

**Università degli Studi di Padova**

DIPARTIMENTO DI MATEMATICA "TULLIO LEVI-CIVITA "

CORSO DI LAUREA IN INFORMATICA



**Realizzazione di un microservizio con Spring  
per un'applicazione per la gestione di  
attività sportive**

*Tesi di laurea triennale*

*Relatore*

Prof. Paolo Baldan

*Laureando*

Lorenzo Matterazzo

---

ANNO ACCADEMICO 2020-2021

*Realizzazione di un microservizio con Spring per un'applicazione per la gestione di  
attività sportive*

Tesi di laurea triennale

Lorenzo Matterazzo, © Dicembre 2021.

# Sommario

Il presente documento descrive il lavoro svolto durante il periodo di stage, della durata di 320 ore, dal laureando Lorenzo Matterazzo presso l'azienda Sync Lab S.r.l., situata a Padova. Lo stage ha avuto come argomento principale l'implementazione di nuove funzionalità nel contesto dell'applicazione ***Sport Will***, una *web app* che dà modo all'utente di divulgare la sua intenzione (*will*<sup>[g]</sup>) di effettuare un'attività sportiva. La piattaforma fa vedere tutto a tutti, rendendo troppo caotica la fruizione, quindi l'esigenza era di dare la possibilità all'utente di creare uno o più gruppi a cui utenti "amici" possano unirsi, vedendo quindi solo le *will* di gruppo. Le attività svolte nel corso dello stage sono due:

- \* la prima è un insieme di attività che sono legate al *backend* della *web app*, come lo sviluppo di tre *microservizi*<sup>[g]</sup> mediante il *framework*<sup>[g]</sup> *Spring*<sup>[g]</sup> Java. In particolare:
  1. il primo microservizio consente l'effettuazione delle funzionalità *Create Read Update Delete (CRUD)* per la gestione dei gruppi e la visualizzazione delle *will* degli utenti appartenenti allo stesso gruppo;
  2. il secondo è l'implementazione di un *API Gateway*<sup>[g]</sup>, che permette di esporre le *Application Program Interface (API)* dei vari servizi presenti in un unico punto di accesso;
  3. il terzo è l'implementazione di un *Eureka Server*<sup>[g]</sup> che contiene le informazioni di tutti i servizi che si registrano nel suo server.

Oltre all'implementazione di nuovi microservizi, quelli esistenti sono stati modificati affinché le *will* possano essere visualizzate o da tutti gli utenti oppure solo dagli utenti appartenenti agli stessi gruppi. Ultimate le attività lato *backend*, è stata effettuata la *containerizzazione*<sup>[g]</sup> di tutti i microservizi su Docker.

- \* La seconda attività è stata la modifica del *frontend*, mediante il *framework* *Angular*<sup>[g]</sup>, per adeguarla alle nuove funzionalità del *backend*.

L'esito dello stage è stato molto positivo: le attività obbligatorie e facoltative sono state portate a termine con successo abbastanza facilmente e con un un po' di anticipo che mi ha permesso di implementare anche la parte *frontend* dell'applicazione.



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

— Confucius

# Ringraziamenti

*Innanzitutto, vorrei esprimere la mia gratitudine al Prof. Paolo Baldan, 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, Dicembre 2021*

Lorenzo Matterazzo



# Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
1.1	Convenzioni tipografiche . . . . .	1
1.2	L'azienda . . . . .	1
1.3	Lo stage proposto . . . . .	2
1.4	Strumenti utilizzati . . . . .	2
1.4.1	Visual Studio Code . . . . .	2
1.4.2	Figma . . . . .	3
1.4.3	Git . . . . .	3
1.4.4	Postman . . . . .	3
1.4.5	DbVisualizer . . . . .	3
1.5	Prodotto ottenuto . . . . .	3
1.6	Accessibilità . . . . .	4
1.7	Organizzazione del testo . . . . .	4
<b>2</b>	<b>Processi e metodologie</b>	<b>5</b>
2.1	Processo sviluppo prodotto . . . . .	5
<b>3</b>	<b>Descrizione dello stage</b>	<b>7</b>
3.1	Introduzione al progetto . . . . .	7
3.2	Analisi preventiva dei rischi . . . . .	7
3.3	Requisiti e obiettivi . . . . .	7
3.4	Pianificazione . . . . .	7
<b>4</b>	<b>Analisi dei requisiti</b>	<b>9</b>
4.1	Casi d'uso . . . . .	9
4.2	Tracciamento dei requisiti . . . . .	10
<b>5</b>	<b>Progettazione e codifica</b>	<b>13</b>
5.1	Tecnologie e strumenti . . . . .	13
5.2	Ciclo di vita del software . . . . .	13
5.3	Progettazione . . . . .	13
5.4	Design Pattern utilizzati . . . . .	13
5.5	Codifica . . . . .	13
<b>6</b>	<b>Verifica e validazione</b>	<b>15</b>
<b>7</b>	<b>Conclusioni</b>	<b>17</b>
7.1	Consuntivo finale . . . . .	17
7.2	Raggiungimento degli obiettivi . . . . .	17

