

Geo-referenced Photo Management System

*Anno Accademico 2022-2023

Giuseppe Pio Salcuni

Matr: 1100090

Università di Bologna

Alessandro Tocco

Matr: 1097880

Università di Bologna

Manuel Placella

Matr: 1099701

Università di Bologna

Abstract—Questo progetto si concentra sull'elaborazione di una sofisticata piattaforma software dedicata alla gestione di file multimediali georeferenziati, in particolare immagini fotografiche. L'obiettivo primario del sistema è di offrire agli utenti la possibilità di catturare nuovi contenuti multimediali attraverso dispositivi mobili, archiviarli in un robusto database e fruirne agevolmente tramite un'interfaccia web intuitiva. Le funzioni principali di questa piattaforma comprendono l'analisi avanzata dei dati spaziali per la visualizzazione delle informazioni geografiche e la creazione di cluster di marcatori geografici correlati alle immagini.

Index Terms—Location awareness, Container, Geo-spatial Querying, Unsupervised clustering

I. INTRODUZIONE

Nel contesto dell'era digitale in costante evoluzione e del crescente rilievo dei dati georeferenziati, ci siamo posti l'obiettivo di sviluppare un progetto ambizioso: la creazione di una sofisticata piattaforma software dedicata alla gestione di contenuti multimediali. Questa piattaforma è stata concepita per soddisfare una serie di esigenze fondamentali per gli utenti, tra cui la possibilità di acquisire nuovi file tramite dispositivi mobili, garantirne una conservazione sicura all'interno di un database, e consentire un accesso intuitivo e flessibile a tali dati attraverso un'interfaccia web all'avanguardia.

II. PROGETTAZIONE

Il nostro progetto è suddiviso in tre componenti fondamentali, ciascuna delle quali ha un ruolo vitale nell'attuazione degli obiettivi stabiliti. Qui di seguito, esamineremo la pianificazione di ciascuna parte del progetto.

A. Applicazione Mobile

L'applicazione deve consentire agli utenti di poter creare o selezionare collezioni esistenti, catturare foto, acquisire automaticamente le coordinate GPS, inviare foto insieme ai nomi delle collezioni e alle relative posizioni al backend e recuperare un determinato numero di foto più vicine alla loro posizione attuale, dove il numero è un parametro configurabile, e abilitare altri utenti alla condivisione della propria galleria.

B. Back-end

Il back-end svolge un ruolo centrale nella gestione e persistenza dei dati. È responsabile della memorizzazione dei file multimediali e dei relativi geo-tag, i quali vengono acquisiti dal client mobile e archiviati in un database dedicato. È

responsabile inoltre della gestione degli utenti e delle varie condivisioni tra loro. Il back-end implementa funzionalità di interrogazione geospaziale, consentendo agli utenti di recuperare informazioni basate sulla posizione mediante query spaziali al fine di trovare foto in posizioni specifiche sulla base di un sistema di filtraggio illustrato nei capitoli successivi.

C. Front-end

Il front-end, rappresentato dalla dashboard web, offre agli utenti un'interfaccia intuitiva per interagire con i dati georiferiti e i file multimediali. Le funzionalità principali comprendono:

- Gestione del Login e della Registrazione: Il front-end offre un sistema di autenticazione per gli utenti, consentendo loro di effettuare il login e la registrazione.
- Condivisione dei Contenuti con Altri Utenti: Gli utenti possono condividere le immagini con altri utenti del sistema. Questa funzionalità promuove la collaborazione e la condivisione all'interno della piattaforma.
- Caricamento di File Multimediali: Gli utenti possono aggiungere file o cartelle contenenti immagini e specificare il geo-tag spaziale per ciascuna foto o per l'intera cartella.
- Gestione di GEOJSON: Il front-end supporta anche l'upload di file GEOJSON contenenti informazioni geografiche, come ad esempio i confini delle regioni geografiche. Questi dati sono utilizzati per filtrare e visualizzare le foto in base a criteri geografici specifici.
- Visualizzazione dei Marker delle Foto: Gli utenti possono visualizzare i marker delle foto sulla mappa e applicare filtri basati sui confini geografici definiti nei file GEOJSON precedentemente caricati. È inoltre possibile colorare le aree sulla mappa in base al numero di foto contenute in ciascuna regione.
- Clustering e Heatmap: Il sistema offre funzionalità avanzate di clustering dei geo-tag utilizzando l'algoritmo K-Means e l'algoritmo DBSCAN. Gli utenti hanno la flessibilità di scegliere tra una configurazione automatica, in cui il sistema determina il numero ottimale di cluster utilizzando il metodo "elbow," e una configurazione manuale, dove possono specificare manualmente il numero di cluster desiderato.

I dati possono essere visualizzati attraverso una heatmap per una migliore comprensione della distribuzione spaziale.

