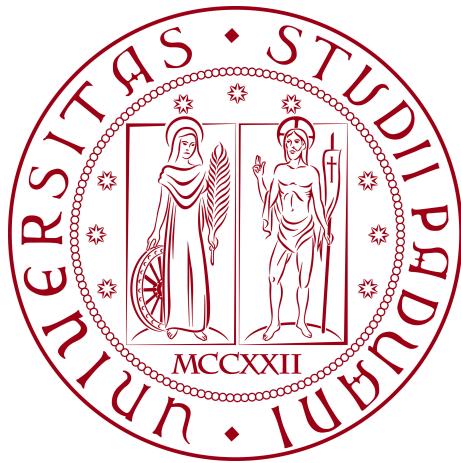


**Università degli Studi di Padova**

DIPARTIMENTO DI MATEMATICA “TULLIO LEVI-CIVITA”

CORSO DI LAUREA IN INFORMATICA



**Studio di Operatori Kubernetes per la gestione  
di applicazioni persistenti.**

*Tesi di Laurea Triennale*

*Relatore*

Prof. Vardanega Tullio

*Laureando*

Rosson Lorenzo

Matricola 2042349

© Rosson Lorenzo, Agosto 2024. Tutti i diritti riservati. Tesi di Laurea Triennale:  
“*Studio di Operatori Kubernetes per la gestione di applicazioni persistenti.*”, Università degli Studi di Padova, Corso di Laurea in Informatica, Dipartimento di Matematica “Tullio Levi-Civita”.

# Sommario

Il presente documento descrive ed esplora il lavoro svolto durante il periodo di *stage*, della durata di trecentoventi ore, dal laureando Lorenzo Rosson presso l’azienda Zucchetti Spa. Lo scopo del tirocinio è quello di apprendere ciò che comprende l’uso di Kubernetes nella gestione di applicazioni persistenti, “*Cloud Native*”, tramite Operatori realizzati appositamente, in modo da poterne valutare potenzialità e limitazioni in ottica di produzione. Per iniziare mi sono dedicato all’acquisizione delle competenze tecniche richieste per poter riuscire nello sviluppo del progetto, attraverso lo studio di documentazione, la lettura approfondita di articoli inerenti e la visione di video esempio. Inizialmente relative all’architettura e funzionamento di *Kubernetes*, in secondo luogo, relative alla sviluppo di operatori e quindi necessariamente, al linguaggio di programmazione individuato per tale attività, Go (*Golang*). Successivamente all’apprendimento delle conoscenze richieste, ho innanzitutto creato un ambiente basilare con *Kubernetes* testandone le funzionalità native al fine di consolidare l’apprendimento teorico con l’esperienza pratica. Sono poi passato alla progettazione dell’operatore andando a valutare e confrontare principalmente varie politiche di natura gestionale con il fine di migliorare le prestazioni dell’applicazione che controlla. Infine ho proceduto con lo sviluppo vero e proprio dell’operatore andando a coprire le funzionalità previste, soffermandomi con maggiore attenzione su quelle relative alla componente adibita alla persistenza. A supporto, da buona prassi, ho svolto in modo incrementale, un’attività di documentazione di tutti i periodi e relativi prodotti del progetto, comprendente progettazione, sviluppo e *testing*.

---

Il documento è suddiviso nei seguenti 4 capitoli:

- **L'azienda:** capitolo che introduce e presenta il contesto ed ambiente aziendale nel quale ho avuto l'opportunità di svolgere il tirocinio. Comprendente un approfondimento sull'azienda, suoi prodotti e clienti, processi interni e di sviluppo, e propensione all'innovazione;
- **Lo stage:** capitolo che illustra lo *stage*, il suo scopo e il suo ruolo nel contesto aziendale. Comprendente elenco e descrizione di vincoli e obiettivi del progetto e rapporto dell'azienda con le attività di *stage*;
- **Il progetto:** TODO;
- **Valutazioni retrospettive:** TODO.

#### **Norme tipografiche**

TODO

# Indice

<b>1 L’azienda</b>	<b>1</b>
1.1 Struttura organizzativa . . . . .	1
1.2 Principali prodotti e clienti . . . . .	2
1.3 Processi interni . . . . .	4
1.3.1 Metodologie in adozione . . . . .	6
1.3.2 Processi di sviluppo . . . . .	7
1.4 Tecnologie in utilizzo . . . . .	9
1.5 Propensione all’innovazione . . . . .	14
<b>2 Lo stage</b>	<b>16</b>
2.1 La visione dell’azienda . . . . .	16
2.2 Il ruolo degli stage nel contesto aziendale . . . . .	17
2.3 Scopo . . . . .	18
2.4 Vincoli e obiettivi . . . . .	21
2.5 Motivazione della scelta . . . . .	29
2.6 Obiettivi personali . . . . .	30
<b>3 Il progetto</b>	<b>33</b>
3.1 Pianificazione . . . . .	33
3.1.1 Metodologia . . . . .	33
3.1.2 Rapporti con tutor e azienda . . . . .	33
3.1.3 Revisioni . . . . .	33
3.2 Analisi dei requisiti . . . . .	34
3.2.1 Scenari e casi d’uso . . . . .	34
3.2.2 Requisiti . . . . .	34

## INDICE

---

3.3 Progettazione . . . . .	34
3.4 Sviluppo . . . . .	34
3.5 Verifica . . . . .	34
3.5.1 Analisi Statica . . . . .	35
3.5.2 Analisi Dinamica . . . . .	35
3.6 Validazione e collaudo . . . . .	35
3.7 Risultati ottenuti . . . . .	35
<b>4 Valutazioni retrospettive</b>	<b>i</b>
4.1 Soddisfacimento degli obiettivi . . . . .	i
4.2 Crescita personale . . . . .	i
4.3 Riflessioni finali . . . . .	i
<b>Bibliografia</b>	<b>ii</b>
<b>Sitografia</b>	<b>iii</b>

# Elenco delle figure

1.1	Torre Zucchetti, sede principale e simbolo dell'azienda . . . . .	2
1.2	Interfaccia Zucchetti inrecruiting, software per la gestione del reclutamento di nuovo personale . . . . .	3
1.3	Processo <i>Scrum</i> , suddiviso in fasi . . . . .	7
1.4	Architettura d'esempio di un'applicazione Zucchetti di base . . . . .	10
1.5	<i>Podman Desktop</i> , interfaccia grafica dello strumento . . . . .	12
1.6	<i>Team Zucchetti</i> ai mondiali di calcio 2022 . . . . .	15
2.1	Conferenze <i>KubeCon &amp; CloudNativeCon</i> , Hong Kong Agosto 2024	17
2.2	Astrazione funzionamento di base degli operatori . . . . .	19
2.3	Astrazione infrastruttura per la gestione delle cartelle condivise	23
2.4	<i>Gantt view</i> del progetto . . . . .	26
2.5	Crescita professionale . . . . .	31

# Elenco delle tavole

2.1	Tabella riassuntiva dei requisiti individuati durante la stesura del piano di lavoro iniziale . . . . .	28
-----	---	----

## ELENCO DELLE TABELLE

---

# Capitolo 1

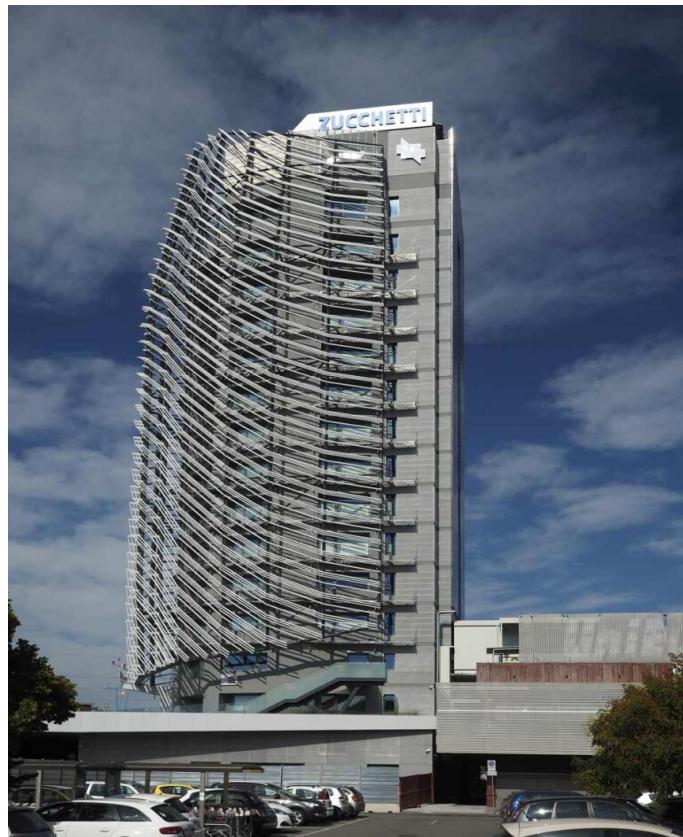
## L’azienda

### 1.1 Struttura organizzativa

Zucchetti venne fondata da Domenico Zucchetti nel 1978 con sede a Lodi, tuttora centro dell’azienda (Figura 1.1). Tra gli anni Ottanta e Duemila, la società ha ampliato progressivamente la propria presenza in diversi settori, tra cui l’*enterprise resource planning* (ERP), sistemi software integrati che consentono alle aziende di gestire e automatizzare molte attività loro quotidiane, come la contabilità, la gestione delle risorse umane, la fornitura, le vendite, la produzione e molte altre, ma anche, l’automazione industriale, l’uso di tecnologie, sistemi di controllo e software per gestire e automatizzare processi industriali, e la robotica. Questo processo di espansione ha incluso anche l’ingresso in alcuni mercati esteri attraverso l’acquisizione di svariate società informatiche. Dal 2008, il Gruppo Zucchetti è guidato dai figli del fondatore, Alessandro e Cristina, la cui *leadership* ha portato l’azienda a crescere significativamente, acquisendo numerose aziende, anche concorrenti. Basti pensare come, la loro ultima operazione, o almeno la sua ufficializzazione<sup>1</sup>, risalga a poco meno di due mesi fà, ovvero contemporaneamente al mio tirocinio. Oggi, Zucchetti può vantare una presenza internazionale che conta svariate sedi, oltre 9.000 dipendenti e uno dei fatturati annui più rilevanti nel settore a livello nazionale.

