



Progetto di Sistemi Distribuiti

Alex Baiardi
Francesca Frattini

Introduzione

Si è pensato di realizzare un'applicazione che consenta di gestire una serra "smart", in modo da poter visualizzare un elenco di tutti i tipi di ortaggi coltivati e gestire il loro ambiente di sviluppo

Per iniziare ad utilizzare l'applicativo, un utente deve **registrarsi**, successivamente avrà la possibilità di:

- visualizzare le piante presenti in serra
- gestire l'irrigazione delle piante
- monitorare i dati relativi alla qualità dell'aria all'interno della serra

Il progetto presenta una componente di Internet of Things, in quanto l'applicazione comunica con "oggetti intelligenti" tramite la connessione di rete

Introduzione

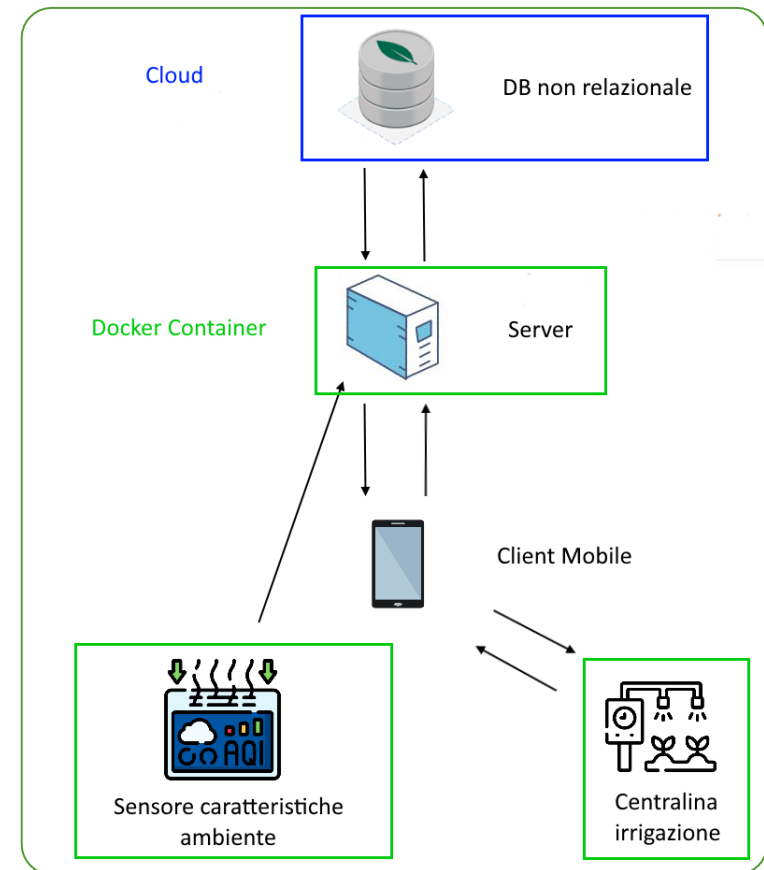
Ogni **utente** ha la possibilità di:

- Aggiungere una nuova pianta alla serra, fornendo le seguenti informazioni:
 - Specie
 - Settore in cui è stata piantata
 - Data di semina
 - Immagine della pianta
- Visualizzare le piante presenti in serra, ordinate in base al settore della serra in cui sono state piantate
- Attivare/disattivare il sistema di irrigazione direttamente dall'applicazione: ogni settore ha un rubinetto dedicato che può essere attivato o disattivato autonomamente
- Visualizzare i valori relativi all'ambiente della serra (temperatura, umidità, pressione atmosferica, quantità di CO₂)

