**ANALISI STRUMENTALE**

L’obiettivo di Serra 4.0 è creare un sistema completo per il monitoraggio e la visualizzazione di dati in tempo reale provenienti dai sensori di una serra automatizzata. Utilizzando Node-RED per l’acquisizione e l’elaborazione dati, PostgreSQL come database per l’archiviazione strutturata e Grafana per la visualizzazione e l’analisi interattiva, si ottiene una soluzione scalabile, modulare e personalizzabile per ottimizzare la gestione ambientale della serra.

**PostgreSQL:** è un database relazionale open-source, scelto per la sua affidabilità, performance e supporto a query complesse.

* Configurazione: Database strutturato con tabelle per dati sensoriali rilevati in Serra 4.0 (es. temperatura, umidità, luminosità, timestamp).
* Accesso: Node-RED interagisce con PostgreSQL tramite il nodo node-postgres.
* Ruolo: Conservazione storica dei dati raccolti, supporto a query aggregative per Grafana.

**Node-RED:** è una piattaforma di programmazione visuale per connettere hardware, API e servizi online.

* Flusso principale:
  + Acquisizione dati da sensori MQTT installati in Serra 4.0.
  + Trasformazione e normalizzazione dei dati (es. conversione unità).
  + Inserimento dei dati nel database PostgreSQL tramite nodo apposito.
* Vantaggi: Facilità di creazione di flussi personalizzati senza programmazione complessa.

**Grafana:** è uno strumento open-source per la creazione di dashboard interattive.

* Configurazione: Collegamento diretto a PostgreSQL come fonte dati per Serra 4.0.
* Dashboard: Creazione di grafici a linee e gauge per visualizzazione in tempo reale l’andamento di temperatura, umidità e altri parametri ambientali.
* Funzionalità aggiuntive: Configurazione alert per notifiche su valori fuori soglia.

**WebSocket : è un protocollo di comunicazione bidirezionale, basato su TCP, che consente uno scambio continuo di dati tra client e server attraverso una connessione persistente**

**Funzionamento e architettura**

* **Una volta elaborati i dati da Node-RED (ricevuti via MQTT), questi vengono inviati tramite nodi WebSocket a client web connessi.**
* **La connessione WebSocket rimane aperta finché client e server lo desiderano, permettendo scambi continui di informazioni in entrambe le direzioni.**

**Caratteristiche tecniche**

* **Protocollo: WebSocket (RFC 6455), basato su TCP.**
* **Connessione persistente: a differenza di HTTP, la connessione rimane attiva dopo il primo handshake.**
* **Integrazione: Node-RED dispone di nodi websocket in e websocket out per la gestione semplificata dei flussi dati.**
* **Supporto client: compatibile con tutti i browser moderni e facilmente integrabile in interfacce JavaScript.**

**Vantaggi**

* **Aggiornamento in tempo reale: ideale per sistemi di monitoraggio.**
* **Efficienza di rete: riduce il carico rispetto a soluzioni basate su richieste periodiche.**
* **Bidirezionalità: consente anche al server (Node-RED) di inviare aggiornamenti proattivamente.**
* **Integrazione web: nativamente supportato da browser, perfetto per dashboard e interfacce personalizzate.**

**Mosquitto :**

**è un broker MQTT open-source progettato per abilitare comunicazioni efficienti tra dispositivi IoT mediante il paradigma *publish/subscribe*. È caratterizzato da leggerezza, affidabilità e semplicità di configurazione, ed è ampiamente usato in sistemi distribuiti e ambienti embedded.**

**Funzionamento e architettura**

* **I sensori agiscono come publisher, inviando messaggi a topic MQTT.**
* **Node-RED agisce come subscriber, ricevendo i dati in tempo reale.**
* **Il broker Mosquitto gestisce le connessioni, inoltra i messaggi e può essere configurato per garantire la persistenza e la sicurezza della comunicazione.**

**Caratteristiche tecniche**

* **Protocollo supportato: MQTT v3.1, v3.1.1 e v5.**
* **Qualità del servizio (QoS): supporta tre livelli (0, 1, 2) per la garanzia di consegna dei messaggi.**
* **Configurabilità: tramite file mosquitto.conf, consente la definizione di listener multipli, autenticazione, crittografia TLS, ACL (Access Control List), log personalizzati e persistenza.**
* **Compatibilità: funziona su Linux, Windows e dispositivi embedded.**

**Vantaggi**

* **Efficienza: basso consumo di risorse, adatto a dispositivi con capacità limitate.**
* **Affidabilità: gestione robusta delle connessioni e dei messaggi.**
* **Sicurezza: supporto a crittografia TLS/SSL, autenticazione utente, autorizzazioni granulari.**
* **Scalabilità: può essere distribuito in architetture IoT complesse, anche con migliaia di dispositivi.**
* **Integrazione diretta: interoperabilità ottimale con Node-RED, facilitando la costruzione di pipeline dati in tempo reale.**