

Università degli Studi di Padova

DIPARTIMENTO DI MATEMATICA “TULLIO LEVI-CIVITA”

CORSO DI LAUREA IN INFORMATICA



Sviluppo di una piattaforma di monitoraggio  
del traffico utente mediante APM e IA

*Tesi di laurea*

*Relatore*

Prof. Marco Zanella

*Laureando*

Marco Cola

*Matricola* 2079237

---

ANNO ACCADEMICO 2024-2025



Lorem ipsum dolor sit amet, consectetur adipiscing elit.

— Oscar Wilde

# Sommario

Il presente documento descrive il lavoro svolto durante il periodo di stage, della durata di 320 ore, dal laureando Marco Cola presso l'azienda Kirey S.r.l.

L'obiettivo principale dello stage è stato lo sviluppo di una piattaforma per il monitoraggio e l'analisi dei dati di navigazione relativi ad una *web application*, con particolare attenzione alla raccolta, indicizzazione ed elaborazione dei *log*.

Il progetto si è articolato in quattro fasi principali:

- Una prima fase di preparazione dell'ambiente di lavoro, comprendente l'individuazione, l'installazione e la configurazione delle componenti *software* necessarie, con successiva verifica del corretto funzionamento e della connettività tra i moduli;
- Una seconda fase di implementazione dell'estrazione dei dati, realizzata attraverso la configurazione di agenti di raccolta, la creazione di *pipeline* di *log* tramite *Logstash* e l'invio dei dati a *Elasticsearch*, con validazione del processo di acquisizione e indicizzazione;
- Una terza fase di elaborazione e rappresentazione grafica dei dati, che ha previsto lo sviluppo di *script* per la generazione di traffico utente e monitoraggio, l'analisi e aggregazione dei dati su *Elasticsearch* e la realizzazione di dashboard avanzate in *Kibana* con metriche di performance, accesso e flussi utente. Sono state inoltre configurate regole di *alerting* e notifiche per la rilevazione in tempo reale di anomalie mediante *Machine Learning*;
- Una fase finale di documentazione tecnica del progetto, contenente la descrizione delle tecnologie e dei prodotti utilizzati, la rappresentazione dei flussi logici dell'applicazione, nonché un'analisi dei pro e contro di ciascuna componente e delle principali criticità riscontrate.

Lo stage ha permesso di acquisire competenze trasversali nell'ambito dell'osservabilità delle applicazioni *web*, approfondendo l'utilizzo dello *stack Elastic* (*Elasticsearch*, *Logstash*, *Kibana*) e l'integrazione con strumenti di monitoraggio automatizzato, con un approccio volto alla creazione di soluzioni scalabili, robuste e orientate al miglioramento continuo delle prestazioni.

*“Quando incontri un uomo con la spada, estrai la tua spada:  
non recitare poesie a chi non è poeta”*

— proverbio Ch'an

## Ringraziamenti

*Innanzitutto, vorrei esprimere la mia gratitudine al Prof. Marco Zanella, 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.*

*Padova, Dicembre 2025*

Marco Cola

# Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
1.1	L'azienda . . . . .	1
1.1.1	Storia e servizi . . . . .	1
1.2	L'idea . . . . .	2
1.3	Organizzazione del testo . . . . .	3
1.3.1	Convenzioni tipografiche . . . . .	3
<b>2</b>	<b>Descrizione dello stage</b>	<b>4</b>
2.1	Introduzione al progetto . . . . .	4
2.2	Pianificazione . . . . .	4
2.3	Analisi preventiva dei rischi . . . . .	5
2.4	Requisiti e obiettivi . . . . .	6
<b>3</b>	<b>Analisi dei requisiti</b>	<b>8</b>
3.1	Tracciamento dei requisiti . . . . .	8
3.2	Requisiti funzionali . . . . .	10
3.3	Requisiti non funzionali . . . . .	11
3.4	Requisiti qualitativi . . . . .	12
3.5	Requisiti di vincolo . . . . .	13
3.6	Riepilogo dei requisiti . . . . .	14
3.7	Rappresentazione Architetturale . . . . .	15
3.7.1	Scelta dell'approccio architetturale . . . . .	15
3.7.2	Diagrammi architetturali principali . . . . .	15
3.7.3	Mappatura dei requisiti . . . . .	15
3.8	Conclusioni . . . . .	15
<b>4</b>	<b>Progettazione e codifica</b>	<b>16</b>
4.1	Tecnologie e strumenti . . . . .	16
4.2	Ciclo di vita del software . . . . .	16
4.3	Progettazione . . . . .	16
4.4	Design Pattern utilizzati . . . . .	16
4.5	Codifica . . . . .	16
<b>5</b>	<b>Verifica e validazione</b>	<b>17</b>
<b>6</b>	<b>Conclusioni</b>	<b>18</b>
6.1	Consuntivo finale . . . . .	18
6.2	Raggiungimento degli obiettivi . . . . .	18
6.3	Conoscenze acquisite . . . . .	18