Struttura del sistema

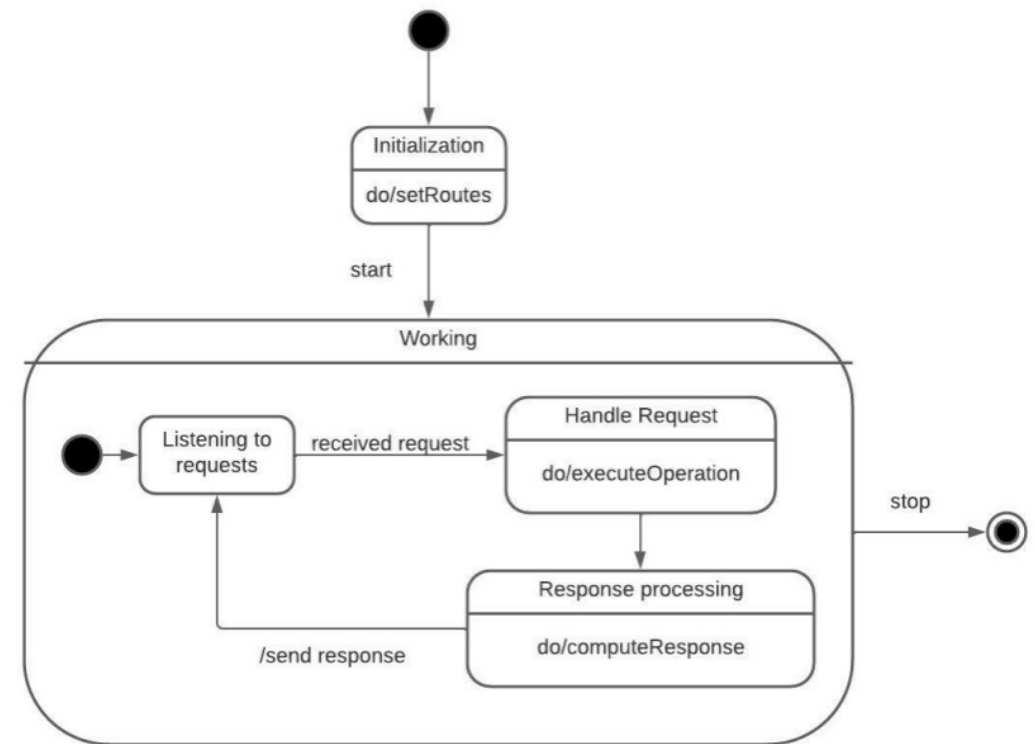
I principali componenti del sistema sono:

- Database non relazionale
- Server
- Client Mobile
- Dispositivi IoT:
 - Centralina irrigazione
 - Sensore ambientale



Server

- Il compito principale del Server è quello di **gestire le richieste** provenienti dal client, attraverso **l'interazione** con il database
- Si tratta di un Server di tipo **stateless**
 - Elabora richieste **HTTP RESTful**
 - Utilizza **ajv** per la validazione
- L'interfaccia del servizio offerto è stata formalizzata tramite **OpenAPI**



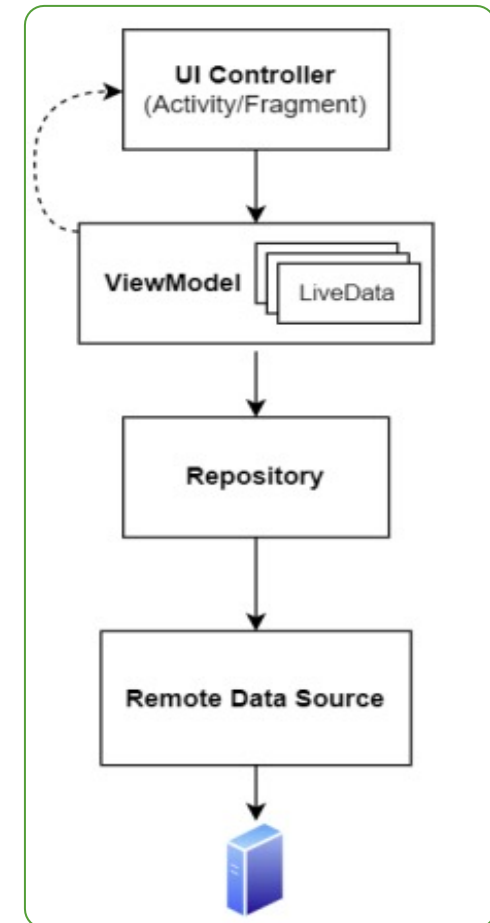
Client Mobile

Il Client si occupa di **gestire l'interazione** con l'utente e di compiere le operazioni che esso richiede tramite l'**invio** di apposite richieste al Server e la **gestione** delle risposte.

Le operazioni che l'utente effettua determinano **il comportamento** del Client.

