

Progetto d'esame: laboratorio di PISSIR

Anno accademico 2022/2023

DESCRIZIONE PROGETTO

Sistema di gestione di scorte d'acqua per l'irrigazione di coltivazioni: requisiti funzionali

Obiettivo del progetto è la progettazione e l'implementazione di un sistema di gestione di acqua per irrigazione di colture.

Gli utenti del sistema sono di due tipi: utenti autorizzati alla gestione delle colture di una specifica azienda, utenti che gestiscono le risorse idriche condivise dalle aziende.

I gestori delle risorse idriche possono aggiornare le informazioni relative alla disponibilità di risorse idriche e ai limiti massimi di erogazione giornaliera globali e per ciascuna azienda; inoltre possono consultare i dati relativi ai consumi (complessivi oppure dettagliati per azienda).

Ciascuna azienda possiede uno o più coltivazioni che devono essere irrigate: gli utenti autorizzati a gestire i dati di una data azienda possono inserire le informazioni relative alle coltivazioni possedute, che dovranno comprendere (almeno) la dimensione del campo (ettari), un parametro che indica se il tipo di coltivazione ha esigenze più o meno grandi di acqua (potrebbero essere per esempio 3 livelli: 1, 2, 3 per indicare poca, media, tanta), un parametro che indica il tipo di irrigazione (almeno 2, per indicare per esempio tradizionale o goccia a goccia ... potete indicarne altri) ed infine un parametro che indica il grado di umidità ideale da mantenere. In base a questi parametri e a misure di umidità e temperatura, ottenute periodicamente da sensori sul campo, si potrà ogni giorno stabilire la quantità d'acqua da erogare su ciascun campo. In funzione del numero e dell'estensione delle proprie coltivazioni, il gestore dell'azienda dovrà impostare le richieste di quantità giornaliera di acqua da utilizzare per l'irrigazione. Tale quantità può essere modificata periodicamente: in caso di aggiornamento la nuova quantità potrà essere accettata o rifiutata dal sistema di gestione delle risorse idriche in base alla disponibilità complessiva d'acqua ed alle richieste complessivamente avanzate da tutte le aziende; la nuova quantità, se accordata, sarà disponibile a partire dal giorno successivo alla conferma di variazione.

L'acqua prenotata viene pagata in ogni caso, anche se non utilizzata (per cui è interesse di ogni azienda fare una stima piuttosto precisa delle esigenze, ed è inoltre utile poter mantenere e consultare uno storico delle quantità prenotate ed effettivamente utilizzate per ciascuna coltivazione nel passato, oltre allo storico delle misure di umidità e temperatura rilevate nel tempo).

Oltre ai dati sulle coltivazioni si dovranno registrare le informazioni sui sensori e attuatori presenti nelle varie coltivazioni, in modo da poter associare le misure ricevute alla coltivazione dove si trova il sensore, e in modo da poter attivare (automaticamente) gli attuatori (irrigatori) nel momento in cui le condizioni (di temperatura ed umidità) lo richiedono.

Struttura del sistema (progettazione)

Il sistema dovrà essere composto da

- un "backend" che espone un'interfaccia di tipo REST e permette di inserire in un database i dati rilevanti sulle risorse idriche disponibili e la loro assegnazione alle aziende, e sulle informazioni relative alle aziende ed alle loro coltivazioni (incluse quelle che variano dinamicamente, come per

esempio i valori delle misure rilevate), ed inoltre permette di modificare o consultare tali dati. Il backend ha accesso esclusivo al DB (chiunque voglia aggiornare o leggere i dati deve chiederlo al backend).

- Una interfaccia utente (un'app, una web-app o una mobile app) che permetterà agli utenti di interagire con il backend (tramite le API-REST di quest'ultimo). Questa potrà essere una semplice interfaccia testuale oppure una interfaccia a finestre (per esempio eseguibile nel browser, quindi una web-app) o una mobile app (se avete seguito il corso di applicazioni mobili).
- Un gestore del sottosistema IoT: si tratta di un componente che può entrare in comunicazione con i sensori e gli attuatori, invia periodicamente le misure rilevate tramite i sensori e agisce sullo stato degli attuatori in base ai requisiti di umidità stabiliti, alle misure rilevate e alla quantità d'acqua effettivamente disponibile. L'idea è che ogni azienda agricola di norma abbia nella propria rete locale (la stessa alla quale si agganciano i dispositivi sensori/attuatori) un gestore di sottosistema IoT. I gestori dei sottosistemi IoT dovranno interagire con gli altri sottosistemi tramite broker MQTT (per esempio per trasmettere misure o indicazioni di quantità d'acqua consumate nell'irrigazione, per ricevere indicazione della quantità d'acqua effettivamente disponibile per la specifica coltivazione, e per inviare eventuali notifiche di allarme per insufficiente disponibilità d'acqua oppure, al contrario, per indicare un surplus/avanzo rispetto alla richiesta iniziale).

