

**Università degli Studi di Padova**

DIPARTIMENTO DI MATEMATICA "TULLIO LEVI-CIVITA "

CORSO DI LAUREA IN INFORMATICA



**Sviluppo di un'applicazione mobile per la  
gestione di eventi sportivi tramite  
framework Flutter**

*Tesi di laurea triennale*

*Relatore*

Prof. Luigi De Giovanni

*Laureando*

Matteo Budai

---

ANNO ACCADEMICO 2020-2021



# Sommario

Il presente documento descrive lo stage da me svolto nel periodo che va dal 28/06/2021 al 20/08/2021, della durata di trecentoventi ore, presso l'azienda Sync Lab s.r.l. nella sede di Padova.

Lo stage riguarda la realizzazione di varie funzionalità per un'applicazione denominata 'Sportwill' che permette la gestione di eventi sportivi.

Gli obiettivi da raggiungere erano molteplici.

In primo luogo era richiesto il ripasso del linguaggio Java SE e dei concetti Web come Servlet, servizi Rest e Json. In secondo luogo era richiesto lo studio dei principi generali, delle best practice, dei widget e dell'architettura di Flutter e lo studio del linguaggio Dart.

In seguito si è passati allo studio del codice esistente dell'applicazione e allo sviluppo di varie funzionalità che hanno permesso di completarla rendendola utilizzabile.

Il seguente documento è stato diviso in 5 capitoli:

- **Capitolo 1:** Descrizione dell'azienda e delle metodologie utilizzate;
- **Capitolo 2:** Presentazione degli obiettivi, del Piano di Lavoro e delle attività svolte con introduzione al progetto;
- **Capitolo 3:** Descrizione del linguaggio Dart e del framework Flutter e presentazione di alcune piccole applicazioni realizzate per lo studio;
- **Capitolo 4:** Descrizione dettagliata dell'applicazione esistente e delle nuove funzionalità apportate;
- **Capitolo 5:** Resoconto conclusivo con valutazione del percorso svolto.



# Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
1.1	L'azienda . . . . .	1
1.2	Metodologie utilizzate e principali prodotti . . . . .	2
<b>2</b>	<b>Descrizione dello stage e obiettivi</b>	<b>5</b>
2.1	Introduzione al progetto e scopo . . . . .	5
2.2	Obiettivi dello stage . . . . .	5
2.3	Pianificazione del lavoro svolto . . . . .	6
2.3.1	Pianificazione iniziale . . . . .	6
2.3.2	Variazioni . . . . .	8
2.4	Strumenti di comunicazione per lo svolgimento del lavoro . . . . .	10
<b>3</b>	<b>Framework Flutter e linguaggio Dart</b>	<b>11</b>
3.1	Sviluppo applicazioni mobile . . . . .	11
3.1.1	App native . . . . .	12
3.1.2	Web app . . . . .	13
3.1.3	App Ibride Web View Wrapper . . . . .	14
3.1.4	App Ibride Compile to Native . . . . .	14
3.2	Flutter . . . . .	15
3.2.1	Dart . . . . .	15
3.2.2	Componenti . . . . .	15
3.2.3	Widget . . . . .	15
3.2.4	Rendering e Layout . . . . .	15
3.2.5	Esempi piccole applicazioni realizzate . . . . .	15
<b>4</b>	<b>Sportwill</b>	<b>17</b>
4.1	Descrizione progetto . . . . .	18
4.2	Analisi dei requisiti . . . . .	18
4.3	Tecnologie e strumenti . . . . .	18
4.3.1	Flutter e Dart . . . . .	18
4.3.2	Android Studio . . . . .	18
4.3.3	GitLab . . . . .	18
4.3.4	Backend . . . . .	18
4.4	Implementazioni . . . . .	18
4.4.1	Filtro di ricerca testo . . . . .	18
4.4.2	Logo . . . . .	18
4.4.3	Pagina Modifica e campi obbligatori . . . . .	18
4.4.4	Mappa percorso . . . . .	18