- Visualizzazione della Galleria: Gli utenti possono selezionare e visualizzare le foto in diverse modalità, tra cui la ricerca per nome della collezione. Questo offre una visione personalizzata dei dati multimediali in base alle preferenze dell'utente.

D. Feature aggiuntive

Nel contesto del nostro progetto, abbiamo introdotto nuove funzionalità per arricchire l'esperienza dell'utente sia nell'applicazione web che nell'app mobile.

1) **Applicazione Web** : Nell'ambito dell'applicazione web, abbiamo implementato una serie di funzionalità:

- Gestione degli Utenti: Abbiamo introdotto un sistema di login e registrazione per garantire un accesso sicuro alle funzionalità del sistema. Gli utenti possono creare un account e accedere in modo personalizzato.
- Gestione delle Collezioni: Ogni immagine deve essere assegnata a una specifica collezione, permettendo agli utenti di organizzare le proprie foto in base a categorie o preferenze. Questa funzionalità contribuisce all'organizzazione dei contenuti.
- Algoritmo di Clustering DBSCAN: Per una migliore analisi dei dati basata sulla densità, abbiamo integrato l'algoritmo DBSCAN. Questo algoritmo è in grado, come vedremo anche nei capitoli successivi, di determinare autonomamente il numero di cluster in base a due parametri: l'epsilon, rappresentante una soglia di distanza tra i punti, e il minPoints, una soglia sul numero di punti vicini a ciascun punto.
- Condivisione dei Contenuti tra Utenti: Gli utenti possono condividere le proprie foto e contenuti con altri utenti iscritti al sistema. Questa funzionalità promuove la collaborazione e la condivisione di contenuti tra gli utenti, contribuendo alla socializzazione della piattaforma.

2) **Applicazione mobile**: Nell'applicazione mobile, abbiamo implementato una serie di funzionalità per rendere l'esperienza dell'utente più ricca e accessibile:

- Login e Registrazione dell'Utente: Abbiamo introdotto la possibilità di effettuare il login e la registrazione direttamente dall'app mobile. Gli utenti possono accedere in modo sicuro alle funzionalità dell'app.
- Visualizzazione delle Foto su Mappa: Gli utenti possono visualizzare le proprie foto su una mappa interattiva direttamente dall'app. Questa funzionalità offre una visualizzazione geografica dei contenuti multimediali, consentendo agli utenti di esplorare le proprie foto in base alla posizione geografica.
- Condivisione dei Contenuti tramite App: Gli utenti possono condividere le proprie foto con altri utenti direttamente dall'app mobile. Questa funzionalità semplifica il processo di condivisione di contenuti multimediali, favorendo la collaborazione tra gli utenti.
- Visualizzazione delle Foto di Altri Utenti: Gli utenti possono anche visualizzare le immagini di altri utenti che hanno abilitato la condivisione dei propri contenuti. Ciò

promuove l'interazione e la visualizzazione di contenuti condivisi, consentendo agli utenti di scoprire foto di altri membri della comunità.

Riteniamo che queste nuove funzionalità migliorino significativamente l'esperienza dell'utente sia sul lato dell'applicazione web che su quello dell'app mobile, contribuendo, secondo la nostra opinione, a rendere la piattaforma più completa e interattiva.

III. IMPLEMENTAZIONE

In questo capitolo verranno descritte le tecnologie utilizzate per la realizzazione del sistema e le scelte fatte per la sua implementazione. Utilizzeremo immagini e diagrammi per illustrare le strutture e i dettagli implementativi, offrendo una panoramica chiara e completa del nostro sistema finale.

A. Applicazione Mobile

1) *Registrazione e Login*: : Quando si avvia l'applicazione mobile, l'utente sarà diretto alla schermata di login. Se l'utente è già registrato, potrà accedere immediatamente. Altrimenti, se è un nuovo utente, avrà l'opzione di andare alla schermata di registrazione, dove potrà inserire le proprie informazioni.

Entrambe queste funzioni si collegano al nostro server backend, che è stato sviluppato in background. Questo server è responsabile di salvare i dati degli utenti nel sistema e di garantire che i nomi utente siano univoci. In altre parole, non è possibile avere più utenti con le stesse credenziali.

È importante notare che abbiamo scelto di sviluppare la nostra applicazione utilizzando il linguaggio Flutter Dart. Questa scelta ci consente di rendere l'applicazione accessibile su dispositivi principali come IOS e Android. Abbiamo condotto il testing sull'emulatore fornito da Android Studio per garantire la funzionalità del nostro prodotto.