<i>INDICE</i>	vi
6.4 Valutazione personale . . . . .	18
<b>A Appendice A</b>	<b>19</b>
<b>Acronimi e abbreviazioni</b>	<b>20</b>
<b>Glossario</b>	<b>21</b>
<b>Bibliografia</b>	<b>22</b>

# Elenco delle figure

1.1	Figura 1.1 - Logo di Kirey S.r.l. . . . .	<a href="#">1</a>
-----	---	-------------------

# Elenco delle tabelle

3.1	Tabella del tracciamento dei requisiti funzionali . . . . .	<a href="#">10</a>
3.2	Tabella del tracciamento dei requisiti non funzionali . . . . .	<a href="#">11</a>
3.3	Tabella del tracciamento dei requisiti qualitativi . . . . .	<a href="#">12</a>
3.4	Tabella del tracciamento dei requisiti di vincolo . . . . .	<a href="#">13</a>
3.5	Riepilogo dei requisiti . . . . .	<a href="#">14</a>



# Capitolo 1

## Introduzione

### 1.1 L'azienda

Kirey Group è uno dei *system integrator* europei più dinamici e in crescita, specializzato nell'accompagnare le imprese nei percorsi di trasformazione digitale e di adozione di modelli *data-driven*. Con sede principale in Italia e una presenza consolidata in diversi Paesi europei ed extraeuropei, il Gruppo conta oltre 1500 dipendenti ed opera in dieci Paesi.

La missione di Kirey è rendere l'innovazione accessibile, trasformando il potenziale tecnologico in valore economico e in nuovi modelli di *business*. L'azienda si distingue per un approccio che unisce affidabilità tecnica, innovazione, competenza centrata sul lavoro delle persone e sinergia cross-funzionale, elementi che costituiscono i valori fondanti del marchio.

Il manifesto del gruppo sintetizza questa filosofia nel concetto “*Data Made Human*”, ovvero la volontà di tradurre la complessità dei dati in soluzioni comprensibili, intuitive e ad alto impatto, mettendo sempre la persona al centro della tecnologia.



**Figura 1.1:** Figura 1.1 - Logo di Kirey S.r.l.

#### 1.1.1 Storia e servizi

La storia del gruppo affonda le radici negli anni Settanta e, attraverso fusioni, acquisizioni e nuove fondazioni, ha portato alla nascita di Kirey Group nel 2016. Negli anni successivi l'azienda ha accelerato la propria espansione internazionale integrando nuove realtà, consolidando così competenze e capacità operative in diversi settori e mercati.

Il portafoglio di servizi è ampio e integrato, con *Data & AI* come filo conduttore e aree principali che comprendono:

- *Cloud & Infrastructure*, con soluzioni ibride e *on-premise*, sicurezza in ambienti *cloud*, migrazione e monitoraggio;
- *Software Development*, che spazia dallo sviluppo *agile* e *mobile* alla *system integration*, con particolare attenzione alla qualità e all'automazione dei *test*;
- *Cybersecurity*, con servizi di consulenza, *audit*, architetture sicure, *managed services* e sistemi antifrode;
- *Data & AI*, che include *data integration*, *data governance*, *analytics*, *machine learning*, *synthetic data*, *forecasting* e soluzioni [Environmental, Social and Governance \(ESG\)](#)<sup>[g]</sup>.