4.4.5	Aggiornamento automatico mappa . . . . .	18
4.4.6	Mappa schermo intero . . . . .	18
4.4.7	Colori . . . . .	18
4.4.8	Logo . . . . .	18
4.4.9	Pagina Modifica . . . . .	18
4.4.10	Eliminazione, Modifica e Aggiunta di un'attività . . . . .	18
4.4.11	Filtro avanzato di ricerca attività . . . . .	18
<b>5</b>	<b>Conclusioni</b>	<b>19</b>
5.1	Consuntivo finale . . . . .	19
5.2	Raggiungimento degli obiettivi . . . . .	19
5.3	Conoscenze acquisite . . . . .	19
5.4	Valutazione personale . . . . .	19
	<b>Bibliografia</b>	<b>23</b>

# Elenco delle figure

1.1	Logo aziendale Sync Lab . . . . .	1
1.2	Sedi Sync Lab . . . . .	2
1.3	SynClinic . . . . .	3
2.1	Trello . . . . .	10
2.2	Google sheets . . . . .	10
3.1	Sviluppo applicazioni mobile . . . . .	11
3.2	Logo Android . . . . .	12
3.3	Logo iOS . . . . .	13
3.4	Logo Windows Phone . . . . .	13
3.5	Logo Flutter . . . . .	15

# Elenco delle tabelle

2.1	Tabella riassuntiva della pianificazione iniziale di stage . . . . .	8
2.2	Tabella riassuntiva della pianificazione di stage con variazioni . . . . .	9





# Capitolo 1

## Introduzione

*In questo capitolo viene descritta l'azienda, le metodologie utilizzate e come viene organizzato il lavoro.*

### 1.1 L'azienda

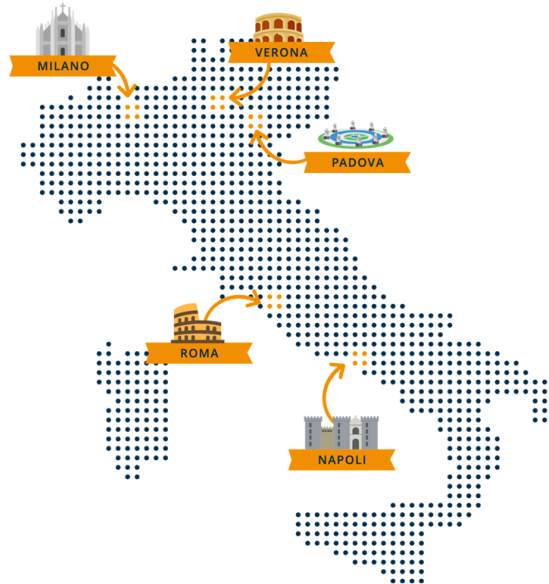
Sync Lab nasce nel 2002 come Software house e si è trasformata rapidamente in System Integrator attraverso uno studiato processo di maturazione delle competenze tecnologiche, metodologiche ed applicative nel dominio del software.



**Figura 1.1:** Logo aziendale Sync Lab

In seguito all'apertura della sede principale di Napoli, Sync Lab è cresciuta esponenzialmente nel mercato ICT e ha consolidato ottimi rapporti con clienti e partner. Attualmente l'azienda ha più di 150 clienti diretti e finali e vanta un organico di oltre

200 dipendenti, una solida base finanziaria e un'ottima diffusione nel territorio italiano attraverso le sue cinque sedi: Napoli, Roma, Milano, Padova e Verona.



**Figura 1.2:** Sedi Sync Lab

Sync Lab, propone sul mercato interessanti e innovativi prodotti software, nati nel proprio laboratorio di ricerca e sviluppo. Attraverso questi prodotti, Sync Lab ha gradualmente conquistato significativamente fette di mercato nei seguenti settori: mobile, videosorveglianza e sicurezza delle infrastrutture informatiche aziendali.

## 1.2 Metodologie utilizzate e principali prodotti

L'azienda adotta un modello di sviluppo agile che pone le proprie basi nel metodo Scrum. Gli stakeholders, infatti, vengono costantemente coinvolti nel processo di sviluppo del prodotto per raccogliere feedback. Gli obiettivi si possono riassumere in tre punti fondamentali:

- Comprendere attentamente il contesto operativo del cliente;
- Fornire al cliente un supporto mirato;
- Accelerare e favorire la formazione di soluzioni.

