



SAPIENZA
UNIVERSITÀ DI ROMA

Air Quality Map

Facoltà di Ingegneria dell'informazione, informatica e statistica
Corso di Laurea in Informatica

Candidato

Gianluca Culaon

Matricola 1846795

A handwritten signature in black ink, appearing to read 'Gianluca Culaon'.

Relatore

Prof. Paolo Bottoni

A handwritten signature in black ink, appearing to read 'Paolo Bottoni'.

Anno Accademico 2022/2023

Air Quality Map

Relazione di Tirocinio. Sapienza – Università di Roma

© 2023 Gianluca Culaon. Tutti i diritti riservati

Questa relazione di tirocinio è stata composta con L^AT_EX e la classe Sapthesis.

Email dell'autore: culaon.1846795@studenti.uniroma1.it

Sommario

Air Quality Map è una Web app progettata e sviluppata per tracciare ed analizzare la qualità dell'aria, mostrandone visivamente i risultati attraverso una mappa interattiva. La relazione è divisa in 5 parti principali di seguito elencate:

1. **Introduzione**, in questa sezione verrà introdotto il problema e la soluzione proposta.
2. **Analisi e Requisiti**, in questa sezione verranno mostrate tutte le funzionalità dell'applicazione.
3. **Tecnologie Utilizzate**, in questa sezione verrà fatta una panoramica delle tecnologie utilizzate.
4. **Descrizione del Progetto e della sua Implementazione**, in questa sezione verrà chiarito ulteriormente l'obiettivo del progetto, in particolare andando a vedere le funzionalità a livello implementativo.
5. **Test e Feedback**, in questa sezione verrà spiegata come è stata gestita la parte relativa ai test sia a livello di design sia a livello implementativo.

Indice

1	Introduzione	1
1.1	Come Monitorare la Qualità Dell'aria?	1
1.1.1	Come funziona una stazione di monitoraggio?	2
1.2	Come Valutare la Qualità dell'Aria?	2
1.3	Air Quality Map	4
2	Analisi e requisiti	5
2.1	Analisi del Contesto e Studio di Fattibilità	5
2.2	Requisiti Funzionali	5
2.3	Requisiti Non Funzionali	9
3	Architettura e Tecnologie	11
3.1	Front-end	12
3.1.1	Javascript & React	12
3.1.2	HTML & CSS	13
3.2	Back-end	13
3.2.1	MongoDB	13
3.2.2	Node & Express	13
3.2.3	API & Librerie	14
4	Il progetto e l'Implementazione	15
4.1	Richiesta Dati	15
4.2	Express	15
4.2.1	Metodi HTTP	15
4.3	Axios	16
4.4	Richiesta Dati: Le Difficoltà Riscontrate	16
4.4.1	Promises	17
4.5	Elaborazione dei Dati	18
4.5.1	geoJSON	18
4.5.2	Struttura di un file GeoJSON	18
4.6	La mappa	18
4.7	Sidebar	20
4.8	Analisi Multivariata	21
4.8.1	Matrice di Correlazione	21
4.8.2	Coefficiente di Pearson	21
4.8.3	Regressione Lineare	23
4.8.4	Regressione Multipla	25
4.9	Control Panel	27
4.9.1	Redux	27

4.9.2	Aggiornamento Dati	28
4.9.3	Layers Inquinanti	28
4.9.4	Layer Vento	29
4.9.5	Layer Vento Scalare	29
4.9.6	Visualizzazione Globo	29
4.9.7	Locazione Stazioni	30
4.9.8	Slider Temporale	30
4.10	Database	30
4.11	Extra	31
4.11.1	Info Modale	31
4.11.2	Github	31
4.11.3	Titoli Modali Sidebar	31
5	Test e Feedback	32
6	Conclusioni	33
6.0.1	Sviluppi Futuri	33
6.0.2	Considerazioni Finali	33
	Bibliografia	34

Capitolo 1

Introduzione

Nella società attuale, la qualità dell'aria è un tema di crescente rilevanza e preoccupazione. L'ambiente in cui viviamo è un bene di inestimabile valore e va protetto per il benessere delle generazioni presenti e future. Il monitoraggio dell'ambiente diventa quindi una responsabilità che richiede attenzione costante e sforzi dedicati. Quando si parla di qualità dell'aria ci si riferisce alla composizione chimica e fisica dell'atmosfera circostante, prestando particolare attenzione alla presenza di inquinanti atmosferici che possono influenzare la salute umana, la biodiversità e il clima.

In un mondo sempre più industrializzato e urbanizzato, le emissioni, soprattutto di natura antropogenica, hanno contribuito ad un aumento significativo dell'inquinamento atmosferico. Questo fenomeno ha conseguenze dirette sulla salute umana, con una vasta gamma di patologie legate all'esposizione prolungata a inquinanti atmosferici. Gli studi [1] riportati dall'OMS, mostrano una correlazione tra gli intervalli giornalieri di concentrazione di particolato (PM) e la mortalità giornaliera, in particolare è stato registrato un aumento dell'indice di morbidità tra i soggetti più delicati [2]. Alcuni studi mostrano addirittura che il particolato può invadere le parti più profonde delle vie aeree raggiungendo quindi più facilmente il flusso sanguigno.

1.1 Come Monitorare la Qualità Dell'aria?

Per monitorare la qualità dell'aria si utilizzano le stazioni di rilevamento. Una stazione di rilevamento della qualità dell'aria è un dispositivo complesso utilizzato per valutare, in una determinata area geografica, la composizione chimica e le concentrazioni di inquinanti dell'aria.

In particolare vengono tracciati 6 inquinanti:

- Ozono (O₃)
- Particolato 2.5 (PM 2.5)
- Particolato 10 (PM 10)
- Monossido di carbonio (CO)
- Biossido di zolfo (SO₂)
- Biossido di azoto (NO₂)

1.1.1 Come funziona una stazione di monitoraggio?

La stazione cattura campioni di aria dall'ambiente circostante, l'aria campionata passa attraverso un sistema di filtraggio che rimuove particelle solide e polveri sospese. I componenti chimici dell'aria, come gli ossidi di azoto, gli idrocarburi, il biossido di zolfo e altri inquinanti, vengono quindi separati e analizzati attraverso metodi chimici e fisici avanzati, come ad esempio la spettroscopia, cromatografia e rilevatori specifici per ciascuna sostanza.

Una volta ottenute le rilevazioni, vengono inviate ai sistemi informatici che provvederanno ad utilizzare i dati raccolti.



Figura 1.1. Stazione di Monitoraggio

1.2 Come Valutare la Qualità dell'Aria?

Una volta ottenuti i dati dalle stazioni di monitoraggio è possibile calcolare l'Air Quality Index (AQI). L'AQI è un sistema di valutazione che aiuta a leggere e categorizzare il livello di un inquinante o in generale dell'aria. Per calcolare l'AQI si utilizza questa formula:

$$I_p = \frac{I_{Hi} - I_{Lo}}{BP_{Hi} - BP_{Lo}} \times (C_p - BP_{Lo}) + I_{Lo} \quad (1.1)$$

Consultando la tabella (figura 1.2) possiamo calcolare i vari elementi dell'equazione:

I_p è l'Indice dell'inquinante,

C_p è la concentrazione dell'inquinante p ,

BP_{Hi} è il valore del breakpoint che è maggiore o uguale a C_p ,

BP_{Lo} è il valore del breakpoint che è minore o uguale a C_p ,

I_{Hi} è il valore AQI corrispondente a BP_{Hi} ,

I_{Lo} è il valore AQI corrispondente a BP_{Lo} .

Il risultato finale ovvero l'indice dell'inquinante, viene chiamato subindex.

These Breakpoints...							...equal this AQI	...and this category
O ₃ (ppm) 8-hour	O ₃ (ppm) 1-hour ¹	PM _{2.5} (µg/m ³) 24-hour	PM ₁₀ (µg/m ³) 24-hour	CO (ppm) 8-hour	SO ₂ (ppb) 1-hour	NO ₂ (ppb) 1-hour	AQI	
0.000 - 0.054	-	0.0 - 12.0	0 - 54	0.0 - 4.4	0 - 35	0 - 53	0 - 50	Good
0.055 - 0.070	-	12.1 - 35.4	55 - 154	4.5 - 9.4	36 - 75	54 - 100	51 - 100	Moderate
0.071 - 0.085	0.125 - 0.164	35.5 - 55.4	155 - 254	9.5 - 12.4	76 - 185	101 - 360	101 - 150	Unhealthy for Sensitive Groups
0.086 - 0.105	0.165 - 0.204	(55.5 - 150.4) ³	255 - 354	12.5 - 15.4	(186 - 304) ⁴	361 - 649	151 - 200	Unhealthy
0.106 - 0.200	0.205 - 0.404	(150.5 - 250.4) ³	355 - 424	15.5 - 30.4	(305 - 604) ⁴	650 - 1249	201 - 300	Very unhealthy
(²)	0.405 - 0.504	(250.5 - 350.4) ³	425 - 504	30.5 - 40.4	(605 - 804) ⁴	1250 - 1649	301 - 400	Hazardous
(²)	0.505 - 0.604	(350.5 - 500.4) ³	505 - 604	40.5 - 50.4	(805 - 1004) ⁴	1650 - 2049	401 - 500	Hazardous

Figura 1.2. Tabella Breakpoints AQI

Una volta ottenuti i subindex per ciascun inquinante, l'AQI della zona relativa alla stazione di monitoraggio coinciderà con il massimo tra i subindex calcolati.

Per capire la qualità dell'aria dal valore ottenuto, si può consultare la tabella (figura 1.3) presa dal sito governativo degli USA gestito dall'agenzia per la protezione ambientale (EPA).

Daily AQI Color	Levels of Concern	Values of Index	Description of Air Quality
Green	Good	0 to 50	Air quality is satisfactory, and air pollution poses little or no risk.
Yellow	Moderate	51 to 100	Air quality is acceptable. However, there may be a risk for some people, particularly those who are unusually sensitive to air pollution.
Orange	Unhealthy for Sensitive Groups	101 to 150	Members of sensitive groups may experience health effects. The general public is less likely to be affected.
Red	Unhealthy	151 to 200	Some members of the general public may experience health effects; members of sensitive groups may experience more serious health effects.
Purple	Very Unhealthy	201 to 300	Health alert: The risk of health effects is increased for everyone.
Maroon	Hazardous	301 and higher	Health warning of emergency conditions: everyone is more likely to be affected.