Kirey Group pone grande attenzione alla sostenibilità, alla trasparenza e all'integrità, adottando pratiche responsabili nei confronti di clienti, partner, dipendenti e *stakeholder*. L'azienda è inoltre attivamente impegnata in progetti sociali, promuove la diversità e l'inclusione, e investe nello sviluppo delle competenze tecnologiche e professionali dei propri collaboratori.

Oggi il gruppo conta oltre 1370 casi di business realizzati, 10 *Innovation Center* attivi, un fatturato di circa 126 milioni di euro e più di 1500 collaboratori distribuiti in 10 paesi.

## 1.2 L'idea

L'idea alla base dello stage consiste nello sviluppo di una piattaforma per il monitoraggio intelligente del traffico utente di un *e-commerce*. L'obiettivo principale è quello di sfruttare algoritmi di Intelligenza Artificiale e di *Machine Learning* per individuare e segnalare automaticamente eventuali anomalie nei dati raccolti.

La piattaforma sarà in grado di analizzare i flussi in tempo reale, rilevando accessi sospetti, rallentamenti e potenziali minacce, così da consentire interventi tempestivi e garantire sia la sicurezza sia le prestazioni ottimali del sistema.

Il progetto è stato organizzato in quattro fasi principali:

- Una prima fase di formazione e preparazione dell'ambiente di lavoro, utile a familiarizzare con le tecnologie e i prodotti utilizzati, che comprende l'individuazione, l'installazione e la configurazione delle componenti necessarie, con successiva verifica della connettività tra i moduli;
- Una fase di analisi e progettazione, in cui saranno definite le specifiche funzionali e la soluzione tecnica tramite la configurazione degli agenti di raccolta, la realizzazione di *pipeline* di *log* e la loro indicizzazione in *Elasticsearch*;
- Una fase di realizzazione e test della piattaforma, con sviluppo di *script* per il monitoraggio sintetico, analisi su *Elasticsearch*, creazione di *dashboard* in *Kibana* e configurazione di regole di *alerting*;
- Infine la stesura della documentazione tecnica e funzionale, contenente la descrizione delle tecnologie adottate, dei flussi logici implementati e delle criticità riscontrate.

Per lo sviluppo saranno impiegati linguaggi come *Python* e *Java*, sistemi *Linux* e i prodotti della suite *Elastic Stack*, strumenti particolarmente adatti per l'elaborazione e il monitoraggio di grandi volumi di dati in tempo reale.

## 1.3 Organizzazione del testo

**Il secondo capitolo** approfondisce il progetto di stage, descrivendo gli obiettivi e le attività svolte, la metodologia di lavoro adottata e un'analisi preventiva dei principali rischi e relative strategie di mitigazione;

**Il terzo capitolo** è dedicato all'analisi dei requisiti della piattaforma e alle motivazioni che ne hanno guidato le scelte progettuali;

**Il quarto capitolo** approfondisce ...

**Il quinto capitolo** approfondisce ...

**Nel sesto capitolo** viene descritta ...

### 1.3.1 Convenzioni tipografiche

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

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

## Capitolo 2

# Descrizione dello stage

*Il capitolo approfondisce il progetto di stage, descrivendo gli obiettivi e le attività svolte, la metodologia di lavoro adottata e un'analisi preventiva dei principali rischi e relative strategie di mitigazione.*

### 2.1 Introduzione al progetto

Lo stage è stato svolto presso l'azienda Kirey Group S.r.l., realtà consolidata nell'ambito della fornitura di prodotti e servizi informatici, con clienti internazionali e una forte specializzazione nei settori *Banking*, *Insurance*, *Oil&Gas* e Pubblica Amministrazione. L'attività si è inserita nel contesto del team [Application Performance Monitoring \(APM\)](#)<sup>[g]</sup> e ha avuto come obiettivo principale la realizzazione e il collaudo di una piattaforma per il monitoraggio delle performance di una *web application*.

Il progetto è stato sviluppato interamente in ambiente *Linux* mediante *Windows Subsystem for Linux*, utilizzando la suite *Elastic Stack* e i suoi principali componenti: *Elasticsearch* per la gestione dei dati, *Kibana* per la visualizzazione, *Logstash* per l'ingestione e la trasformazione dei log, *Beats* e *Fleet Server* per la raccolta distribuita delle metriche e *APM Server/Agent* per il tracciamento delle *performance* applicative. Sono stati realizzati sviluppi in *Python* e *Java* per l'estensione delle funzionalità e l'integrazione di algoritmi di [Artificial Intelligence \(AI\)](#)<sup>[g]</sup> e *Machine Learning*, con l'obiettivo di rilevare automaticamente anomalie e problematiche di prestazione.