In base a questi principi Sync Lab raggiunge i propri obiettivi grazie a:

- Consulenza;
- Fornitura;
- Sviluppo;

- Manutenzione.

Nell'ambito di prodotti e innovazioni, l'azienda ne può vantare un buon numero. Tra questi troviamo:

- **SynClinic** che è un software integrato per la gestione delle strutture sanitarie, che permette di gestire, organizzare e monitorare tutte le fasi del percorso di cura del paziente;



**Figura 1.3:** SynClinic

- **DPS 4.0** che permette di gestire la General Data Protection Regulation (GDPR) Privacy in pochi semplici passi con una soluzione guidata per aggiornare e modificare i documenti di privacy in modo conforme agli standard di riferimento;
- **StreamLog** che permette di gestire la compliance al provvedimento del Garante per la protezione dei dati personali relativo agli Amministratori di Sistema (AdS). In particolare, permette di soddisfare requisiti fissati dal Garante;
- **StreamCrusher** che è una tecnologia che aiuta ad essere bene informati su quando bisogna prendere decisioni di business, ad identificare velocemente criticità ed a riorganizzare i processi in base a nuove esigenze;
- **Wave** che si propone come integrazione tra i mondi della Videosorveglianza e quello dei Sistemi Informativi Territoriali (GIS) abilitando il controllo totale dell'area da sorvegliare;
- **Seastream** che mette a disposizione un sistema di monitoraggio avanzato delle flotte armatoriali operative in tutto il mondo e una piattaforma integrata di servizi per gli operatori in ambito portuale.



## Capitolo 2

# Descrizione dello stage e obiettivi

*In questo capitolo viene introdotto il progetto e viene descritto come è stato organizzato il lavoro in azienda con gli obiettivi iniziali e le variazioni rispetto a quanto pianificato inizialmente.*

### 2.1 Introduzione al progetto e scopo

Lo scopo del progetto di stage è lo sviluppo di una piattaforma web-mobile per la gestione di eventi sportivi. È stata effettuata una fase iniziale di analisi e progettazione, basata sull'utilizzo del framework Flutter in linguaggio Dart, seguita dalla realizzazione di alcune parti dell'interfaccia mobile. Regolarmente, ci sono stati incontri diretti con il tutor aziendale Fabio Pallaro per verificare lo stato di avanzamento, chiarire eventualmente gli obiettivi, affinare la ricerca e aggiornare il piano di lavoro.

### 2.2 Obiettivi dello stage

Come obiettivi è stato richiesto di:

1. Realizzare le funzionalità indicate;
2. Produrre un documento Tecnico che descriva le funzionalità realizzate;
3. Rilasciare il codice sul repository che verrà indicato dall'azienda.

### Notazione

Per gli obiettivi delle stage si farà riferimento ai requisiti secondo le seguenti notazioni:

- *O* per i requisiti obbligatori, vincolanti in quanto obiettivo primario richiesto dal committente;
- *D* per i requisiti desiderabili, non vincolanti o strettamente necessari, ma dal riconoscibile valore aggiunto;

- $F$  per i requisiti facoltativi, rappresentanti valore aggiunto non strettamente competitivo.

Le sigle precedentemente indicate saranno seguite da una coppia sequenziale di numeri, identificativo del requisito.

### Obiettivi fissati

Si prevede lo svolgimento dei seguenti obiettivi:

- Obbligatori
  - O01: Acquisizione delle competenze sulle tematiche sopra descritte e sulle attività svolte;
  - O02: Capacità di raggiungere gli obiettivi richiesti in autonomia seguendo il cronoprogramma;
  - O03: Portare a termine le implementazioni previste con una percentuale di superamento pari all'80%.
- Desiderabili
  - D01: Portare a termine le implementazioni previste con una percentuale di superamento pari al 100%.
- Facoltativi
  - F01: Realizzazione di una nuova funzionalità per l'app che prevede la gestione Signin con il protocollo Oath2.

## 2.3 Pianificazione del lavoro svolto

In questa sezione viene mostrato come è stato il pianificato il lavoro e le variazioni apportate in seguito agli incontri regolari con il tutor aziendale Fabio Pallaro.

