

SPECIFICA

OBIETTIVO DEL PROGETTO

Progettazione e implementazione di un sistema di gestione di acqua per irrigazione di colture.

DEFINIZIONI E TERMINI

| TERMINE | DEFINIZIONE | DESCRIZIONE |
|--------------------------------|---|--|
| Utente | | Persone che utilizzano il sistema |
| Gestore Azienda | GA – Utente Agricoltore | Possono inserire le informazioni relative alle coltivazioni possedute. Dovrà impostare le richieste di quantità giornaliera di acqua da utilizzare per l'irrigazione. Tale quantità può essere modificata periodicamente |
| Gestore Risorse Idriche | GSI – Utente Gestore | 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 |
| Campo | Appartiene all'azienda agricola che lo deve irrigare | Ha diversi parametri come l'estensione in ettari |
| Coltura | Tipo di coltivazione di un campo | Ogni coltura ha le sue esigenze di acqua, tipo di irrigazione e umidità / temperatura |
| Sensore | Dispositivo passivo che rileva le misure | Dispiegato nei vari campi misura temperatura o umidità |
| Attuatore | Dispositivo attivo che permette l'interazione con il campo a cui si riferisce | Dispiegato nei vari campi può essere di tipo irrigatore, riscaldamento/raffreddamento |
| Piano | Definisce gli orari di attivazione e disattivazione, le modalità automatica o manuale di un attuatore | |
| Acqua Totale | È l'acqua globale a disposizione di un fornitore che la ripartisce tra le aziende che ha associate | Nel caso si volesse diminuire non può mai andare sotto la soglia dell'acqua già assegnata alle aziende |
| Acqua Azienda | È l'acqua disposizione dell'azienda che la ripartisce tra i suoi campi | Nel caso si volesse diminuire non può mai andare sotto la soglia dell'acqua già assegnata ai campi |

REQUISITI NON FUNZIONALI

LINGUAGGIO

Il linguaggio scelto per il progetto è Javascript, poiché è in comune sia per frontend che per il backend., inoltre ha i seguenti vantaggi:

- **Indipendenza dalla piattaforma:** a differenza di altri linguaggi di programmazione, è possibile inserire JavaScript in qualsiasi pagina Web e utilizzarlo con molti altri framework e linguaggi di sviluppo Web. Una volta scritto, è possibile eseguire il codice JavaScript su qualsiasi macchina. Pertanto, JavaScript rende lo sviluppo delle applicazioni indipendente dalla piattaforma.
- **Riduzione del carico del server:** è possibile utilizzare JavaScript per ridurre il carico del server e la congestione della rete, perché può eseguire operazioni logiche e svolgere gran parte del lavoro del server sul client stesso.
- **Miglioramento dell'interfaccia utente:** crea siti Web eleganti che consentono di trovare ed elaborare informazioni complesse, gli sviluppatori lo applicano per estendere la funzionalità e la leggibilità oltre che a rendere più efficiente l'interazione con l'utente
- **Supporto della simultaneità:** può eseguire diverse serie di istruzioni in parallelo; infatti, sul backend, Node.js è in grado di gestire ed elaborare le risposte del server su vasta scala senza consumare la stessa quantità di larghezza di banda.

