

# ANALISI TECNICA

## Architettura del sistema

L'applicazione è una Single Page Application (SPA), ovvero un'applicazione web che funziona come un'unica pagina, caricando inizialmente tutto il codice necessario (HTML, CSS e JavaScript) e aggiornando dinamicamente solo le parti della pagina che cambiano, anziché ricaricare l'intera pagina ad ogni interazione.

È strutturata in modo modulare, con componenti separati e riutilizzabili — ad esempio componenti diversi per mostrare la temperatura, il vento, i grafici e le notifiche.

L'applicazione si interfaccia con un server ThingsBoard, utilizzato per la raccolta, gestione e visualizzazione dei dati IoT. ThingsBoard fornisce l'infrastruttura per ricevere i dati dai dispositivi, elaborarli e renderli disponibili alla SPA tramite API, permettendo così l'aggiornamento in tempo reale delle informazioni mostrate all'utente.

## Tecnologia di frontend

Per il frontend utilizzo:

- HTML5 per la struttura e i contenuti.
- CSS3 per lo stile, il layout che si adatta a diversi schermi (responsive) e le animazioni.
- JavaScript (ES6+) per la logica dell'app, aggiornare i dati e modificare la pagina senza ricaricarla.

Come framework o libreria si possono usare React, Vue.js o Angular, che aiutano a gestire i componenti e lo stato dell'app. Inoltre, si usa un sistema di routing interno per passare da una schermata all'altra senza caricare la pagina da capo.

## Visualizzazione grafica

Per i grafici si può usare Chart.js, che è semplice, leggero e si integra facilmente. Permette di creare grafici a linee, a barre e altri tipi base. Se in futuro servono grafici più complessi, si può pensare a librerie come D3.js o ApexCharts.

I grafici mostrano dati come temperatura, umidità e pressione nel tempo, mentre le precipitazioni sono visualizzate con barre. Per il vento e le condizioni del cielo, si usano icone animate o sovrapposizioni.

## Responsività e compatibilità

Il layout e i componenti sono pensati per adattarsi automaticamente a dispositivi diversi (computer, tablet, smartphone). L'app è testata per funzionare sui browser più usati come Chrome, Firefox, Safari ed Edge.

## Gestione degli errori e stato

È importante gestire errori come problemi di rete o dati non disponibili, mostrando messaggi chiari e utili all'utente. Lo stato dell'app (cioè i dati e informazioni condivise tra componenti) viene gestito in modo centralizzato, usando strumenti come Redux o Vuex, per evitare incongruenze.

**Performance e ottimizzazione**

Si cerca di ottimizzare il caricamento iniziale riducendo la quantità di codice JavaScript da scaricare. I dati si aggiornano senza ricaricare la pagina completamente, e quando possibile si usano sistemi di caching per migliorare la velocità di risposta.

**Sicurezza**

La comunicazione tra frontend e backend avviene sempre con protocollo sicuro HTTPS. Inoltre, l'app è protetta contro attacchi comuni come XSS (cross-site scripting) e CSRF (cross-site request forgery), sia lato frontend che backend.