---

<sup>1</sup> Acquisizione dell’azienda Ranocchi. URL: <https://www.zucchetti.it/it/cms/news-e-eventi/comunicati/zucchetti-acquisisce-il-gruppo-ranocchi.html>.



**Figura 1.1:** Torre Zucchetti, sede principale e simbolo dell'azienda

**Fonte:** [www.promozioneacciaio.it](http://www.promozioneacciaio.it)

Sono inoltre numerose le certificazioni ed i riconoscimenti ottenuti nel corso degli anni, tra cui: la certificazione ISO 9001:2015 per il sistema di gestione della qualità, la ISO 22301 per la gestione della continuità operativa, la ISO/IEC 27018 per la protezione delle informazioni personali nei servizi *cloud* pubblici e moltissime altre.

## 1.2 Principali prodotti e clienti

Il Gruppo Zucchetti è *leader* nel settore delle soluzioni *software* e dei servizi tecnologici, offrendo una gamma completa di prodotti per soddisfare le esigenze di diverse tipologie di aziende.

Tra i principali prodotti, possiamo trovare:

- **Zucchetti ERP:** Un *software* integrato per la gestione delle risorse aziendali che copre la contabilità, la gestione delle risorse umane, il controllo di gestione e la logistica;
- **HR Infinity:** Una *suite* completa per la gestione del personale, che include strumenti per la pianificazione delle presenze, il calcolo delle retribuzioni e l'amministrazione del personale;



**Figura 1.2:** Interfaccia Zucchetti inrecruiting, software per la gestione del reclutamento di nuovo personale

**Fonte:** [hr.zucchetti.it](http://hr.zucchetti.it)

- **Infinity Safety:** Software per la gestione della salute e sicurezza sul lavoro, che permette la valutazione dei rischi e la gestione degli incidenti;
- **Zucchetti e-Fattura:** Una soluzione per la creazione, l'invio e la conservazione delle fatture elettroniche, conforme alle normative italiane ed europee;
- **Zucchetti Automation:** Sistemi per l'automazione dei processi produttivi, comprendendo soluzioni di robotica e *Internet of Things* (IoT), con-

cetto tecnologico che si riferisce al dominio di dispositivi fisici connessi a Internet e in grado di raccogliere, scambiare e analizzare dati.

- **Zucchetti Cloud:** Servizi *cloud*, ovvero risidenti sull'Internet, e *data center*, strutture fisiche utilizzate per ospitare sistemi informatici critici e componenti correlati, come server, storage, reti e dispositivi di sicurezza, per l'*hosting*, la gestione e la sicurezza dei dati aziendali;
- **Infinity SitePainter:** Una piattaforma per la creazione di portali web aziendali e siti internet, permettendo una gestione semplice e intuitiva dei contenuti online.

L'azienda serve un'ampia varietà di clienti, che spaziano dalle piccole e medie imprese a grandi enti e organizzazioni, operando in diversi settori. Tra i principali clienti di Zucchetti si trovano alcune delle più grandi catene di supermercati italiane come Esselunga, che utilizza le loro soluzioni per la gestione dei punti vendita e il controllo degli *stock*; ma anche multinazionali come Ferrero, nel settore dolciario, e il Gruppo Calzedonia, nel settore moda, che beneficiano delle soluzioni della casa di sviluppo per l'automazione e la gestione delle risorse umane. L'azienda collabora inoltre con leader nel settore automobilistico come *Fiat Chrysler Automobiles* (FCA), che implementa le soluzioni ERP e di automazione industriale, e con Ferrovie dello Stato per la gestione delle risorse e la sicurezza. Infine, tra i clienti di spicco figurano anche l'Ospedale San Raffaele, per la gestione delle risorse sanitarie, e Banca Mediolanum, per la gestione dei servizi finanziari e la sicurezza dei dati. Zucchetti fornisce soluzioni personalizzate e supporto dedicato, caratteristiche che gli hanno permesso di costruire un portfolio di clienti prestigiosi e diversificati, consolidando la propria posizione di *leadership* nel mercato.

### 1.3 Processi interni

Le informazioni che riporto, derivano dalla mia esperienza di *stage* in una delle sedi del reparto di ricerca e sviluppo (RD) dell'azienda. Tali potrebbero quindi non valere o valere solo in parte per gli altri reparti e sedi.

L'azienda presenta una struttura organizzativa ben definita e processi interni progettati per ottimizzare l'efficienza e garantire elevati standard di qualità in tutte le sue operazioni. I principali processi interni dell'azienda comprendono:

- **Consulenza:** Per la quale l'azienda dispone un servizio clienti progettato per offrire assistenza tempestiva e professionale attraverso supporto tecnico dedicato, che comprende *help desk*, assistenza telefonica e *online*, e servizi di consulenza personalizzata;
- **Fornitura:** Processo che gestisce l'approvvigionamento e la distribuzione dei prodotti, assicurando che le soluzioni siano consegnate ai clienti in modo efficiente e conforme agli standard di qualità aziendali. Zucchetti da molta importanza alla trasparenza con il cliente, all'interno dell'azienda questo processo è infatti strettamente legato a quello di consulenza e controllo della qualità.
- **Controllo della qualità:** Per la quale l'azienda adotta rigorosi standard di controllo per assicurare che tutti i prodotti e servizi soddisfino le aspettative dei clienti. Il processo include test approfonditi e controlli degli *standard* interni che seguono certificazioni secondo le normative internazionali come la già citata ISO 9001:2015;
- **Sviluppo delle Competenze del Personale:** Processo focalizzato sul reclutamento e sulla crescita professionale dei dipendenti. Zucchetti promuove un ambiente di lavoro stimolante e collaborativo, offrendo programmi di formazione continua per lo sviluppo delle competenze.
- **Gestione delle Infrastrutture IT:** Comprende la gestione delle infrastrutture tecnologiche, la manutenzione dei sistemi e la gestione dei dati. Zucchetti utilizza soluzioni avanzate di *cloud computing* e *data center* per garantire sicurezza ed affidabilità operativa.
- **Manutenzione:** Il processo di manutenzione include la gestione continua delle soluzioni software e hardware fornite ai clienti. L'azienda assicura che i suoi prodotti siano sempre aggiornati e funzionanti, fornendo supporto

per eventuali problemi tecnici e implementando miglioramenti periodici per rispondere alle esigenze del mercato.

Questi processi interni sono supportati da una cultura aziendale orientata all'innovazione, alla qualità e al servizio eccellente, che puntano a rispondere in modo efficace alle esigenze in continua evoluzione dei suoi clienti.

Pilastro di ogni *software house*, vengono poi i processi di sviluppo e gestione del progetto.

### 1.3.1 Metodologie in adozione

Per la gestione dei progetti e lo sviluppo *software*, l'azienda, fa uso di una metodologia di stampo *Agile*<sup>2</sup>, approccio pensato per affrontare le sfide ricorrenti nei progetti software, come la necessità di flessibilità, la rapida evoluzione dei requisiti e l'importanza della collaborazione e trasparenza continua tra team e clienti.

Il modello *Agile* si basa su una serie di principi e pratiche che mirano a migliorare la reattività e l'efficacia dei team di sviluppo. Essi stati formalizzati nel Manifesto *Agile*, pubblicato nel 2001, che enfatizza valori fondamentali come:

- **Valorizzare gli individui e le interazioni** rispetto ai processi e agli strumenti;
- **Preferire il *software* funzionante** alla documentazione dettagliata;
- **Promuovere la collaborazione con il cliente** piuttosto che focalizzarsi sulla negoziazione dei contratti;
- **Adattarsi al cambiamento** anziché seguire rigidamente un piano.

Nello specifico, come ho potuto constatare nella sede nella quale ho svolto il tirocinio, l'azienda adotta un modello che risulta avvicinarsi molto a quello rappresentato dal *framework Scrum*<sup>3</sup>.

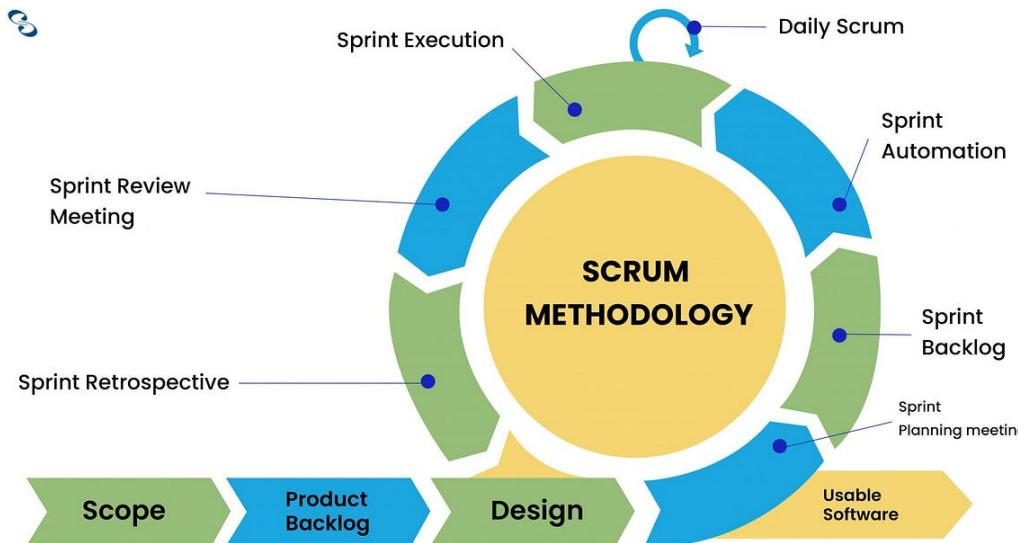
---

<sup>2</sup>Manifesto Agile. URL: <http://agilemanifesto.org/>.