### 2.3.1 Pianificazione iniziale

All'inizio è stata fatta una pianificazione iniziale basata su 8 settimane con una pianificazione di 40 ore per ciascuna settimana per un totale di 320 ore. La pianificazione iniziale è la seguente:

- **Prima Settimana (40 ore)**
  - Presentazione strumenti di lavoro per la condivisione del materiale di studio e per la gestione dell'avanzamento;
  - Condivisione scaletta di argomenti;
  - Ripasso del linguaggio Java SE;

- Ripasso concetti Web (Servlet, servizi Rest, Json ecc.).
- **Seconda Settimana (40 ore)**
  - Studio principi generali di Flutter e linguaggio Dart;
  - Studio delle best practice Flutter;
  - Studio delle architetture per i services in Flutter.
- **Terza Settimana (40 ore)**
  - Studio dei widget in Flutter;
  - Studio del prototipo di app SportWill oggi esistente.
- **Quarta Settimana (40 ore)**
  - Implementazione del login/signup con salvataggio credenziali su localStorage con aggiunta di foto ed altre info dell'utente corrente;
  - Modifica dell'interfaccia grafica nella visualizzazione 'Elenco Uscite' con filtri di ricerca.
- **Quinta Settimana (40 ore)**
  - Modifica funzionalità 'Modifica Uscita';
  - Implementazione della funzionalità 'Inserisci mappa percorso';
  - Implementazione della funzionalità 'Uscita in esecuzione/archiviata'.
- **Sesta Settimana (40 ore)**
  - Implementazione della funzionalità 'WaitForMe' per le uscite in esecuzione.
- **Settima Settimana (40 ore)**
  - Termine implementazione funzionalità 'WaitForMe'.
- **Ottava Settimana (40 ore)**
  - Termine integrazioni e collaudo finale.

La pianificazione iniziale, in termini di quantità di ore di lavoro, è stata così distribuita:

Durata in ore	Descrizione dell'attività
<b>40</b>	<b>Formazione sulle tecnologie</b>
<b>80</b>	<b>Definizione architettura di riferimento e relativa documentazione</b>
13	<i>Studio principi generali di Flutter e linguaggio Dart</i>
13	<i>Studio delle best practice Flutter</i>
14	<i>Studio delle architetture per i services in Flutter</i>
20	<i>Studio dei widget in Flutter</i>
20	<i>Studio del prototipo di app SportWill oggi esistente</i>
<b>160</b>	<b>Implementazioni</b>
20	<i>Implementazione del login/signup con salvataggio credenziali su localStorage con aggiunta di foto ed altre info dell'utente corrente</i>
20	<i>Modifica dell'interfaccia grafica nella visualizzazione 'Elenco Uscite' con filtri di ricerca</i>
10	<i>Modifica funzionalità 'Modifica Uscita'</i>
15	<i>Implementazione della funzionalità 'Inserisci mappa percorso'</i>
15	<i>Implementazione della funzionalità 'Uscita in esecuzione/archiviata'</i>
40	<i>Implementazione della funzionalità 'WaitForMe' per le uscite in esecuzione.</i>
40	<i>Termine implementazione funzionalità 'WaitForMe'</i>
<b>40</b>	<b>Termine integrazioni e collaudo finale.</b>
<b>Totale ore: 320</b>	

**Tabella 2.1:** Tabella riassuntiva della pianificazione iniziale di stage

### 2.3.2 Variazioni

Rispetto all'inizio sono variate solamente le funzionalità da realizzare in quanto quelle già presenti sono state suddivise con un altro stagista. Le variazioni in particolar modo sono dalla quarta alla settima settimana e sono le seguenti:

- **Quarta Settimana (40 ore)**

- Configurazione iniziale;
- Modifica logo;
- Modifica dell'interfaccia grafica nella visualizzazione 'Elenco Uscite' con filtri di ricerca.

- **Quinta Settimana (40 ore)**

- Modifica funzionalità 'Modifica Uscita' con vari fix e cambio colori;
- Implementazione della funzionalità 'Visualizza mappa percorso'.



- **Sesta Settimana (40 ore)**

- Implementazione della funzionalità che permette di vedere la mappa a schermo intero;
- Implementazione della funzionalità che permette l'aggiornamento automatico della mappa.

- **Settima Settimana (40 ore)**

- Modifica dell'interfaccia grafica nella visualizzazione 'Elenco Uscite' con filtri di ricerca avanzati.