7.3	Conoscenze acquisite . . . . .	17
7.4	Valutazione personale . . . . .	17
<b>A</b>	<b>Appendice A</b>	<b>19</b>
	<b>Bibliografia</b>	<b>23</b>



# Elenco delle figure

1.1	Logo dell'azienda . . . . .	<a href="#">1</a>
4.1	Use Case - UC0: Scenario principale . . . . .	<a href="#">9</a>

# Elenco delle tabelle

4.1	Tabella del tracciamento dei requisiti funzionali . . . . .	<a href="#">11</a>
4.2	Tabella del tracciamento dei requisiti qualitativi . . . . .	<a href="#">11</a>
4.3	Tabella del tracciamento dei requisiti di vincolo . . . . .	<a href="#">11</a>



# Capitolo 1

## Introduzione

### 1.1 Convenzioni tipografiche

Durante 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](#)<sup>[g]</sup>;
- \* i termini in lingua straniera o facenti parti del gergo tecnico sono evidenziati con il carattere *corsivo*.

### 1.2 L'azienda

Sync Lab S.r.l. nasce a Napoli nel 2002 come *software house* ed è rapidamente cresciuta nel mercato dell'[Information and Communications Technology \(ICT\)](#), tramutatasi in *System Integrator* e conquistando significative fette di mercato nei settori *mobile*, videosorveglianza e sicurezza delle infrastrutture informatiche aziendali.

Attualmente, Sync Lab S.r.l. ha più di 150 clientidiretti e finali, con un organico aziendale di 200 dipendenti distribuiti tra le 5 sedi dislocate in tutta Italia.

Sync Lab S.r.l. si pone come obiettivo principale quello di supportare il cliente nella realizzazione, messa in opera e governance di soluzione IT, sia dal punto di vista tecnologico, sia nel governo del cambiamento organizzativo.



**Figura 1.1:** Logo dell'azienda

## 1.3 Lo stage proposto

“Lo sport dà il meglio di sé quando ci unisce.”

Frank Deford

**Sport Will** permette agli utenti di condividere le proprie *will*, e gli utenti che hanno intenzione di unirsi in questa attività vi possono partecipare.

Dal momento che allo stato dell'arte attuale non esiste ancora la suddivisione delle *will* per gruppi, lo stage proposto da Sync Lab S.r.l. consiste nell'integrare alla piattaforma già esistente la suddivisione delle visualizzazioni delle *will* solo agli utenti che appartengono agli stessi gruppi.

Gli obiettivi da raggiungere nel corso dello stage sono principalmente due:

- \* sviluppo di un *microservizio* utilizzando il *framework Spring* Java per la creazione dei gruppi;
- \* modifica dei *microservizi* esistenti affinché permettano la visualizzazione solo delle *will* di utenti appartenenti allo stesso gruppo.

Non è richiesta la modifica del *frontend* in modo da adeguarlo alle nuove funzionalità del *backend*, a meno che non rimanga tempo da investire su questa attività.

## 1.4 Strumenti utilizzati

### 1.4.1 Visual Studio Code

Visual Studio Code è un editor di codice sorgente sviluppato da Microsoft per Windows, Linux e macOS. Include il supporto per debugging, un controllo per Git integrato, Syntax highlighting, IntelliSense, Snippet e refactoring del codice.

Punto di forza di Visual Studio Code sono le estensioni grazie alle quali è possibile ampliare notevolmente le funzionalità del programma.

Le estensioni utilizzate nel corso dello stage sono:

- \* **GitLens**: permette di ampliare le funzionalità di Git integrate in Visual Studio Code;
- \* **Spring Boot Extension Pack**: raccolta di estensioni per lo sviluppo con Spring Boot Application;
- \* **W3C Web Validator**: permette di controllare la validità del *markup* dei documenti html e css;
- \* **Web Accessibility**: permette di verificare l'accessibilità dei documenti html, evidenziando gli elementi che si potrebbe prendere in considerazione di cambiare e dando suggerimenti su come potrebbe modificato;
- \* **Docker Extension Pack**: raccolta di estensioni per la gestione dei *container* Docker, Docker images, Dockerfile e file Docker-compose;
- \* **Angular Extension Pack**: raccolta di estensioni per lo sviluppo con Angular;
- \* **Extension Pack for Java**: raccolta di estensioni popolari che possono aiutare a scrivere, testare e fare il *debugging* di applicazioni Java.

### 1.4.2 Figma

Figma è un *tool* per la progettazione di interfacce, che si rivolge principalmente ai *web designer* che hanno bisogno di un software studiato appositamente per realizzare il *design* di siti web e applicazioni.

Nel contesto dello stage è stato utilizzato per la realizzazione delle pagine web per la visualizzazione, creazione e modifica di un gruppo.

### 1.4.3 Git

*Software* di versionamento utile a tracciare modifiche e cambiamenti di insiemi di file.

### 1.4.4 Postman

Si tratta di uno strumento che permette di eseguire richieste HTTP ad un *server* di *backend*. Quando si lavora con un altro sviluppatore *backend* è possibile condividere le [API](#), ma la sua vera forza è quella di farci sapere tutto di una richiesta HTTP.