<sup>3</sup>Scrum. URL: <http://scrum.org/>.

### 1.3.2 Processi di sviluppo

Lo sviluppo di ogni progetto, compreso quello relativo al mio *stage*, attraversa varie fasi che si rifanno a quanto previsto da questa metodica, osservabili in figura 1.3 seguente.



**Figura 1.3:** Processo Scrum, suddiviso in fasi

**Fonte:** blog.stackademic.com

Inizialmente, il progetto viene suddiviso in periodi, detti *sprint*, nel mio caso di durata settimanale, ai quali corrispondono determinati obiettivi riconducibili solitamente a specifiche *feature* del prodotto finale. Tali obiettivi vengono raccolti in un documento denominato ***Product Backlog***, una lista prioritaria di funzionalità, *bug* e miglioramenti che dovranno essere implementati.

Nel caso concreto del mio *stage*, ciò si rispecchia nella redazione dell'iniziale piano di lavoro.

Similmente, l'azienda ne produce uno a partire dalle esigenze e richieste del cliente o dall'idea e obiettivi del progetto, nel caso quest'ultimo sia di ricerca. Ogni *sprint* inizia quindi con una fase detta, di ***Sprint Planning***, durante la quale il *team* di sviluppo decide quali elementi del *backlog*, devono essere affrontati durante l'iterazione. Durante questa fase, il *team* discute e definisce

chiaramente gli obiettivi e i criteri di accettazione per ogni *ticket*, attività.

Come ho avuto modo di osservare l'azienda definisce questi ultimi solitamente durante un breve meeting che comprende principalmente gli sviluppatori, per poi suddividersi le varie attività, assegnando i compiti e gestendo il progetto attraverso vari strumenti di supporto che descriverò successivamente.

Corrispondentemente a quanto io abbia fatto ad ogni settimana, aggiornando in modo incrementale il preventivato piano di lavoro.

Durante lo *sprint* poi, il *team* si incontra quotidianamente in brevi riunioni chiamate **Daily Stand-up** o **Daily Scrum** con lo scopo di sincronizzare le attività del *team*, risolvere eventuali impedimenti e pianificare il lavoro giornaliero.

Per quanto riguarda la mia esperienza essi sono associabili alle interazioni quotidiane, anche se non calendarizzate ma in base alle necessità di ambe le parti, con il mio *tutor* aziendale. L'azienda predilige invece, quando possibile, la loro calendarizzazione. Inoltre i meeting avvengono sia tra gli sviluppatori, che talvolta, con clienti e collaboratori.

Successivamente, al termine di ogni *sprint*, si tiene una **Sprint Review**, durante la quale il *team* dimostra le funzionalità completate agli *stakeholder* e raccoglie *feedback*. Questo momento è cruciale per assicurarsi che il prodotto soddisfi le aspettative dei clienti e per apportare eventuali correzioni o miglioramenti. Anche questa penultima fase è riconducibile a quanto ho vissuto durante il tirocinio. Alla fine di ogni settimana di lavoro infatti, attraverso un *meeting* scadenzato, illustravo al mio tutor quanto fatto e raggiunto, raccogliendo *feedback* e consigli. Tale *meeting* andava poi ad impattare sul successivo *sprint planning*, rivedendo il piano di lavoro in base alle nuove priorità.

A questo proposito, l'azienda sembra tenere questi incontri più raramente, o perlomeno con una minore frequenza. Infine, ogni *sprint* si conclude con una **Sprint Retrospective**, una riunione interna del *team*, essenziale per il miglioramento costante delle pratiche e delle dinamiche di squadra, volta a riflettere su ciò che è andato bene, ciò che potrebbe essere migliorato e come apportare miglioramenti concreti al processo di sviluppo. Nel corso della mia esperienza ho avuto modo di confrontarmi settimanalmente, oltre che con il mio tutor aziendale, e con me stesso, con il mio relatore. Andando sostanzialmente in

parte, a replicare questa fase, ripercorrendo quanto fatto e affrontato. All'interno dell'azienda invece, come mi è stato possibile osservare, solitamente queste retrospettive coinvolgono gli sviluppatori incaricati e i loro punti di riferimento, come altri sviluppatori *senior*, ovvero a loro superiori in esperienza, o *project manager*.

## 1.4 Tecnologie in utilizzo

La sua vasta espansione e la numerosa acquisizione di altri enti nel corso della sua storia, ha portato l'azienda a diversificarsi in moltissimi settori con un'ampia varietà di prodotti e di servizi offerti. Ciò si riflette nel utilizzo di una vasta gamma di linguaggi di programmazione, strumenti di supporto e *framework*. Inoltre, per mantenere il suo vantaggio competitivo e rispondere alle evoluzioni del mercato, Zucchetti investe significativamente nella formazione continua dei propri dipendenti per, e nell'adozione di moderne e nascenti tecnologie.

Durante il mio *stage* ho avuto modo di osservare da vicino alcune delle tecnologie e strumenti, da loro, più comunemente utilizzati. Tra questi, spiccano i seguenti:

- **Java<sup>4</sup>**: linguaggio di programmazione ad oggetti ampiamente utilizzato per la sua robustezza, portabilità e scalabilità. In Zucchetti, impiegato principalmente per lo sviluppo di applicazioni enterprise, grazie alla sua capacità di gestire grandi volumi di dati e transazioni complesse, rendendolo ideale per i *software* gestionali e ERP, ma anche come linguaggio di programmazione *server-side*, ovvero che si occupa di tutto ciò che avviene e viene gestito su un *server*, nelle applicazioni *web*;
- **JavaScript<sup>5</sup>**: linguaggio di scripting versatile utilizzato sia lato client che lato server. Per l'azienda fondamentale per lo sviluppo di interfacce web interattive e dinamiche, migliorando l'esperienza utente nelle applicazioni web;

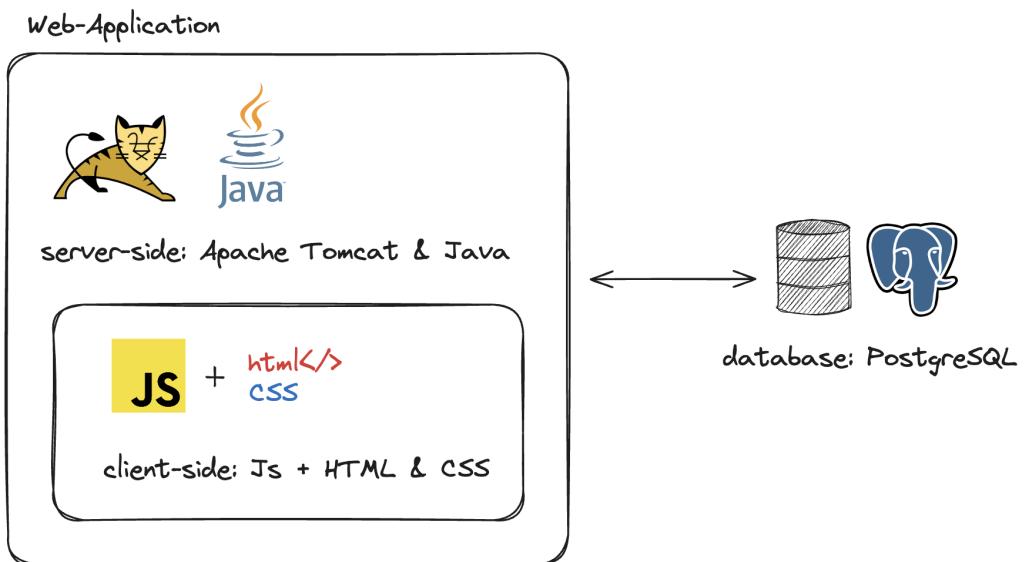
---

<sup>4</sup>Java. URL: <http://www.java.com/>.

<sup>5</sup>JavaScript. URL: <http://www.javascript.com/>.

- **Postgresql<sup>6</sup>**: sistema di gestione di database relazionale *open-source* conosciuto per la sua affidabilità e le sue avanzate capacità di gestione dei dati. Zucchetti utilizza *PostgreSQL* per archiviare e gestire dati nelle sue applicazioni, sfruttando le sue potenti funzionalità di *query* e la sua compatibilità con altri strumenti di analisi dati;
- **Apache Tomcat<sup>7</sup>**: *server* applicativo *open-source* per l'esecuzione di applicazioni Java, in particolare quelle basate su *servlet*, classe Java utilizzata per estendere le funzionalità di un server web, e *JSP*, tecnologia che consente di creare pagine web dinamiche utilizzando Java. Utilizzato dall'azienda per il *deployment* di applicazioni *web Java*, fornendo un ambiente stabile e sicuro per l'esecuzione dei servizi;

Durante il tirocinio, ho avuto la possibilità di usare un'applicazione base dell'azienda per alcuni test utili al mio progetto, che approfondirò in seguito. Allo stesso tempo, ciò mi ha permesso di capire come le tecnologie da loro utilizzate siano integrate le une con le altre. Nell'immagine 1.4 sottostante, è possibile vederne un'approssimazione.



**Figura 1.4:** Architettura d'esempio di un'applicazione Zucchetti di base

---

<sup>6</sup> Postgresql. URL: <http://www.postgresql.org/>.

<sup>7</sup> Apache Tomcat. URL: <http://tomcat.apache.org/>.

Nello specifico nella *web-application* dell'azienda: *HTML*<sup>8</sup> e *CSS*<sup>9</sup> gestiscono la struttura e lo stile delle pagine *web*. *JavaScript* è utilizzato per la logica *client-side*, come la gestione degli eventi e le richieste al *server*. Mentre *Java* è utilizzato per la logica di *business*. Mentre i *servlet*, gestiscono le richieste *HTTP*<sup>10</sup>, interagiscono con il *database* basato su *PostgreSQL*, e generano risposte, spesso sotto forma di pagine *JSP* o dati *JSON*, formato di testo "leggero", per lo scambio di dati.