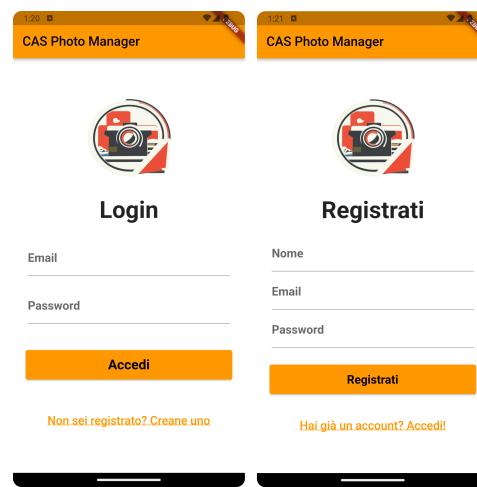


Fig. 1. Login e Registrazione

2) *Dashboard e Gallery Page*: : Dopo aver effettuato l'accesso, l'utente verrà reindirizzato alla sua galleria personale. La schermata principale consente di sfogliare le collezioni e visualizzare le foto in ciascuna collezione. Le collezioni caricate dall'utente saranno di colore giallo, mentre le collezioni

condivise saranno di colore celeste.

Nella parte inferiore, troverà una dashboard per la navigazione tra le varie pagine, nonché un pulsante per scattare nuove foto, che spiegheremo più avanti. Questa dashboard sarà sempre presente in tutte le pagine dell'applicazione.



Fig. 2. Gallery

Le principali funzionalità includono:

- Creazione di nuove collezioni e aggiunta di foto
- Rimozione di foto da una collezione tenendo premuto su di esse
- Eliminazione completa di una collezione tramite un'azione di pressione prolungata
- Condivisione della galleria con altri utenti tramite l'apposito pulsante posto in alto a sinistra. In questo modo, le collezioni e le foto saranno visibili nella galleria dell'utente condiviso.

È essenziale notare che l'utente deve concedere l'autorizzazione per accedere alla galleria del proprio smartphone al fine di consentire l'aggiunta di foto alle collezioni, poiché questo è fondamentale per preservare la privacy. Senza tali autorizzazioni, non sarà possibile caricare nuove foto.

3) Upload Page: : La sezione di caricamento consente all'utente di:

- Aggiungere foto a collezioni esistenti
- Creare nuove collezioni e caricare foto al loro interno.

Le foto e le collezioni caricate saranno disponibili anche per gli utenti con cui è condivisa la galleria.

Anche in questo caso, è essenziale che l'utente conceda le autorizzazioni per accedere alla galleria del proprio smartphone al fine di consentire il caricamento delle foto



Fig. 3. Upload Page

4) Camera Preview Page: : L'utente ha la possibilità di catturare una foto direttamente dalla fotocamera del suo dispositivo, oltre a caricare immagini dalla galleria. Un pulsante dedicato consente di scattare una foto tramite la fotocamera, e successivamente, questa foto verrà visualizzata nella pagina successiva, insieme alla registrazione della posizione in cui è stata scattata. L'utente può decidere di salvarla nella collezione desiderata se la foto gli piace, altrimenti può optare per un nuovo scatto.

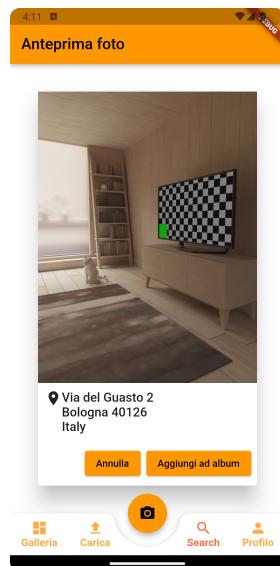


Fig. 4. Image Preview Page

Un aspetto fondamentale in questa funzionalità è la richiesta dei permessi per accedere alla fotocamera e alla posizione del dispositivo. Questo è necessario per garantire un funzionamento corretto e rispettare la privacy dell'utente.

durante la cattura e la gestione delle foto. Se l'utente non concede queste autorizzazioni, non sarà possibile salvare la fotografia scattata.

5) *Search Page*: : La pagina di ricerca presenta principalmente una mappa interattiva in cui l'utente può visualizzare i marker delle foto caricate nell'applicazione, riconoscibili dalle icone blu. Se l'utente fa clic su uno di questi marker, verrà visualizzata una miniatura dell'immagine associata e i dettagli sulla posizione. Un'icona gialla rappresenta la posizione corrente dell'utente sulla mappa.

Nella parte superiore della schermata è presente un bottomsheet che permette di accedere a un elenco di tutti i marker delle foto. Cliccando su uno di essi, l'utente verrà portato direttamente alla posizione del marker sulla mappa.

Le funzionalità principali in questa pagina includono:

- Recupero della posizione corrente tramite un pulsante nella parte inferiore destra
- Filtraggio delle foto in base alla posizione attuale
- . Quando attivata, verrà visualizzato un popup in cui l'utente può specificare il numero di foto da filtrare in base alla sua posizione corrente. I risultati saranno mostrati nella sezione iniziale della mappa, e facendo clic su uno di essi, verrà visualizzato il marker corrispondente sulla mappa.

