

Università degli Studi di Padova

DIPARTIMENTO DI MATEMATICA "TULLIO LEVI-CIVITA"

CORSO DI LAUREA IN INFORMATICA



Tecnologie web per lo sviluppo di
applicazioni mobile ibride

Tesi di laurea

Relatore

Prof. Paolo Baldan

Laureando

Nicola Salvatore

ANNO ACCADEMICO 2019-2020

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 Salvatore Nicola presso l'azienda Alternative Studio di Costa Francesco.

L'obiettivo dello stage è stato il rifacimento della applicazione per smartphone a disposizione di UCIS (Unità Cinofile Italiane da Soccorso), che si occupa di registrare la geolocalizzazione delle unità cinofile durante esercitazioni, addestramenti, e anche operazioni di emergenza.

“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, Settembre 2020

Nicola Salvatore

Indice

| | | |
|----------|---|-----------|
| 1 | Introduzione | 1 |
| 1.1 | Convenzioni tipografiche | 1 |
| 1.2 | Scopo del documento | 1 |
| 1.2.1 | Organizzazione del testo | 1 |
| 1.3 | L'azienda | 1 |
| 1.4 | L'idea | 2 |
| 1.5 | Tecnologie utilizzate | 2 |
| 1.5.1 | Ionic | 2 |
| 1.5.2 | Angular | 3 |
| 1.5.3 | Android e Java | 3 |
| 1.6 | Metodo di Lavoro | 3 |
| 1.6.1 | Processi di sviluppo | 3 |
| 1.6.2 | Strumenti di supporto allo sviluppo | 4 |
| 2 | Processi e metodologie | 7 |
| 2.1 | Processo sviluppo prodotto | 7 |
| 3 | Descrizione dello stage | 9 |
| 3.1 | Introduzione al progetto | 9 |
| 3.2 | Analisi preventiva dei rischi | 9 |
| 3.3 | Requisiti e obiettivi | 9 |
| 3.4 | Pianificazione | 9 |
| 4 | Analisi dei requisiti | 11 |
| 4.1 | Casi d'uso | 11 |
| 4.2 | Tracciamento dei requisiti | 12 |
| 5 | Progettazione e codifica | 15 |
| 5.1 | Tecnologie e strumenti | 15 |
| 5.2 | Ciclo di vita del software | 15 |
| 5.3 | Progettazione | 15 |
| 5.4 | Design Pattern utilizzati | 15 |
| 5.5 | Codifica | 15 |
| 6 | Verifica e validazione | 17 |
| 7 | Conclusioni | 19 |
| 7.1 | Consuntivo finale | 19 |
| 7.2 | Raggiungimento degli obiettivi | 19 |

| | | |
|----------|---------------------------------|-----------|
| 7.3 | Conoscenze acquisite | 19 |
| 7.4 | Valutazione personale | 19 |
| A | Appendice A | 21 |
| | Acronyms | 25 |
| | Glossary | 27 |
| | Bibliografia | 29 |

Elenco delle figure

| | | |
|-----|---|----|
| 1.1 | Logo di WebStorm | 5 |
| 1.2 | Logo di Android Studio | 5 |
| 4.1 | Use Case - UC0: Scenario principale | 11 |

Elenco delle tabelle

| | | |
|-----|--|----|
| 4.1 | Tabella del tracciamento dei requisiti funzionali | 13 |
| 4.2 | Tabella del tracciamento dei requisiti qualitativi | 13 |
| 4.3 | Tabella del tracciamento dei requisiti di vincolo | 13 |

Capitolo 1

Introduzione

1.1 Convenzioni tipografiche

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*.

1.2 Scopo del documento

Qui devo scrivere i contenuti di ciò che dirò.

1.2.1 Organizzazione del testo

Il secondo capitolo descrive ...

Il terzo capitolo approfondisce ...

Il quarto capitolo approfondisce ...