La pianificazione, in termini di quantità di ore di lavoro dopo la variazione è stata così distribuita:

Durata in ore	Descrizione dell'attività
<b>40</b>	<b>Formazione sulle tecnologie</b>
<b>80</b>	<b>Definizione architettura di riferimento e relativa documentazione</b>
13	<i>Studio principi generali di Flutter e linguaggio Dart</i>
13	<i>Studio delle best practice Flutter</i>
14	<i>Studio delle architetture per i services in Flutter</i>
20	<i>Studio dei widget in Flutter</i>
20	<i>Studio del prototipo di app SportWill oggi esistente</i>
<b>160</b>	<b>Implementazioni</b>
15	<i>Configurazione iniziale</i>
5	<i>Modifica logo</i>
20	<i>Modifica dell'interfaccia grafica nella visualizzazione 'Elenco Uscite' con filtri di ricerca</i>
10	<i>Modifica funzionalità 'Modifica Uscita' con vari fix e cambio colori</i>
30	<i>Implementazione della funzionalità 'Visualizza mappa percorso'</i>
10	<i>Implementazione della funzionalità che permette di vedere la mappa a schermo intero</i>
30	<i>Implementazione della funzionalità che permette l'aggiornamento automatico della mappa.</i>
40	<i>Modifica dell'interfaccia grafica nella visualizzazione 'Elenco Uscite' con filtri di ricerca avanzati.</i>
<b>40</b>	<b>Termine integrazioni e collaudo finale.</b>
<b>Totale ore: 320</b>	

**Tabella 2.2:** Tabella riassuntiva della pianificazione di stage con variazioni

## 2.4 Strumenti di comunicazione per lo svolgimento del lavoro

Per ottimizzare lo svolgimento del lavoro sono stati usati i seguenti strumenti di comunicazione:

- **Trello** che mi ha permesso di organizzare e di gestire il progetto tramite la creazione di una bacheca condivisibile;

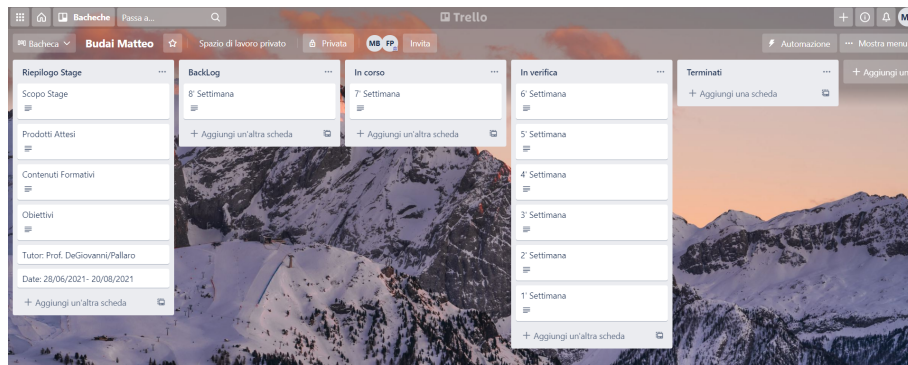


Figura 2.1: Trello

- **Google sheets** in cui venivano riportate ogni giorno le attività che si andavano a svolgere. Veniva segnalato con una spunta se l'attività era corretta oppure nella colonna aggiustamenti veniva spiegato in che modo o cosa svolgere;