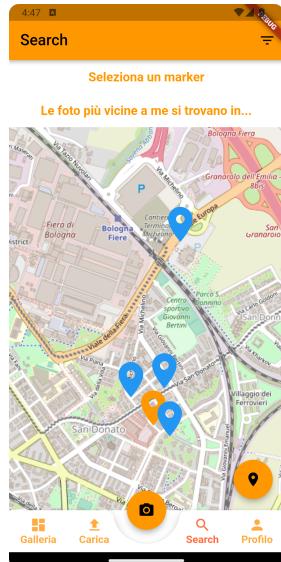


Fig. 5. Search Page

6) *Profile Page*: : L'ultima sezione dell'applicazione è la pagina del profilo, che mostra una foto del profilo, i dati dell'utente e il conteggio delle foto e delle collezioni caricate nell'applicazione.

L'utente ha anche la possibilità di terminare la sessione effettuando il logout. Facendo clic su un apposito pulsante, verrà reindirizzato alla pagina di login, chiudendo così la sua sessione.

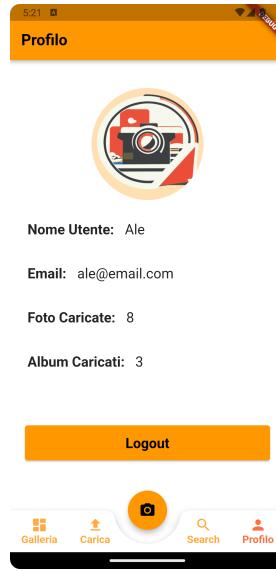


Fig. 6. Profile Page

B. Backend

Nel contesto del nostro progetto, il backend svolge un ruolo essenziale non solo nell'implementazione delle richieste dell'applicazione web, ma anche nella creazione del database Postgres con l'estensione PostGIS. Quest'ultima è un'estensione open source per il sistema di gestione di database PostgreSQL, progettata per supportare dati geospatiali. Aggiunge funzionalità di geolocalizzazione avanzate a PostgreSQL, consentendo la memorizzazione, la query e l'analisi di dati geografici all'interno del database. La struttura dettagliata del database è riportata nell'immagine seguente:

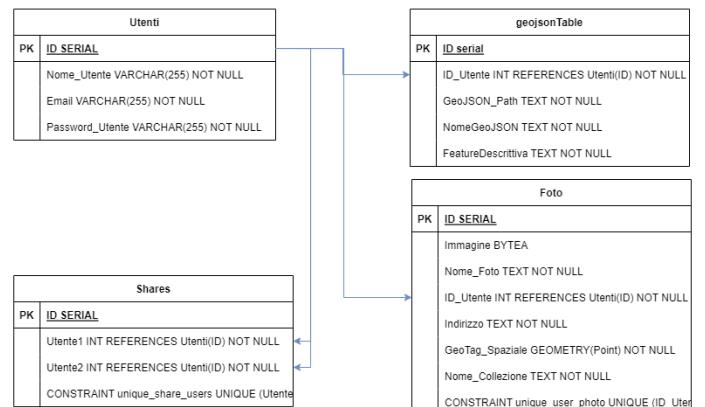


Fig. 7. Schema del DB

Per garantire scalabilità e velocità all'applicazione web, abbiamo scelto Node.js, un runtime JavaScript noto per la sua efficienza e velocità. Questa tecnologia ci permette di gestire le richieste dell'applicazione in modo efficiente e di fornire un'esperienza utente reattiva. Qui di seguito sono presentate le principali richieste fondamentali per il funzionamento dell'applicazione, implementate attraverso un'API

REST all'interno dei file `frontend.js` e `queryDB.js`. Questa divisione è stata realizzata per separare le query al database dalle richieste dell'applicazione. L'API REST rappresenta un'architettura standard che offre una struttura efficiente per l'interazione tra il frontend e il backend, garantendo una comunicazione fluida e scalabile.

