# **Metadati incontro con azienda**

**Data incontro:** 10/11/2023

**Partecipanti:** All

**Partecipanti azienda:** A.Dorigo, F.Pallaro, D.Zorzi

**Tempo:**

# **Spunti per Analisi dei requisiti**

# **SC**

## **Obiettivi del prodotto**

L’obiettivo del prodotto è la realizzazione di una piattaforma di monitoraggio, che metta insieme i dati prodotti da dispositivi (Iot e non) e sensori di una città. Questa piattaforma permetterà alle autorità del caso di ottenere una visione d’insieme della condizione della città e, in base a questo, prendere decisioni informate e tempestive sulla gestione delle risorse e sull'implementazione di servizi.

I dati visualizzati da questa piattaforma dovranno venire prodotti con una simulazione randomica che sia verosimile. In alternativa, potranno essere utilizzati dati reali accessibili liberamente.

## **Funzioni del prodotto**

Il prodotto sarà costituito dai seguenti componenti:

* **L’implementazione dei simulatori**, con relativa documentazione di funzionamento. Questa dovrà garantire la produzione di dati, che siano il più realistici possibili in base al tipo a cui appartengono.
* **Un database OLAP** che verrà utilizzato per lo storage dei dati.
* **L’integrazione di una piattaforma di stream processing**. Questo con l’obbiettivo di idisaccoppiare lo stream di informazioni raccolto.
* La **visualizzazione**, sottoforma di **dashboard**, di una panoramica sullo stato della città, sulla base dei dati archiviati di recente, mantenendo la visualizzazione aggiornata sullo stato attuale.

### **Criteri di comportamento (non so in che sezione vanno inseriti)**

* test che dimostrino il corretto funzionamento dei servizi e delle funzionalità previste:
  + copertura di test >= 80% correlata di report;
* documentazione su:
  + scelte implementative e progettuali effettuate e relative motivazioni;
  + problemi aperti e eventuali soluzioni proposte da esplorare.

La **Consegna minima** prevede: simulazione, invio dati, salvataggio su un DB e visualizzazione in una dashboard di dati prodotti da almeno un sensore.

La **Consegna completa** prevede:

* la simulazione di più sorgenti;
* l’evidenziazione di relazione tra i dati di diverse sorgenti;
* la previsione di eventi;
* Un sistema di segnalazioni (avanzato?);
* Una dashboard avanzata con: mappa città, widget ed informazioni dei sensori (**cosa si vuole sapere sui sensori?**);

## **Caratteristiche degli utenti**

Il prodotto è destinato alle autorità amministrative, consentendo loro di ottenere una panoramica sulle condizioni della città. Questo permetterà loro di prendere decisioni ben informate sulla gestione delle risorse e sull'implementazione dei servizi, inclusa la possibilità di rispondere prontamente in situazioni di emergenza e di identificare opportunità di miglioramento che potrebbero altrimenti passare inosservate.

L'utente finale potrebbe anche essere un dirigente aziendale interessato a ottenere informazioni rilevanti per gli obiettivi e gli interessi dell'azienda (**da chiarire**).

L’utente potrebbe anche essere un cittadino qualsiasi nel caso in cui si scelga di rendere la piattaforma accessibile pubblicamente, tramite portali online o applicazioni mobile (**da chiarire**).

# **NK**

**Tecnologie di containerizzazione:**

**Script di simulazione:**

* Simulazione di Dati Realistici: Gli script devono essere in grado di simulare dati realistici, replicando il comportamento dei sensori reali in un contesto di utilizzo realistico;
* Configurabilità dei Dati Simulati: Gli script devono consentire la configurazione dei parametri dei dati simulati inclusi valori di base, frequenza di campionamento e variabilità;
* Varietà di Sensori: Gli script devono supportare la simulazione di dati provenienti da una varietà di tipi di sensori, come sensori di temperatura, umidità, polveri sottili e altri [...];
* Verifica dei Dati Simulati: Gli script devono includere una funzionalità di verifica per assicurare che i dati simulati rispettino i limiti e le specifiche desiderate;
* Efficienza Computazionale: Gli script devono essere ottimizzati per l'esecuzione senza causare congestioni o rallentamenti del sistema e devono essere in grado di gestire volumi di dati simulati significativi senza intoppi o sovraccarichi.

