo di sonde del test	UC1

Tabella 4.2: Tabella del tracciamento dei requisiti qualitativi

Requisito	Descrizione	Use Case
RQD-1	Le prestazioni del simulatore hardware deve garantire la giusta esecuzione dei test e non la generazione di falsi negativi	-

Tabella 4.3: Tabella del tracciamento dei requisiti di vincolo

Requisito	Descrizione	Use Case
RVO-1	La libreria per l'esecuzione dei test automatici deve essere riutilizzabile	-

Capitolo 5

Strumenti e tecnologie

L'obiettivo principale di questo capitolo è l'illustrazione delle tecnologie degli strumenti ausiliari utilizzati per raggiungere lo scopo finale del progetto

5.1 Strumenti

Gli strumenti di supporto adoperati per il progetto sono elencati nella lista seguente.

- Visual Studio Code (Fig. 5.1): è un ambiente di sviluppo integrato, disponibile per Linux, macOS e Windows. È un'applicazione che supporta la maggior parte dei linguaggi di programmazione ed è quindi molto vantaggiosa per lavorare ad un progetto multi-linguaggio senza dover cambiare ambiente. Un'altra delle funzionalità principali è la fornitura di numerose estensioni che semplificano il processo di scrittura e verifica del codice.

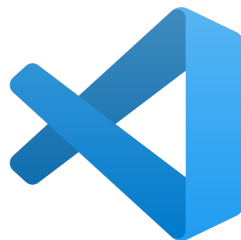


Figura 5.1: Logo di Visual Studio Code

- PhpMyAdmin (Fig. 5.2): è una web-app scritta utilizzando il linguaggio di programmazione PHP, che offre la capacità di gestione di un database MySQL attraverso un browser qualsiasi. Consente la creazione di tabelle, l'inserimento, la modifica e l'interrogazione dei dati. Fornisce un'interfaccia grafica per la visione d'insieme e per le operazioni amministrative.



Figura 5.2: Logo di phpMyAdmin

- Remote Ripple (Fig. 5.3): è un software per l'accesso remoto, che viene utilizzato dallo studente per avviare i programmi presenti nel server aziendale.



Figura 5.3: Logo di Remote Ripple

5.2 Tecnologie

Nelle sezioni seguenti viene data una spiegazione di tutte le tecnologie utilizzate.

5.2.1 Python

Versione: 3.9.0

Python è un linguaggio di programmazione ampiamente utilizzato in settori come il data mining e l'intelligenza artificiale. Offre solide basi per l'integrazione con disparati linguaggi, ma il vantaggio più di risalto è la presenza di numerose librerie che aiutano lo sviluppatore a velocizzare il processo di codifica; in particolare sono state utilizzate le seguenti librerie:

5.2.2 CUDA

Versione: 11.2

Descrizione Tecnologia 1

5.2.3 Docker

Versione: 27.2.0

Descrizione Tecnologia 2

5.2.4 DDEV

Versione: 1.23.2

Descrizione Tecnologia 3

5.2.5 WSL

Versione: 2.3.24.0

Descrizione Tecnologia 4

5.2.6 Laravel

Versione: 11.29.0

Descrizione Tecnologia 5

5.2.7 Filament

Versione: 3.2.121

Descrizione Tecnologia 6

5.2.8 MySQL

Versione: 8.0.36

Descrizione Tecnologia 7

5.3 Ciclo di vita del software

5.4 Progettazione

Namespace 1

Descrizione namespace 1.

Classe 1: Descrizione classe 1

Classe 2: Descrizione classe 2

5.5 Design Pattern utilizzati

5.6 Codifica

Capitolo 6

Progettazione e codifica

Capitolo 7

Conclusioni

7.1 Consuntivo finale

7.2 Raggiungimento degli obiettivi

7.3 Conoscenze acquisite

7.4 Valutazione personale

Appendice A

Appendice A

Citazione

Autore della citazione

Acronimi e abbreviazioni

UML [Unified Modeling Language](#). [13](#), [26](#)

Glossario

UML in ingegneria del software *UML*, *Unified Modeling Language* (ing. linguaggio di modellazione unificato) 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. Gran parte della letteratura di settore usa tale linguaggio per descrivere soluzioni analitiche e progettuali in modo sintetico e comprensibile a un vasto pubblico. [25](#)

Bibliografia

Riferimenti bibliografici

James P. Womack, Daniel T. Jones. *Lean Thinking, Second Editon*. Simon & Schuster, Inc., 2010.

Siti web consultati

Manifesto Agile. URL: <http://agilemanifesto.org/iso/it/>.