- **Python**<sup>11</sup>: linguaggio di programmazione flessibile e dinamico, noto per la sua applicazione in vari campi come il *machine learning*, l'automazione e lo *scripting*, che l'azienda utilizza principalmente in progetti innovativi e di ricerca;
- **Jenkins**<sup>12</sup>: tool di integrazione continua *open-source* che automatizza il processo di *build*, test e *deployment* delle applicazioni. In Zucchetti utilizzato per assicurare che il *software* venga continuamente integrato e verificato, preservandone la qualità;
- **EsLint**<sup>13</sup>: Uno strumento di *linting* per *JavaScript* che aiuta a identificare e risolvere problemi nel codice. L'azienda lo integra nel processo di sviluppo per garantire che il codice *JavaScript* sia conforme agli standard interni e sia privo di errori potenziali, migliorando la manutenzione e la leggibilità del codice;

---

<sup>8</sup> *HTML*. URL: <https://html.spec.whatwg.org/multipage/>.

<sup>9</sup> *CSS*. URL: <https://www.w3.org/Style/CSS/Overview.en.html>.

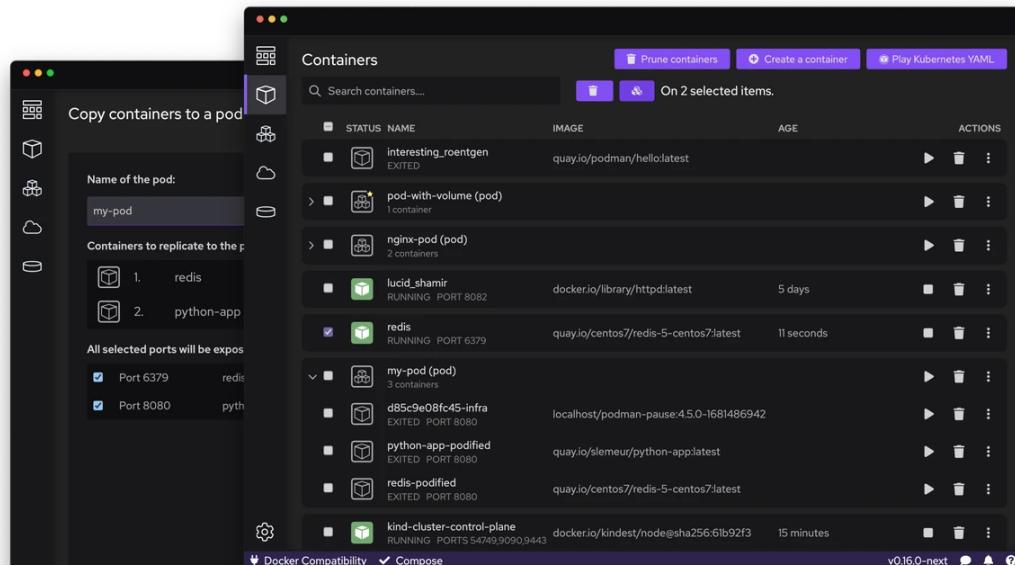
<sup>10</sup> *Protocollo HTTP*, sito di riferimento. URL: <https://www.w3.org/Protocols/>.

<sup>11</sup> *Python*. URL: <http://www.python.org/>.

<sup>12</sup> *Jenkins*. URL: <http://www.jenkins.io/>.

<sup>13</sup> *EsLint*. URL: <http://eslint.org/>.

- **Podman**<sup>14</sup>: Tecnologia di containerizzazione che permette di creare, distribuire e gestire applicazioni in ambienti isolati e portabili, detti *container*. Zucchetti ne fa uso per garantire che le applicazioni siano facilmente scalabili e distribuibili su diverse piattaforme, migliorando l'efficienza operativa e riducendo le incompatibilità tra ambienti di sviluppo e produzione;



**Figura 1.5:** *Podman Desktop*, interfaccia grafica dello strumento

**Fonte:** [podman.io](https://podman.io)

- **Kubernetes**<sup>15</sup>: Una piattaforma *open-source* per l'orchestrazione di *container* che automatizza la distribuzione, la gestione e il scaling delle applicazioni containerizzate. Utilizzato dall'azienda per gestire in modo efficace le applicazioni distribuite su larga scala, in progetti di ricerca e sviluppo.

Per quanto riguarda la gestione dei progetti e la collaborazione tra team, Zucchetti utilizza una serie di strumenti avanzati che facilitano il lavoro di squadra, la comunicazione e il versionamento del *software*. Questi strumenti sono fondamentali per mantenere l'efficienza operativa e garantire che tutti i membri del team siano allineati sugli obiettivi del progetto.

---

<sup>14</sup>Podman. URL: <http://podman.io/>.

<sup>15</sup>Kubernetes. URL: <http://kubernetes.io/>.

- **GitHub<sup>16</sup>**: Una piattaforma di *hosting* per il controllo di versione e la gestione del codice sorgente basata su *Git*. In Zucchetti, GitHub è utilizzato per la gestione collaborativa del codice, consentendo ai team di sviluppo di lavorare insieme in modo efficiente;
- **GitLab<sup>17</sup>**: Simile a *GitHub*, *GitLab* è un'altra piattaforma basata su *Git*, ma con un'attenzione particolare all'integrazione continua e alla distribuzione continua (CI/CD);
- **La suite Office di Microsoft<sup>18</sup>**: Un insieme di applicazioni di produttività che includono *Word*, *Excel*, *PowerPoint* e *Outlook*. In Azienda, la *suite* è utilizzata per una vasta gamma di attività amministrative e operative, come la redazione di documenti, l'analisi dei dati, la preparazione di presentazioni e la gestione della corrispondenza elettronica;
- **Microsoft Teams<sup>19</sup>**: Una piattaforma di collaborazione che integra *chat*, video conferenze, archiviazione di file e applicazioni di lavoro. Utilizzato dall'azienda per facilitare la comunicazione e la collaborazione sia tra i team, soprattutto in un contesto di lavoro remoto o distribuito, sia con esterni, quali clienti o collaboratori.

Degno di nota è anche l'adozione di applicazioni sviluppate internamente da Zucchetti, progettate per ottimizzare la visualizzazione e la gestione dei progetti. Questi strumenti personalizzati si integrano perfettamente con le piattaforme già in uso, come *GitHub*, permettendo all'azienda di sfruttare al meglio le proprie tecnologie e risorse, garantendo al contempo una gestione più efficiente e su misura delle attività di sviluppo.

---

<sup>16</sup> GitHub. URL: <http://github.com/>.

<sup>17</sup> GitLab. URL: <http://about.gitlab.com/>.

<sup>18</sup> Microsoft Office. URL: <http://www.microsoft.com/microsoft-365/>.

<sup>19</sup> Microsoft Teams. URL: <http://www.microsoft.com/microsoft-teams/>.

## 1.5 Propensione all'innovazione

Zucchetti, forte della sua affermazione e solidità finanziaria derivanti da una storia di numerosi successi, pone grande attenzione e risorse nell'innovazione, sia nell'ambito tecnologico che industriale. Questa propensione all'innovazione si riflette in diversi aspetti della sua organizzazione e delle sue operazioni.

In particolare, secondo i dati aziendali, dei 9000 dipendenti di Zucchetti, quasi un terzo è impiegato nel settore di ricerca e sviluppo (RD). Questo impegno significativo dimostra la volontà dell'azienda di investire costantemente in nuove tecnologie e soluzioni innovative. La sede di Padova, dove ho avuto l'occasione di svolgere il mio tirocinio, è un esempio tangibile di questo impegno, essendo essa stessa un centro dedicato alla ricerca e sviluppo. Questa esperienza mi ha permesso di osservare da vicino l'importanza che l'azienda attribuisce ai progetti e processi orientati all'innovazione.

Durante il mio *stage*, ho potuto constatare che molti dei progetti in corso erano focalizzati su aree all'avanguardia come l'automazione industriale, l'intelligenza artificiale e il *cloud computing*.

Zucchetti inoltre collabora attivamente con numerose entità esterne, tra cui università, centri di ricerca e aziende private a livello internazionale. Queste collaborazioni non solo ampliano le competenze e le risorse a disposizione dell'azienda, ma favoriscono anche uno scambio continuo di conoscenze e idee. Alcune delle collaborazioni degne di nota includono partnership con il Politecnico di Milano, l'Università di Bologna e l'Istituto Italiano di Tecnologia.

L'azienda partecipa regolarmente a progetti di ricerca europei e internazionali, ottenendo finanziamenti e riconoscimenti per le sue soluzioni innovative. Questi progetti coprono una vasta gamma di settori, dall'ICT (*Information and Communication Technology*) alla *smart manufacturing*, confermando la capacità di Zucchetti di adattarsi e innovare in vari contesti.

Uno dei più recenti e importanti, quello per la sicurezza degli stadi ai mondiali di calcio 2022. Durante i quali Zucchetti ha avuto la responsabilità di gestire gli accessi a 6 degli 8 stadi per la durata di tutta la competizione. Un team di esperti, visibili in figura 1.6, si è occupato dell'installazione e monitoraggio dei tornelli automatici per l'affluenza del pubblico.



**Figura 1.6:** Team Zucchetti ai mondiali di calcio 2022

**Fonte:** zucchetti.it

Inoltre, l'azienda, promuove e partecipa a conferenze, *workshop* e *hackathon*, eventi che incentivano la creatività e l'innovazione tra i suoi dipendenti e nella comunità tecnologica più ampia. Questi eventi offrono opportunità per esplorare nuove idee, sviluppare prototipi e collaborare con esperti del settore.

Grazie a questa strategia integrata di investimenti in RD, collaborazioni esterne e coinvolgimento attivo nella comunità tecnologica, Zucchetti riesce a mantenere un alto livello di innovazione, rispondendo efficacemente alle esigenze in continua evoluzione dei suoi clienti e del mercato globale.