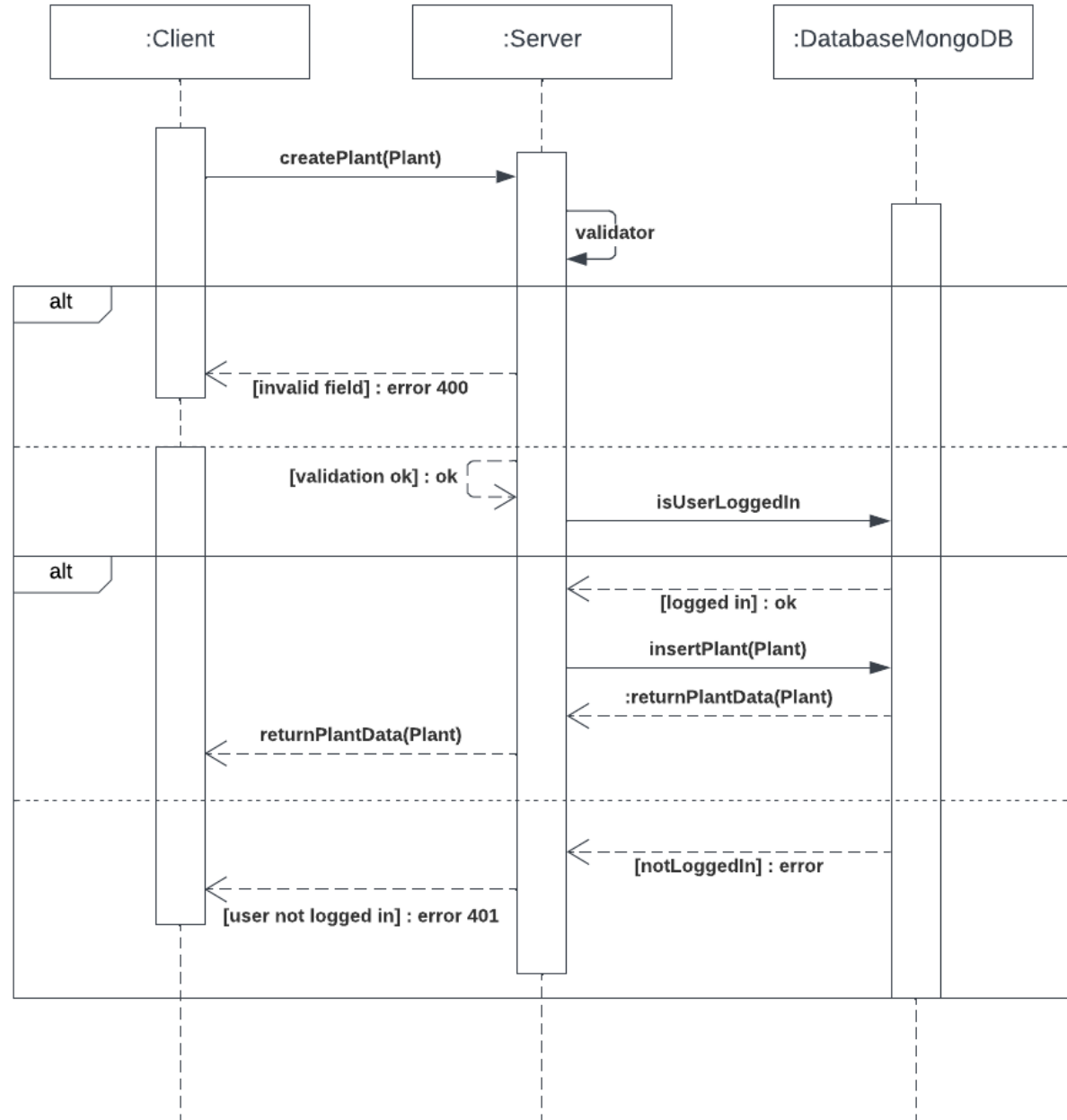
È stata realizzata un' applicazione **Android** sviluppata in linguaggio **Java**

- Suddivisione in **moduli** in adesione al pattern **MVVM**
- Aspetti di networking gestiti con la libreria **Volley**
- Richiede una versione di Android ≥ 9.0 (SDK 28)



