

Università degli Studi di Padova

DIPARTIMENTO DI MATEMATICA "TULLIO LEVI-CIVITA"

CORSO DI LAUREA IN INFORMATICA



Revisione dell'interfaccia grafica di un software CRM con l'uso di AngularJS

Tesi di laurea triennale

Relatore

Prof. Paolo Baldan

Laureando

Davide Stocco

ANNO ACCADEMICO 2017-2018

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

— Oscar Wilde

Dedicato a ...

Sommario

Il presente documento descrive il lavoro svolto durante il periodo di stage, della durata di circa trecento ore, dal laureando Pinco Pallino presso l'azienda Azienda S.p.A. Gli obbiettivi da raggiungere erano molteplici.

In primo luogo era richiesto lo sviluppo di ... In secondo luogo era richiesta l'implementazione di un ... Tale framework permette di registrare gli eventi di un controllore programmabile, quali segnali applicati Terzo ed ultimo obbiettivo era l'integrazione ...

“Life is really simple, but we insist on making it complicated”

— Confucius

Ringraziamenti

Innanzitutto, vorrei esprimere la mia gratitudine al Prof. NomeDelProfessore, relatore della mia tesi, per l'aiuto e il sostegno fornitomi durante la stesura del lavoro.

Desidero ringraziare con affetto i miei genitori per il sostegno, il grande aiuto e per essermi stati vicini in ogni momento durante gli anni di studio.

Ho desiderio di ringraziare poi i miei amici per tutti i bellissimi anni passati insieme e le mille avventure vissute.

Padova, 26 Settembre 2018

Daide Stocco

Elenco delle figure

1.1	Logo di Sanmarco Informatica	1
1.2	Logo di StageIT	2
4.1	Use Case - UC0: Scenario principale	9

Elenco delle tabelle

4.1	Tabella del tracciamento dei requisiti funzionali	11
4.2	Tabella del tracciamento dei requisiti qualitativi	11
4.3	Tabella del tracciamento dei requisiti di vincolo	11

Organizzazione del testo

Il primo capitolo descrive ...

Il secondo capitolo descrive ...

Il terzo capitolo approfondisce ...

Il quarto capitolo approfondisce ...

Il quinto capitolo approfondisce ...

Il sesto capitolo approfondisce ...

Nel settimo capitolo descrive ...

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

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

Capitolo 1

Introduzione

Introduzione al contesto applicativo.

Esempio di utilizzo di un termine nel glossario
[Application Program Interface \(API\)](#).

Esempio di citazione in linea
`site:agile-manifesto`.

1.1 L'azienda



Figura 1.1: Logo di Sanmarco Informatica

Sanmarco Informatica Spa è un'azienda italiana leader nella progettazione e realizzazione di soluzioni a supporto della riorganizzazione di tutti i processi aziendali e professionali.

Ha sede a Grisignano di Zocco (VI), è attiva da 35 anni, può contare su più di 400 dipendenti e collaboratori, per seguire quotidianamente più di 1000 aziende clienti.

Il punto di forza dell'azienda sta nella formazione del personale e nella ricerca, campi nei quali viene investito in media il 20% del fatturato annuo.

, Il prodotto di punta di Sanmarco Informatica è *JGalileo*, un software gestionale *ERP*^[g] utilizzato, oltre che da aziende europee, anche in USA, Russia e Cina.

Nel centro di Sviluppo dell'azienda viene adottato il metodo Agile: esso nasce nel 2001 e definisce i dodici principi fondamentali per sviluppare con tempistiche e procedure ottimizzate un software che sia per il cliente una soluzione di successo. La metodologia "Agile" privilegia *gli individui e le interazioni più che i processi e gli strumenti, il software funzionante più che la documentazione esaustiva, la collaborazione col cliente più che la negoziazione dei contratti, rispondere al cambiamento più che seguire un piano*.

L'obiettivo è quello di garantire all'azienda tutta la flessibilità e la versatilità necessaria alla realizzazione di progetti efficaci. Questo metodo di lavoro si concentra su una collaborazione più face-to-face fra colleghi con cadenza regolare e ravvicinata, permettendo di mantenere e perfezionare man mano il focus sui progetti, valorizzando i nuovi punti di vista e le nuove risposte alle problematiche da affrontare.

In particolare, il metodo di gestione progetti "SCRUM" prevede di dividere il progetto in blocchi (Sprint), all'interno dei quali vengono sviluppate delle funzioni/moduli/applikazioni complete (denominate storie) pronte per la potenziale installazione al cliente.

Il termine Scrum è mutuato dal termine del Rugby che indica il pacchetto di mischia ed è una metafora del team di sviluppo che deve lavorare insieme in modo che tutti gli attori del progetto spingano nella stessa direzione, agendo come un'unica entità coordinata.

1.2 L'offerta di Stage

Da anni l'azienda collabora con l'università di Padova alla ricerca di neolaureati da formare ed inserire nel proprio organico e, grazie ad iniziative come StageIt le possibilità di incontro tra studenti ed aziende sono molto più concrete e produttive.



Figura 1.2: Logo di StageIT

Gli studenti hanno infatti l'occasione di conoscere svariate realtà, anche piccole, che operano nel settore nei più disparati rami applicativi mentre per le aziende, oltre ad essere un'importante vetrina ed un momento di confronto con potenziali partner e concorrenti, costituisce anche la possibilità di conoscere tanti giovani studenti in un breve arco temporale.

