

# Analisi dei requisiti

v0.0



7Last



## Versioni

Ver.	Data	Autore	Verificatore	Descrizione
0.0	28/03/2024	Matteo Tiozzo		Stesura struttura del documento

# Indice

<b>1</b>	<b>Introduzione</b>	<b>4</b>
1.1	Scopo del documento . . . . .	4
1.2	Glossario . . . . .	4
1.3	Riferimenti . . . . .	4
1.3.1	Normativi . . . . .	4
<b>2</b>	<b>Descrizione del prodotto</b>	<b>4</b>
2.1	Obiettivi del prodotto . . . . .	4
2.2	Funzionalità del prodotto . . . . .	5
2.3	Caratteristiche degli utenti . . . . .	6
2.4	Tecnologie . . . . .	6
<b>3</b>	<b>Casi d'uso</b>	<b>7</b>
3.1	Introduzione . . . . .	7
3.2	Attori . . . . .	7
3.3	Elenco dei casi d'uso . . . . .	7
3.3.1	UC1: Registrazione . . . . .	7
<b>4</b>	<b>Requisiti</b>	<b>7</b>
4.1	Requisiti funzionali . . . . .	7
4.2	Requisiti qualitativi . . . . .	7
4.3	Requisiti di vincolo . . . . .	7
4.4	Requisiti prestazionali . . . . .	7
4.5	Tracciamento . . . . .	7
4.5.1	Requisito - Fonte . . . . .	7
4.5.2	Fonte - Requisito . . . . .	7
4.6	Riepilogo . . . . .	7

**Elenco delle tabelle**

**Elenco delle figure**



# 1 Introduzione

## 1.1 Scopo del documento

Questo documento ha lo scopo di illustrare i casi d'uso e i requisiti del capitolato proposto da *Sync Lab S.r.l.*, a seguito di un'attenta analisi del capitolato stesso e di un confronto tenuto con l'azienda.

## 1.2 Glossario

Per evitare qualsiasi ambiguità o malinteso sui termini utilizzati nel seguente documento, è stato adottato un glossario, contenente le definizioni necessarie. È possibile individuare ogni termine presente nel glossario grazie ad uno stile specifico:

- Ad ogni parola presente sarà aggiunta una "G" al pedice della stessa.
- Verrà fornito il link al glossario online (v.1.0) per ciascuna parola.

## 1.3 Riferimenti

BYTEOPS METTONO SOLO RIFERIMENTI NORMATIVI

### 1.3.1 Normativi

- Capitolato C6 - SyncCity: Smart city monitoring platform.  
LINK AL CAPITOLATO
- Regolamento di progetto didattico.  
LINK AL REGOLAMENTO
- Norme di progetto.  
LINK ALLE NORME DI PROGETTO

# 2 Descrizione del prodotto

## 2.1 Obiettivi del prodotto

Sviluppare una piattaforma di monitoraggio per una "Smart City" mediante l'utilizzo di Apache Kafka, Python come simulatore di dati e Docker come ambiente di containerizzazione. Questa piattaforma permetterà la gestione di una città in modo smart tramite



l'utilizzo di vari tipi di sensori, quali umidità, quantità di polveri sottili, temperatura, traffico, livelli di acqua, stato di riempimento delle isole ecologiche, guasti elettrici. Questi dati dovranno essere conservati in un database che permetterà la visualizzazione in una dashboard. La dashboard sarà formata da widget e grafici che permetteranno una visione d'insieme delle condizioni della città. Nel complesso questo strumento permetterà alle autorità di prendere decisioni immediate e informate sulla gestione delle risorse e sull'implementazione di servizi, coinvolgendo anche i cittadini nella gestione e nel miglioramento della città.

L'implementazione di una città monitorata fornisce una solida base per il concetto di città del futuro, permettendo una migliore gestione, ottimizzazione dei servizi pubblici, gestione del traffico, sicurezza e sostenibilità ambientale.