### 1.4.5 DbVisualizer

DbVisualizer è uno strumento multi-database intuitivo e ricco di funzionalità per sviluppatori, analisti e amministratori di database, che fornisce un'unica, potente interfaccia su un'ampia gamma di sistemi operativi. Grazie alla sua interfaccia chiara e facile da usare, DbVisualizer si è dimostrato uno strumento di database molto conveniente, che funziona su tutti i principali sistemi operativi e supporta molte varietà di database.

## 1.5 Prodotto ottenuto

Al termine dello stage le integrazioni delle funzionalità con la *web-app* sono state realizzata con successo. Tutte le chiamate alle [API](#) sono state testate e sono state integrate anche nel *frontend*. L'integrazione delle nuove funzionalità permettono alla *web-app* di:

- \* visualizzare le [will](#) appartenenti agli utenti che partecipano agli stessi gruppi;
- \* visualizzare le [will](#) con visibilità globale (quindi che non sono visibili solo agli utenti che partecipano agli stessi gruppi);
- \* visualizzare i gruppi a cui partecipa un utente;
- \* visualizzare e modificare i gruppi creati da un utente;
- \* visualizzare e cercare i gruppi;
- \* creare nuovi gruppi.

## 1.6 Accessibilità

Durante la realizzazione del sito è stata resa la navigazione più efficace attraverso l'uso di *accesskey*.

Le immagini sono tutte marcate con gli appositi *tag alt*, che sono stati lasciati vuoti nel caso servissero solo per il layout.

È presente una barra di navigazione che aiuta a navigare nel sito.

Ogni *link* è stato reso distinguibile da ogni altro elemento tramite appositi CSS, *hover* e *visited*, in modo da aiutare l'utente ad orientarsi.

Inoltre, per evitare *link* circolari, nella barra di navigazione le pagine non contengono *link* che navigano alla pagina corrente.

I form contengono dei tag label per ogni input.

Non sono stati aggiunti *tag optgroup* o *fieldset* in quanto sono utili nel caso di *form* molto grandi, ma essendo presenti solo *form* di piccole dimensioni è stato ritenuto non necessario.

Sono presenti gli attributi *accesskey* con chiave uguale alla prima lettera della parola del *tag label* associato per migliorare l'accessibilità alle *form* da tastiera senza l'uso del *mouse*. Sono presenti degli aiuti contestuali che mostrano gli errori nel caso fossero presenti. Non sono stati aggiunti *tabindex* nei form dato che l'ordine di tabulazione è già corretto.

Sono stati evitati i tag ed attributi deprecati.

È presente del testo nascosto utile agli utenti con disabilità visive, come il *link* con la funzione di saltare al contenuto, ovvero di permettere di non far leggere allo *screen reader* la barra di navigazione, passando direttamente al contenuto, ed è presente all'inizio della navigazione per segnalare che le scorciatoie da tastiera sono attive.

Inoltre per migliorare la navigazione dello scroll, viene mostrato un pulsante in basso a destra dello schermo e, se un utente lo clicca, viene effettuato uno scroll verso l'alto fino all'inizio della pagina. Per agevolare la navigazione è stato inserito un pulsante in basso a destra che, se cliccato, permette di visualizzare l'inizio della pagina. Il pulsante è un *link* con testo nascosto e come immagine di *background* una freccia, in modo che lo *screen reader* riesca comunque a leggere il testo. Le parole in lingua straniera sono state racchiuse dentro un *tag* con l'attributo *lang*, in modo da permettere allo *screen reader* di leggere la parola correttamente.

## 1.7 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 ...

## 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 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



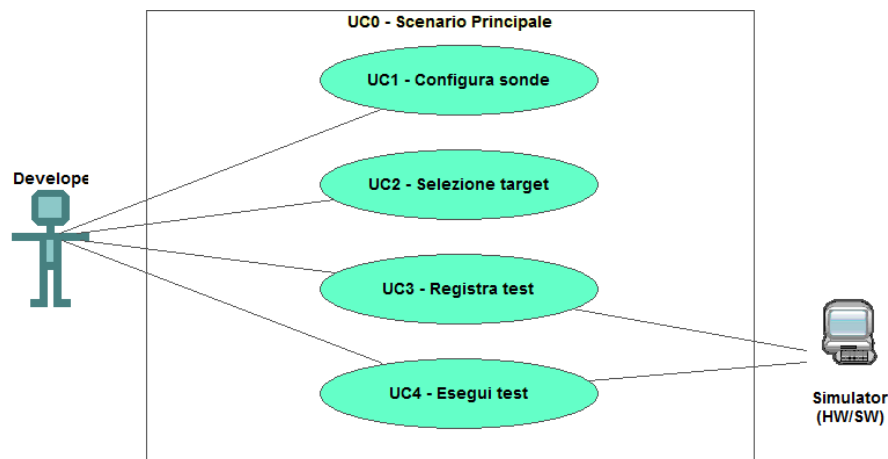
## Capitolo 4

# 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'IDE.

**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









# Bibliografia