La finalità complessiva è quella di fornire un sistema scalabile, proattivo e ben documentato, capace di garantire prestazioni ottimali e un monitoraggio continuo della *web application*.

### 2.2 Pianificazione

Tutte le attività sono state condotte in affiancamento ad un tutor aziendale che ha curato sia la parte di formazione che di indirizzamento delle attività. A tal fine sono stati svolti dei momenti di confronto settimanali per la valutazione dello stato di avanzamento delle attività e momenti quotidiani di confronto sulle problematiche riscontrate.

Le attività proposte sono state collocate all'interno di un progetto più ampio portato avanti in Kirey da un team di persone eterogeneo.

Al termine dello stage sono stati presentati i risultati ottenuti a tutto il team. L'infrastruttura tecnologica e le piattaforme su cui girerà l'applicazione sono state messe a disposizione da Kirey.

## 2.3 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. Inesperienza nella suite Elastic

**Descrizione:** La limitata esperienza iniziale nell'utilizzo della *Elastic Stack* (*Elasticsearch*, *Kibana*, *Logstash*, *APM*) potrebbe comportare difficoltà nella configurazione e nell'integrazione delle componenti, rallentando lo sviluppo del progetto.

**Impatto:** Alto.

**Probabilità:** Alta.

**Soluzione:** Organizzazione di momenti di confronto con il tutor aziendale e studio personale della documentazione, al fine di acquisire le competenze necessarie.

### 2. Integrazione tra componenti Elastic

**Descrizione:** La comunicazione tra i diversi moduli della *suite Elastic* (*Elasticsearch*, *Kibana*, *Logstash*, *APM*) potrebbe presentare problemi di configurazione, causando ritardi o malfunzionamenti nell'acquisizione dei dati.

**Impatto:** Medio.

**Probabilità:** Media.

**Soluzione:** Esecuzione di test di connettività e validazione progressiva delle *pipeline*, con il supporto del *tutor* aziendale per la risoluzione dei problemi.

### 3. Qualità e coerenza dei dati raccolti

**Descrizione:** I dati acquisiti dagli agenti potrebbero risultare incompleti, duplicati o non coerenti, compromettendo le analisi e le *dashboard*.

**Impatto:** Alto.

**Probabilità:** Media.

**Soluzione:** Definizione di regole di filtraggio e validazione all'interno delle *pipeline Logstash* ed esecuzione di test di integrità sugli indici *Elasticsearch*.

### 4. Scalabilità e carico del sistema

**Descrizione:** L'aumento del volume dei dati e delle richieste potrebbe impattare sulle prestazioni della piattaforma, riducendo l'efficienza del monitoraggio.

**Impatto:** Medio.

**Probabilità:** Media.

**Soluzione:** Implementazione di strategie di *scaling* orizzontale e utilizzo di metriche di *Kibana* per monitorare l'impatto del carico in tempo reale.

### 5. Implementazione di algoritmi di Machine Learning

**Descrizione:** L'integrazione di modelli di *Machine Learning* per il rilevamento delle anomalie potrebbe richiedere competenze specifiche e tempi di sviluppo più lunghi del previsto.

**Impatto:** Alto.

**Probabilità:** Media.

**Soluzione:** Formazione preliminare su tecniche di *Machine Learning* e utilizzo di librerie e *framework* consolidati per accelerare lo sviluppo.

### 6. Problemi di configurazione delle pipeline Logstash

**Descrizione:** Errori di configurazione nelle *pipeline* di *Logstash* potrebbero causare la perdita, la duplicazione o la trasformazione errata dei dati raccolti, compromettendo l'affidabilità delle analisi.

**Impatto:** Alto.

**Probabilità:** Media.

**Soluzione:** Adozione di un approccio con *test* di validazione a ogni modifica delle *pipeline* e utilizzo di ambienti di prova per verificare la correttezza del flusso dei dati prima della messa in produzione.