- `/register` Richiesta POST che gestisce la registrazione degli utenti in un'applicazione web. I parametri Nome, Email e Password vengono estratti dal corpo della richiesta. Dopo l'inserimento dei dati nel database, la risposta al cliente varia in base all'esito della `INSERT` nel database.
- / Endpoint di login. Viene gestita una richiesta POST con parametri Email e Password. Il server verifica le credenziali nel database e risponde con uno stato 200 se il login è riuscito,
- `/dashboard/:userId/loadGeoJSON` Richiesta POST che gestisce il caricamento di un file GeoJSON sul server. Viene utilizzato il modulo multer per gestire il caricamento dei file. La richiesta accetta il file e informazioni aggiuntive, quali il nome del file e una descrizione. Prima del caricamento, viene verificato se il file è stato effettivamente caricato. Successivamente, i dati del file e l'ID dell'utente vengono estratti dalla richiesta. Il server interagisce con il database per archiviare il file GeoJSON.
- `/dashboard/:userId/downloadGeoJSON` Questo endpoint di tipo POST accetta un parametro `userId` per identificare l'utente e richiede il percorso del file GeoJSON tramite il body della richiesta. Dopo aver ottenuto il percorso del file, il server utilizza il modulo fs per leggere il contenuto del file in formato testuale (utf8). Questa richiesta consente quindi di scaricare dati geospatiali rappresentati nel formato GeoJSON dal server.
- `/dashboard/:userId/loadImage` Questa richiesta di tipo POST gestisce il caricamento di un'immagine in un'applicazione. L'immagine viene temporaneamente memorizzata in memoria utilizzando il modulo multer prima di essere archiviata nel database. Quindi l'id dell'utente che effettua il caricamento insieme ad altri diversi dettagli dell'immagine, come nome del file, indirizzo, coordinate geografiche e associazione a una collezione, vengono estratti dalla richiesta. L'immagine viene temporaneamente archiviata come un buffer in memoria e successivamente memorizzata nel database.
- `/dashboard/:userLog/:userView/photos` Richiesta GET utilizzata per recuperare dal database tutte le foto caricate da un utente specifico nell'applicazione. Identifica quindi l'utente di cui si desidera visualizzare le foto (`userView`). La query esegue una ricerca nel database per ottenere le foto caricate dall'utente specificato (`userView`) e restituisce i risultati.
- `/dashboard/:userLog/:userView/photosInGeoJSON` Richiesta POST che consente di

ottenere foto da un GeoJSON filtrate per area specifica. Accetta sia l'utente loggato (da cui prelevare il GeoJSON) che l'utente di cui si desidera visualizzare le foto (`userView`). Dal corpo della richiesta vengono estratti il GeoJSON (da cui poi si preleva il percorso del file GeoJSON), la feature che rappresenta l'area e infine la stessa area da cui si desiderano ottenere le foto. Dopo aver letto il contenuto del file GeoJSON e averlo analizzato in formato JSON, vengono eseguite una serie di operazioni:

- Verifica se il GeoJSON contiene features valide.
- Filtra le feature all'interno del GeoJSON in base all'area specificata o a tutte le feature con geometrie valide, a seconda del valore del parametro "area".
- Interagisce con il database per ottenere le foto associate a ciascuna feature geometrica (di tipo geometry).
- Elimina eventuali duplicati dalle foto raccolte.

Restituisce al termine una risposta contenente un messaggio di successo e l'elenco delle foto filtrate.

- `/dashboard/:userId/:userView/fotoCollection` Questa richiesta è di tipo POST ed è utilizzata per il download di foto di una collezione. Richiede l'identificativo dell'utente di cui si desidera visualizzare le foto (`userView`). Il nome della collezione viene estratta dal corpo della richiesta. Successivamente, esegue una query al database per ottenere le foto associate a questa collezione specifica.
- `/dashboard/:userId/:userView/kmeans` Questa richiesta di tipo POST è utilizzata per l'analisi di cluster con l'algoritmo K-Means di cui successivamente verrà spiegato il funzionamento. La procedura di analisi prevede i seguenti passaggi:

- Recupero dei Dati: Le coordinate latitudine e longitudine delle foto caricate da un utente specifico vengono estratte dal database.
- Determinazione del Numero di Cluster: Il numero di cluster viene recuperato dal corpo della richiesta. Nel caso in cui il numero di cluster non è definito viene utilizzato l'Elbow Method (implementato in una funzione apposita) per calcolare il numero ottimale di cluster. Questo passaggio è essenziale per definire quanti gruppi distinti di foto dovrebbero essere creati.
- Esecuzione di K-Means: L'algoritmo K-Means viene avviato con il numero di cluster identificato nel passaggio precedente. Le coordinate delle foto vengono quindi suddivise in cluster basati sulla loro vicinanza spaziale.
- Visualizzazione dei Risultati: I risultati dell'analisi vengono presentati in un formato facilmente comprensibile, fornendo una panoramica dei cluster identificati e delle foto contenute in ciascun cluster.

Il metodo utilizzato per calcolare il numero ottimale di cluster (Elbow Method) verrà discusso più avanti in questo paragrafo.