Figura 1.3. Tabella valori AQI

1.3 Air Quality Map

Il sistema descritto in questa relazione ha quindi come scopo quello di fornire all'utente una Web application per il monitoraggio ambientale. Ciò include anche l'informare sui rischi legati all'inquinamento dell'aria e quindi alla sensibilizzazione su questo argomento. Per raggiungere questi obiettivi, Air Quality Map cerca di sfruttare la semplicità di una mappa interattiva e dell'interfaccia e, soprattutto, cerca di elaborare dati complessi rendendoli facilmente comprensibili all'utenza tramite l'utilizzo di grafici e matrici.

Capitolo 2

Analisi e requisiti

Prima di passare alla progettazione è stato svolto un lavoro di analisi per capire il problema e valutare sia l'efficacia del progetto proposto sia la fattibilità.

2.1 Analisi del Contesto e Studio di Fattibilità

Lo scopo di questo progetto è quello di monitorare l'ambiente fornendo agli utenti una visione semplificata e immediata della qualità dell'aria.

Sono già presenti sul mercato applicazioni Web con questo scopo; ciò che ha spinto alla creazione di questo progetto è la mancanza di una buona interazione, di una UI immediata e soprattutto di una parte analitica che potesse dare qualche informazione aggiuntiva.

Per valutare la fattibilità del progetto, sono state fatte ricerche e progetti di prova, soprattutto per testare la reperibilità dei dati, i quali spesso in questi ambiti non sono offerti gratuitamente.

2.2 Requisiti Funzionali

Di seguito sono elencati i requisiti funzionali dell'applicazione:

1. **Visualizzazione della Mappa:** La mappa è la funzionalità chiave di questo progetto, l'idea nasce proprio dal realizzare una choropleth map, ovvero una mappa in cui paesi o stati/regione sono colorati in base ad un valore.

Nel contesto della mia applicazione questo valore è l'AQI di cui si è discusso nel Capitolo 1, mentre la scala di colori è la stessa riportata da AirNow [4], il quale è uno dei principali siti di riferimento su questo tema.

In tal modo, la navigazione dell'utente risulta agevolata grazie al mantenimento della coerenza visiva dei colori.

2. **Interazione della Mappa:** La mappa è interagibile, questo significa che l'utente non solo può visualizzare visivamente l'inquinamento di una zona, ma può anche interagire con essa, tramite la funzione di hover o tramite la funzione di click.

L'hover comporta la comparsa di un pop-up mobile che segue il cursore, al cui interno ci sono le informazioni chiave relative a quella zona, questo per far sì che la navigazione risulti veloce ed efficiente nel caso in cui all'utente non voglia visualizzare informazioni dettagliate.

Il click invece comporta la comparsa della sidebar descritta nel punto successivo.

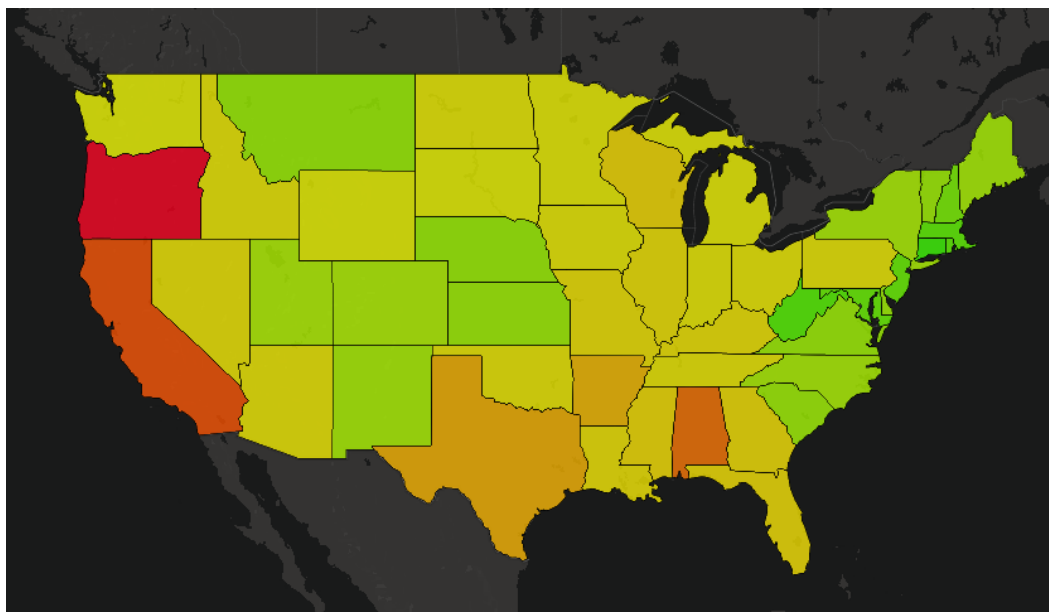


Figura 2.1. Mappa di Air Quality Map

3. **Sidebar:** La sidebar è un elemento a comparsa in cui vengono mostrate tutte le informazioni riguardo il paese o lo stato/regione cliccato. Oltre le informazioni, la sidebar mostra dati meteorologici, grafici e parte di analisi.

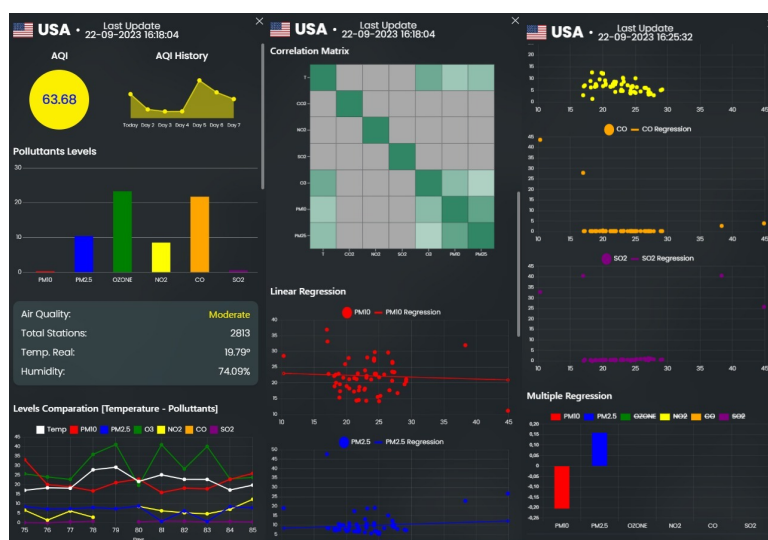


Figura 2.2. Sidebar di Air Quality Map

I grafici sono stati sviluppati in modo tale da risultare anch'essi interagibili. L'utente potrà quindi cliccare un inquinante nella legenda per disattivarlo/attivarlo, con conseguente animazione nel grafico. Questa possibilità di interazione nella parte di analisi risulta ancor più fondamentale. Ad esempio nella regressione multipla, si dà l'opportunità all'utente di decidere quali inquinanti includere nel calcolo della regressione.

4. **Control Panel:** Il Control Panel è invece il cuore delle funzionalità dell'applicazione in termini end-to-user.

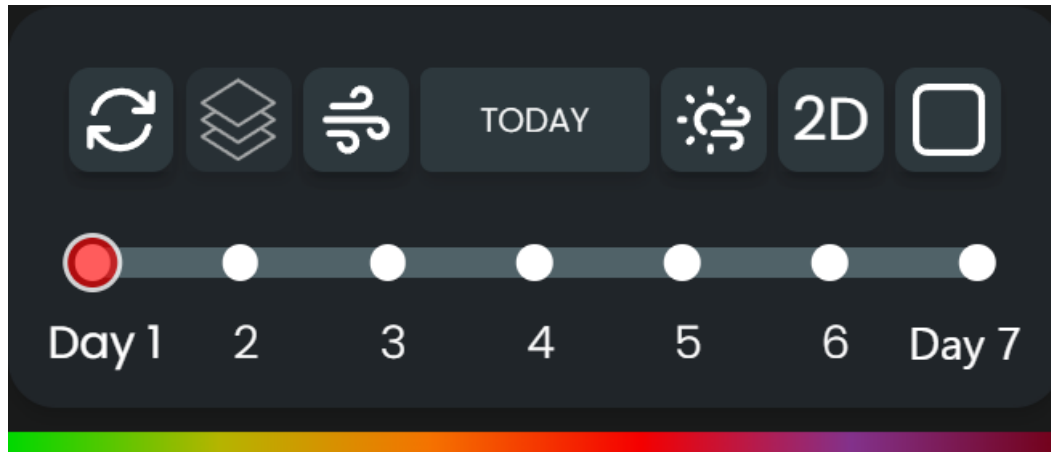


Figura 2.3. Pannello di Controllo di Air Quality Map

Di seguito vengono elencate tutte le funzionalità presenti al suo interno:

- Visualizzazione del vento, questa funzionalità permette all'utente di attivare un layer che si occupa di mostrare direzione e forza del vento tramite particelle animate.

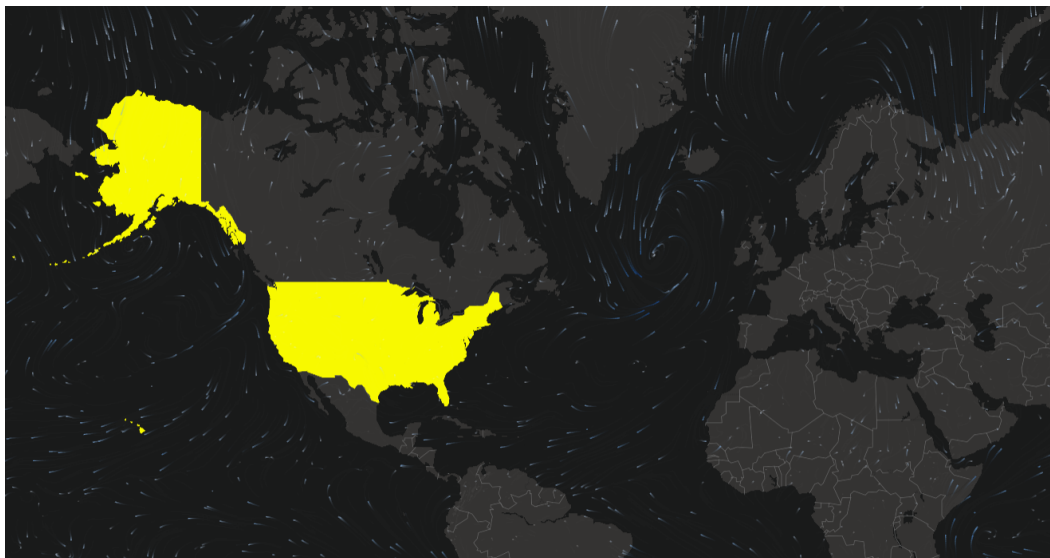


Figura 2.4. Layer Vento Attivo

- Slider temporale, questa funzionalità permette all'utente di impostare il giorno "visualizzato". Ogni giorno è legato ai suoi dati, modificando il giorno visualizzato, vengono quindi modificati anche i dati visualizzati.
- Reloading dei dati, questa funzionalità permette di ottenere le ultime misurazioni, aggiornando i dati nel database.
- Visualizzazione alternativa degli stati/regioni, questa funzionalità permette all'utente di selezionare uno fra i sei inquinanti per impostare la colorazione della mappa in base solamente al subindex dell'inquinante selezionato.
- Visualizzazione del vento con dati scalari, questa funzionalità permette all'utente di attivare un layer in cui viene mostrato il vento in modo alternativo, utilizzando una heatmap.
- Visualizzazione del globo, questa funzionalità consente all'utente di scegliere il tipo di visualizzazione della mappa, 2D o 3D. ¹
- Visualizzazione stazioni, questa funzionalità consente all'utente di visualizzare, tramite puntini luminosi o heatmap, tutte le stazioni di monitoraggio coinvolte nel rilevamento dati della giornata selezionata.