# Capitolo 2

## Lo stage

### 2.1 La visione dell’azienda

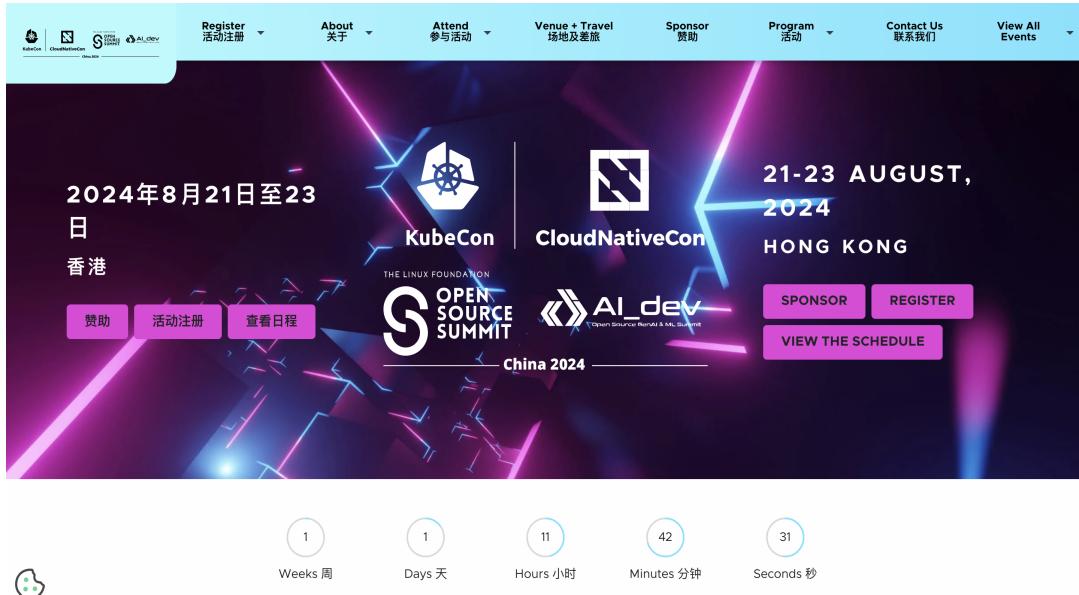
L’azienda, da sempre orientata all’innovazione tecnologica, ha riconosciuto da tempo l’importanza strategica del mondo *Cloud Native*<sup>1</sup>. Questa filosofia di sviluppo, che promuove la creazione di applicazioni resilienti, scalabili e facilmente gestibili nel *cloud*, rappresenta per Zucchetti un passo fondamentale verso il futuro dell’informatica aziendale. Zucchetti, come già descritto, fornisce ai suoi clienti una vasta gamma di servizi che richiedono aggiornamenti continui, operazioni di manutenzione e, talvolta, personalizzazioni su richiesta. Questi interventi non solo necessitano di risorse significative, ma possono anche avere tempi di applicazione variabili. L’obiettivo dell’azienda è garantire non solo l’eccellenza del software, ma anche la massima continuità ed affidabilità del servizio, mantenendo elevata l’efficienza e riducendo i consumi. Questo approccio non solo consente di beneficiare di un miglioramento economico, ma promuove anche la sostenibilità, un valore a cui l’azienda si è dedicata con impegno negli anni. Come illustrerò a breve, in questo contesto, il progetto che mi è stato affidato assume un ruolo centrale.

La visione dell’azienda è chiara: creare strumenti che permettano ai clienti di sfruttare appieno le potenzialità del *cloud*, senza doversi preoccupare di complessità tecniche o di gestione.

---

<sup>1</sup>Linux Foundation, CloudNative. URL: <https://www.cncf.io/>.

Attraverso questo progetto, Zucchetti punta ad innovarsi ulteriormente. Nello specifico la mia esperienza rappresenterà uno dei primi passi di ricerca verso questo mondo, quello del *Cloud Native*. Attualmente infatti, l'azienda è ancora in fase di esplorazione di queste tecnologie emergenti, che si ritrovano nel pieno del loro sviluppo. Basti pensare come, una delle più importanti conferenze 2.1 al riguardo, si terrà a breve, rispetto alla stesura di questo stesso documento.



**Figura 2.1:** Conferenze *KubeCon & CloudNativeCon, Hong Kong Agosto 2024*

**Fonte:** [events.linuxfoundation.org](https://events.linuxfoundation.org)

Riassumendo, lo *stage* mira a fornire esempi concreti, test e risultati che aiuteranno Zucchetti ad ottenere una visione più chiara e informata sul potenziale e sull'applicabilità di tali soluzioni nel loro contesto.

## 2.2 Il ruolo degli stage nel contesto aziendale

Gli *stage* rappresentano per l'azienda non solo una risorsa strategica per esplorare e innovare in ambiti tecnologici avanzati, ma anche un'opportunità per lo *scouting*, reclutamento, e formazione di nuovi giovani. Zucchetti utilizza gli *stage* come un canale per la sperimentazione di nuove tecnologie, approfittando di progetti che richiedono tempo e risorse i quali non sempre, le aziende

in generale, sono disposte o hanno la possibilità di investire.

L’azienda dimostra una visione lungimirante nel trattare gli *stage* come vere e proprie opportunità di sviluppo tecnologico. I progetti assegnati agli stagisti sono frequentemente orientati verso le ultime innovazioni, come l’intelligenza artificiale, il *cloud computing*, e l’Internet delle Cose (IoT), che rappresentano le frontiere più avanzate del settore tecnologico. Questi progetti non solo contribuiscono al progresso dell’azienda, ma permettono anche agli stagisti di lavorare su tecnologie all’avanguardia e di acquisire competenze preziose.

Durante il mio tirocinio, ho avuto l’opportunità di osservare da vicino come Zucchetti gestisce questi processi. Ho visto come l’azienda valorizza le capacità e le idee dei giovani, integrandoli in progetti reali e offrendo loro un ambiente di apprendimento stimolante. Allo stesso modo, ho potuto constatare il valore che gli altri stagisti apportano, contribuendo a progetti paralleli e acquisendo esperienza pratica nelle stesse aree innovative.

In alcuni casi, Zucchetti riconosce il potenziale degli stagisti e offre loro opportunità di lavoro al termine dello stage. Questo approccio consente all’azienda di accogliere talenti già formati e familiarizzati con i progetti e la cultura aziendale, garantendo una continuità di competenze e contribuendo alla crescita e all’innovazione continua.

In questo contesto, gli *stage* non sono solo un’opportunità per la crescita professionale degli stagisti, ma anche una strategia per l’azienda per rimanere all’avanguardia e alimentare una cultura di innovazione continua.

### 2.3 Scopo

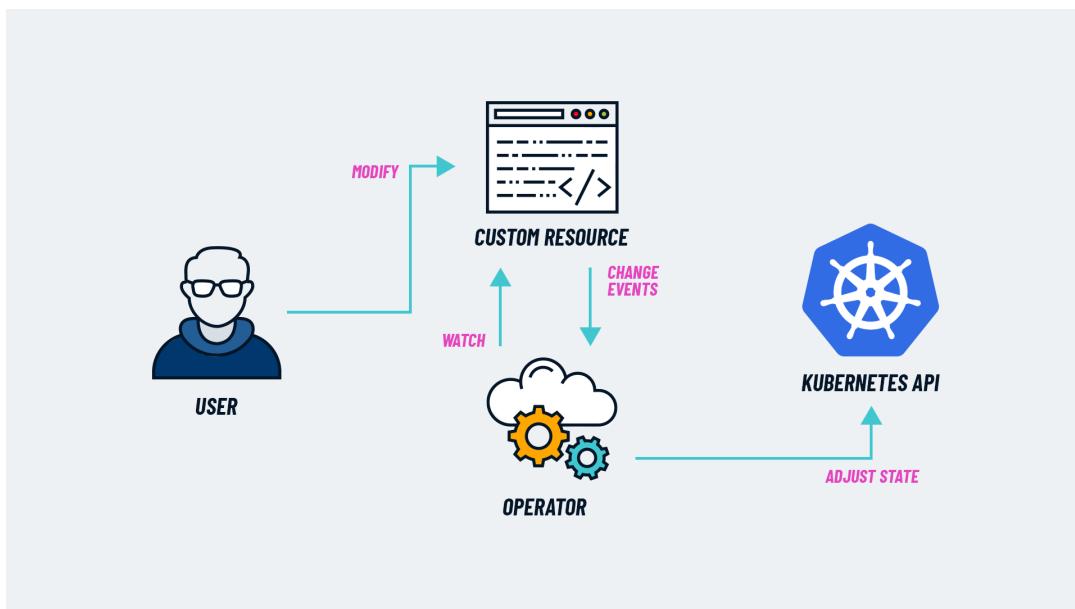
Il progetto previsto dal mio *stage* ha l’obiettivo di compiere uno studio di ricerca sugli operatori per *Kubernetes*, andandone a sviluppare uno che possieda alcune *feature* specificate dall’azienda. Gli operatori non sono altro che programmi che ne estendono le capacità native, per gestire applicazioni e servizi complessi attraverso la definizione e l’automazione di logiche di business specifiche e personalizzate.

Nello specifico questi ultimi, hanno il ruolo di automazzire alcuni processi che

soltanente richiederebbero l'intervento di un lavoratore addetto.

Essi sono composti da due componenti principali:

- **Custom Resource Definitions (CRD)**: che permettono di estendere l'API di *Kubernetes* per creare e gestire nuove risorse personalizzate.
- **Custom Controllers**: responsabili della gestione e del controllo dello stato delle risorse personalizzate definite attraverso i CRD. Un controller è un programma che osserva lo stato delle risorse personalizzate e applica le azioni necessarie per mantenerle nello stato desiderato.



