

# Piattaforma IoT

---

- [Piattaforma IoT](#)
  - [Introduzione](#)
  - [Funzionalità](#)
    - [Impianti](#)
    - [Nodi](#)
    - [Sensori](#)
    - [MQTT](#)
    - [Allarmi e soglie](#)
    - [Interfaccia grafica](#)
  - [Autenticazione ed autorizzazione](#)
  - [Backend](#)
  - [Hosting](#)

## Introduzione

---

**Obiettivo:** creazione di un'applicazione web per lo stoccaggio, la visualizzazione ed il trasferimento di dati derivanti da una rete di dispositivi (ciascuno connesso ad un certo numero di sensori) ed acquisiti tramite protocollo MQTT.

## Funzionalità

---

### Impianti

- la rete di dispositivi potrebbe dover monitorare più impianti di una certa azienda
- l'app dovrà quindi essere in grado di visualizzare gli impianti registrati e permettere di inserirne di nuovi
- nella visualizzazione impianti dovranno essere elencati anche tutti i dispositivi ad essi associati

### Nodi

- arruolamento di nuovi nodi
  - inserimento manuale del codice identificativo
  - QR code
- modifica delle informazioni
- spegnimento volontario di un nodo
- **informazioni aggiuntive**
  - ogni nodo sarà dotato di un solo tipo di sensore per una determinata grandezza fisica  
es: un nodo non potrà avere due sensori di temperatura installati al suo interno
  - la locazione di un nodo all'interno di un impianto non viene memorizzata nella piattaforma: l'utente dovrà assegnare un nome significativo al nodo affinché ne riconosca posizione e tipologia

## Sensori

- tipologia
  - temperatura
  - pressione
  - umidità
  - CO2
  - luminosità (?)

## MQTT

- protocollo utilizzato per scambiare i dati acquisiti da ogni nodo con i servizi backend della piattaforma
- tipi di payload

```
// Payload nominale
"payload": {
  "date": "18/01/2023",
  "node_id": "12345",
  "battery": "{1,2,3,4,5}", // livello batteria nodo
  "values": {               // valori acquisiti dai sensori
    "temp": 23.5,           // temperatura
    "press": 46.0,          // pressione
    "hum": 12.3,            // umidità
    "co2": 326.45,          // CO2
    "lum": 125.67,          // luminosità
  }
}

// Payload posizionale
"payload": "18/01/2023,12345,23.5,..."
```

## Allarmi e soglie

- diversificare tipi di allarme
  - **permanenti** -> disabilitabili solo dall'utente
  - **temporanei** -> validi fino a che non si rientra nel range accettabile
- **esempi di allarmi**
  - batteria di un nodo scarica (*allarme temporaneo?*)
  - disconnessione di un nodo (*allarme permanente?*)
    - persa connessione MQTT
- possibilità di notificare gli utenti tramite mail quando scatta un allarme
- impostazione di un **set point**
  - soglia oltre la quale (o sotto la quale) può scattare un allarme
  - il backend pubblica su un topic MQTT (canale comandi) -> comunicazione bidirezionale
- **idea**: impostazione di regole automatiche
  - es: se la soglia sale o scende sotto un certo valore, allora spegni il nodo X

## Interfaccia grafica

- deve essere responsive, ovvero capace di adattarsi a varie tipologie di schermi/dispositivi
- deve permettere la visualizzazione dei dati acquisiti dai sensori, in particolare tramite l'utilizzo di grafici opportuni
  - le grandezze fisiche devono essere visualizzate sullo stesso grafico (ovviamente con possibilità di mascherarne alcune)
  - ogni grandezza dovrà avere il proprio asse delle ordinate (sempre visibile), mentre quello delle ascisse è comune a tutte e rappresenta il tempo trascorso
- deve consentire di scaricare i dati in locale nell'intervallo temporale desiderato nei formati `.xlsx`, `.csv` e `.json`
- deve permettere la massima personalizzazione possibile per ogni caso d'uso con i minimi cambiamenti necessari da effettuare

## Autenticazione ed autorizzazione

---

- la piattaforma deve consentire l'accesso solo ad utenti autenticati al sistema
- saranno presenti diversi livelli di autenticazione:
  - **utente**: generalmente può soltanto visualizzare dati e schermate senza modificare nulla
  - **admin**: gestisce utenti, impianti, dispositivi, allarmi e soglie
  - **gestore (?)**: utente con qualche autorizzazione in più (es: permesso di marcare un allarme come "*solved*")
- autenticazione mediante username e password
- servizio per recupero password tramite mail (?)

## Backend

---

- tecnologie impiegate:
  - Node.js `v18.13.0`
  - MongoDB `v6.0.4`
  - Mosquitto `v2.0.15`
- dati utente da memorizzare:
  - username
  - email
  - password
  - ruolo

## Hosting

---

Per la comparazione tra i vari servizi di hosting esaminati vedere [qui](#).