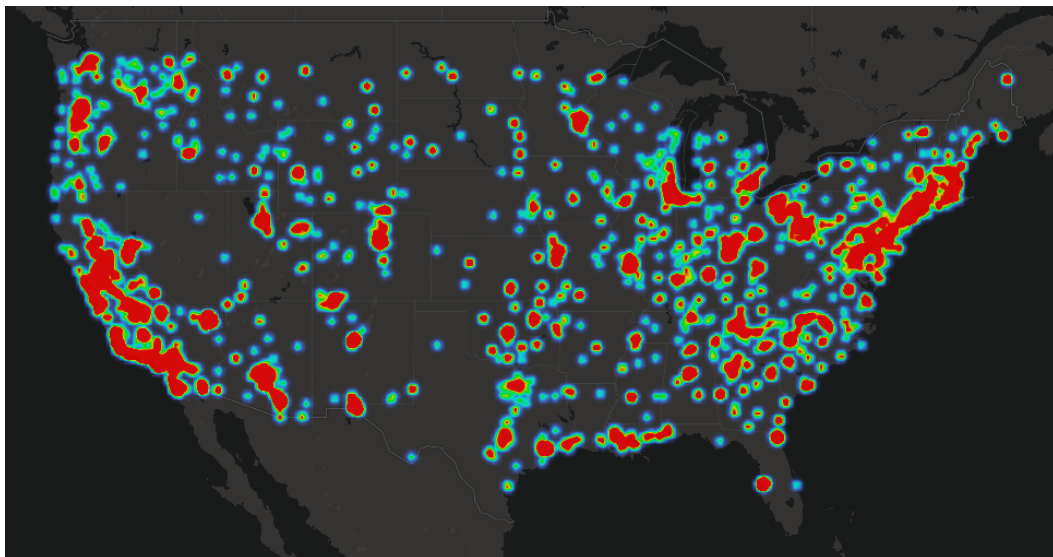


Figura 2.5. Locazione delle Stazioni con Visualizzazione Heatmap

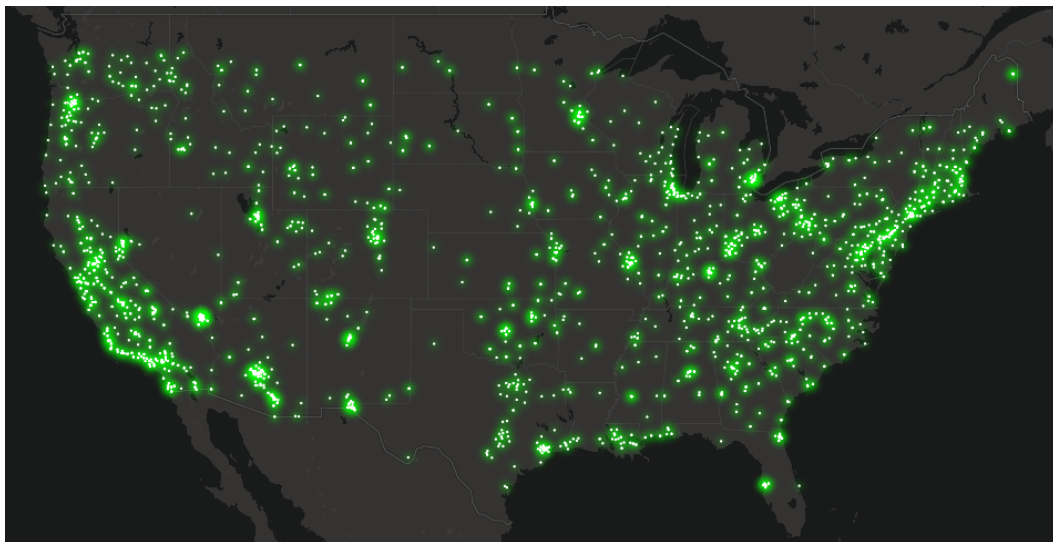


Figura 2.6. Locazione delle Stazioni con Visualizzazione Dots

- Infine il pannello di controllo ha in basso la legenda che rappresenta la colorazione della mappa.

2.3 Requisiti Non Funzionali

Di seguito sono elencati i requisiti non funzionali dell'applicazione:

1. **Color-Blind Mode:** Funzione di accessibilità, consente agli utenti daltonici di cambiare i colori visualizzati, sia dell'intera mappa sia dei grafici, consentendo quindi agli utenti, che non riescono a vedere correttamente il verde ed il rosso (il tipo di daltonismo più diffuso), di poter visualizzare comunque una scala cromatica.

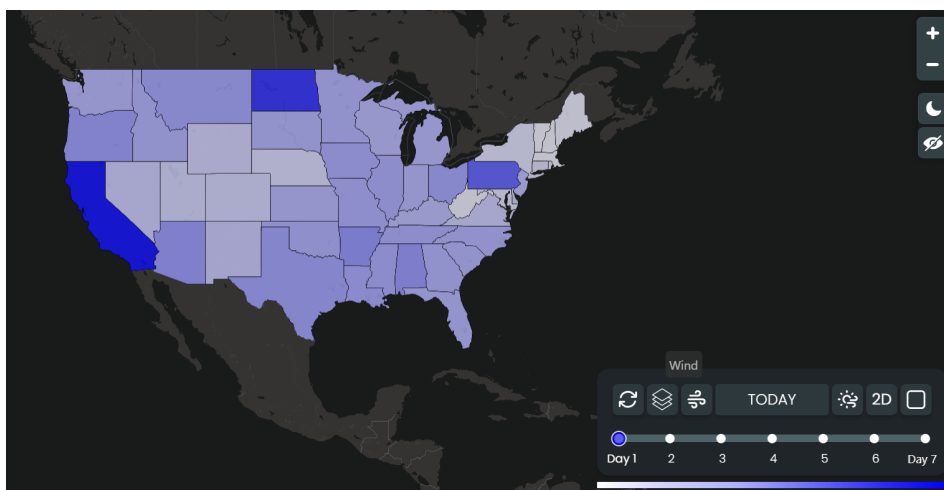


Figura 2.7. Visualizzazione per Daltonici

¹NOTA: Se la visualizzazione del globo è 3D, il vento, in entrambe le sue visualizzazioni, viene bloccato a causa di un estremo calo delle performance

La Color-Blind mode è stato il requisito non funzionale più complesso da sviluppare poichè, nel caso specifico di questo progetto, i colori svolgono un ruolo fondamentale e sono costantemente presenti.

2. **Night Mode:** Funzione di accessibilità, consente all'utente di scegliere se utilizzare un tema scuro o chiaro, scelta che si riflette sia sull'interfaccia, sia sui grafici sia sulla mappa stessa.
3. **Responsiveness:** Il progetto è stato sviluppato tenendo conto di tutte le tipologie di dispositivi. Ciò consente all'utente di poter utilizzare Air Quality Map anche da tablet e smartphone.
4. **Scalabilità:** Il progetto attualmente ottiene i dati e quindi offre i suoi servizi solo per gli USA. Il motivo dietro a questa scelta è puramente economico, poichè già solo ottenere i dati degli USA comporta un grosso carico per le API che offrono questo servizio gratuitamente, esagerando con l'ammontare di dati richiesto, i server delle API restituiscono messaggi di errore rendendo impossibile ottenere i dati per l'intero globo. Utilizzando API a pagamento questo problema non sussiste.

Nonostante ciò il progetto è pensato e sviluppato per poter essere esteso a tutto il mondo, anche a livello di funzionalità. Durante lo sviluppo, la scalabilità è stata una scelta prioritaria. Ad esempio per la funzionalità dello slider temporale il quale permette all'utente di scegliere quale giorno "visualizzare", è stato sviluppato per poter essere esteso ad oltre i 7 giorni attuali. La scalabilità, in termini di storico dei dati, non è utile solo alla funzionalità dello slider temporale ma anche per la parte di analisi, poichè i grafici utilizzano proprio lo storico dei dati per poter effettuare analisi sempre più accurate.

Capitolo 3

Architettura e Tecnologie

Air Quality Map utilizza un'architettura client-server ed è stata progettata per fornire una piattaforma robusta e scalabile per la visualizzazione e l'analisi dei dati ambientali.

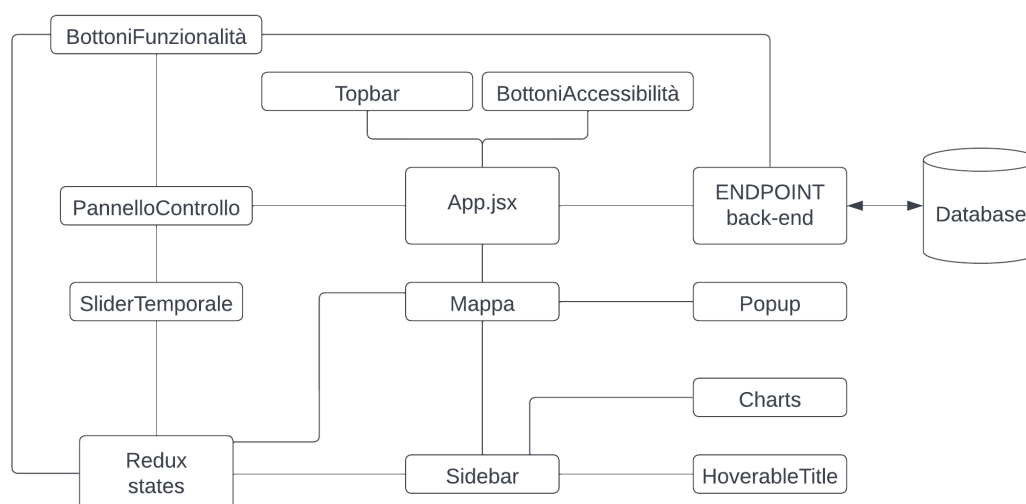


Figura 3.1. Diagramma dei Componenti di Air Quality Map

Come rappresentato nella figura 3.1, l'applicazione è suddivisa in componenti, ognuno con un ruolo specifico all'interno del sistema. Il componente principale è **App.jsx**, il quale contiene i componenti **Mappa**, **Sidebar** e **ControlPanel**. Questi 3 componenti madre a sua volta contengono i componenti figli a cui sono legati nel diagramma 3.1. Alcuni componenti, quali **Sidebar**, **Mappa**, **BottoniFunzionalità** e **SliderTemporale** interagiscono con lo store di **Redux** per il cambio degli stati.