## 2.2 Funzionalità del prodotto

Il software di monitoraggio "SyncCity" è disegnato per offrire una serie di funzionalità di fondamentale importanza per la città del futuro, tra cui:

- **Monitoraggio in tempo reale dei dati:** il software raccoglierà in tempo reale tutti i dati provenienti dal simulatore. Sarà così in grado di fornire in modo continuativo lo stato aggiornato della città.
- **Memorizzazione dei dati:** i dati raccolti verranno immagazzinati in un database. Questo permetterà l'accesso ad essi anche in caso di necessità future e per poter avere una storia della città stessa.
- **Visualizzazione attraverso Dashboard:** sarà implementata una dashboard per poter accedere comodamente ed intuitivamente a tutti i dati raccolti dal software. Inoltre si potranno vedere informazioni cruciali come le condizioni della città in tempo reale, così da sapere dove e come intervenire per migliorare e ottimizzare la città. I dati saranno rappresentati tramite widget e grafici.
- **Visualizzazione mappa dei sensori:** nella dashboard sarà inclusa una mappa che mostrerà la posizione di tutti i sensori presenti, ciascuno con le proprie caratteristiche e tipologia. Inoltre ci sarà la possibilità di vedere lo stato di funzionamento dei vari sensori.
- **Visualizzazione punteggio di salute:** il software calcolerà un indice di benessere della città, valutato su una scala da zero a cento in base all'ultima rilevazione di ciascun sensore. Un punteggio più alto corrisponderà a condizioni di vita migliori.



- **Supporto alle decisioni:** il software fornirà a chi di dovere strumenti per prendere decisioni informate e tempestive sulla gestione delle risorse e sull'implementazione di servizi presenti nella città.
- **Analisi dettagliata delle misurazioni:** il software permetterà di filtrare le misurazioni in base a multipli parametri come intervalli temporali, aree della mappa, sensori specifici e soglie di rilevamento. Questo permetterà di esaminare i dati in modo mirato, sia nel tempo che nello spazio, fornendo un'analisi dettagliata e rilevante per le esigenze specifiche.
- **Sistema di notifica:** il software invierà notifiche in tempo reale alle autorità competenti quando un sensore rileverà una misurazione che supera i valori preimpostati come soglia critica. Questo permetterà di garantire una risposta tempestiva ed efficace di fronte a situazioni che richiedono un'azione immediata, ottimizzando i tempi.

## 2.3 Caratteristiche degli utenti

(DA RIVEDERE) **Autorità locali:** gli utenti principali sono le autorità locali responsabili della gestione e del monitoraggio della Smart City. Questi utenti devono essere in grado di prendere decisioni consapevoli sulla base delle informazioni raccolte e analizzate dal sistema.

- L'utente dovrà utilizzare un dispositivo (Desktop o Mobile) connesso alla reteG per poter accedere alla piattaformaG.

## 2.4 Tecnologie

- **Python:** come simulatore di dati provenienti dai sensori.
- **Apache Kafka:** broker per disaccoppiareG lo streamG di informazioni provenienti dai simulatori dei sensori.
- **Docker:** per la containerizzazione dell'ambiente di sviluppo.
- **Grafana:** piattaforma di Data Visualization per permettere il monitoraggio della città e la visualizzazione delle informazioni raccolte dai sensori.



## **3 Casi d'uso**

### **3.1 Introduzione**

Questa sezione descrive i casi d'uso delineati dopo l'analisi del capitolato, il confronto con il proponente e le discussioni tra gli analisti del gruppo. Ognuno di essi è identificato da un codice, la cui struttura è dettagliata nel documento "Norme di Progetto" (più precisamente in questa sezione).

### **3.2 Attori**

### **3.3 Elenco dei casi d'uso**

#### **3.3.1 UC1: Registrazione**

## **4 Requisiti**

### **4.1 Requisiti funzionali**

### **4.2 Requisiti qualitativi**

### **4.3 Requisiti di vincolo**

### **4.4 Requisiti prestazionali**

### **4.5 Tracciamento**

#### **4.5.1 Requisito - Fonte**

#### **4.5.2 Fonte - Requisito**

### **4.6 Riepilogo**