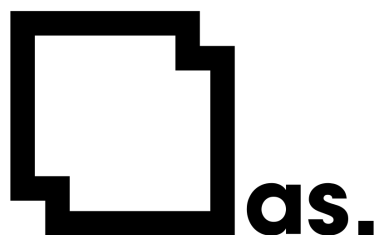
Il quinto capitolo approfondisce ...

Il sesto capitolo approfondisce ...

Nel settimo capitolo descrive ...

1.3 L'azienda

Alternative Studio è una web agency che fornisce soluzioni professionali su misura, costruite secondo le esigenze del cliente. Si occupa principalmente di sviluppo web e marketing. È un'azienda piccola che raccoglie poche risorse umane, ma molte energie



che continuano a spingere per crescere. Opera da appena sei anni nel settore del web development, ma ha abbracciato anche altre iniziative, collaborando in progetti più grandi con altre realtà. Negli ultimi anni l'azienda si è cimentata nello sviluppo di un gestionale per l'organizzazione [Unità Cinofile da Soccorso \(UCIS\)](#). Quest ultimo possiede un'[Application Program Interface \(API\)](#) volta alla ricezione e all'elaborazione di attività registrate durante addestramento, soccorso o esercitazioni.

1.4 L'idea

L'attuale applicazione per smartphone che si occupa di registrare le attività succitate è stata sviluppata molti anni fa ed essendo molto instabile è nata l'esigenza di un refactoring di quest'ultima. Volendo utilizzare tecnologie moderne per lo sviluppo mobile e crossplatform è nata l'esigenza di un'indagine preventiva sulle tecnologie da utilizzare.

Alternative Studio ha pensato, quindi, di aprire una posizione perfetta per un laureando, che cerchi un progetto formante, che lo metta continuamente alla prova.

1.5 Tecnologie utilizzate

Lo stage si è focalizzato sull'utilizzo di tecnologie mobile per lo sviluppo di applicazioni per smartphone. Queste sono spesso nuove e in continua evoluzione, è per questo che Alternative Studio ha avviato uno studio approfondito per scegliere quelle adatte allo sviluppo della propria applicazione.

1.5.1 Ionic

[Ionic](#) è un [framework](#) open source che fornisce strumenti agli sviluppatori, come librerie grafiche e plugins nativi per dialogare con [Android](#) e [iOS](#). Esso permette lo sviluppo mobile utilizzando tecnologie standard web come Angular, React e Vue, evitando lo sviluppo in linguaggio nativo dei singoli sistemi operativi. Questo è reso possibili da wrapper esistenti su ogni piattaforma volti a eseguire l'applicazione.

1.5.2 Angular

Come framework web per lo sviluppo [frontend](#) si è utilizzato Angular, evoluzione del noto AngularJS, sviluppato in prevalenza da Google, ma con distribuzione [open source](#). Le applicazioni Angular sono eseguite direttamente lato client dal web browser e quindi non vengono reinviolate indietro al web server. Inoltre sono nativamente responsive, cioè i toolkit utilizzate si adattano al dispositivo sul quale sono eseguite.

Nonostante l'esperienza di Alternative Studio sull'analogo Vue, si è deciso di adottare Angular dato che Ionic-Vue era ancora in fase di beta e quindi potenzialmente instabile e soggetto a cambiamenti.

TypeScript

TypeScript è un linguaggio implementato da Microsoft nel 2012, derivato da JavaScript, al quale aggiunge il concetto di tipizzazione e di orientamento agli oggetti. Nonostante JS sia un linguaggio [Tipizzato](#), esso non fa nessun tipo di controllo statico sui tipi effettuando sempre una conversione dinamica. Questo produce spesso degli errori difficili da trovare e correggere, per questo TypeScript introduce la compilazione che non fa altro che tradurre il codice in JavaScript, eseguendo prima un controllo dei tipi. Grazie a queste caratteristiche TypeScript non è stato utilizzato solo per il [frontend](#), ma anche per parte della logica dell'applicazione. Per questo e per il fatto che è il linguaggio adottato nativamente da Angular sarà quello più utilizzato durante lo sviluppo.