I componenti favoriscono una chiara separazione delle responsabilità all'interno del codice. Ogni componente si concentra su una specifica funzionalità o visualizzazione, rendendo il codice più organizzato e manutenibile, inoltre scomponendo l'interfaccia utente in piccoli componenti, è più facile gestire la complessità dell'applicazione. Ogni componente può essere sviluppato, testato e mantenuto separatamente.

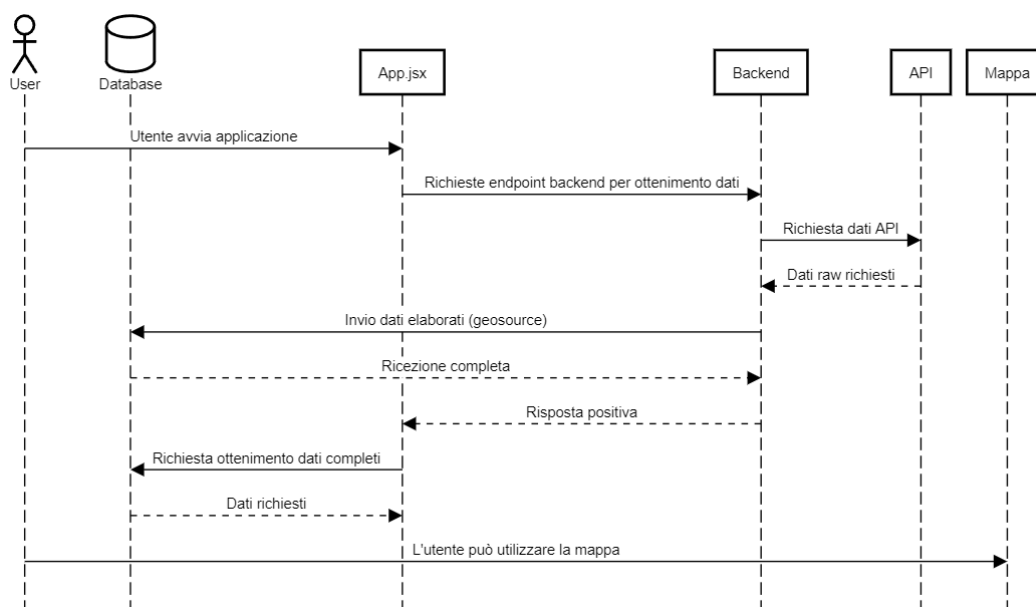


Figura 3.2. Diagramma di sequenza di Air Quality Map

Nella figura 3.2 viene rappresentato il Sequence Diagram ovvero una rappresentazione grafica utilizzata nell'ambito dell'ingegneria del software per modellare e visualizzare l'interazione tra gli oggetti o le componenti di un sistema durante un certo periodo di tempo. Nel caso di questo progetto, il diagramma mostra quello che succede una volta che l'utente avvia l'avvia applicazione. Inizialmente vengono fatte le richieste API tramite gli endpoint del back-end, ottenendo i dati raw. Successivamente questi dati vengono elaborati ed inseriti nel json di risposta alla richiesta iniziale del front-end. A questo punto il componente principale ovvero App.jsx può passare questi dati elaborati a tutti i componenti che li richiedono, in tal modo l'applicazione diventa pronta per l'utilizzo e l'utente può iniziare ad interagire con la mappa.

In linea con le più comuni pratiche di sviluppo, il sistema è stato suddiviso in front-end e back-end.

3.1 Front-end

Di seguito sono elencate le tecnologie scelte per la creazione dell'interfaccia utente.

3.1.1 Javascript & React

JavaScript [5] è un linguaggio di programmazione cruciale nel panorama dello sviluppo Web. È stato creato per consentire l'aggiunta di interattività e dinamicità alle pagine Web, trasformandole da statiche ad interattive. JavaScript è la chiave per la manipolazione del Document Object Model (DOM) questo significa che gli sviluppatori possono utilizzare JavaScript per aggiungere, rimuovere o modificare elementi HTML e CSS in risposta a eventi utente, è proprio questo a rendere possibile la creazione di interfacce utente dinamiche e reattive. Su questo linguaggio si basa React.

React [6] è una libreria JavaScript open-source ampiamente utilizzata per lo sviluppo Web. Questa libreria è stata progettata per risolvere alcune sfide comuni nello

sviluppo di applicazioni Web moderne, come la gestione efficiente del DOM e la creazione di componenti riutilizzabili. React è una libreria e non un framework e questo la rende facilmente integrabile nella maggior parte degli ambienti.

Per questo progetto era necessario uno strumento che fosse ampiamente supportato, flessibile e con una curva di apprendimento relativamente piccola, inoltre è stato fondamentale il riutilizzo dei componenti.

3.1.2 HTML & CSS

Acronimo di HyperText Markup Language, HTML è il linguaggio di marcatura utilizzato per creare pagine Web. La sua funzione principale è quella di definire la struttura e l'organizzazione del contenuto all'interno di una pagina Web. HTML svolge un ruolo cruciale in quanto fornisce un mezzo standardizzato per descrivere come i diversi elementi di una pagina Web, dovrebbero essere presentati e interconnessi. Senza HTML, la comunicazione e la condivisione di informazioni su Internet non sarebbero possibili, proprio per questo è il fondamento su cui si basa il Web moderno. HTML viene spesso utilizzato con CSS.

Acronimo di Cascading Style Sheets, CSS è un linguaggio utilizzato per definire l'aspetto e la formattazione di una pagina Web. Mentre HTML si occupa della struttura e del contenuto della pagina, CSS si concentra sull'aspetto estetico. CSS è fondamentale per la progettazione e la presentazione efficace delle pagine Web. Senza CSS, le pagine sarebbero prive di stile e formattazione, rendendo difficile per gli utenti interagire e comprendere il contenuto.

3.2 Back-end

Di seguito sono elencate le tecnologie scelte per la richiesta, l'elaborazione ed il salvataggio di dati.

3.2.1 MongoDB

MongoDB [7] è un sistema di gestione di database (DBMS) di tipo NoSQL, noto per la sua flessibilità e scalabilità. A differenza dei tradizionali database relazionali, MongoDB è basato su un modello di dati flessibile chiamato "documento". In MongoDB, i dati sono archiviati in documenti JSON-like, che possono contenere campi di dati di tipo diverso senza una struttura rigorosamente definita. Non essendo necessario un database relazionale, ho scelto MongoDB per le sue caratteristiche di scalabilità ed agilità, oltre che per la sua facilità di utilizzo. Nel contesto del progetto, essendo gli oggetti JSON-like, la scelta di questa tecnologia sembrava la più adeguata.

3.2.2 Node & Express

Node [8] è un ambiente runtime open source per eseguire codice javascript, viene utilizzato per la programmazione lato server, consentendo agli sviluppatori di utilizzare JavaScript per il codice sia lato client sia lato server. La sua caratteristica distintiva è la capacità di gestire operazioni asincrone in modo efficiente, rendendolo ideale per applicazioni Web in tempo reale, API RESTful, e molte altre applicazioni server-side. Nel contesto di questa applicazione, node è stata una scelta obbligata poichè per il back-end ho scelto di utilizzare il framework Express.

Express [9] è un framework Web per Node.js ampiamente utilizzato per la creazione di applicazioni Web scalabili e veloci. Questo framework offre un ambiente flessibile per la gestione delle richieste HTTP e la creazione di API Web. La sua popolarità è cresciuta rapidamente grazie alla sua semplicità d'uso e alla sua capacità di consentire agli sviluppatori di creare applicazioni Web robuste. In questo progetto è stato necessario per comunicare tra front-end e back-end creando endpoint GET e POST.

3.2.3 API & Librerie

Infine sono state utilizzate alcune API [11] e librerie fondamentali per lo sviluppo di questo progetto:

- Redux [10] è una libreria di gestione dello stato in JavaScript, la sua funzione è quella di mantenere uno stato globale in un'applicazione e gestire le modifiche allo stato in modo prevedibile e coerente.
- Mapbox GL JS [12] ovvero una libreria JavaScript open-source per la creazione di mappe interattive e personalizzate. Questa libreria offre agli sviluppatori gli strumenti necessari per integrare mappe di alta qualità e soprattutto offre la possibilità di personalizzare la mappa e di integrarla con dati personalizzati.
- Chart.js [13] è una libreria JavaScript open-source che consente agli sviluppatori di creare grafici interattivi e visualizzazioni dati dinamiche per le applicazioni Web. In questo progetto è stata essenziale per riuscire a far comprendere all'utente i risultati della parte analitica.
- Math.js [16] è una libreria JavaScript open-source che fornisce funzionalità avanzate di calcolo matematico e algebrico per applicazioni Web. È progettata per semplificare operazioni matematiche complesse e manipolazioni di dati numerici.
- Regression.js [17] è una libreria JavaScript specializzata nella regressione statistica. È utilizzata per modellare relazioni tra variabili e prevedere valori in base a dati esistenti.

Capitolo 4

Il progetto e l'Implementazione

Una volta conclusa la fase di analisi è iniziata la fase di progettazione. Prima di iniziare lo sviluppo vero e proprio, è stato necessario effettuare molteplici test con diverse librerie geospaziali per capire quale si adattasse meglio alle mie esigenze. Una volta scelto lo stack tecnologico da usare, è iniziato lo sviluppo del progetto.

4.1 Richiesta Dati

Al suo avvio, l'applicazione effettua le richieste API per ottenere sia le misurazioni odierne, sia i dati meteorologici. Entrambi questi dati sono ottenuti tramite le API di AirNow. Per effettuare le chiamate API dal front-end, vengono richiamati gli endpoint del back-end creati con Express.

4.2 Express

Express.js è un framework Web basato su Node.js che semplifica la creazione di applicazioni Web e API. Segue un modello di progettazione basato su middleware che permette di definire rotte, gestire richieste HTTP e implementare funzionalità specifiche del server. Ogni endpoint è associato a un percorso specifico (ad esempio, /api/data) e ad un metodo HTTP.