**Figura 2.2:** Astrazione funzionamento di base degli operatori

**Fonte:** [www.cncf.io](http://www.cncf.io)

L'operatore pensato dall'azienda, avrà il compito di facilitare la gestione di applicazioni persistenti, all'interno dell'ambiente *Kubernetes*, garantendo non solo l'efficienza operativa, ma anche la continuità del servizio e la sicurezza dei dati.

Nello specifico l'azienda ha voluto porre l'attenzione su alcuni macro argomenti:

- **Scalabilità**: Rappresenta la capacità di un sistema di gestire un aumento della richiesta di risorse o di carico di lavoro senza compromettere le sue

prestazioni o stabilità. L'operatore dovrà supportare la crescita dinamica e la distribuzione equa delle risorse per rispondere efficacemente a variazioni nel carico di lavoro.

- **Gestione degli aggiornamenti:** Fondamentale per mantenere le applicazioni e l'infrastruttura al passo con le ultime versioni e patch. L'operatore dovrà facilitare e automatizzare l'aggiornamento delle applicazioni senza causare interruzioni del servizio, garantendo che i cambiamenti siano applicati in modo sicuro e senza impatti negativi sull'operatività.
- **Persistenza:** Capacità di mantenere i dati e lo stato delle applicazioni anche in caso di riavvii o guasti dei nodi (*server*). L'operatore deve assicurare che i dati siano memorizzati in modo duraturo e che l'accesso ai dati persistenti sia sempre affidabile e sicuro, anche in scenari di *failover* o recupero di emergenza.

Esso sarà sviluppato in *Go (Golang)*<sup>2</sup>, linguaggio di programmazione *open-source* sviluppato da *Google*, progettato per essere semplice, efficiente e altamente performante, attraverso l'uso del *Operator SDK*<sup>3</sup>, strumento che fornisce una serie di *tools* e *framework* per semplificare la creazione, il test e la distribuzione degli operatori.

---

<sup>2</sup>*Go (Golang)*. URL: <https://go.dev/>.

<sup>3</sup>*Operator SDK*. URL: <https://sdk.operatorframework.io/>.

## 2.4 Vincoli e obiettivi

Il primo *step* per avviare il tirocinio, prevede la stesura di un piano di lavoro che definisce vincoli e obiettivi dello *stage* e relativo progetto. Esso viene redatto dallo studente, insieme al proprio tutor aziendale, in modo preventivo. In quanto tale, lo stesso, subisce solitamente, come anche nel mio caso, modifiche che seguono e si adattano all'avanzare del progetto. Inoltre il documento deve venire approvato da tutor aziendale, tutor interno (relatore della tesi) e dall'amministratore degli stage universitari.

### Vincoli temporali:

Il tirocinio prevedeva una durata di 8 settimane per un totale di 320 ore, 40 l'una, a partire dal 10 Giugno 2024 fino al 2 Agosto 2024. L'orario di lavoro si teneva dalle 09:00 alle 18:00 con un'ora di pausa pranzo tra le 13:00 e le 14:00 e alcune pause ordinarie di breve durata, previste per tutti i dipendenti. Nello specifico tutti i giorni di lavoro sono stati svolti in loco ad eccezione di 2, svolti in modalità smart per necessità personali.

### Vincoli tecnologici:

L'azienda ha inoltre voluto definire il dominio tecnologico che avrei dovuto innanzitutto studiare, e successivamente, utilizzare per la realizzazione del progetto ma anche sua gestione. Sebbene fosse necessario aderire a certe tecnologie, l'azienda mi ha anche concesso un'ampia libertà su altre scelte, specialmente per le tecnologie complementari. Tra le tecnologie utilizzate vi sono:

- **GitHub:** Per il versionamento, salvataggio e condivisione di codice e documentazione prodotta;
- **Docker/Podman:** Strumenti essenziali per la containerizzazione delle applicazioni. Sono stati utilizzati per creare, gestire e distribuire i container, risultando necessari per garantire l'integrazione con *Kubernetes*;
- **Kubernetes:** piattaforma di orchestrazione container utilizzata per gestire e scalare le applicazioni containerizzate. Centrale per la gestione dei

servizi e dell'applicazione distribuita all'interno del progetto.

- **Minikube e Kubectl:** Strumenti proposti dal sottoscritto e successivamente approvati dall'azienda, per la gestione e la simulazione di un *cluster Kubernetes* in ambiente di sviluppo. *Minikube*<sup>4</sup> consente di eseguire un *cluster Kubernetes* locale, mentre *Kubectl*<sup>5</sup> è utilizzato per interagire con il *cluster* e gestire le risorse.
- **Go** (Golang): Per lo *scripting* dell'operatore.
- **MinIO**<sup>6</sup>: *software open-source* progettato per fornire una soluzione di storage compatibile con l'API di *Amazon S3*<sup>7</sup>. È utilizzato principalmente come *server* di *storage* per dati non strutturati, come file, oggetti e immagini. Usato nel progetto per la gestione di cartelle condivise, utilizzate nell'applicazione per gestire una parte della persistenza.
- **Datashim**<sup>8</sup>: *framework* utilizzato per mappare i *bucket*, contenitori utilizzati per memorizzare e organizzare oggetti in servizi di storage, con *MinIO* a dei *Persistent Volume Claim* utilizzabili successivamente dai *pod* dell'applicazione. I *Persistent Volume Claim* sono risorse *Kubernetes* che consentono alle applicazioni di richiedere e utilizzare spazio di archiviazione persistente all'interno di un *cluster*, rappresentato dai *Persistent Volumes*. Mentre i *pod* sono una raccolta di uno o più *container* che condividono lo stesso spazio di rete e le stesse risorse di storage. Rappresentano l'unità più piccola e basilare che può essere distribuita e gestita in un *cluster*.

---

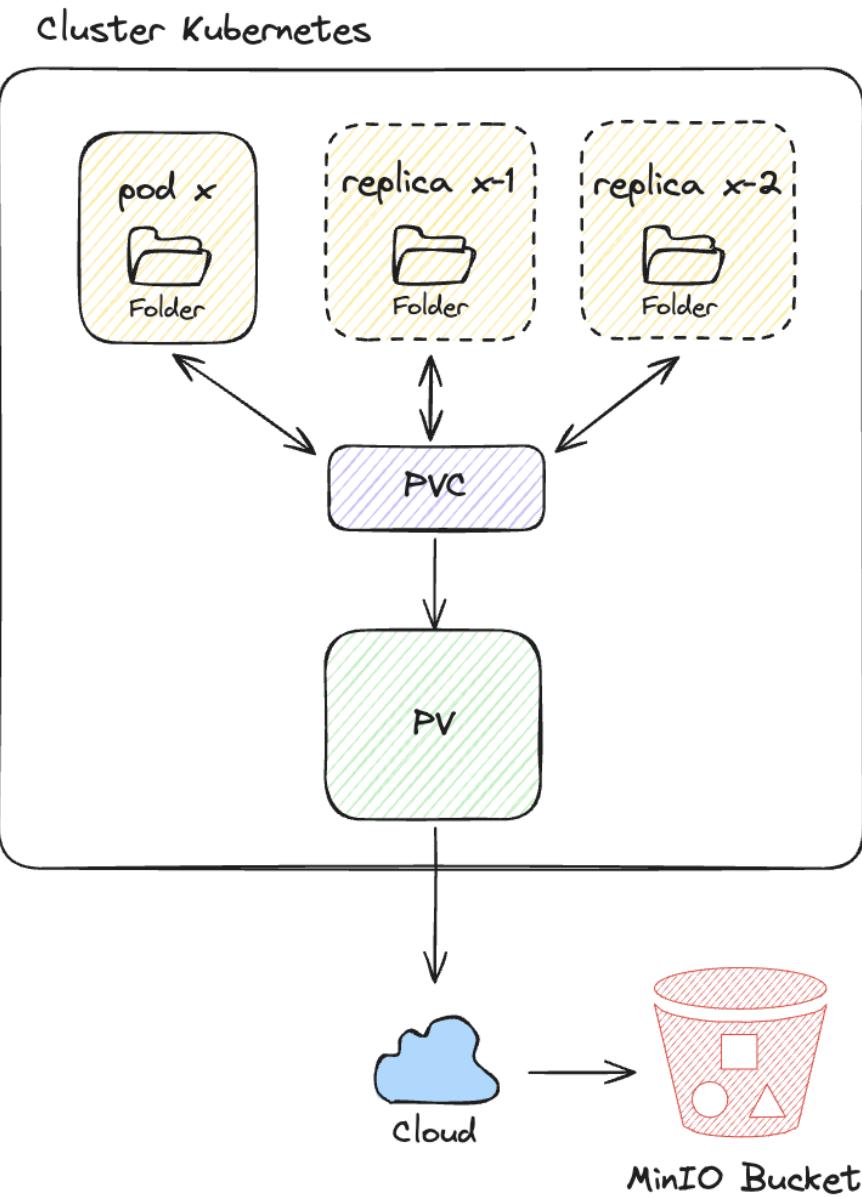
<sup>4</sup> *Minikube*. URL: <https://minikube.sigs.k8s.io/>.

<sup>5</sup> *Kubectl*. URL: <https://kubernetes.io/docs/reference/kubectl/>.

<sup>6</sup> *Minio*. URL: <https://min.io/>.

<sup>7</sup> *Amazon S3*. URL: <https://aws.amazon.com/it/pm/serv-s3/>.

<sup>8</sup> *Datashim*. URL: <https://github.com/datashim-io/datashim>.



**Figura 2.3:** Astrazione infrastruttura per la gestione delle cartelle condivise

Ed altre, per lo sviluppo di un applicazione di test a libera scelta. Per la quale, nello specifico, ho utilizzato:

- **React<sup>9</sup>**: Una libreria *JavaScript* per costruire interfacce utente interattive e dinamiche, utilizzata per sviluppare il *front-end* dell'applicazione di test;

---

<sup>9</sup>React. URL: <https://react.dev/>.