- /dashboard/:userId/:userView/dbSCAN
Questa richiesta di tipo POST è utilizzata per l'analisi di cluster con l'algoritmo DBSCAN (Density-Based Spatial Clustering of Applications with Noise) di cui successivamente verrà spiegato il funzionamento. Questo algoritmo è viene applicato alle coordinate geografiche delle foto caricate dall'utente loggato. Inizialmente, vengono quindi estratte le coordinate latitudine e longitudine delle foto dal database. L'utente ha la possibilità di configurare due parametri fondamentali: "epsilon", che rappresenta il raggio dell'intorno, e "minPoints", che indica il numero minimo di punti richiesti per formare un cluster. Prima di avviare l'algoritmo DBSCAN, vengono effettuate verifiche per garantire che ci siano dati sufficienti da analizzare e che i parametri specificati siano validi. L'algoritmo DBSCAN viene quindi eseguito, valutando la densità dei dati nello spazio e identificando cluster basati sulla vicinanza dei punti. I risultati dell'analisi DBSCAN vengono presentati in un formato comprensibile, evidenziando i cluster identificati e i punti dati associati a ciascun cluster.
- /dashboard/:userId/addShare
Questa richiesta POST è utilizzata per aggiungere la condivisione tra due utenti. L'utente che desidera effettuare la condivisione, viene identificato tramite "userId" mentre l'utente con cui si intende condividere il contenuto, indicato come "condividicon", viene prelevato dal body della richiesta. Una volta effettuata la condivisione, il sistema restituirà una risposta appropriata in base all'esito dell'operazione. Se la condivisione ha successo, verrà confermato il corretto inserimento. In caso contrario, se esiste già una condivisione con l'utente specificato, verrà segnalato l'eventuale duplicazione dell'operazione.
- /dashboard/:userId/getUserFriends
Richiesta GET per ottenere tutti gli amici di un utente.
Questa richiesta permette agli utenti di recuperare un elenco completo dei loro amici registrati nell'applicazione. Utilizzando il parametro "userId" per identificare l'utente di interesse, vengono recuperati dal database gli utenti che hanno attivato una condivisione con l'utente loggato. Se sono presenti amici, viene restituito un elenco contenente i loro nomi utente e identificatori. In caso contrario, viene comunicato che non sono stati trovati amici.

Come precedentemente discusso, alcuni algoritmi e metodi necessitano di una spiegazione dettagliata qui di seguito riportata.

1) *K-Means*: Il K-Means è un algoritmo di clustering ampiamente utilizzato nell'analisi dei dati e nell'apprendimento automatico. Il suo scopo principale è quello di suddividere un insieme di dati in gruppi omogenei chiamati "cluster" in base alla similarità delle loro caratteristiche.

L'algoritmo K-Means opera attraverso i seguenti passi principali:

- Inizializzazione dei centroidi: L'algoritmo inizia selezio-

nando casualmente K punti dal set di dati come centroidi iniziali. K rappresenta il numero desiderato di cluster.

- Assegnazione ai cluster: Ogni punto dati viene assegnato al cluster il cui centroide è più vicino in base a una misura di distanza, spesso la distanza euclidea. Questo passo crea i primi cluster.
- Aggiornamento dei centroidi: Calcolando la media delle caratteristiche di tutti i punti assegnati a ciascun cluster, si sposta il centroide del cluster in questa nuova posizione.
- Iterazione: I passi 2 e 3 vengono ripetuti iterativamente fino a quando i centroidi non cambiano significativamente. Questo processo è noto come "convergenza".
- Risultati finali: Alla fine, si ottengono K cluster di dati, ognuno contenente punti simili tra loro e differenti dai punti in altri cluster.

I vantaggi principali del K-Means includono la sua semplicità, la velocità di convergenza, e l'efficacia nella gestione di grandi quantità di dati. Tuttavia, ha alcune limitazioni, come la sensibilità alla scelta iniziale dei centroidi e l'assunzione che i cluster abbiano forma sferica e dimensioni simili.

Come già menzionato in precedenza, è possibile determinare il valore del parametro "k," che rappresenta il numero di cluster, in due modi: mediante l'input dell'utente o utilizzando il metodo del "gomito," che verrà spiegato di seguito.

2) *Elbow Method*: Il metodo del "gomito" (implementato nella funzione calculateOptimalClusterNumber) è un approccio di selezione del numero ottimale di cluster in un algoritmo di clustering basato su K-Means. Questa tecnica coinvolge il calcolo della Somma degli Errori Quadratici (SSE) per diversi valori di K (il numero di cluster). L'SSE misura la variazione totale all'interno di un cluster e, di conseguenza, quantifica quanto bene i dati sono suddivisi nei cluster. Quando aumentiamo il numero di cluster, l'SSE tende a diminuire poiché i cluster sono più specifici. Tuttavia, oltre un certo punto, la diminuzione dell'SSE rallenta. Il punto in cui l'SSE smette di diminuire rapidamente e inizia a stabilizzarsi è noto come "gomito". Questo punto rappresenta il numero ottimale di cluster, poiché ulteriori cluster non porterebbero miglioramenti significativi nella suddivisione dei dati. Il metodo del "gomito" offre una guida oggettiva nella scelta del numero di cluster, contribuendo a evitare under-clustering (troppi cluster) o over-clustering (troppo pochi cluster), migliorando l'efficacia dell'analisi dei dati.

3) *DBSCAN*: Il DBSCAN è un algoritmo di clustering basato sulla densità progettato per identificare cluster nei dati senza richiedere una specifica anticipata del numero di cluster desiderati. Il suo funzionamento è suddiviso in una serie di passaggi chiave:

- Scelta di un punto iniziale: L'algoritmo inizia selezionando un punto qualsiasi dal set di dati come punto di partenza. Questo punto viene etichettato come il primo "nucleo."
- Espansione dei cluster: Il DBSCAN esplora la vicinanza del nucleo in base a una misura di distanza specifica chiamata "distanza di raggio" (epsilon, ϵ). Il raggio ϵ

rappresenta la massima distanza entro cui il DBSCAN ricerca punti vicini al nucleo.