4.2.1 Metodi HTTP

HTTP è un protocollo di comunicazione che stabilisce regole e convenzioni per lo scambio di informazioni tra client e un server. Di seguito sono elencati i metodi più utilizzati:

- **GET:** Il metodo GET è utilizzato per richiedere dati da una risorsa specifica come un database. Nel caso di questo progetto il metodo GET è utilizzato ad esempio per ottenere i dati dei giorni precedenti.
- **POST:** Il metodo POST è impiegato per inviare dati a un server. Nel caso di questo progetto il metodo POST è utilizzato ad esempio per inviare la geosource della giornata attuale al database così da inserirla nello storico dei dati.

- **PUT**: Il metodo PUT è utilizzato per aggiornare una risorsa esistente o creare una nuova risorsa se non presente nel database.
- **DELETE**: Il metodo DELETE consente di richiedere la rimozione di una risorsa specifica dal server.

Nel back-end di questo progetto sono stati creati diversi endpoint, sia per la scrittura che per la lettura di dati. Dal front-end si accede a questi endpoint attraverso l'uso di fetch o della libreria axios, riuscendo quindi a scrivere e leggere nel server o effettuare chiamate API.

```
app.get("/bulk-datas", async (req, res) => {  
  try {  
    const db = client.db(dbName);  
    const collection = db.collection("bulkdata");  
    const bulkData = await collection.findOne();  
    if (bulkData) {  
      res.json(bulkData);  
    } else {  
      res.status(500).send(false);  
    }  
  } catch (error) {}  
});
```

Figura 4.1. Esempio Endpoint Back-end Creato con Express

4.3 Axios

Axios è una libreria JavaScript che semplifica la gestione delle richieste HTTP dal tuo front-end al tuo server back-end fornendo un'interfaccia semplice e intuitiva per effettuare richieste GET, POST, PUT, DELETE e altre operazioni HTTP.

4.4 Richiesta Dati: Le Difficoltà Riscontrate

Le difficoltà riscontrate sono di natura economica. Essendo API gratuite, hanno un rate limit molto basso ed inoltre non riescono a gestire carichi eccessivi. A causa di questi limiti è risultato impossibile anche solo ottenere, con un'unica chiamata API, i dati degli USA.

Per rimediare a questo problema sono stati utilizzati gli endpoint boundingbox. Questo endpoint permette di selezionare tramite 4 coordinate, un'area geografica per la quale ottenere i dati. Grazie a questo endpoint, è stato possibile suddividere gli USA in 5 zone più piccole effettuando per ognuna di queste zone una chiamata API. Così facendo, il server, gestendo chiamate più piccole, ha iniziato a restituire i dati in modo corretto.

«Divide et Impera»

Per le chiamate API viene utilizzato il costrutto "await" [14] in modo tale che, i passaggi successivi che richiedono l'utilizzo dei dati ottenuti dalle API, siano effettuati solo dopo il termine delle chiamate e quindi dopo l'ottenimento di tali dati. Questo però ha portato a dei problemi di performance notevoli, poichè le chiamate API sono risultate molto lente, questo perchè le chiamate venivano effettuato in modo sequenziale.

Per ovviare a questo problema sono state introdotte le "Promises" [15].

4.4.1 Promises

Le promesse sono un costrutto utilizzato per gestire operazioni asincrone in modo più efficace e organizzato. Una "promessa" rappresenta un valore che potrebbe non essere disponibile immediatamente, ma che sarà disponibile in futuro, o potrebbe non essere disponibile affatto.

In questo progetto la "promessa" viene effettuata sulle chiamate API per l'ottenimento dei dati delle stazioni di misurazione. In tal modo, essendo tutte le chiamate asincrone, il processo risulta estremamente velocizzato.

```
//endpoint to get daily datas from API
app.get("/get-daily-datas", async (req, res) => {
  try {
    const boundingBoxes = [
      "https://www.airnowapi.org/aq/data/?parameters=Oz",
      "https://www.airnowapi.org/aq/data/?parameters=Oz",
      "https://www.airnowapi.org/aq/data/?parameters=Oz",
      "https://www.airnowapi.org/aq/data/?parameters=Oz",
      "https://www.airnowapi.org/aq/data/?parameters=Oz"
    ];

    const promises = boundingBoxes.map((url) =>
      fetch(url).then((response) => response.json())
    );
    const responses = await Promise.all(promises);
    res.json(responses);
  } catch (error) {}
});
```

Figura 4.2. Codice Promises API di Air Quality Map

4.5 Elaborazione dei Dati

Una volta ottenuti i dati, è necessario elaborarli e trasformati nel geoJSON di occorrenza.

4.5.1 geoJSON

GeoJSON è un formato di dati geospaziali basato su JSON che viene utilizzato per rappresentare informazioni geografiche e geospaziali. È ampiamente utilizzato per memorizzare, scambiare e visualizzare dati geografici in un formato leggibile dalle macchine e facilmente interpretabile dagli sviluppatori.

4.5.2 Struttura di un file GeoJSON

I file GeoJSON sono organizzati come segue:

- **Tipo di oggetto (Type):** Il campo "type" indica il tipo di oggetto geografico rappresentato nel file, come ad esempio "Feature", "FeatureCollection", "Point", "LineString", "Polygon", ecc. In questo progetto sono riuscito a sperimentare quasi tutti i type citati.
- **Geometria (Geometry):** Se il tipo di oggetto è "Point", "LineString" o "Polygon", il file conterrà un campo "geometry" che descrive la geometria dell'oggetto, ovvero le coordinate geografiche che definiscono la forma e la posizione dell'oggetto.
- **Proprietà (Properties):** Oltre alla geometria, un oggetto GeoJSON può includere un campo "properties" che contiene informazioni aggiuntive associate all'oggetto. Nel caso di questo progetto, il campo proprietà è stato usato proprio per inserire le informazioni principali da memorizzare, come ad esempio il nome, l'AQI etc...
- **Collezione di oggetti (FeatureCollection):** Un file GeoJSON può anche contenere una raccolta di oggetti geografici (ad esempio punti, linee o poligoni). In questo caso, il tipo di oggetto sarà "FeatureCollection", e ciascun oggetto all'interno della collezione sarà un oggetto "Feature" che include sia la geometria che le proprietà associate. Questo è il caso dei geoJSON per i paesi, i quali conterranno all'interno gli oggetti feature, ovvero gli stati.

Il geoJSON ottenuto viene dato in input alla mappa.

4.6 La mappa

La mappa è stata sviluppata utilizzando la libreria mapbox. La prima sfida affrontata è stata quella di creare una choropleth map, ovvero una mappa colorata in modo gradiente a partire da una scala di colori-valori. Per far ciò, mapbox mette a disposizione la funzionalità di painting, la quale richiede la geosource ottenuta in precedenza. Grazie a quest'ultima, la funzionalità disegna sulla mappa dei poligoni in base alle coordinate all'interno del geoJSON, colorandoli con la modalità fill, ottenendo quindi l'effetto desiderato.

```
map.on("load", () => {
  let show = false;
  map.style.stylesheet.layers.forEach(function (layer) {
    if (layer.type === "symbol") { ...
  }
});
map.addSource("aqi", {
  type: "geojson",
  data: dataR,
});
map.addLayer({
  id: "state-aqi",
  source: "aqi",
  minzoom: zoomThreshold,
  type: "fill",
  paint: {
    "fill-color": [
      "interpolate",
      ["linear"],
      ["get", "AQI"],
      ...colorsLayers,
    ],
    "fill-opacity": [
      "case",
      ["boolean", ["feature-state", "hover"], false],
      1,
      0.75,
    ],
    "fill-outline-color": "rgba(0, 0, 0, 1)",
  },
});
```

Figura 4.3. Codice della Mappa

Nella figura 4.3 è rappresentato l'inizializzazione della mappa. La funzione load permette di aggiungere la geosource, una volta aggiunta è possibile aggiungere tutti i layer desiderati e personalizzarli come più si desidera.

Una volta colorata la mappa il passo successivo è stato quello di renderla interagibile. Per far ciò mapbox mette a disposizione degli event listener che si possono legare ai layer della mappa. Questo ha permesso la creazione degli effetti di hover e di click.

L'hover comporta la comparsa di un popup nel quale sono disponibili i dati base ovvero il nome della zona, l'AQI ed un grafico che mostra i valori degli inquinanti.

Il click invece comporta la comparsa o l'aggiornamento della sidebar.


```

mapRef.current.on("mousemove", "state-aqi", (e) => {
  mapRef.current.getCanvas().style.cursor = "pointer";
  if (e.features.length > 0) {
    if (hoveredPolygonId == null) {
      hoveredPolygonId = e.features[0].id;
      mapRef.current.setFeatureState(
        { source: "aqi", id: hoveredPolygonId },
        { hover: true }
      );
      setHoveredState([dataR, e.features[0].id, false]);
      setHoveredStateColor(e.features[0].layer.paint["fill-color"]);
    } else if (e.features[0].id !== hoveredPolygonId) {
      mapRef.current.setFeatureState(
        { source: "aqi", id: hoveredPolygonId },
        { hover: false }
      );
      hoveredPolygonId = e.features[0].id;
      mapRef.current.setFeatureState(
        { source: "aqi", id: hoveredPolygonId },
        { hover: true }
      );
      setHoveredState([dataR, e.features[0].id, false]);
      setHoveredStateColor(e.features[0].layer.paint["fill-color"]);
    }
    setPopupPosition({
      x: e.originalEvent.clientX,
      y: e.originalEvent.clientY,
    });
    setShowPopup(true);
  }
});

```

Figura 4.4. Event Listener della Mappa

Come si può vedere nella figura 4.4 si utilizza l'evento "mousemove" legando questo evento al layer "state-aqi", quindi questa porzione di codice viene eseguita ogni volta che il mouse si muove su quel layer. Inoltre, si occupa di impostare uno stato in modo tale che al suo interno abbia le informazioni riguardanti la zona cliccata.

4.7 Sidebar

La sidebar è un elemento chiave di questo progetto, poichè è il fulcro delle informazioni. Svilupparla è stato il passaggio che ha richiesto più tempo a causa di molteplici fattori:

- **Quantità di elementi:** Questo componente è ricco di elementi anche molto diversi tra loro. Ciò ha comportato, per ognuno di essi, uno studio sia per l'implementazione che per il design. Ci sono dati generici e dati più rilevanti quali lo storico del valore AQI di quella zona e dati meteorologici. Ci sono grafici semplici che mostrano i livelli degli inquinanti e ci sono i grafici complessi per la parte di analisi multivariata.