- **Express<sup>10</sup>**: *framework* minimalista per *Node.js<sup>11</sup>*, usato per gestire le richieste HTTP e costruire l'API del *back-end* dell'applicazione di test;
- **Postgresql**: DBMS per *database* relazionali avanzato e *open-source*, utilizzato per memorizzare e gestire i dati dell'applicazione di test.

Per la stesura della documentazione invece, non avendo alcuna specifica, ho adottato principalmente il *Markdown<sup>12</sup>*, un linguaggio leggero e semplice che si integra perfettamente con *GitHub*.

### Piano di lavoro:

Il documento prevede poi l'organizzazione dei periodi di progetto in attività, che riporto per com'è stata pensata inizialmente:

- **Prima Settimana - Introduzione e studio:**
  - Incontro con persone coinvolte nel progetto per discutere i requisiti e le richieste relativamente al sistema da sviluppare;
  - Verifica credenziali e strumenti di lavoro assegnati;
  - Presa visione dell'infrastruttura esistente;
  - Formazione sulle tecnologie adottate.
- **Seconda Settimana - Analisi e comprensione:**
  - Individuazione dominio di applicazioni *stateful*;
  - Individuazione requisiti e vincoli in relazione alle suddette applicazioni.
- **Terza Settimana - Primo approccio e scalabilità verticale:**
  - Primo approccio pratico al progetto;
  - Creazione e clonazione di *pod* con presistenza dei dati;

---

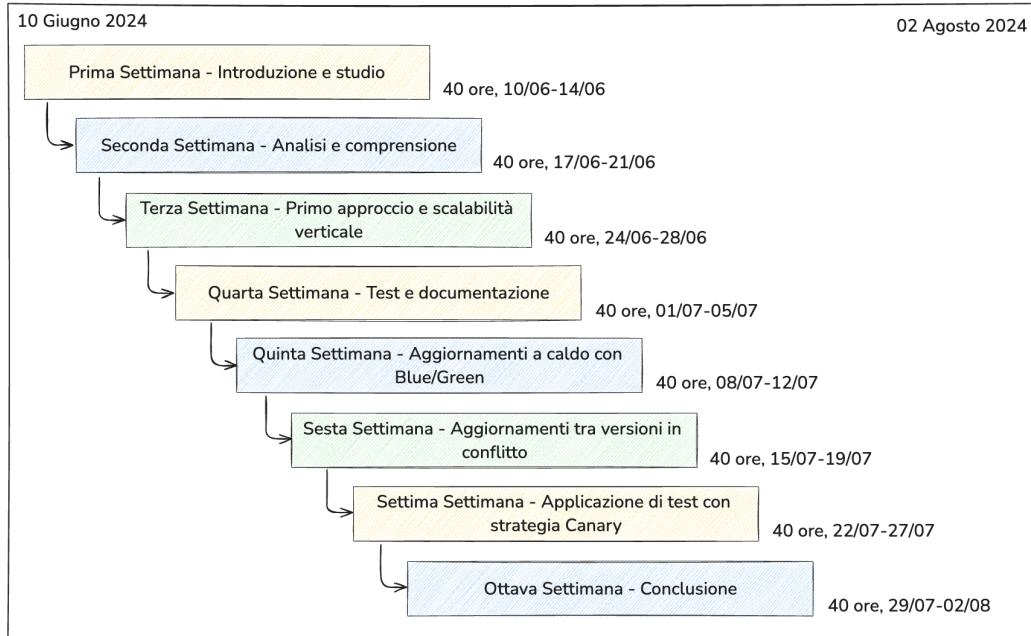
<sup>10</sup>Express. URL: <https://expressjs.com/>.

<sup>11</sup>Node. URL: <https://nodejs.org/en>.

<sup>12</sup>Markdown. URL: <https://www.markdownguide.org/>.

- *Scheduling* e scalabilità verticale dei pod con persistenza dei dati;
  - Sviluppo della prima versione dell'operatore.
- **Quarta Settimana - Test e documentazione:**
    - Test sulla clonazione di *pod* con persistenza dei dati;
    - Test sulla scalabilità verticale dei *pod* con persistenza dei dati;
    - Stesura di documentazione relativa ai progressi ottenuti.
  - **Quinta Settimana - Aggiornamenti a caldo con Blue/Green:**
    - Implementazione dello "switch" tra *pod*, a caldo;
    - Applicazione della strategia *blue/green*;
    - Gestione della persistenza a caldo.
  - **Sesta Settimana - Aggiornamenti tra versioni in conflitto:**
    - Implementazione degli aggiornamenti, a caldo;
    - Gestione delle differenze tra versioni delle immagini *Docker* contenute nei pod;
    - Valutazione e implementazione di strategie per la gestione delle conversioni tra versioni differenti.
  - **Settima Settimana - Applicazione di test con strategia *Canary*:**
    - Applicazione della strategia *Canary* per gli aggiornamenti;
    - Identificazione ed implementazione di test dei pod temporanei.
  - **Ottava Settimana - Conclusione:**
    - Test e verifica delle nuove funzionalità;
    - Stesura di documentazione relativa ai progressi ottenuti.

## CAPITOLO 2. LO STAGE



**Figura 2.4:** *Gantt view* del progetto

Come già accennato, il piano di lavoro ha subito alcune modifiche durante lo *stage* adattandosi allo sviluppo del progetto. Nel complesso le attività delineate preventivamente non hanno comunque subito degli stravolgimenti. Riporto le settimane e relative attività che hanno subito dei cambiamenti:

- **Seconda Settimana - Analisi e comprensione:** Alla quale è stata aggiunta l'attività: "Sviluppo di una basilare applicazione di test, che seguia l'architettura specificata dall'azienda.";
- **Terza Settimana - Primo approccio e scalabilità orizzontale:** La cui attività: "Scheduling e scalabilità veritcale dei *pod* con presistenza dei dati." è stata modificata in: "Scheduling e scalabilità orizzontale dei *pod* con presistenza dei dati.";
- **Quarta Settimana - Test e documentazione:** La cui attività: "Test sulla scalabilità veritcale dei *pod* con presistenza dei dati." è stata modificata in: "Test sulla scalabilità orizzontale dei *pod* con presistenza dei dati.". Alla quale, inoltre, è stata aggiunta l'attività: "Realizzazione di una pre-

sentazione, a scopo informativo, sul linguaggio di programmazione *Go*", poi esposta in azienda;

- **Ottava Settimana - Conclusione:** Alla quale è stata aggiunta l'attività: "Collaudo dell'operatore per la gestione di un'applicazione dell'azienda." .

**Obiettivi:**

Tali attività hanno portato all'individuazione di alcuni requisiti e obiettivi, stilati con la collaborazione del tutor aziendale, e approvati dal mio tutor interno e relatore, che riporto:

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 un codice sequenziale di numeri, identificativo del requisito.

Codice	Descrizione	Tipologia
RO1	Dominio e analisi dei requisiti di applicazioni stateful utili	Documentale
RO2	Gestione dei pod con operazioni di base quali creazione e eliminazione	Funzionale
RO3	Sviluppo di un operatore per kubernetes per automatizzare la gestione dei pod	Funzionale
RO4	Scalabilità verticale e scheduling dei pod	Funzionale
Continua nella prossima pagina...		

**Tabella 2.1 – Continuo della tabella**

Codice	Descrizione	Tipologia
RO5	Aggiornamento a caldo con strategia Blue/Green	Funzionale
RO6	Gestione di versioni incopatibili con attenzione alla persistenza dei dati	Funzionale
RO7	Applicazione della strategia Canary	Funzionale
RO8	Ideazione e implementazione di test su nuovi pod temporanei	Funzionale
RO9	Documentazione del codice e dei progressi ottenuti	Documentale
RD1	Test su pod e immagini con architetture differenti	Funzionale

**Tabella 2.1:** Tabella riassuntiva dei requisiti individuati durante la stesura del piano di lavoro iniziale

I requisiti individuati preventivamente durante la stesura del piano di lavoro, sono stati sottoposti durante lo svolgimento del progetto, ad un adattamento e raffinamento, in risposta al cambiamento di alcune attività.

Nello specifico:

- **RO4** è stato modificato, la nuova descrizione è divenuta: "Scalabilità orizzontale e scheduling dei pod.";
- **RO10** nuovo requisito funzionale: "Sviluppo di un'applicazione di test.";
- **RO11** nuovo requisito funzionale: "Gestione della persistenza dei dati, tramite cartelle condivise.";
- **RO12** nuovo requisito funzionale: "Gestione della persistenza dei dati, tramite *database*.";
- **RO6** è stato suddiviso in:

- **RO6A:** "Gestione di versioni incopatibili con attenzione alla persistenza dei dati - Cartelle condivise. ";
- **RO6B:** "Gestione di versioni incopatibili con attenzione alla persistenza dei dati - *Database*. ".

#### Prodotti Attesi:

Infine il documento elenca i prodotti attesi:

- Documentazione relativa al codice sviluppato complessivamente;
- Documentazione relativa allo studio iniziale delle tecnologie;
- *Environment* Kubernetes per il *testing* dell'operatore;
- Operatore realizzato in *Go*, munito di tutte le *feature* previste.

Alcuni di essi sono stati rivisti durante il corso dello *stage*. Sono state quindi apportate le seguenti modifiche:

- Aggiunto: "Presentazione a scopo informativo sul linguaggio di programmazione Go.".

## 2.5 Motivazione della scelta

Durante Stage-IT, evento organizzato dall'università che offre a noi studenti l'opportunità di incontrare e conoscere diverse aziende, tra le quali Zucchetti, ho avuto la possibilità di interagire direttamente con alcuni rappresentanti dell'azienda e di approfondire le loro proposte di *stage*. In particolare, Zucchetti presentava tre progetti distinti: il primo focalizzato sull'intelligenza artificiale, il secondo dedicato allo studio di tecnologie per la creazione di infografiche e il terzo, che ho scelto, relativo al mondo del *CloudNative*.