A	B	C	D	E	F
DATA	DESCRIZIONE		AGGIUSTAMENTI		
28/06/2021	Ripasso Java del corso "altri paradigmi di programmazione"	✓			
29/06/2021	Fine ripasso Java	✓			
30/06/2021	Ripasso Concetti Web/JSON, servlet e REST	✓			
01/07/2021	Corso flutter Udemy(studio basi Dart)	✓			
02/07/2021	Corso flutter Udemy(studio basi Dart)	✓			
05/07/2021	Corso flutter Udemy(best Practice e architettura)	✓			
06/07/2021	Corso flutter Udemy(best Practice e architettura)	✓			
07/07/2021	Corso flutter Udemy	✓			
08/07/2021	Corso flutter Udemy	✓			
09/07/2021	Corso flutter Udemy	✓			
12/07/2021	Corso flutter Udemy	✓			
13/07/2021	Corso flutter Udemy	✓			
14/07/2021	Fine corso flutter Udemy e studio codice esistente	✓			
15/07/2021	Studio codice esistente	✓			
16/07/2021	Studio codice esistente	✓			
19/07/2021	Studio implementazione per filtri	✓			
20/07/2021	Riconrollo filtri, cambiato logo app, modifica interfaccia modifica con cambio pulsante e sistemazione problema minuti	✓			
21/07/2021	Inizio studio per inserimento mappa	✓			
22/07/2021	Funzionalità mappa	✓			
23/07/2021	Funzionalità mappa	✓			
26/07/2021	Funzionalità mappa con leaflet	✓			
27/07/2021	Funzionalità mappa	✓			
28/07/2021	Funzionalità mappa	✓			
29/07/2021	Funzionalità mappa	✓			

Figura 2.2: Google sheets

- **Discord** che mi ha permesso di interfacciarmi direttamente con il tutor aziendale Fabio Pallaro sfruttando più canali di comunicazione vocali e testuali, divisi per argomenti.

# Framework Flutter e linguaggio Dart

### 3.1 Sviluppo applicazioni mobile

11

Esistono quattro diversi approcci di implementazione:

- App native;
- Web app;
- App Ibride Web View Wrapper;
- App Ibride Compile to Native.

### 3.1.1 App native

Il metodo nativo dà la possibilità all'applicazione di integrarsi con la parte hardware del dispositivo, sfruttando così tutte le funzionalità del sistema operativo. Le app native vengono realizzate utilizzando gli strumenti di sviluppo software e la documentazione fornita dai produttori del sistema operativo per il quale si ha l'intenzione di sviluppare. Questo metodo è scelto soprattutto dagli sviluppatori attenti alle prestazioni e alle performance dell'applicazione. I vantaggi principali di sviluppare App native sono:

- Maggiore velocità, affidabilità e reattività;
- Accesso diretto alla parte hardware e al software installato nel device;
- Notifiche dirette;
- Funzionamento offline.

Attualmente i sistemi operativi più utilizzati sono:

- Android;
- iOS;
- Windows Phone.

## Android



**Figura 3.2:** Logo Android

Android è il sistema operativo più utilizzato e diffuso. È stato sviluppato da Google ed è stato scelto da multinazionali importanti come Samsung, Huawei e Amazon per il funzionamento dei loro dispositivi. Il linguaggio per sviluppare un'applicazione Android è Java. Negli ultimi anni è nato anche Kotlin che è un altro linguaggio ufficiale per la progettazione di applicazioni Android che è più moderno, meno complesso ma performante e compatibile con l'ambiente Android quanto Java.

## iOS

### iOS

**Figura 3.3:** Logo iOS

iOS è il sistema operativo sviluppato da Apple per dispositivi iPhone, iPod touch e iPad. Per un lungo periodo il linguaggio per sviluppare un'applicazione iOS è stato Objective-C che deriva da C e C++. Per aumentare la produttività Apple ha lanciato un linguaggio di più alto livello ovvero Swift. Swift è veloce, più leggibile e meno prolisso. Nonostante ciò, Objective-C viene ancora preferito quando si sta lavorando più a basso livello.

## Windows Phone



**Figura 3.4:** Logo Windows Phone

Windows Phone è il sistema operativo sviluppato da Microsoft. Il linguaggio utilizzato per sviluppare un'applicazione Windows Phone è C# che è un linguaggio semi-compilato orientato agli oggetti. Con Android e iOS in ambito mobile non c'è paragone. Invece per quanto riguarda sistemi desktop Windows risulta una delle migliori.

### 3.1.2 Web app

Una web app è un'applicazione che funziona come un sito web adattandosi al dispositivo utilizzato. Queste applicazioni non necessitano di essere installate sugli smartphone e quindi non andranno ad aumentare la memoria utilizzata nel dispositivo. Inoltre non possono essere nemmeno pubblicate sugli Store e quindi non godono di questa enorme visibilità. I principali framework e librerie per creare una web app sono:

- Angular;
- PolymerJS;
- React.