### 7. Accesso limitato a funzionalità premium di Elastic

**Descrizione:** Durante le prime settimane di lavoro in locale, la mancata possibilità di operare su *Elastic Cloud* e di accedere a funzionalità premium come il *Machine Learning* potrebbe limitare l'analisi dei dati e rallentare la validazione di alcune funzionalità previste dal progetto.

**Impatto:** Medio.

**Probabilità:** Bassa.

**Soluzione:** Esecuzione preventiva delle attività in locale con dati di test, studio della documentazione sulle funzionalità premium e pianificazione di un passaggio successivo a *Elastic Cloud* non appena disponibile, con supporto del tutor aziendale.

## 2.4 Requisiti e obiettivi

Gli obiettivi del progetto sono stati classificati in base alla loro priorità secondo le seguenti notazioni:

- **Ob** (Requisiti Obbligatori) - requisiti essenziali e imprescindibili per il successo del progetto, vincolanti in quanto obiettivo primario richiesto dal committente;
- **D** (Requisiti Desiderabili) - requisiti importanti ma non critici, la cui assenza non compromette il progetto, non vincolanti o strettamente necessari, ma dal riconoscibile valore aggiunto;

- **Op** (Requisiti Opzionali) - requisiti desiderabili ma non essenziali, rappresentanti valore aggiunto non strettamente competitivo.

Durante il periodo di stage si prevede il raggiungimento dei seguenti obiettivi:

- **Preparazione dell'ambiente per l'implementazione della soluzione:**
  - **Ob1.1:** Individuazione delle componenti ed eventuali librerie da utilizzare;
  - **Ob1.2:** Installazione e configurazione delle componenti;
  - **Ob1.3:** Verifica del corretto funzionamento dell'ambiente e test di connettività tra componenti.
- **Implementazione estrazione dati dalla web application:**
  - **Ob2.1:** Configurazione *agent* (*Beats/APM*) per la raccolta dati della navigazione;
  - **Ob2.2:** Implementazione *pipeline* di *log* tramite *Logstash* per filtraggio e inoltro dati in *Elasticsearch*;
  - **Ob2.3:** Verifica della corretta acquisizione dei dati e loro indicizzazione in *Elasticsearch*.
- **Implementazione elaborazione dati e rappresentazione grafica dei dati:**
  - **Ob3.1:** Sviluppo *script* *Python/Java* per il monitoraggio sintetico (*Selenium*) e generazione di traffico *log*;
  - **Ob3.2:** Analisi e aggregazione dati in *Elasticsearch*, con *query* e visualizzazioni preliminari;
  - **Ob3.3:** Creazione dashboard avanzate su *Kibana* con metriche di *performance*, accesso e flussi utente;
  - **Op3.4:** Configurazione regole di *alerting* e notifiche per anomalie rilevate in tempo reale.
- **Documentazione dettagliata:**
  - **Ob4.1:** Descrizione delle tecnologie e prodotti utilizzati;
  - **Ob4.2:** Descrizione dei flussi logici del progetto e delle funzionalità dell'applicazione;
  - **D4.3:** Pro/contro di ogni componente e criticità nell'applicazione.

Nel capitolo successivo verrà approfondita l'analisi dei requisiti identificati in questa sezione, al fine di definirne in modo più strutturato la natura e le caratteristiche.

## Capitolo 3

# Analisi dei requisiti

*Il capitolo è dedicato all'analisi dei requisiti della piattaforma, con l'obiettivo di fornire una visione completa e dettagliata delle funzionalità e delle caratteristiche attese dal sistema. Verranno illustrate le esigenze degli utenti e del contesto operativo, evidenziando le specifiche tecniche e le funzionalità che hanno guidato le scelte progettuali.*

### 3.1 Tracciamento dei requisiti

A seguito di un'attenta attività di analisi del progetto e degli obiettivi tecnici e funzionali prefissati, sono state redatte le tabelle di tracciamento che riassumono in modo strutturato i requisiti individuati.