**Kafka**:

* Specifiche per Topic e Formato dei Messaggi: L'architettura specifica distinti topic Kafka per diversi tipi di dati IoT, come temperatura, umidità, polveri sottili e altri [...]. I messaggi all'interno dei topic Kafka rispetteranno un formato di messaggio standardizzato;
* Verifica dei Messaggi: Il sistema convaliderà i messaggi in ingresso per assicurarsi che rispettino il formato di messaggio specificato, rifiutando eventuali messaggi non conformi;
* Archiviazione dei Dati di Stato: I dati relativi allo stato dei sensori e ai dati di ingestione dovrebbero essere archiviati in modo affidabile, consentendo la ricostruzione del flusso dei dati in caso di “guasto”;
* Inserimento di Dati Reali: Il sistema supporterà l'inserimento di dati reali provenienti da sensori fisici. I dati provenienti dai sensori saranno ricevuti tramite protocolli specifici e integrati nel flusso dei dati esistente gestito da Kafka;
* Politiche di Retention: Deve essere definita una politica di retention dei dati per specificare quanto a lungo i dati dovrebbero essere conservati nei topic Kafka prima di essere eliminati o archiviati;
* Politiche di Sicurezza per il Trasferimento dei Dati: Il sistema dovrà implementare rigorose politiche di sicurezza per garantire la protezione dei dati durante il trasferimento attraverso Kafka.

**ClickHouse**:

* Strategia di Ingestione dei Dati: L'architettura definirà una strategia di ingestione dei dati, specificando come i dati fluiscono in ClickHouse. Le opzioni possono includere il flusso continuo di dati da Kafka, script personalizzati o altri metodi;
* Elaborazione dei Dati in Tempo Reale: L'architettura ClickHouse supporterà l'elaborazione dei dati in tempo reale, ottimizzando l'esecuzione delle query e garantendo un uso efficiente del motore di archiviazione;
* Elaborazione dei Dati Mancanti o Incompleti: ClickHouse dovrebbe essere in grado di gestire dati mancanti o incompleti provenienti da Kafka in modo elegante, consentendo analisi su dati parziali.

**Grafana**:

* Creazione delle Dashboard: Gli utenti devono poter creare dashboard personalizzate, consentendo loro di selezionare le metriche, le visualizzazioni e le posizioni dei pannelli;
* Pannelli Interattivi: Le dashboard devono supportare pannelli interattivi, come grafici a serie temporali, mappe, tabelle e altro, per una visualizzazione efficace dei dati;
* Meccanismo di Allerta: Il sistema fornirà un meccanismo di allerta all'interno di Grafana, consentendo agli utenti di impostare allarmi basati su specifiche metriche ClickHouse. Gli allarmi attiveranno notifiche quando vengono soddisfatte condizioni predefinite;
* Gestione delle Versioni: Deve essere possibile tenere traccia delle versioni delle dashboard per il controllo delle modifiche nel tempo.

## **Risposte**

* Sul piano di progetto deve essere documentata la dimensione dello sprint (nel senso anche di motivazioni) (2settimane è la scelta presa);
* Si parte con lo studio/sviluppo in contemporanea;
* Obbiettivo sprint:
  + Avanzare i documenti;
  + Prima struttura del compose;
  + Primo simulatore semplicissimo per definire un formato.
  + Creare backlog di user stories
    - Definire cosa fare nel prossimo sprint.
* Chiarimento tecnico:
  + Un solo script per la creazione di falsi sensori (consigliata l’asincronia, uso di thread);
  + Ci sarà un CSV con una lista di sensori (ID, codice di fabbrica, tipo di sensore, quantità di sensori, posizione geospaziale[coordinate]);
  + Lo script Python andrà a produrre un messaggio, contenente i dati simulati, in formato JSON;
* I prossimi meet (non di fine sprint, ma proprio di stato avanzamento lavori), li impostiamo noi anche ogni settimana (5,6 giorni), per fare bene i lavori (hanno dato disponibilità a riguardo);
* Analisi:
  + Dati real time;
  + Query di analisi;
  + Req. prestazionali;
  + Req. qualità dei dati;
  + Req. di sicurezza;
  + Req. scalabilità;
  + Req. tecnici (non funzionale).
  + (consiglio) Dividere in:
    - req. utente;
    - req. applicativi;
    - (scremare entrambi in opzionali e obbligatori)
* Il **target di utenza** sono solo amministratori / tecnici comunali.