I vantaggi di sviluppare una web app sono:

- Scritte con Markup HTML;
- non essendo pubblicate sul Market non devono essere sottoposte al processo di approvazione;
- Minor tempo di sviluppo.

### 3.1.3 App Ibride Web View Wrapper

Questo tipo di metodo permette di creare applicazioni senza alcuna conversione del codice in base al sistema operativo. In pratica l'applicazione rileva inizialmente il sistema operativo utilizzato e successivamente imita l'aspetto dell'interfaccia utente utilizzando CSS, Sass...

Le piattaforme più usate sono:

- Ionic;
- Apache Cordova;
- PhoneGap.

I vantaggi principali di Ionic e delle App Ibride Web View Wrapper sono:

- Riutilizzo facile del codice;
- Ionic utilizza JavaScript e fornisce un supporto per Angular;
- Aggiornamento automatico in base alla piattaforma;

Lo svantaggio principale di questo tipo di applicazioni sta in termini di velocità di calcolo e quindi avranno prestazioni inferiori a quelle compilate in nativo.

### 3.1.4 App Ibride Compile to Native

Le applicazioni ibride che compilano in nativo utilizzano un unico linguaggio di programmazione per la scrittura del codice. Una volta compilato i componenti dell'interfaccia utente del codice vengono convertiti nei componenti dell'interfaccia utente nativi. Ad esempio se bisogna creare un nuovo Widget questo una volta compilato verrà tradotto nel Widget nativo del sistema operativo di esecuzione. Le principali piattaforme utilizzate per compilare in nativo sono:

- React Native;
- NativeScript;
- Xamarin;
- Flutter.

I vantaggi principali di utilizzare piattaforme che compilano in nativo sono:

- Anche se minore delle App Ibride Web View Wrapper hanno un'elevata riutilizzabilità del codice;
- Elevato numero di librerie utilizzabili;
- Compilando in nativo offrono prestazioni elevate.

## 3.2 Flutter



Figura 3.5: Logo Flutter

### 3.2.1 Dart

### 3.2.2 Componenti

Framework

Engine

Skia

Embedder

### 3.2.3 Widget

Stateless widget

Stateful widget

Librerie Material e Cupertino

Principali Widget

### 3.2.4 Rendering e Layout

### 3.2.5 Esempi piccole applicazioni realizzate





## Capitolo 4

# Sportwill

*In questo capitolo viene descritto il progetto, effettuata un'analisi iniziale dei requisiti per poi definire dettagliatamente come le funzionalità sono state implementate e che tecnologie e strumenti e sono stati usati.*

## 4.1 Descrizione progetto

## 4.2 Analisi dei requisiti

## 4.3 Tecnologie e strumenti

### 4.3.1 Flutter e Dart

### 4.3.2 Android Studio

### 4.3.3 GitLab

### 4.3.4 Backend

## 4.4 Implementazioni

### 4.4.1 Filtro di ricerca testo

### 4.4.2 Logo

### 4.4.3 Pagina Modifica e campi obbligatori

### 4.4.4 Mappa percorso

### 4.4.5 Aggiornamento automatico mappa

### 4.4.6 Mappa schermo intero

### 4.4.7 Colori

### 4.4.8 Logo

### 4.4.9 Pagina Modifica

### 4.4.10 Eliminazione, Modifica e Aggiunta di un'attività

### 4.4.11 Filtro avanzato di ricerca attività

## Capitolo 5

# Conclusioni

5.1 Consuntivo finale

5.2 Raggiungimento degli obiettivi

5.3 Conoscenze acquisite

5.4 Valutazione personale







# Bibliografia

1 <https://www.synclab.it/> 2 <https://www.bintmusic.it/uso-smartphone-statistiche-controllo/>  
3 <https://pan-webdesign.it/pages/sviluppo-applicazioni-mobile-possibili-scenari-metodi-applicativi.html> 4 <http://www.mr-apps.com/it/blog/la-differenza-tra-app-native-app-ibride-e-web-app> 5 <https://www.icicletech.com/blog/react-native-flutter-ionic-xamarin-nativescript> 6 7 <https://flutter.dev/> 8 <https://dart.dev/> 9  
<https://www.motivonetwork.it/i-sistemi-operativi-mobile-piu-usati-al-mondo/>