

Simone Da Re

Esame Finale - 1° Prova TSAC

Il seguente applicativo ha lo scopo di monitorare un impianto per il riciclo dei materiali. L'impianto funziona attraverso dei nastri trasportatori.

I nastri causano problemi quando:

- il consumo supera i 60 watt
- la velocità supera i 60 metri al minuto

Progettare una soluzione Web per monitorare l'impianto

Soluzione Applicata

Quando avverrà una delle due condizioni citate sopra, l'operatore riceverà un avviso tramite UI dell'applicazione Web. Una volta che il valore tornerà nei parametri considerati sicuri il messaggio rosso di pericolo sparirà.

L'inserimento nella UI dell'ID (evidenziato dal simbolo #) aiuterà l'utente a comunicare ai propri colleghi i dati essenziali per intervenire:

- ID del nastro
- ID della sezione
- Ultimi valori registrati

Strategie applicate, il Cloud

Durante l'analisi della consegna ho capito come uno punti più importanti di questa applicazione fosse quello della **velocità nel comunicare i guasti**. Per questo ho utilizzato una sistema di code basato su **RabbitMQ**. In particolare ho utilizzato **due code**:

- coda *General* in cui vengono gestite le rilevazioni con valori nella norma
- coda *Warning* per i messaggi contenenti rilevazioni di pericolo.

Il motivo di questa scelta architetturale è molto semplice: un messaggio di pericolo non può aspettare i messaggi meno importanti, che potrebbero rallentare l'intervento sulla macchina.

Come sarà possibile notare nello schema architetturale il sistema farà uso di un **load balancer**, utilizzato davanti ai server che riceveranno le **richieste HTTP provenienti dai sensori**.

Per non creare colli di bottiglia il Database potrà essere portato su una istanza **RDS**, in quanto viene offerto il supporto completo a **PostgreSQL**.

Nello schema architetturale ho teorizzato l'utilizzo di Lambda Functions altamente scalabili, al momento l'applicazione viene hostata da una **EC2 AWS**.

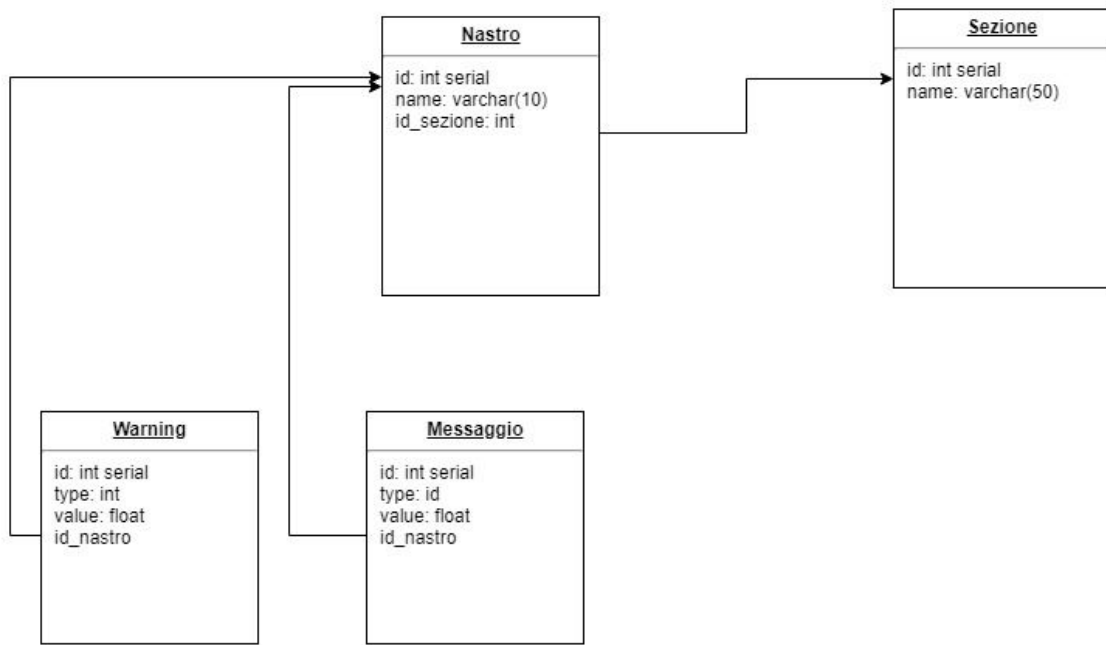
Se il numero di sensori dovesse aumentare le code potranno essere facilmente portate su un servizio managed sempre su AWS.

Per supportare un numero maggiore di nastri ho ipotizzato l'utilizzo di un **load balancer**.