- **Comunicazione tra Componenti:** Uno dei problemi che mi sono trovato ad affrontare maggiormente è proprio il passaggio di dati e la comunicazione tra componenti. Per poter mostrare i dati della zona cliccata, era necessario far comunicare il componente Mappa con il componente Sidebar. Uno dei metodi più comuni è l'utilizzo delle props. In React, "props" è una contrazione di "properties" ed è un meccanismo per passare dati da un componente ad un altro.

Un'altra possibilità è quella di utilizzare la libreria Redux. Per esercizio accademico ho scelto di usare entrambi nel corso del progetto.

- **Analisi Multivariata:** Dentro la sidebar è presente anche la parte analitica di questo progetto. Ciò ha comportato un enorme lavoro sia in termini di studio sia in termini di implementazione.

4.8 Analisi Multivariata

L'Analisi Multivariata è una metodologia statistica avanzata che si concentra sull'esplorazione, l'interpretazione e la comprensione di relazioni complesse tra più variabili simultaneamente. Queste variabili possono essere correlate tra loro o influenzate da diverse condizioni o fattori. L'obiettivo principale dell'analisi multivariata è esplorare la struttura e le interazioni all'interno di un insieme di variabili complesse. Questo può aiutare a individuare schemi, tendenze o associazioni nascoste nei dati. Ci sono diversi metodi per effettuare un'analisi multivariata, per questo progetto ho scelto di utilizzare la matrice di correlazione, la regressione lineare e la regressione multipla.

4.8.1 Matrice di Correlazione

La matrice di correlazione è un metodo statistico utilizzato per esaminare le relazioni lineari tra le variabili. In questo contesto, è stata applicata per valutare la correlazione tra i livelli di inquinanti e la temperatura. Nella matrice di correlazione, ogni cella rappresenta un coefficiente di Pearson.

4.8.2 Coefficiente di Pearson

Il coefficiente di Pearson è una misura statistica utilizzata per valutare la forza della relazione lineare tra due variabili quantitative. Viene calcolato come il rapporto tra la covarianza delle due variabili (X e Y) e il prodotto delle loro deviazioni standard:

$$r = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2} \sqrt{\sum_{i=1}^n (Y_i - \bar{Y})^2}}$$

dove:

- r = coefficiente di correlazione.
- x_i = valori della variabile x in un campione.
- \bar{x} = media dei valori della variabile x .
- y_i = valori della variabile y in un campione.
- \bar{y} = media dei valori della variabile y .

Ponendo in correlazione gli inquinanti e la temperatura, si ottiene una tabella simmetrica in cui ogni cella rappresenta il grado di associazione lineare tra due variabili.

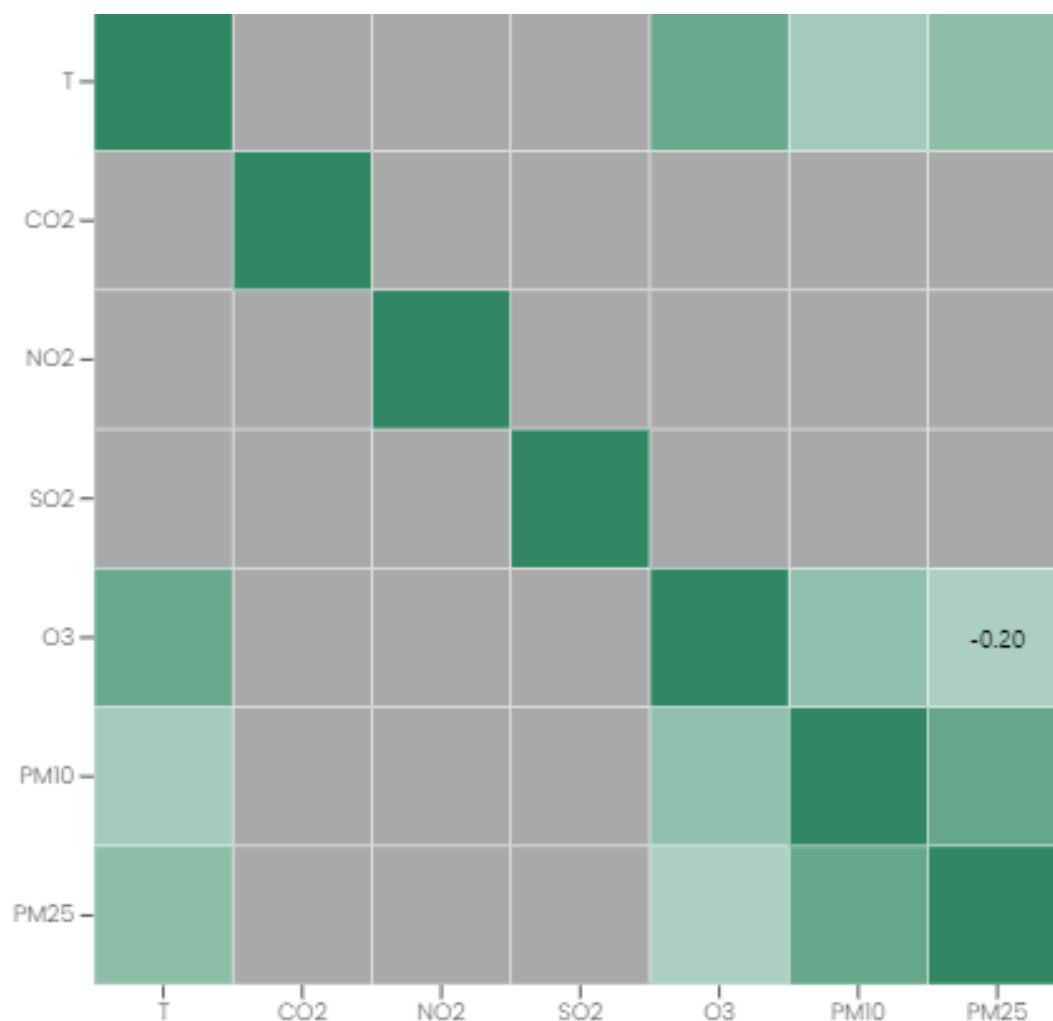


Figura 4.5. Matrice di Correlazione

I coefficienti di correlazione possono variare da -1 ad 1:

- Più un coefficiente si avvicina a -1 più indica una correlazione negativa perfetta, ovvero all'aumentare di uno, l'altro diminuisce.
- Più un coefficiente si avvicina a 1 più indica una correlazione perfetta, ovvero all'aumentare di uno aumenta anche l'altro.
- Più un coefficiente si avvicina a 0 più indica che non esiste correlazione tra i due elementi.

I coefficienti di correlazione compaiono sulla casella in cui il cursore è presente. Inoltre, per facilitarne la comprensione, la matrice è colorata con una scala cromatica in cui maggiore è la forza di correlazione più scuro sarà il colore. Il grigio indica che non ci sono state abbastanza rilevazioni per calcolare la correlazione tra i due elementi.

```
//Calcolare il coeff. di corr. di Pearson tra due array
const calculateCorrelationPearson = (xData, yData) => {
  const n = xData.length;
  const xSum = xData.reduce((acc, val) => acc + val, 0);
  const ySum = yData.reduce((acc, val) => acc + val, 0);
  const xySum = xData.reduce(
    (acc, val, index) => acc + val * yData[index], 0);
  const xSquaredSum = xData.reduce((acc, val) =>
    acc + val * val, 0);
  const ySquaredSum = yData.reduce((acc, val) =>
    acc + val * val, 0);
  const numerator = n * xySum - xSum * ySum;
  const denX = n * xSquaredSum - xSum * xSum;
  const denY = n * ySquaredSum - ySum * ySum;
  const correlation = numerator / Math.sqrt(denX * denY);
  return correlation;
};
```

Figura 4.6. Calcolo Correlazione di Pearson

4.8.3 Regressione Lineare

La regressione lineare è una tecnica statistica che viene utilizzata per analizzare la relazione tra una variabile dipendente e una o più variabili indipendenti. La regressione lineare produce un modello matematico che rappresenta la relazione lineare tra le variabili coinvolte. Consente inoltre, di stimare i coefficienti che quantificano l'effetto di ciascuna variabile indipendente sulla variabile dipendente. Viene calcolata con la seguente formula:

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

Dove:

- Y è la variabile dipendente.
- X è la variabile indipendente.
- β_0 è l'intercetta, che rappresenta il valore di Y quando X è uguale a zero.
- β_1 è il coefficiente di regressione, che rappresenta la pendenza della retta di regressione.
- ε rappresenta l'errore residuo, che rappresenta la differenza tra i valori osservati e i valori predetti.

Per calcolare i coefficienti β_0 e β_1 , si utilizzano le seguenti formule:

$$\beta_1 = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sum_{i=1}^n (X_i - \bar{X})^2}$$

$$\beta_0 = \bar{Y} - \beta_1 \bar{X}$$

Dove:

- n è il numero di osservazioni nel dataset.
- X_i e Y_i sono i valori delle variabili X e Y per l' i -esima osservazione.
- \bar{X} e \bar{Y} sono le medie dei valori di X e Y rispettivamente.

Nel contesto di questo progetto, la regressione lineare è stata utilizzata per esaminare la forza e la direzione della correlazione tra inquinanti e temperatura.

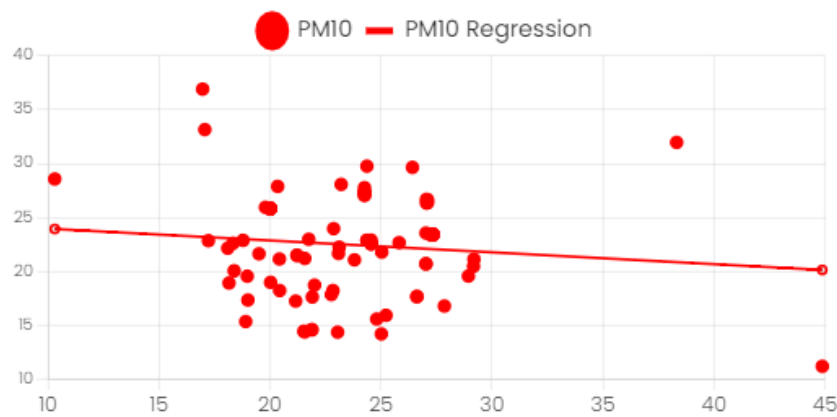


Figura 4.7. Regressione Lineare 1/6

A differenza della matrice di correlazione, per questa funzionalità è stata utilizzata una libreria matematica, chiamata `regression.js`, che consente lo sviluppo di una regressione lineare.

```
// Calcola la regressione lineare per l'inquinante corrente
const result = regression.linear(
  | temperatures.map((temp, index) => [temp, pollutantValues[index]])
);

const slope = result.equation[0];
const intercept = result.equation[1];

// Genera i punti per la linea di regressione
const regressionLine = [
  {
    x: Math.min(...temperatures),
    y: slope * Math.min(...temperatures) + intercept,
  },
  {
    x: Math.max(...temperatures),
    y: slope * Math.max(...temperatures) + intercept,
  },
];
```

Figura 4.8. Utilizzo della Libreria Regression

4.8.4 Regressione Multipla

La regressione multipla è un'estensione della regressione lineare che permette di analizzare la relazione tra una variabile dipendente e più di una variabile indipendente contemporaneamente.

