

Analisi Dei Requisiti

ANALISI DEI REQUISITI

Requisiti funzionali

- **Acquisizione dati dal sensore**
Il sistema deve raccogliere i dati meteo dal sensore locale, come temperatura, umidità, pressione, velocità e direzione del vento.
I dati devono aggiornarsi automaticamente con una frequenza che si può configurare (per esempio ogni 3 ore).
- **Visualizzazione dati in tempo reale**
I dati raccolti devono essere mostrati in tempo reale o quasi, senza che l'utente debba aggiornare manualmente la pagina.
- **Dashboard principale**
Deve esserci una schermata principale che riassume chiaramente i dati più importanti: temperatura, umidità, pressione e vento.
Da qui si deve poter interagire per vedere informazioni più dettagliate.
- **Visualizzazione previsioni meteo**
L'utente deve poter vedere le previsioni meteo dettagliate, sia orarie per 24 ore, sia giornaliere per 7 giorni.
Le previsioni saranno mostrate con grafici e icone facili da capire e interattive.
- **Gestione notifiche e allerte**
Il sistema deve avvisare l'utente in caso di eventi meteo importanti (come temporali, vento forte o neve).
L'utente deve poter impostare le soglie per ricevere queste notifiche.
- **Personalizzazione**
L'utente deve poter scegliere le unità di misura (Celsius/Fahrenheit, km/h o mph, mm o pollici).
Deve poter selezionare il tema grafico (chiaro o scuro).
Deve poter scegliere la lingua dell'interfaccia.
- **Esportazione dati**
L'utente deve poter esportare i dati storici per fare analisi esterne, in formati comuni come CSV o JSON.

Requisiti non funzionali

- **Usabilità**
L'interfaccia deve essere semplice e intuitiva, facile da usare anche per chi non è esperto.
Le informazioni devono essere chiare e immediate.
- **Responsività**
L'interfaccia deve adattarsi bene a tutti i dispositivi: computer, tablet e smartphone.
- **Compatibilità**
Il sistema deve funzionare correttamente sui principali browser: Chrome, Firefox, Safari ed Edge.
- **Affidabilità**
Deve esserci un sistema che gestisce gli errori: se manca la connessione

al sensore o i dati non arrivano, deve comparire un messaggio chiaro, ad esempio “Errore di connessione con il sensore”.

- **Performance**

I dati devono aggiornarsi e mostrarsi rapidamente, senza rallentamenti evidenti.

- **Sicurezza**

I dati devono essere protetti quando viaggiano tra backend e frontend, usando protocolli sicuri come HTTPS.

Requisiti di interazione

- **Navigazione**

L'utente deve poter spostarsi facilmente tra le varie schermate (dashboard, dettagli previsioni, mappe, impostazioni).

Deve esserci il supporto per gesture touch, come lo swipe per cambiare giorno o ora.

- **Filtri e selezione**

L'utente deve poter scegliere intervalli di tempo per vedere i dati storici, per esempio giornalieri, settimanali o mensili.

- **Accessibilità**

Il sistema deve seguire le linee guida base per l'accessibilità, come un buon contrasto dei colori, testi leggibili e possibilità di navigare usando solo la tastiera.

Analisi Funzionale

ANALISI FUNZIONALE

Scopo del frontend del meteo

Il frontend del meteo serve a fornire agli utenti informazioni sul tempo in modo semplice, veloce e interattivo. Deve permettere di vedere i dati attuali, le previsioni a breve e lungo termine e eventuali allerte, così da aiutare a prendere decisioni basate sulle condizioni del tempo.

Il frontend è l'interfaccia che collega l'utente al backend, che raccoglie e elabora i dati meteorologici provenienti da un sensore locale.

Funzionalità principali

- **Dashboard principale**

È la schermata iniziale dove si vedono le informazioni più importanti per il posto in cui è installato il sensore.