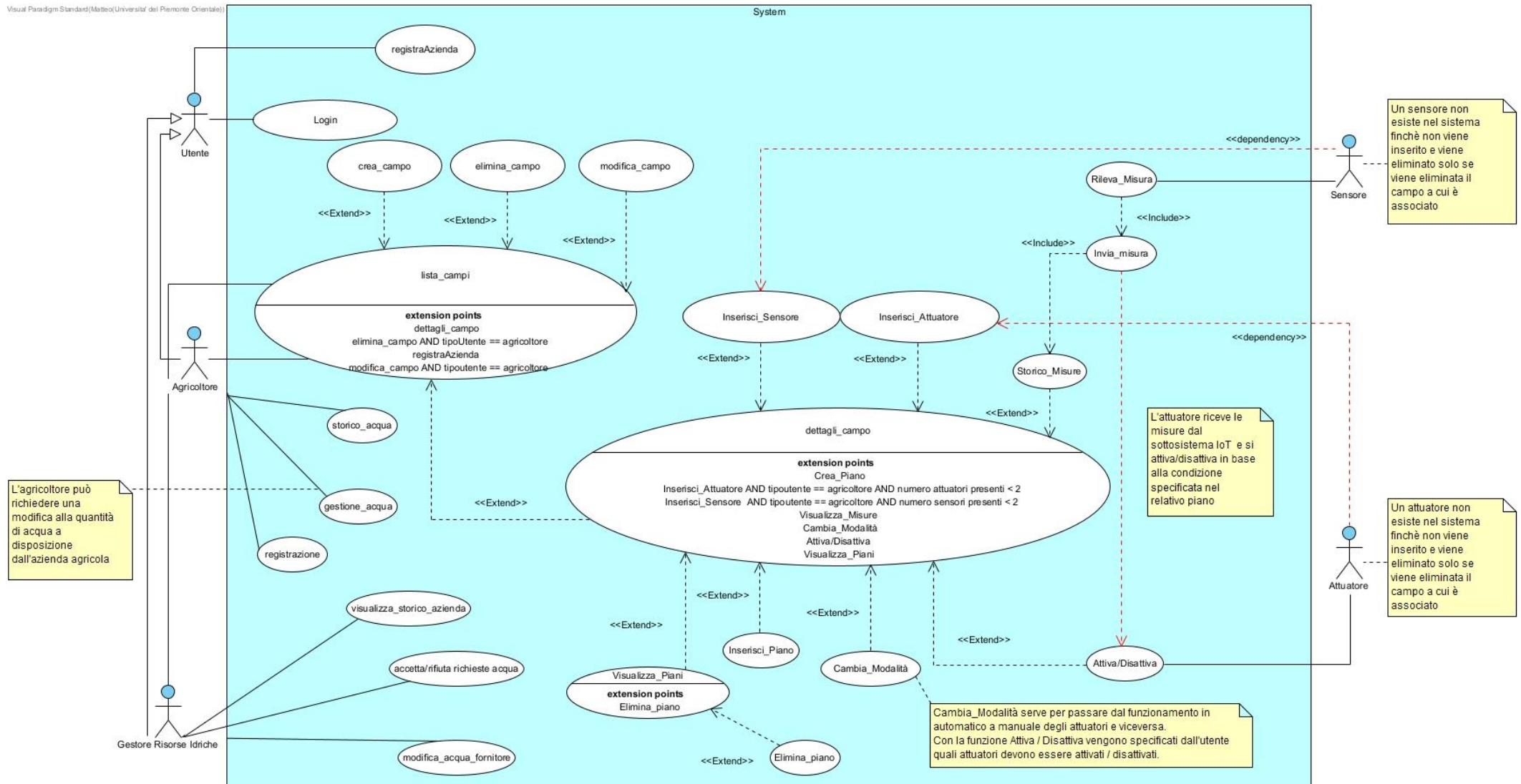
FRAMEWORK

- **Express:** è un framework leggero per Node.js che semplifica lo sviluppo di applicazioni web e API. Utilizza il concetto di middleware per gestire le richieste e le risposte HTTP. Offre funzionalità come il routing, la gestione degli errori e l'autenticazione. È flessibile e ampiamente utilizzato nella comunità di sviluppatori web.
- **Morgan:** è un middleware per Node.js che registra le informazioni dei log per le richieste HTTP, come il metodo, l'URL e lo stato della risposta.
- **MQTT:** è un protocollo di messaggistica leggero e affidabile per la comunicazione tra dispositivi IoT, con basso overhead di rete e bassa latenza. Utilizza un modello di pubblicazione/sottoscrizione per scambiare messaggi tra dispositivi.
- **Passport:** è un framework di autenticazione per Node.js che semplifica l'implementazione di strategie di autenticazione come username/password e nelle applicazioni web.
- **Bootstrap:** è un framework front-end open-source che fornisce componenti e stili predefiniti per la creazione di interfacce web responsive e moderne.
- **jQuery** è una libreria JavaScript leggera e veloce che semplifica la manipolazione del DOM, la gestione degli eventi e le chiamate AJAX nelle pagine web.
- **SQLite3:** è una libreria di database relazionale leggera e veloce che consente di creare e gestire database locali senza la necessità di un server separato. È compatibile con diverse piattaforme e linguaggi di programmazione.

CASI D'USO

DIAGRAMMA UML

Visual Paradigm Standard (Matteo/Università del Piemonte Orientale)



CASI D'USO COMUNI

- **Login:** verifica di username e password dell'utente che vuole accedere al sistema
- **Lista Campi:** vengono visualizzati tutti i campi di un'azienda (in caso di utente GSI vengono visualizzati i campi per aziende)
- **Dettagli Campo:** vengono visualizzati i dettagli del campo relativi a coltura, ettari, acqua richiesta, ecc.... e gli attuatori / sensori del campo
- **Storico Misura:** vengono visualizzate tutte le misure con data e valore di un sensore
- **Invio Misura:** il sensore rileva sistematicamente le misure e salvate nel sistema

CASI D'USO GESTORE AZIENDA AGRICOLA

- **Registrazione:** l'utente GA compila il form, seleziona un'azienda già registrata e conclude la registrazione della nuova utenza per la gestione dell'azienda
- **Registrazione Azienda:** pagina dedicata alla registrazione di una nuova azienda
- **Gestione Acqua:** serve per richiedere una variazione della quantità di acqua a disposizione dell'azienda (approvata dal fornitore)
- **Storico Acqua:** si seleziona un campo dell'azienda e viene visualizzato lo storico dei consumi di acqua, in ordine decrescente di data, del campo richiesto
- **Crea Campo:** aggiunge un nuovo campo all'azienda
- **Modifica Campo:** modifica uno o più valori del campo
- **Elimina Campo:** elimina un campo dell'azienda
- **Inserisci Sensore:** aggiunge un sensore ad uno specifico campo
- **Inserisci Attuatore:** aggiunge un attuatore ad uno specifico campo
- **Visualizza Piani:** visualizza i piani di un attuatore
- **Inserisci Piano:** inserisce un nuovo piano per uno specifico attuatore
- **Elimina Piano:** elimina un piano per uno specifico attuatore
- **Cambia Modalità:** cambia modalità dell'attuatore attivo/disattivo o viceversa

CASI D'USO GESTORE RISORSE IDRICHE

- **Visualizza Storico Azienda:** visualizza lo storico dei consumi di acqua di un'azienda
- **Accetta / Rifiuta Richieste Acqua:** vengono accettate/rifiutate le richieste di modifica dell'acqua assegnata ad una specifica azienda
- **Modifica Acqua Fornitore:** modifica l'acqua a disposizione di un fornitore

CLASSI DEL DOMINIO

DIAGRAMMA UML

Visual Paradigm Standard (Matteo/Università del Piemonte Orientale)