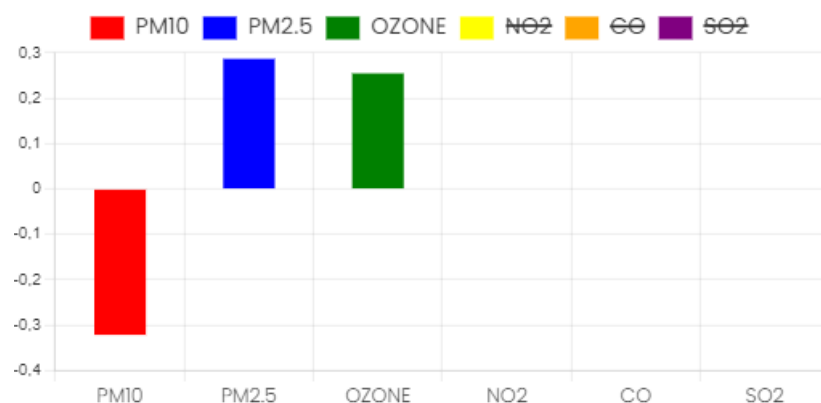


Figura 4.9. Regressione Multipla

L'obiettivo di questo tipo di analisi è quello di misurare l'effetto combinato di diverse variabili indipendenti sulla variabile dipendente, tenendo conto delle loro interazioni.

Di seguito viene mostrata la formula per il suo calcolo:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n + \varepsilon$$

Dove:

- Y è la variabile dipendente.
- X_1, X_2, \dots, X_n sono le variabili indipendenti.
- β_0 è l'intercetta, che rappresenta il valore di Y quando tutte le variabili indipendenti sono uguali a zero.
- $\beta_1, \beta_2, \dots, \beta_n$ sono i coefficienti di regressione, che rappresentano le pendenze associate alle rispettive variabili indipendenti.
- ε rappresenta l'errore residuo.

Per calcolare i coefficienti di regressione $\beta_0, \beta_1, \beta_2, \dots, \beta_n$, si può utilizzare il metodo dei minimi quadrati:

$$\beta_0 = \bar{Y} - \beta_1 \bar{X}_1 - \beta_2 \bar{X}_2 - \dots - \beta_n \bar{X}_n$$

$$\beta_1 = \frac{\sum_{i=1}^N (X_{1i} - \bar{X}_1)(Y_i - \bar{Y})}{\sum_{i=1}^N (X_{1i} - \bar{X}_1)^2}$$

$$\beta_2 = \frac{\sum_{i=1}^N (X_{2i} - \bar{X}_2)(Y_i - \bar{Y})}{\sum_{i=1}^N (X_{2i} - \bar{X}_2)^2}$$

$$\vdots$$

$$\beta_n = \frac{\sum_{i=1}^N (X_{ni} - \bar{X}_n)(Y_i - \bar{Y})}{\sum_{i=1}^N (X_{ni} - \bar{X}_n)^2}$$

Dove:

- N è il numero di osservazioni nel dataset.
- $X_{1i}, X_{2i}, \dots, X_{ni}$ sono i valori delle variabili indipendenti per l' i -esima osservazione.
- $\bar{X}_1, \bar{X}_2, \dots, \bar{X}_n$ sono le medie dei valori delle rispettive variabili indipendenti.
- \bar{Y} è la media dei valori della variabile dipendente.

Questa regressione è stata particolarmente difficile da implementare, poichè il grafico che la rappresenta è interattivo, ciò permette all'utente di selezionare il gruppo di variabili indipendenti su cui calcolare la regressione multipla. Quando un inquinante viene attivato o disattivato, la regressione viene ricalcolata sul gruppo di inquinanti rimanente.

```
// Calcola i coefficienti di regressione usando la formula OLS
const Xtranspose = math.transpose(X);
const XtX = math.multiply(Xtranspose, X);
const XtXinverse = math.inv(XtX);
const XtY = math.multiply(Xtranspose, Y);
const coefficients = math.multiply(XtXinverse, XtY);

// Crea un array di oggetti contenenti il coefficiente e l'inquinante corrispondente
const results = coefficients.map((coefficient, index) => ({
  coefficient: coefficient,
  pollutant: pollutants[index],
}));
return results;
```

Figura 4.10. Utilizzo Math.js per calcolo Regression Multipla

Per quest'ultima analisi è stato utilizzato un approccio ibrido. I calcoli sono stati eseguiti da una libreria matematica chiamata Math.js ed i risultati ottenuti sono stati impiegati per calcolare il valore finale della regressione.

4.9 Control Panel

Il pannello di controllo è un componente globale sempre presente sullo schermo. Si occupa di fornire all'utente svariate funzionalità con il quale può modificare la mappa, aggiungendo o cambiando visualizzazioni. Lo sviluppo di questo componente ha richiesto l'uso di stati globali, per cui è stato necessario integrare la libreria Redux.

4.9.1 Redux

Redux è una libreria open-source di gestione dello stato che viene spesso utilizzata in applicazioni Web in particolare con React. La principale caratteristica di Redux è fornire un approccio prevedibile e centralizzato per gestire lo stato dell'applicazione. Redux promuove il concetto di "architettura a flusso di dati unidirezionale", ovvero il flusso dei dati nell'applicazione segue una direzione predefinita. Per utilizzare Redux è necessaria la creazione di diversi punti:

- **Store:** Il cuore di Redux, rappresenta lo stato globale dell'applicazione. Lo store contiene tutti i dati che l'applicazione utilizza e fornisce metodi per accedervi e modificarli. Ad esempio in questo progetto lo store comprendeva le variabili per tenere traccia della night/colorblind mode, del giorno visualizzato, etc...
- **Actions:** Le azioni sono oggetti JavaScript che ti permettono di modificare lo stato.
- **Reducer:** I reducer sono funzioni pure che specificano come cambierà lo stato dell'applicazione in risposta a un'azione. Un reducer prende uno stato precedente con un'azione e restituisce un nuovo stato.

In Redux quindi le azioni vengono inviate allo store, i reducer elaborano le azioni e aggiornano lo stato. Questo flusso di dati unidirezionale rende più facile comprendere come cambia lo stato dell'applicazione.


```
//stati globali
const initialState = {
  sliderValue: 0,
  layerToShow: "AQI",
  currentLayer: "country",
  wind: false,
  windHeatmap: false,
  map3d: false,
  sidebar: false,
  nightMode: true,
  colorBlindMode: false,
};

//reducer
const sidebarReducer = (state = initialState.sidebar, action) => {
  if (action.type === "SET_SIDE BAR") {
    return action.payload;
  }
  return state;
};

//azione
export const setSidebar = (value) => {
  return {
    type: "SET_SIDE BAR",
    payload: value,
  };
};
```

Figura 4.11. Esempio di stati, riduttori ed azioni di Redux

4.9.2 Aggiornamento Dati

Funzionalità che si occupa di aggiornare i dati del giorno attuale con le ultime misurazioni salvando l'aggiornamento nel database. Per farlo, viene effettuata una chiamata alle API, quando i dati cambiano automaticamente viene aggiornata tutta l'applicazione. Con delle API che consentono l'aggiornamento real-time dei dati, questa funzione deve essere eliminata.

4.9.3 Layers Inquinanti

Funzionalità che si occupa di cambiare visualizzazione della mappa, colorandola in base all'inquinante scelto. Il suo sviluppo è basato sullo stato globale del layer da visualizzare. A seconda di questo stato, la mappa mostra una colorazione diversa. Quando l'utente sceglie un inquinante da visualizzare, cambiando questo stato globale, cambia anche la visualizzazione del resto dell'applicazione.

4.9.4 Layer Vento

Funzionalità che si occupa di mostrare un layer animato che rappresenta il vento. Questo layer consente di visualizzare dati relativi al vento in modo dinamico. Questa funzionalità è stata inserita perchè uno dei fattori che influenza la qualità dell'aria è proprio il vento. Per creare l'effetto vento si generano casualmente una serie di particelle virtuali che rappresentano il flusso del vento. Queste particelle sono posizionate su una griglia tridimensionale che copre l'area geografica di interesse. Una volta generate, le particelle vengono animate in base ai dati del vento. Ogni particella segue un percorso determinato dalla direzione e dall'intensità del vento in quel punto della griglia. Questa animazione è ciò che crea il movimento visivo del vento sulla mappa. Per ottenere prestazioni ottimali e un aspetto fluido, il rendering delle particelle viene effettuato utilizzando il motore grafico OpenGL. Questo permette di sfruttare l'accelerazione hardware e di gestire un gran numero di particelle in modo efficiente. I dati relativi al vento, come direzione e intensità, sono solitamente ottenuti da fonti esterne, come servizi meteorologici o modelli di previsione.¹

4.9.5 Layer Vento Scalare

Funzionalità che si occupa di applicare a schermo una immagine semitrasparente che rappresenta il vento scalare.