Durante questa fase sono state identificate differenti tipologie di requisiti, distinte sia in base alla loro categoria (funzionale, non funzionale, qualitativo o di vincolo), sia in base alla loro priorità di implementazione (obbligatorio, desiderabile o opzionale). Per garantire una tracciabilità chiara e univoca, a ciascun requisito è stato assegnato un codice identificativo composto da lettere che ne descrivono la tipologia e l'importanza, secondo la seguente convenzione:

**R** = Requisito

**F** = Funzionale

**N** = Non funzionale

**Q** = Qualitativo

**V** = Vincolo

**O** = Obbligatorio

**D** = Desiderabile

**Z** = Opzionale



Ogni requisito analizzato sarà identificato univocamente dalla seguente notazione:

**R[Priorità][Categoria]-[Numero]**

Dove:

- **Priorità** indica la priorità di implementazione, che può essere O = Obbligatorio, D = Desiderabile, Z = Opzionale;
- **Categoria** la categoria di appartenenza, che può essere F = Funzionale, N = Non funzionale, Q = Qualitativo, V = Vincolo;
- **Numero** un numero progressivo che identifica in modo univoco il requisito all'interno della sua categoria.

Nelle tabelle [3.1](#), [3.2](#), [3.3](#) e [3.4](#) sono riportati in modo sintetico tutti i requisiti emersi dall'analisi, classificati in base alla loro priorità e accompagnati da una breve descrizione della relativa funzionalità o vincolo tecnico.

## 3.2 Requisiti funzionali

Descrivono cosa deve fare il sistema. Sono le funzionalità concrete che la soluzione deve offrire per raggiungere gli obiettivi del progetto.

**Tabella 3.1:** Tabella del tracciamento dei requisiti funzionali

Codice	Descrizione	Classificazione
RO1	Individuazione delle componenti ed eventuali librerie da utilizzare	Obbligatorio
RO2	Installazione e configurazione delle componenti	UC2
RO3	Verifica del corretto funzionamento dell'ambiente e test di connettività tra componenti	UC3
RO4	Configurazione agent (Beats/APM) per la raccolta dati della navigazione	UC3
RO5	Implementazione pipeline di log tramite Logstash per filtraggio e inoltro dati in Elasticsearch	UC3
RO6	Verifica della corretta acquisizione dei dati e loro indicizzazione in Elasticsearch	UC3
RO7	Sviluppo script Python/Java per il monitoraggio sintetico (Selenium) e generazione di traffico log	UC3
RO8	Analisi e aggregazione dati in Elasticsearch, con query e visualizzazioni preliminari	UC3
RO8	Analisi e aggregazione dati in Elasticsearch, con query e visualizzazioni preliminari	UC3
RO9	Creazione dashboard avanzate su Kibana con metriche di performance, accesso e flussi utente	UC3
RO10	Descrizione delle tecnologie e prodotti utilizzati	UC3
RO11	Descrizione dei flussi logici del progetto e delle funzionalità dell'applicazione	UC3

### 3.3 Requisiti non funzionali

Definiscono come il sistema deve comportarsi, cioè le sue proprietà di qualità interna. Non aggiungono nuove funzioni, ma impongono vincoli di prestazioni, sicurezza, disponibilità, scalabilità, affidabilità, manutenibilità.

**Tabella 3.2:** Tabella del tracciamento dei requisiti non funzionali

Codice	Descrizione	Use Case
RD1	Pro/controllo di ogni componente e criticità nell'applicazione	-

## 3.4 Requisiti qualitativi

Specificano proprietà qualitative che influenzano l'esperienza d'uso, la manutenibilità o l'efficienza. Sono simili ai non funzionali, ma si concentrano su aspetti percepibili o progettuali, come semplicità, chiarezza, flessibilità o estendibilità.

**Tabella 3.3:** Tabella del tracciamento dei requisiti qualitativi

Codice	Descrizione	Use Case
RO1	Configurazione regole di alerting e notifiche per anomalie rilevate in tempo reale	-

## 3.5 Requisiti di vincolo

Impongono limitazioni o condizioni esterne al progetto: ambienti, tecnologie, compatibilità, strumenti, standard aziendali o legali.

**Tabella 3.4:** Tabella del tracciamento dei requisiti di vincolo

Codice	Descrizione	Use Case
RO1	Configurazione regole di alerting e notifiche per anomalie rilevate in tempo reale	-

## 3.6 Riepilogo dei requisiti

**Tabella 3.5:** Riepilogo dei requisiti

Tipologia	Quantità
RF-O1	Il sistema deve raccogliere metriche tramite agenti OpenTelemetry.
RF-O2	Le metriche devono essere inviate a Elasticsearch per l'analisi.