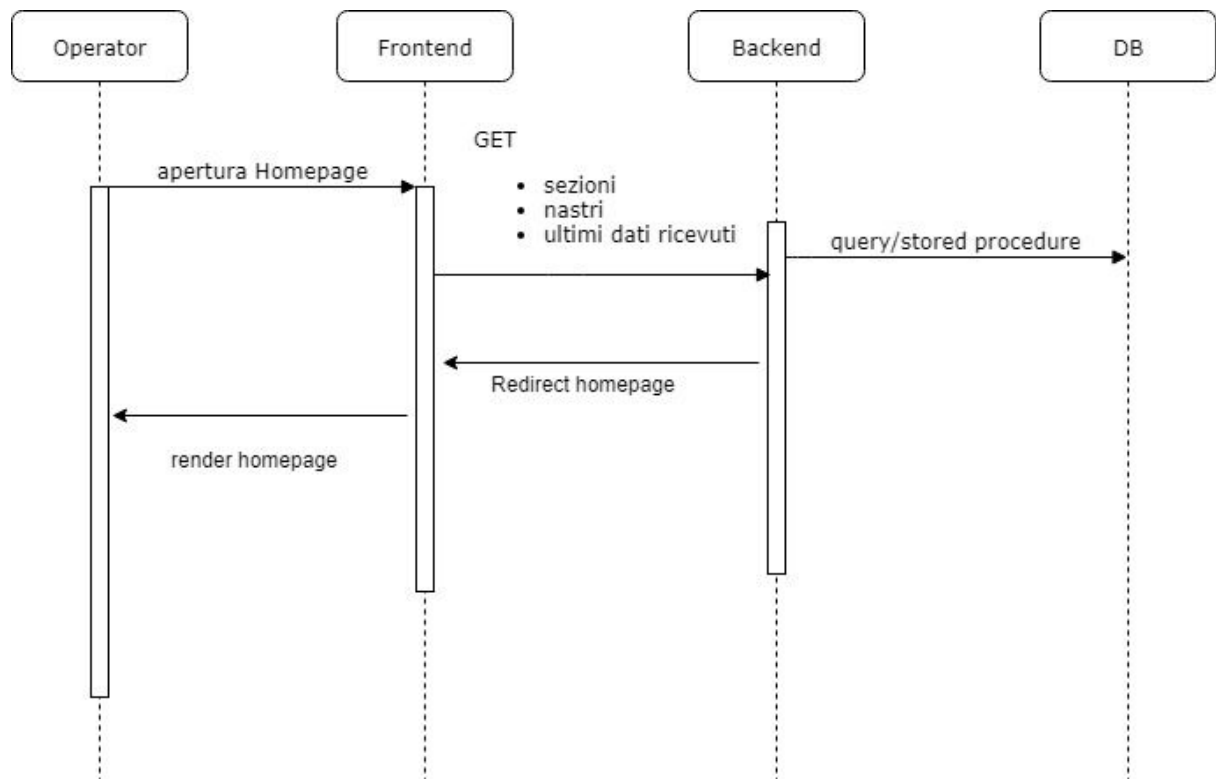
Context Diagram



Diagramma delle classi



Sequence Diagram



Use case diagram

Application

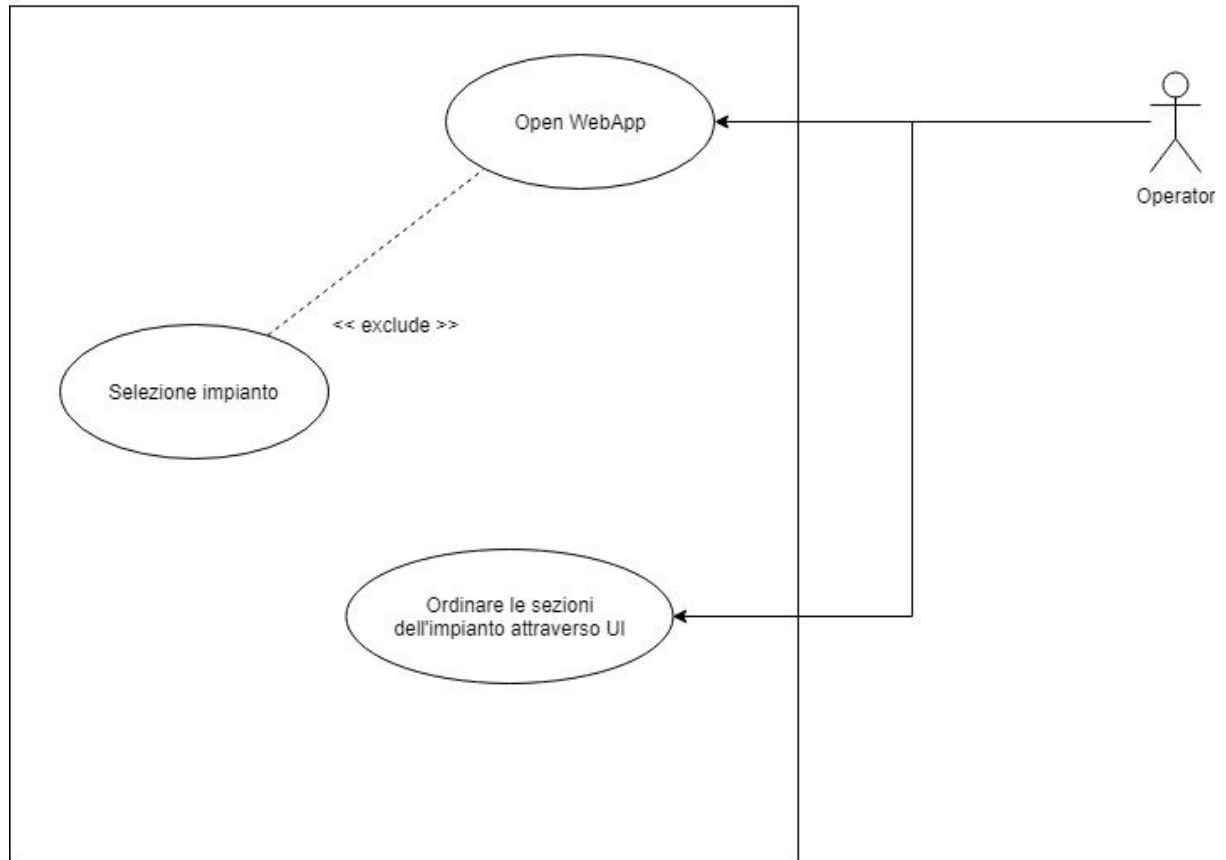


Diagramma Architeturale