La prima cosa che mi colpì positivamente fu l'approccio dell'azienda nel presentare i progetti come linee guida piuttosto che come vincoli rigidi. Zucchetti ha incentivato fin da subito la proposta di idee innovative e la personalizzazione dei progetti, incoraggiando noi candidati a esplorare soluzioni originali e a contribuire con il nostro punto di vista. Questo approccio aperto e flessibile ha

reso l'opportunità di stage ancora più stimolante e allettante.

Inizialmente, ero maggiormente attratto dai progetti che trattavano l'intelligenza artificiale, tema molto discusso e ricorrente tra le varie aziende, i quali però tendevano a presentare molte somiglianze ed a risultare quindi, perlomeno per il sottoscritto, meno interessanti. Ciò unito alla mia curiosità per un ambito a me sconosciuto e meno esplorato di altri in campo universitario, anche in ottica magistrale, unita all'opportunità di confrontarmi con una tecnologia emergente e molto richiesta come quella del *cloud native*, mi ha spinto a scegliere il progetto relativo a Kubernetes. Questa scelta, a mio parere, si è rivelata estremamente gratificante. Oltre infatti ad avere appreso molto dal punto di vista tecnologico, ho avuto anche il modo di approcciarmi ad un ambiente lavorativo, che ho trovato, professionale, accogliente e soprattutto stimolante a contatto con persone disponibili, interessate, e di istruttiva esperienza, in un'azienda grande e rinomata come Zucchetti.

## 2.6 Obiettivi personali

Ho sempre pensato allo stage come ad un'opportunità di crescita sia professionale che personale di fondamentale importanza. Ha rappresentato il mio primo approccio al mondo del lavoro in un contesto aziendale, un ambiente che si discosta notevolmente dalla realtà accademica alla quale noi studenti siamo abituati. Sebbene non avessi la certezza esatta di cosa avrei affrontato, il mio obiettivo principale era massimizzare l'apprendimento e trarre il massimo vantaggio da questa esperienza.



**Figura 2.5:** Crescita professionale

**Fonte:** it.vecteezy.com

### Obiettivi Professionali:

- **Acquisizione di Competenze Tecniche:** Uno dei miei principali obiettivi era acquisire competenze pratiche in ambiti tecnologici di mio interesse e soprattutto, a me sconosciuti, così da ampliare il portfolio delle mie conoscenze.
- **Esperienza Pratica:** Essendo la prima volta che lavoravo in un contesto aziendale, volevo familiarizzare con le dinamiche di lavoro di questo ambiente. L'obiettivo era comprendere come le teorie apprese durante il percorso accademico si traducono in attività quotidiane e sfide concrete all'interno di un'azienda.
- **Conoscenza delle Pratiche Aziendali:** Ero inoltre curioso di apprendere come un'azienda gestisce i suoi progetti, coordina le attività tra i membri del team, e implementa soluzioni tecnologiche.

### Obiettivi Personal:

- **Sviluppo delle Competenze Trasversali:** Oltre alle competenze tecniche, puntavo a migliorare abilità trasversali come la comunicazione, il problem-solving e la gestione del tempo. Penso che questi aspetti che ho avuto modo di scoprire appena, in ambito universitario, principalmente tramite il corso di Ingegneria del Software, siano essenziali nel mondo del lavoro.
- **Adattamento a Nuovi Ambienti:** Volevo testare la mia capacità di adattarmi a un nuovo ambiente lavorativo e alle sue dinamiche. Questo includeva l'integrazione in gruppo già formato, l'approccio alle sfide e la capacità di lavorare sotto scadenze e direttive.
- **Crescita Professionale:** Inoltre volevo sfruttare l'ambiente di lavoro e l'opportunità di interagire con professionisti esperti per ricevere feedback costruttivi, consigli preziosi e ispirazioni che potessero orientare e arricchire la mia futura carriera.

Sono convinto che lo stage abbia rappresentato una fase cruciale per fare il ponte tra la formazione teorica e l'applicazione pratica, e sono grato per l'opportunità di averlo potuto vivere in un contesto così stimolante e innovativo.

# **Capitolo 3**

## **Il progetto**

### **3.1 Pianificazione**

Desc: Descrizione di come ho impostato la pianificazione sia inizialmente con il piano di progetto, sia durante lo stage stesso, rivedendo talvolta quanto pianificato precedentemente all'inizio dello stage.

#### **3.1.1 Metodologia**

Desc: Descrizione e elencazione della metodologia e strumenti utilizzati per la pianificazione del progetto (in relazione alla sua evoluzione)

#### **3.1.2 Rapporti con tutor e azienda**

Desc: Descrizione dei rapporti con il tutor relativi alla pianificazione e comunicazione, e in alcuni casi con altri dipendenti o supervisori dell'azienda.

#### **3.1.3 Revisioni**

Desc: Descrizione di come e quanto spesso sono state effettuate delle revisioni congiunte con il tutor e/o, altri supervisori dell'azienda.

## 3.2 Analisi dei requisiti

Desc: Descrizione di quanto fatto in relazione all'attività di analisi dei requisiti del progetto di stage, suddividendo applicazione di test e operatore kubernetes. (La prima essendo meno importante avrà meno spazio rispetto all'operatore)

### 3.2.1 Scenari e casi d'uso

Desc: Descrizione ed elencazione degli scenari e casi d'uso analizzati

### 3.2.2 Requisiti

Desc: Descrizione ed elencazione dei requisiti trovati e selezionati

## 3.3 Progettazione

Desc: Descrizione di quanto fatto in relazione all'attività di progettazione del progetto di stage, suddividendo applicazione di test e operatore kubernetes. (La prima essendo meno importante avrà meno spazio rispetto all'operatore)

## 3.4 Sviluppo

Desc: Descrizione di quanto fatto in relazione all'attività di sviluppo del progetto di stage, suddividendo applicazione di test e operatore kubernetes. (La prima essendo meno importante avrà meno spazio rispetto all'operatore)

## 3.5 Verifica

Desc: Descrizione di quanto fatto in relazione all'attività di verifica del progetto di stage, suddividendo analisi statica e dinamica. (Relativa solamente all'operatore)

### **3.5.1 Analisi Statica**

Desc: Descrizione di quanto fatto in relazione all'attività di verifica del progetto di stage, per ciò che riguarda l'analisi statica

### **3.5.2 Analisi Dinamica**

Desc: Descrizione di quanto fatto in relazione all'attività di verifica del progetto di stage, per ciò che riguarda l'analisi dinamica

## **3.6 Validazione e collaudo**

Desc: Descrizione di quanto fatto in relazione all'attività di validazione del progetto di stage, e successivo collaudo

## **3.7 Risultati ottenuti**

Desc: Descrizione ed elencazione dei risultati ottenuti

# **Capitolo 4**

## **Valutazioni retrospettive**

### **4.1 Soddisfacimento degli obiettivi**

Desc: Descrizione ed elencazione degli obiettivi soddisfatti e non

### **4.2 Crescita personale**

Desc: Descrizione della maturazione personale confrontando il prima e il dopo stage

### **4.3 Riflessioni finali**

Desc: Riflessioni finali sulle competenze necessarie rispetto a quelle fornitemi dal corso di studi

# Bibliografia

# Sitografia

*Acquisizione dell’azienda Ranocchi.* URL: <https://www.zucchetti.it/it/cms/news-e-eventi/comunicati/zucchetti-acquisisce-il-gruppo-ranocchi.html> (cit. a p. 1).

*Amazon S3.* URL: <https://aws.amazon.com/it/pm/serv-s3/> (cit. a p. 22).

*Apache Tomacat.* URL: <http://tomcat.apache.org/> (cit. a p. 10).

*CSS.* URL: <https://www.w3.org/Style/CSS/Overview.en.html> (cit. a p. 11).

*Datashim.* URL: <https://github.com/datashim-io/datashim> (cit. a p. 22).

*EsLint.* URL: <http://eslint.org/> (cit. a p. 11).

*Express.* URL: <https://expressjs.com/> (cit. a p. 24).

*GitHub.* URL: <http://github.com/> (cit. a p. 13).

*GitLab.* URL: <http://about.gitlab.com/> (cit. a p. 13).

*Go (Golang).* URL: <https://go.dev/> (cit. a p. 20).

*HTML.* URL: <https://html.spec.whatwg.org/multipage/> (cit. a p. 11).

*Java.* URL: <http://www.java.com/> (cit. a p. 9).

*JavaScript.* URL: <http://www.javascript.com/> (cit. a p. 9).

*Jenkins.* URL: <http://www.jenkins.io/> (cit. a p. 11).

*Kubectl.* URL: <https://kubernetes.io/docs/reference/kubectl/> (cit. a p. 22).

*Kubernetes.* URL: <http://kubernetes.io/> (cit. a p. 12).

*Linux Foundation, CloudNative.* URL: <https://www.cncf.io/> (cit. a p. 16).

*Manifesto Agile.* URL: <http://agilemanifesto.org/> (cit. a p. 6).

*Markdown.* URL: <https://www.markdownguide.org/> (cit. a p. 24).

*Microsoft Office.* URL: <http://www.microsoft.com/microsoft-365/> (cit. a p. 13).

*Microsoft Teams.* URL: <http://www.microsoft.com/microsoft-teams/> (cit. a p. 13).

*Minikube.* URL: <https://minikube.sigs.k8s.io/> (cit. a p. 22).

*Minio.* URL: <https://min.io/> (cit. a p. 22).

*Node.* URL: <https://nodejs.org/en> (cit. a p. 24).

*Operator SDK.* URL: <https://sdk.operatorframework.io/> (cit. a p. 20).

*Podman.* URL: <http://podman.io/> (cit. a p. 12).

*Postgresql.* URL: <http://www.postgresql.org/> (cit. a p. 10).

*Protocollo HTTP, sito di riferimento.* URL: <https://www.w3.org/Protocols/> (cit. a p. 11).

*Python.* URL: <http://www.python.org/> (cit. a p. 11).

*React.* URL: <https://react.dev/> (cit. a p. 23).

*Scrum.* URL: <http://scrum.org/> (cit. a p. 6).