1.5.3 Android e Java

[Android](#) è il sistema operativo mobile più diffuso nel pianeta e di proprietà di [Google](#). È stato scelto come riferimento durante lo sviluppo, nonostante [Ionic](#) offra la possibilità di sviluppare in crossplatform con lo stesso codice. Al contrario alcune funzionalità che si interfacciano con il sistema operativo, come ad esempio la componente GPS, devono essere sviluppate in linguaggio nativo, che nel caso di [Android](#) è [Java](#).

1.6 Metodo di Lavoro

Dato le contenute risorse umane a disposizione Alternative Studio adotta un ciclo di sviluppo software [Incrementale](#) con qualche introduzione di processi da quello [Agile](#), in particolare dal metodo [Scrum](#). Durante lo stage lo studente è stato incaricato, anche di introdurre qualche concetto di questi cicli di vita all'interno del contesto aziendale, come quello di [Sprint](#) e di [Daily Scrum](#).

1.6.1 Processi di sviluppo

La mia figura è subentrata durante la fase di Analisi dei requisiti. Per questo i primi compiti affidatomi sono stati quelli di studio delle tecnologie adatte da utilizzare durante il progetto. Durante questa fase si sono svolte alcune riunioni con il tutor, tramite videochiamata, e redatto alcuni documenti di report su quest'ultime. Inoltre il lavoro individuale (come compilare un "Hello World" con Ionic) si è svolto tramite tasks. Le riunioni con il tutor sono proseguite anche durante la fase di progettazione architetturale, durante la quale abbiamo chiarito la visione generale dell'applicazione.

1.6.2 Strumenti di supporto allo sviluppo

Ci sono alcuni software da citare utilizzati nella gestione del progetto.

Gestione progetto e Versione

Gitlab è un software open source per la gestione di repository [Git](#) e supporto alla Continuous Integration.

Per la prima parte è stato importante il suo strumento di [Issue Tracking System](#) con il quale si sono gestite le tasks e le loro scadenze. Infatti, in questo caso le issues sono state utilizzate come metodo per tracciare i compiti da svolgere, piuttosto che per segnalare problemi all'interno del software, come bug e affini.

Ogni task possiede un titolo significativo, una descrizione approfondita aggiornabile in caso di cambiamenti durante l'esecuzione e una scadenza. Inoltre è possibile assegnarla a più membri del team, aggiungere informazioni o dialogare con il tutor attraverso la sezione commenti e specificare la milestone alla quale è collegata.

Comunicazione

Causa il telelavoro, durante il progetto sono stati fondamentali gli strumenti di comunicazione:

- * Slack: per le comunicazioni ufficiali e come strumento di notifica per i cambiamenti nella repo.
- * Telegram: come servizio di messaggistica istantanea, diretta con il tutor.
- * Skype: per le videochiamate, essenziale per lo svolgersi delle riunioni a distanza.

Codifica

In dotazione ad Alternative Studio ho utilizzato WebStorm, potente IDE parte della suite di JetBrains. Una delle sue caratteristiche principali sono la sua modularità, grazie allo store di plugins disponibili che aggiungono funzionalità, come il conteggio delle ore di programmazione e gestione del repository [Git](#) locale.

Inoltre si integra molto bene con le tecnologie web, come Angular, mettendo a disposizione il suo ambiente di building e di testing direttamente all'interno dell'IDE.



Figura 1.1: Logo di WebStorm

Android Studio

Android Studio è un IDE disegnato appositamente per lo sviluppo Android. Esso deriva da IntelliJ IDEA, software molto diffuso prodotto dal team JetBrains, come lo stesso WebStorm utilizzato.