- Identificazione dei punti del cluster: Ogni punto di dati situato all'interno del raggio ϵ dal nucleo corrente viene considerato parte del cluster attuale. Questi punti vengono etichettati come membri del cluster.
- Densità e nuovi nuclei: Per ogni punto incluso nel cluster, il DBSCAN verifica se ci sono abbastanza punti (almeno "minPts") entro il raggio ϵ . Se il numero minimo di punti è soddisfatto, il punto viene considerato un nuovo nucleo, e il processo si ripete a partire da questo punto. Questo consente l'espansione del cluster in modo incrementale.
- Punti di rumore: I punti che non soddisfano i requisiti per essere un nucleo e che non sono inclusi in alcun cluster vengono etichettati come "punti di rumore."
- Iterazione: Il processo di esplorazione e assegnazione ai cluster continua fino a quando tutti i punti sono stati assegnati a un cluster o etichettati come punti di rumore. Questo assicura che tutti i punti siano presi in considerazione nell'analisi del clustering.

Il DBSCAN è noto per la sua capacità di identificare cluster di forma arbitraria e adattarsi a diverse densità dei dati. Non richiede la specifica preventiva del numero di cluster, il che lo rende adatto a situazioni in cui questo valore è sconosciuto. Il DBSCAN potrebbe non essere altrettanto efficace in dataset di alta dimensionalità a causa del "curse of dimensionality."

C. Frontend

In questa parte procederemo a fornire una dettagliata spiegazione sull'implementazione del front-end, inclusa un'esaustiva descrizione delle funzionalità chiave che sono state integrate. Inoltre, useremo delle immagini per mostrarvi chiaramente come abbiamo sviluppato il nostro sito web e le soluzioni che abbiamo scelto. Per l'implementazione abbiamo scelto di utilizzare il framework React per il frontend al fine di massimizzare la riusabilità del codice e dei componenti. React è un framework JavaScript open-source sviluppato da Facebook, noto per la sua capacità di creare componenti riutilizzabili. Per la gestione delle mappe nel frontend, abbiamo impiegato la libreria Leaflet, un framework JavaScript open-source specializzato nella creazione di mappe interattive. Per il design grafico abbiamo adottato il framework Material-UI, un framework JavaScript open-source che ci ha permesso di creare componenti grafici in linea con lo stile di Material Design.

1) *Login / Registrazione*: La prima pagina del nostro sito è dedicata al processo di accesso, offrendo agli utenti la possibilità di entrare nel sistema. Da questa pagina, gli utenti hanno la comodità di accedere alla pagina di registrazione tramite un link apposito. La pagina di registrazione consente agli utenti di iscriversi al sistema semplicemente inserendo un nome utente, un indirizzo email e una password. Abbiamo scelto di adottare un processo di registrazione rapido e diretto, pur garantendo rigorosi controlli sulla validità dell'indirizzo email, sulla sicurezza della password e in generale sulla correttezza dei dati inseriti dagli utenti. La nostra priorità è

fornire un'esperienza di registrazione fluida, senza trascurare la sicurezza e l'integrità dei dati utente.

Fig. 8. Login e Registrazione

2) *Dashboard*: La nostra dashboard è strutturata in tre componenti chiave, ciascuna svolge un ruolo specifico: una sidebar sulla sinistra, una mappa centralizzata e una galleria sulla destra. Nella parte superiore, troviamo una navbar che offre opzioni per personalizzare l'esperienza dell'utente, tra cui la scelta tra due tipi di mappe (OpenStreetMap o Stamen) e la possibilità di selezionare il profilo di un "amico" per visualizzare i suoi contenuti.

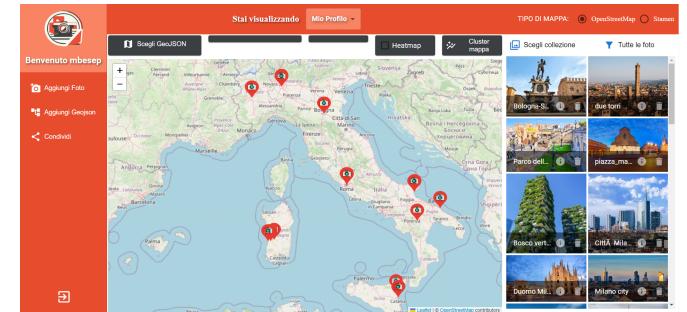


Fig. 9. Dashboard

Per una comprensione approfondita delle operazioni eseguibili nella nostra dashboard, sono di seguito descritte le funzionalità disponibili nella sidebar, sulla mappa e nella galleria. Ciascuna di queste aree offre diverse opzioni e strumenti per interagire con i dati e personalizzare l'esperienza dell'utente.