Interazione

Diagramma di sequenza per
l'inserimento di una pianta
all'interno del database



Internet of Things

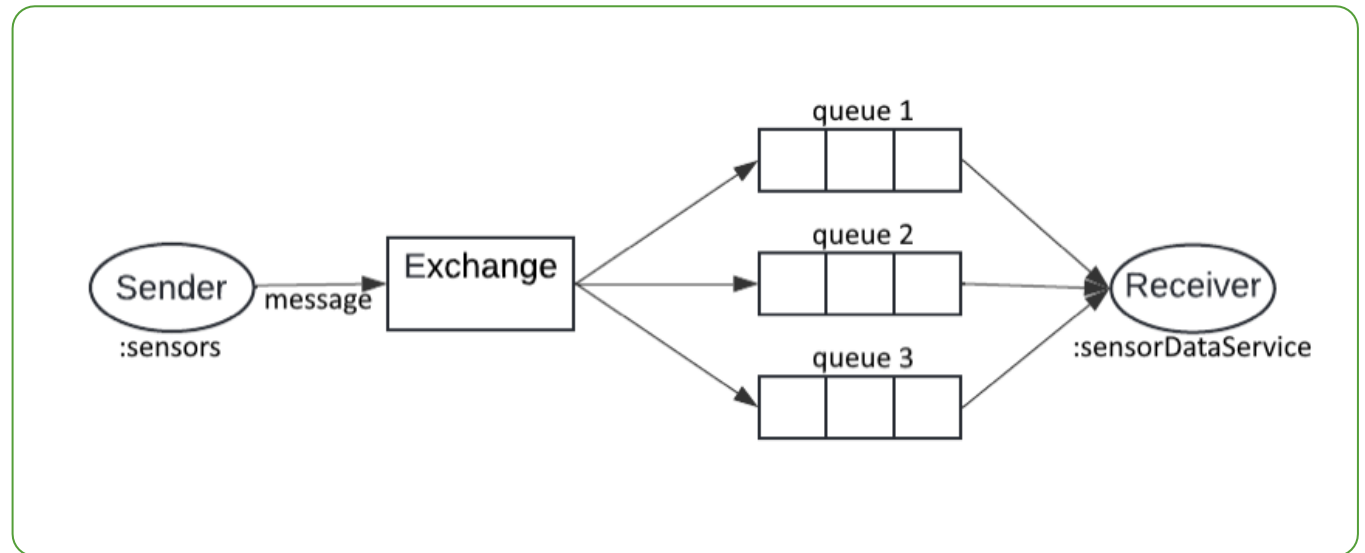
Sensore ambientale

Rileva valori relativi a

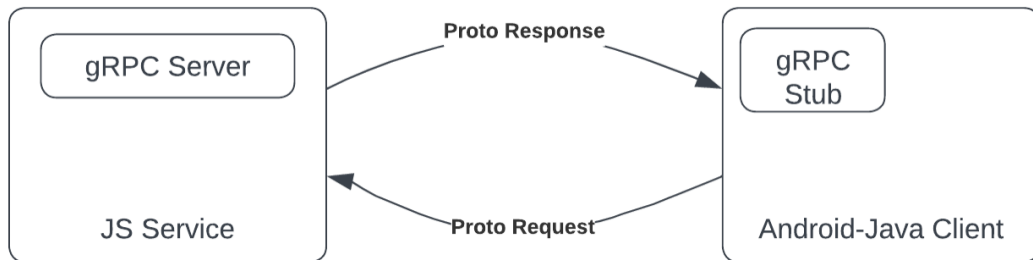
- Temperatura
- Umidità
- Pressione atmosferica
- Quantità CO₂

Comprende due moduli: sensors e sensorDataService

- Essi comunicano tra loro mediante il broker **RabbitMQ** con una comunicazione di tipo publish-subscribe



Internet of Things



Centralina irrigazione

Permette di gestire l'irrigazione nei vari settori della serra

Comunica direttamente con l'applicazione mobile mediante **gRPC**

Per accoppiare una centralina con il relativo utente è stata inserita una rotta API che memorizza l'associazione tra l'utente e l'indirizzo IP della centralina

Fault Tolerance

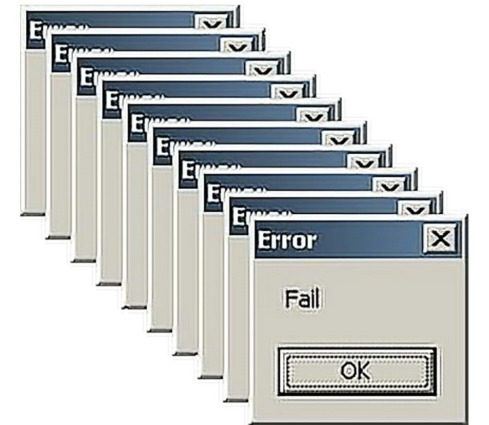
Grazie alla proprietà di instradamento di RabbitMQ, ogni utente ha una coda dedicata sulla quale la relativa centralina pubblica i dati

Per segnalare al consumer l'aggiunta di nuove centraline, è stata riservata una coda a parte sulla quale ogni produttore (periodicamente) segnala la sua presenza.