COMPONENTI ACCESSORI (facoltativi):

Il sistema potrebbe inglobare anche alcuni componenti accessori: seguono alcuni esempi.

- un sistema esterno di autenticazione / autorizzazione (es. keycloak) da utilizzare per permettere l'accesso ai soli utenti autorizzati (previa autenticazione);
- un sistema esterno che fornisce le previsioni meteo in modo da poter basare le richieste di acqua giornaliera anche in funzione della probabilità di avere pioggia, oppure che le temperature siano molto alte;
- un "mercato dell'acqua" che potrebbe permettere ad aziende che hanno un esubero di acqua di cedere "crediti d'acqua" a chi invece non riesce a coprire tutte le proprie esigenze di irrigazione.

SENSORI E ATTUATORI: i sensori e gli attuatori potranno essere reali o emulati, in base alla disponibilità di strumenti fisici. In ogni caso per poter effettuare più facilmente i test sul sottosistema IoT e i test d'integrazione occorrerà predisporre delle piccole applicazioni in grado di fornire misure fittizie (che siano rappresentative di scenari realistici, e che potrebbero essere memorizzate e quindi lette da un file: ciò sarebbe molto utile sia per lo svolgimento dei test che in occasione della demo da mostrare all'esame) ma anche di recepire i comandi per gli attuatori e visualizzarli in formato testo o grafico (una possibilità è usare l'emulatore delle Philips Hue assegnando a ciascuna lampadina emulata un significato, come rappresentante di un attuttore fisico, e in base al comando ricevuto accenderla, spegnerla o farle cambiare colore).

FASI DEL LAVORO

- **Specifica:** in base ai requisiti elencati sopra, eventualmente dettagliati grazie a interviste che i gruppi potranno organizzare con il committente (i docenti del corso), occorrerà definire con sufficiente dettaglio il dominio e i casi d'uso. Per evitare ambiguità si dovranno usare schemi chiari preferibilmente utilizzando i diagrammi UML (diagramma delle classi del dominio e diagramma dei casi d'uso, corredati di brevi commenti testuali complementari ai diagrammi che spiegano in cosa consiste il caso d'uso; a vostra discrezione potrete usare altri tipi di diagrammi per specificare meglio l'articolazione in termini di successione di passi dei diversi casi d'uso).
- **Progettazione:** in questa fase si dovrà scegliere il tipo di architettura (che dovrà essere basata su microservizi), si dovranno individuare i sottosistemi, le loro interfacce e le interazioni. Inoltre

all'interno di ogni gruppo si dovranno definire dei sottogruppi a cui attribuire la responsabilità della progettazione di 1 o 2 sottosistemi (AI GRUPPI PIU' NUMEROSI SARA' RICHiesto DI REALIZZARE UNA DELLE FUNZIONALITA' ACCESSORIE). Per ciascun sottosistema dovrà essere preparato un documento che definisca in modo sufficientemente preciso le classi che si dovranno implementare nella fase successiva: per evitare ambiguità la documentazione dovrà contenere schemi chiari, possibilmente utilizzando diagrammi UML. In particolare dovrete schematizzare il diagramma delle classi (questa volta sarà il diagramma delle classi che dovranno essere implementate) e dei package, e i diagrammi di sequenza per descrivere le interazioni tra i componenti (sia all'interno di uno stesso microservizio, sia tra microservizi diversi). In questa fase si dovranno anche definire gli end-point delle API REST, con gli eventuali parametri e il formato delle informazioni scambiate, sia i topic MQTT con il formato dei messaggi inviati sui diversi topic

- **Implementazione:** Consigliamo di implementare in Java le classi definite in fase di progettazione, ma potete usare altri linguaggi a voi più congeniali (il supporto da parte dei docenti potrebbe non essere ugualmente proficuo se si scelgono altri linguaggi). I diversi microservizi possono essere implementati con linguaggi diversi (ciascun sottogruppo può implementare nel linguaggio preferito il proprio microservizio: l'importante è che le interfacce siano rispettate).
- **Test:** Lo sviluppo dovrà prevedere anche il test dei componenti e dei sottosistemi in isolamento. Quando il gruppo avrà completato tutti i sottosistemi dovrà svolgere anche test di integrazione. I test dovranno coprire un numero adeguato di casistiche tipiche, e saranno anche la base per la dimostrazione di funzionamento che dovrà essere illustrata durante la parte dell'esame relativa al progetto.
- **Demo:** durante l'esame sulla parte pratica dovrà essere preparata una demo che permetta di mostrare tutte le funzionalità implementate e che sia esempio significativo di alcuni dei test più rilevanti realizzati prima della consegna.