Tra le varie proposte di stage che quest'azienda offriva, mi è stato proposto di collaborare a quello riguardante il loro software CRM.

Un software CRM, acronimo di Customer Relationship Management, è un prodotto che permette all'utilizzatore di mantenere una rete di relazioni con clienti e potenziali clienti, non soltanto al fine di mantenere una relazione commerciale ma di fiducia reciproca che si possa protrarre nel tempo, anche attraverso l'impiego di strumenti di fidelizzazione come newsletters, campagne ed eventi.

In questo contesto si inserisce lo stage cui ho preso parte e che aveva come obiettivo la ridefinizione della form principale di tale software, dapprima realizzato in *Java* con traduzione in *JavaScript* a cura della libreria *GWT* ed ora richiesto in *JavaScript* nativo, utilizzando una libreria da definire in fase di svolgimento dello stage dopo

un'adeguata analisi delle alternative open-source disponibili sul mercato¹.

Capitolo 2

Processi e metodologie

Brevissima introduzione al capitolo

2.1 Processo sviluppo prodotto

Capitolo 3

Descrizione dello stage

Breve introduzione al capitolo

3.1 Introduzione al progetto

3.2 Analisi preventiva dei rischi

Durante la fase di analisi iniziale sono stati individuati alcuni possibili rischi a cui si potrà andare incontro. Si è quindi proceduto a elaborare delle possibili soluzioni per far fronte a tali rischi.

1. Performance del simulatore hardware

Descrizione: le performance del simulatore hardware e la comunicazione con questo

¹queste ed altre tecnologie verranno discusse nel dettaglio nei capitoli seguenti.

potrebbero risultare lenti o non abbastanza buoni da causare il fallimento dei test.

Soluzione: coinvolgimento del responsabile a capo del progetto relativo il simulatore hardware.

3.3 Requisiti e obiettivi

3.4 Pianificazione

Capitolo 4

Analisi dei requisiti

Breve introduzione al capitolo

4.1 Casi d'uso

Per lo studio dei casi di utilizzo del prodotto sono stati creati dei diagrammi. I diagrammi dei casi d'uso (in inglese *Use Case Diagram*) sono diagrammi di tipo [Unified Modeling Language \(UML\)](#) dedicati alla descrizione delle funzioni o servizi offerti da un sistema, così come sono percepiti e utilizzati dagli attori che interagiscono col sistema stesso. Essendo il progetto finalizzato alla creazione di un tool per l'automazione di un processo, le interazioni da parte dell'utilizzatore devono essere ovviamente ridotte allo stretto necessario. Per questo motivo i diagrammi d'uso risultano semplici e in numero ridotto.

UC0: Scenario principale

Attori Principali: Sviluppatore applicativi.

Precondizioni: Lo sviluppatore è entrato nel plug-in di simulazione all'interno dell'I-DE.

Descrizione: La finestra di simulazione mette a disposizione i comandi per configurare, registrare o eseguire un test.

Postcondizioni: Il sistema è pronto per permettere una nuova interazione.

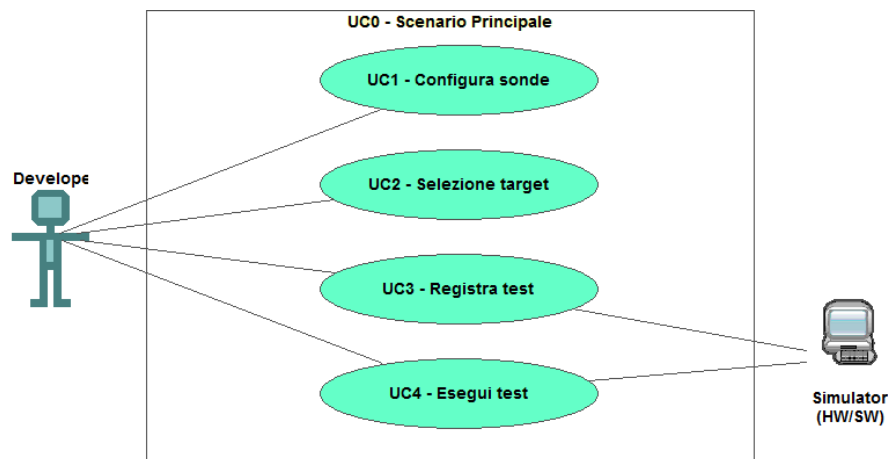


Figura 4.1: Use Case - UC0: Scenario principale

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	-

Capitolo 5

Progettazione e codifica

Breve introduzione al capitolo

5.1 Tecnologie e strumenti

Di seguito viene data una panoramica delle tecnologie e strumenti utilizzati.

Tecnologia 1

Descrizione Tecnologia 1.

Tecnologia 2

Descrizione Tecnologia 2

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	-

5.2 Ciclo di vita del software

5.3 Progettazione

Namespace 1

Descrizione namespace 1.

Classe 1: Descrizione classe 1

Classe 2: Descrizione classe 2

5.4 Design Pattern utilizzati

5.5 Codifica

Capitolo 6

Verifica e validazione

Capitolo 7

Conclusioni

7.1 Consuntivo finale

7.2 Raggiungimento degli obiettivi

7.3 Conoscenze acquisite

7.4 Valutazione personale

Appendice A

Appendice A