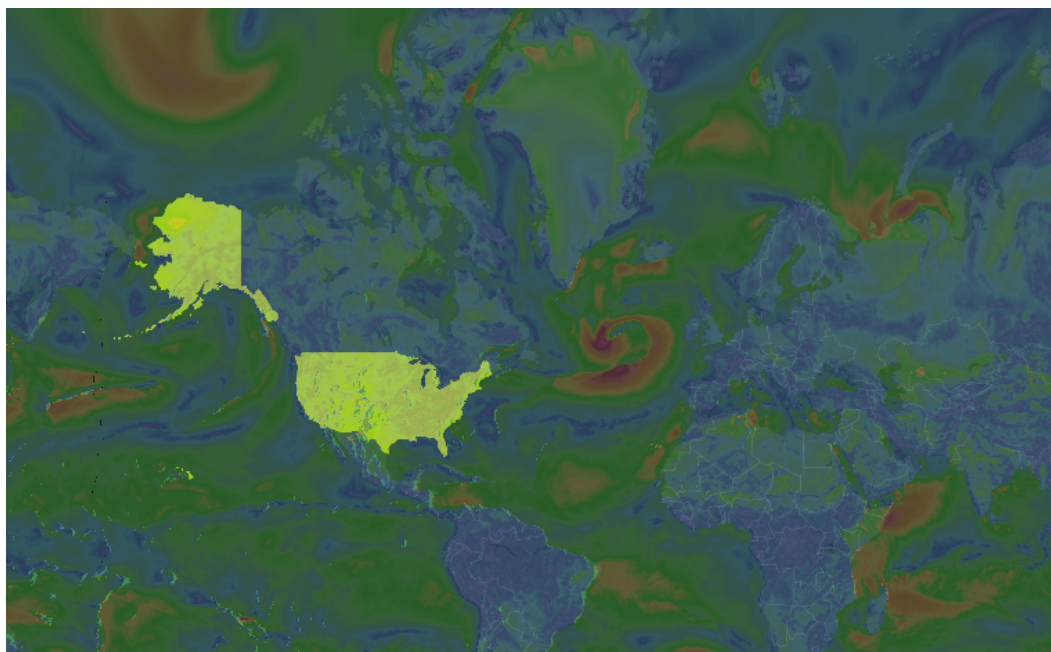


Figura 4.12. Funzionalità di Vento scalare su mappa 2D

4.9.6 Visualizzazione Globo

Funzionalità che permette la scelta, da parte dell'utente, di visualizzare la mappa 2D o 3D. Per farlo si utilizza uno stato globale e durante la creazione della mappa si controlla questo stato per impostare il tipo di visualizzazione richiesto.

¹I dati del vento in questo progetto sono sample statici

4.9.7 Locazione Stazioni

Funzionalità che si occupa di mostrare la locazione delle stazioni coinvolte nel rilevamento del giorno visualizzato. Le visualizzazioni sono di due tipologie: heatmap oppure dots.

I dati delle misurazioni sono dati delle singole stazioni di rilevamento, ciò significa che ogni misurazione contiene le coordinate della stazione stessa. Per sviluppare questa funzionalità è stata creata una lista in cui vengono aggiunte tutte le coordinate della giornata visualizzata. Questa lista viene salvata insieme al resto dei dati nel database. In tal modo, l'effetto dello sliding temporale, può avere effetto anche per questa funzionalità.

4.9.8 Slider Temporale

Funzionalità che permette all'utente di selezionare il giorno visualizzato. Il suo sviluppo ha richiesto l'implementazione di un database per immagazzinare le varie "giornate", consentendo di registrare fino a 7 giorni passati (estendibile all'infinito). Quando l'utente modifica il giorno visualizzato, lo stato globale cambia, renderizzando nuovamente la mappa, mostrando quindi i colori aggiornati relativi alla giornata selezionata. Anche i dati presenti nella sidebar ed in generale nei grafici vengono aggiornati. L'obiettivo è quello di consentire all'utente di visualizzare la variazione della qualità dell'aria nel tempo.

4.10 Database

Per questo progetto non era necessario un database relazionale e volendo provare una nuova tecnologia la scelta è ricaduta su MongoDB. L'utilizzo di questo database è risultato piuttosto semplice. Nel contesto dell'applicazione viene utilizzato per la funzionalità di sliding temporale e per avere una massiccia quantità di dati da passare in input all'analisi multivariata, così da avere campioni di dati ogni giorno più grandi.

Lo storico dei dati attualmente tiene traccia di 7 giorni, ognuno nei quali è salvato la geosource che poi viene passata alla mappa di mapbox.

Nel back-end è impostato il collegamento con il database in modo da poter soddisfare le richieste di scrittura e lettura in MongoDB.

Il database inoltre ha dei triggers che si occupano di impostare un booleano a false ogni mezzanotte. Ad ogni avvio della Web app, viene controllato questo booleano e se risulta false, viene effettuato il fetch dei dati tramite le chiamate API. Al termine di questo processo, il booleano viene impostato a true in modo che il ciclo sia automatico. In tal modo, l'applicazione riconosce quando è passato un giorno ed è quindi necessario ottenere nuovi dati. Anche questa funzionalità, con l'utilizzo di API a pagamento e quindi la possibilità di avere dati real-time, verrà eliminata.

4.11 Extra

Per completezza sono state sviluppate funzionalità che vanno ad aiutare l'utente:

4.11.1 Info Modale

Nella parte destra della topbar è presente un bottone informativo che apre una finestra modale al centro dello schermo con dentro le informazioni generali relative all'applicazione Web.



Figura 4.13. Finestra Modale Informazioni per Utenti

4.11.2 Github

Sempre nella topbar, vicino al bottone informativo, è presente un pulsante che porta alla repo di github nel caso l'utente volesse consultare la modalità di sviluppo.

4.11.3 Titoli Modali Sidebar

Nella sidebar ogni grafico ha un titolo, se il cursore è sopra uno di questi titoli, si apre una finestra che spiega all'utente come interpretare e capire ciò che sta guardando.

Capitolo 5

Test e Feedback

Terminato lo sviluppo, l'applicazione è passata alla fase di testing, nel quale sono state testate le funzionalità, anche a gruppi, per verificare l'interazione tra esse (ad esempio modalità notte e modalità daltonici attive contemporaneamente). Appurato il corretto funzionamento, è stato chiesto a diverse persone di utilizzare l'applicazione in modo completo, ottenendo i seguenti risultati:

- Alcune persone hanno riscontrato un eccessivo numero di elementi sullo schermo, questo perchè inizialmente lo slider temporale era separato dal control panel. A seguito di questi suggerimenti, sono state unite le due funzionalità creando un design più pulito ed intuitivo, senza sacrificare funzionalità.
- Alcune persone hanno trovato difficoltà nel comprendere l'obiettivo della Web app. Per risolvere questo problema è stato creato il pulsante di info modale nel quale vengono fornite tutte le informazioni necessarie riguardanti Air Qaulity Map ed i suoi obiettivi.
- Alcune persone hanno suggerito di far visualizzare l'AQI di una zona nei suoi 7 giorni passati in modo più rapido, per soddisfare questa esigenza è stato utilizzato un grafico ad area.

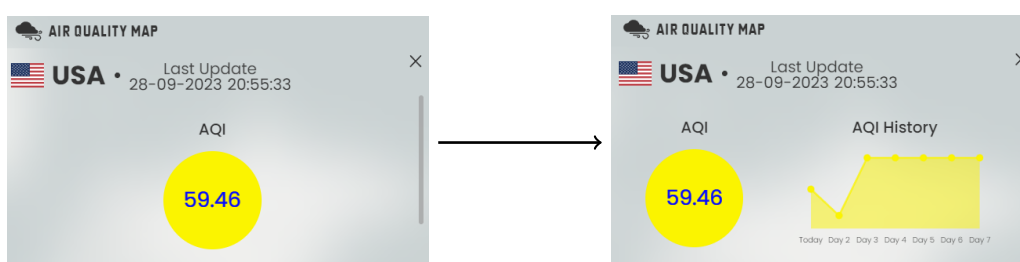


Figura 5.1. Sidebar prima e dopo il feedback

Questa fase di feedback ha consentito il miglioramento e l'aggiunta di funzionalità, ma anche la verifica del corretto funzionamento dell'applicazione e del suo design.

Una volta terminata questa fase, sono stati eseguiti ulteriori stress test, i quali hanno dato risultati soddisfacenti.

Capitolo 6

Conclusioni

L'obiettivo di questo progetto era sviluppare una Web app per il monitoraggio ambientale, creando una mappa quanto più semplice, immediata ed interattiva possibile. Anche il design e la user experience erano obiettivi importanti poiché agevolano la navigazione dell'utente aiutando quindi nello scopo finale di sensibilizzare su questo tema.

La sfida più ardua è stata senza dubbio lo studio per lo sviluppo della parte analitica, ma anche la sua rappresentazione. Ottenere dei dati e riuscire a creare una visualizzazione semplice ed intuitiva era fondamentale.

Grazie a questa esperienza ho potuto apprendere le basi dello sviluppo Web ed approfondire lo studio di un tema complesso ed attuale come quello dell'inquinamento.

6.0.1 Sviluppi Futuri

Gli strumenti utilizzati si sono rivelati adatti allo sviluppo di questa applicazione Web, nonostante ciò l'applicazione ha grandi margini di miglioramento, sia in termini di funzionalità sia in termini di performance. Le funzionalità che si possono aggiungere sono molteplici, prima tra tutte per livello di importanza è l'aggiornamento in real time dei dati. Sotto l'aspetto dell'ottimizzazione è possibile snellire la quantità di dati presenti nel database rendendo così più efficienti e veloci le chiamate sia in scrittura che in lettura. Attraverso uno studio più approfondito delle tecnologie utilizzate, anche la parte implementativa può essere sicuramente migliorata.

6.0.2 Considerazioni Finali

Finito questo percorso cercherò di portare l'applicazione ad un livello professionale, l'obiettivo è quello di pubblicarla ed utilizzarla per il suo scopo originale.

Spero quindi che questa applicazione possa risultare uno strumento utile per la comunità, aiutando ad informare e sensibilizzare sul tema della qualità dell'aria.

Bibliografia

- [1] OMS, Air Pollution Threat to Health,
https://www.who.int/health-topics/air-pollution#tab=tab_1
- [2] Roya Kelishadi, Parinaz Poursafa , Air pollution and non-respiratory health hazards for children
<https://pubmed.ncbi.nlm.nih.gov/22371790/>
- [3] LSI-Lastem, Weather observations Station,
<https://www.lsi-lastem.com/applications/weather/weather-observations/>
- [4] AirNow, Air Quality Index Basics
<https://www.airnow.gov/aqi/aqi-basics/>
- [5] JavaScript, " MDN Web Docs, Mozilla Developer Network
<https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- [6] React Dev Blog, 2023, Introducing to React
<https://react.dev/blog/2023/03/16/introducing-react-dev>
- [7] "MongoDB." MongoDB Documentation
<https://www.mongodb.com/docs/>
- [8] "Node.js." Node.js Documentation
<https://nodejs.org/en/docs>
- [9] "Express.js." Express.js
<https://expressjs.com/>
- [10] Redux, Getting Sartetd
<https://redux.js.org/>
- [11] IBM Topics, Cos'è un API?
<https://www.ibm.com/it-it/topics/api>
- [12] "Mapbox GL JS." Mapbox Documentation
<https://www.mapbox.com/mapbox-gljs>
- [13] "Chart.js." Chart.js Documentation
<https://www.chartjs.org/docs/latest/>
- [14] MDN Web Docs, Await
<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/await>

-
- [15] MDN Web Docs, Promise
https://developer.mozilla.org/enUS/docs/Web/JavaScript/Reference/Global_Objects/Promise
 - [16] Math.js Library, Math.js
<https://mathjs.org/>
 - [17] Regression.js Library, Regression Package
<https://www.npmjs.com/package/regression>