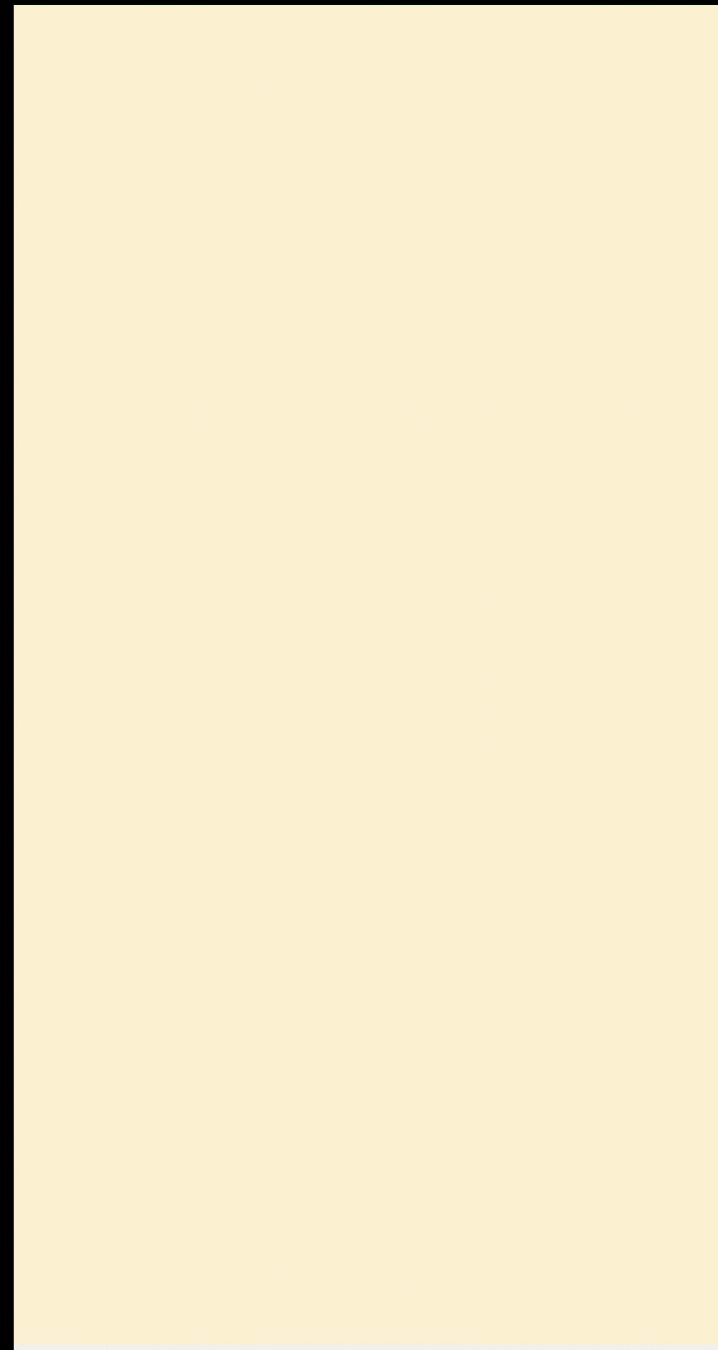
L'invio ciclico di messaggi su questa coda riservata è una sorta di **heartbeat** che segnala al receiver chi sta inviando e permette di ristabilire la lettura dalle code in caso di guasto

All'interno dell'applicazione, invece, in caso di un errore network con Volley o gRPC viene mostrata una schermata di errore che evita il blocco dell'applicazione stessa

There was an error



Esempio di utilizzo



Conclusioni

Le funzionalità realizzate sono state **testate** sia mediante l'utilizzo di test **automatici** che **manuali** e, da un'analisi di quanto fatto, possiamo affermare che il sistema soddisfa gli obiettivi ed i requisiti individuati e che si è complessivamente soddisfatti del risultato raggiunto.

Possibili funzionalità aggiuntive dell'applicazione, che possono essere realizzate in futuro possono essere le seguenti:

- Integrare con una parte di IA per il riconoscimento delle piante tramite un'immagine
- Migliorie nella sezione di IoT per consentire all'utente di impostare una determinata temperatura all'interno della serra
- Includere la possibilità di pianificare l'irrigazione in modo automatizzato creando programmi basati sul giorno della settimana e sull'orario

GRAZIE PER
L'ATTENZIONE!