- Mostra la località (con possibilità futura di geolocalizzazione).
- Condizioni attuali: temperatura, stato del cielo (soleggiato, nuvoloso, pioggia, neve) con un'icona rappresentativa.
- Dati extra come umidità, pressione, velocità e direzione del vento, visibilità.
- Mini-grafici o icone che riassumono l'evoluzione prevista nelle ore o giorni successivi.
- L'utente può cliccare su qualsiasi elemento per vedere dettagli o scorrere per cambiare giorno o ora.

- **Previsioni meteo**

Mostra i dati delle previsioni dettagliate:

- Visualizzazione oraria delle ultime 24 ore e giornaliera degli ultimi 7 giorni.
- Grafici a linee per temperatura, umidità e pressione.
- Barre per precipitazioni previste.
- Icone animate per vento e nuvole.
- Si può selezionare un giorno per vedere i dati ora per ora e passare sopra i grafici per vedere i valori precisi.

- **Mappe meteo interattive (funzionalità futura)**

Mappe semplificate per mostrare visivamente dati come intensità del vento o variazioni di temperatura.

- **Notifiche e allerte**

- Notifiche push o visive per eventi importanti (es. temporali, vento forte, neve).
- Cronologia delle allerte consultabile dall'interfaccia.
- Possibilità di personalizzare le soglie di notifica.

- **Dati storici e analisi**

- Grafici con dati storici di temperatura, precipitazioni, vento e umidità.
- Confronti tra periodi diversi (es. oggi vs ieri, mese corrente vs media storica).

- Possibilità di esportare dati per analisi esterne.
- **Personalizzazione e impostazioni**
 - Scelta delle unità di misura: Celsius o Fahrenheit, km/h o mph, mm o pollici.
 - Tema dell'interfaccia chiaro o scuro, oppure adattato alle condizioni meteo (es. cielo sereno = sfondo azzurro).
 - Lingua selezionabile per testi e descrizioni.
 - Attivazione, disattivazione e personalizzazione delle notifiche.

Requisiti d'interfaccia

- **Layout**

L'interfaccia deve essere organizzata in sezioni chiare e facili da navigare, ad esempio con una barra in alto per selezionare la località, l'area centrale per le condizioni meteo e pannelli laterali o in basso per dettagli e impostazioni.
- **Colori**

Uso di una palette coerente con le condizioni meteo, come blu per la pioggia o giallo per il sole, con attenzione al contrasto per garantire una buona leggibilità.
- **Icone**

Icone intuitive per rappresentare condizioni come sole, nuvole, pioggia, neve e vento, e per i comandi di navigazione.
- **Adattabilità**

Interfaccia responsive, cioè ottimizzata per funzionare bene su computer, tablet e smartphone.

Gestione account e accessi

Per garantire una fruizione personalizzata e sicura dell'applicazione meteo, il sistema supporta la gestione di account utente con diversi livelli di accesso.

Tipi di account

- **Utente base**
 - Può visualizzare le informazioni meteo in tempo reale e le previsioni.
 - Può personalizzare la visualizzazione (tema, lingua, unità di misura).
 - Riceve notifiche secondo le impostazioni predefinite o personalizzate.
- **Amministratore**
 - Può gestire gli account utente (creazione, modifica, rimozione).
 - Può configurare i parametri del sensore locale e aggiornare il sistema.
 - Accesso completo a tutte le funzionalità e dati.
 - Gestione delle soglie di allerta globali e delle impostazioni predefinite.

Autenticazione e sicurezza

- Login tramite email e password con possibilità di recupero credenziali.
- Supporto per autenticazione a due fattori (2FA) (opzionale).
- Connessioni sicure tramite HTTPS.
- Protezione dei dati personali secondo normative GDPR.

Profilo utente

Ogni utente ha accesso a una sezione "Profilo" dove può:

- Modificare le proprie preferenze.
- Gestire le notifiche attive.
- Visualizzare cronologia degli accessi.
- Vedere e gestire i dispositivi collegati (es. più browser o app mobile).

Analisi Tecnica

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.