## **3.7 Rappresentazione Architettuale**

La rappresentazione architettuale funge da ponte tra cosa il sistema deve fare e come sarà progettato.

### **3.7.1 Scelta dell'approccio architettuale**

### **3.7.2 Diagrammi architettureali principali**

### **3.7.3 Mappatura dei requisiti**

Collega i requisiti definiti nella sezione 2 con i moduli architettureali che li soddisfano.

## **3.8 Conclusioni**

## Capitolo 4

# Progettazione e codifica

*Breve introduzione al capitolo*

### 4.1 Tecnologie e strumenti

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

#### **Tecnologia 1**

Descrizione Tecnologia 1.

#### **Tecnologia 2**

Descrizione Tecnologia 2

### 4.2 Ciclo di vita del software

### 4.3 Progettazione

#### **Namespace 1**

Descrizione namespace 1.

**Classe 1:** Descrizione classe 1

**Classe 2:** Descrizione classe 2

### 4.4 Design Pattern utilizzati

### 4.5 Codifica



## Capitolo 5

# Verifica e validazione

## Capitolo 6

# Conclusioni

6.1 Consuntivo finale

6.2 Raggiungimento degli obiettivi

6.3 Conoscenze acquisite

6.4 Valutazione personale

Appendice A

Appendice A

Citazione

---

Autore della citazione

# Acronimi e abbreviazioni

**AI** [Artificial Intelligence](#). [4](#), [21](#)

**APM** [Application Performance Monitoring](#). [4](#), [21](#)

**ESG** [Environmental, Social and Governance](#). [2](#), [21](#)

**UML** [Unified Modeling Language](#). [21](#)

# Glossario

**AI** *Artificial Intelligence* (ing. Intelligenza Artificiale) è un ramo dell'informatica che si occupa della creazione di sistemi in grado di svolgere compiti che normalmente richiederebbero l'intelligenza umana, come il riconoscimento vocale, la visione artificiale, l'apprendimento automatico e la risoluzione di problemi complessi. L'obiettivo principale dell'AI è sviluppare algoritmi e modelli che permettano alle macchine di apprendere dai dati, adattarsi a nuove situazioni e prendere decisioni autonome, migliorando così l'efficienza e l'efficacia in vari settori, tra cui la medicina, la finanza, l'industria e i servizi.. [20](#)

**APM** *Application Performance Monitoring* (ing. Monitoraggio delle Prestazioni delle Applicazioni) è un insieme di pratiche e strumenti utilizzati per monitorare, misurare e gestire le prestazioni e la disponibilità delle applicazioni software. L'obiettivo principale dell'APM è garantire che le applicazioni funzionino in modo ottimale, offrendo un'esperienza utente fluida e senza interruzioni. Ciò include il monitoraggio di vari parametri come tempi di risposta, tassi di errore, utilizzo delle risorse e throughput, nonché l'identificazione e la risoluzione di problemi che potrebbero influire sulle prestazioni dell'applicazione.. [20](#)

**ESG** *Environmental, Social, Governance*, (ing. Ambientale, Sociale e di Governance) è un acronimo che indica i criteri utilizzati per valutare la sostenibilità e la responsabilità di un'azienda. L'aspetto ambientale riguarda pratiche come riduzione delle emissioni, uso delle risorse e tutela del clima; quello sociale include rapporti con dipendenti, clienti e comunità, promuovendo inclusione e condizioni di lavoro eque; infine, la governance si riferisce ai meccanismi di gestione, trasparenza, etica e correttezza nei processi decisionali.. [20](#)

**UML** in ingegneria del software *UML, Unified Modeling Language* (ing. linguaggio di modellazione unificato) è un linguaggio di modellazione e specifica basato sul paradigma object-oriented. L'*UML* svolge un'importantissima funzione di "lingua franca" nella comunità della progettazione e programmazione a oggetti. Gran parte della letteratura di settore usa tale linguaggio per descrivere soluzioni analitiche e progettuali in modo sintetico e comprensibile a un vasto pubblico. [20](#)

# Bibliografia

## Riferimenti bibliografici

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

## Siti web consultati

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