Figura 1.2: Logo di Android Studio

Capitolo 2

Processi e metodologie

Brevissima introduzione al capitolo

2.1 Processo sviluppo prodotto

Per

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 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.

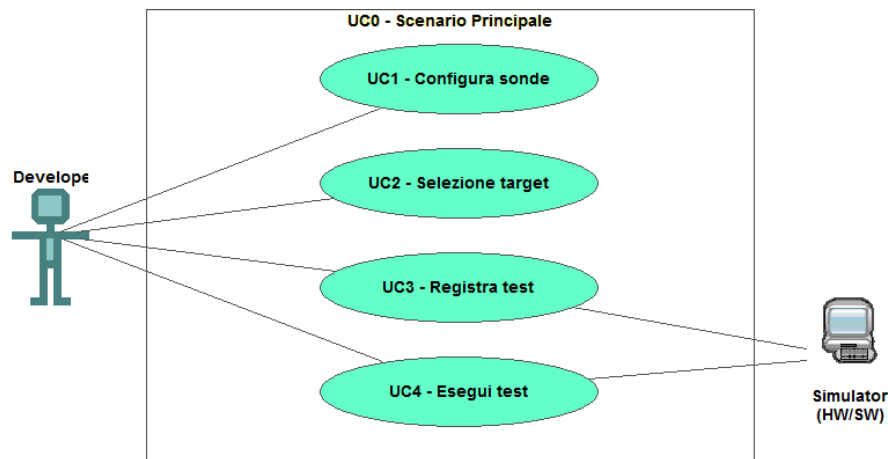


Figura 4.1: Use Case - UC0: Scenario principale

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.

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

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

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

Citazione

Autore della citazione

Acronimi

API [Application Programming Interface](#). 2, 23, 27

ASD [Alternative Studio](#). 23

UCIS [Unità Cinofile da Soccorso](#). 2, 23, 28

UML [Unified Modeling Language](#). 11, 23, 27

XML [eXtensible Markup Language](#). 23, 27

Glossario

API in informatica con il termine *Application Programming Interface API* (ing. interfaccia di programmazione di un'applicazione) si indica ogni insieme di procedure disponibili al programmatore, di solito raggruppate a formare un set di strumenti specifici per l'espletamento di un determinato compito all'interno di un certo programma. La finalità è ottenere un'astrazione, di solito tra l'hardware e il programmatore o tra software a basso e quello ad alto livello semplificando così il lavoro di programmazione. [23](#), [25](#)

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. [23](#), [25](#)

XML FGGSDF . [23](#), [25](#)

Agile FGGSDF . [3](#), [23](#)

Android FGGSDF . [2](#), [3](#), [23](#)

Daily Scrum FGGSDF . [3](#), [23](#)

framework FGGSDF . [2](#), [23](#)

frontend FGGSDF . [3](#), [23](#)

Git FGGSDF . [4](#), [5](#), [23](#)

Google FGGSDF . [3](#), [23](#)

Incrementale FGGSDF . [3](#), [23](#)

Ionic FGGSDF . [2](#), [3](#), [23](#)

iOS FGGSDF . [2](#), [23](#)

Issue Tracking System dfasdga . [4](#), [23](#)

Java fsdf s sadf . [3](#), [23](#)

open source FGGSDF . [3](#), [23](#)

Scrum FGGSDF . [3](#), [23](#)

Sprint FGGSDF . [3](#), [23](#)

Tipizzato fasdfa . [3](#), [23](#)

UCIS **UCIS** È un'Associazione Nazionale di Volontariato, inserita nell'Albo istituito presso il Dipartimento di Protezione Civile. Raggruppa, tutela e coordina i Soccorritori Cinofili presenti sul Territorio Nazionale. [23](#), [25](#)

UCIS Report Tool Applicazione in dotazione attualmente a UCIS sviluppata 5 anni fa e pubblicata nel 2018. [23](#)

Bibliografia

Riferimenti bibliografici

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

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

Siti web consultati

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

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