Sidebar

All'interno della nostra sidebar, gli utenti possono svolgere una serie di operazioni fondamentali. Ecco un'illustrazione dettagliata delle azioni disponibili:

- Aggiungi Foto: Attraverso questo pulsante, gli utenti possono avviare un processo interattivo che consente loro di caricare una foto sulla piattaforma. Inizialmente, gli

utenti scelgono il file immagine che desiderano caricare. Questa fase permette loro di importare l'immagine desiderata direttamente dal proprio dispositivo. Dopo aver selezionato l'immagine, gli utenti inseriscono l'indirizzo associato a quella posizione specifica. Questa funzione è resa possibile grazie all'integrazione dell'API di Google Maps, che semplifica il processo di geocodifica. Prima di completare il caricamento, agli utenti viene richiesto di selezionare una collezione in cui desiderano memorizzare l'immagine. Possono anche optare per la creazione di una nuova collezione semplicemente inserendo un nome. Questa caratteristica è progettata per organizzare le immagini in categorie pertinenti, consentendo agli utenti di accedere facilmente ai propri contenuti.

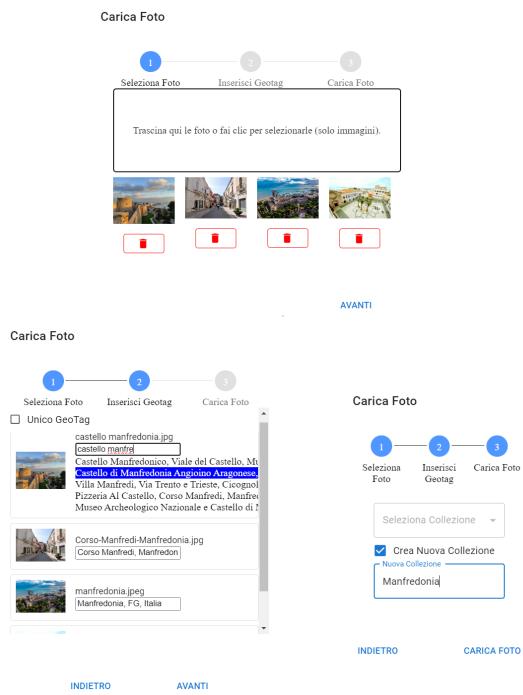


Fig. 10. Photo upload

- Carica GeoJSON: Facendo clic sul pulsante "Aggiungi GeoJSON", si apre un secondo popup che guida gli utenti attraverso un processo interattivo per l'importazione di file GeoJSON. Inizialmente, gli utenti devono scegliere il file GeoJSON che desiderano caricare. Questo file contiene dati geografici rilevanti, come regioni o confini geografici, di tipo multi-polygon. Successivamente, viene richiesto agli utenti di assegnare un nome significativo al file GeoJSON che stanno caricando. Questo passo aiuta a identificare e organizzare i file all'interno della piattaforma. Un'importante caratteristica di questo processo è la possibilità per gli utenti di selezionare una "feature" all'interno del file GeoJSON che meglio descrive le aree rappresentate. Ad esempio, se il file GeoJSON contiene dati delle regioni italiane, gli utenti possono selezionare la "regione" (o un nome che richiama quel concetto) come

feature principale. Questo passaggio è cruciale per la corretta interpretazione dei dati geografici successivamente sulla mappa.

- Condividi: Al click del pulsante "Condividi", si apre un popup semplice al cui interno gli utenti possono scegliere uno degli utenti registrati nel sistema con cui desiderano condividere le proprie foto. Questa funzione consente agli utenti di gestire in modo diretto e intuitivo la condivisione delle proprie immagini con persone specifiche
- Logout: L'icona del logout, situata nella parte inferiore della sidebar, permette agli utenti di uscire dalla loro dashboard e tornare alla pagina di accesso.

Mappa

La mappa costituisce il fulcro del nostro progetto, rivestendo un ruolo di primaria importanza. Al momento del login, la mappa si inizializza con tutti i marker corrispondenti alle foto caricate dall'utente autenticato. Ciascun marker è posizionato con precisione in base al geotag della foto associata. Cliccando su uno qualsiasi dei marker, si apre un popup informativo che mostra le foto caricate in quella posizione specifica, offrendo una visualizzazione chiara e immediata dei contenuti geograficamente correlati. La mappa è altamente interattiva, sfruttando la potenza di Leaflet. Gli utenti possono facilmente ingrandire, rimpicciolire e navigare sulla mappa utilizzando il mouse, garantendo un'esperienza fluida e personalizzata.

All'interno di ciascun popup, oltre alle immagini, vengono visualizzate altre informazioni di rilievanza, tra cui l'indirizzo inserito durante il caricamento delle foto. Questo arricchisce ulteriormente il contesto delle immagini, fornendo dettagli geografici significativi che arricchiscono l'esperienza dell'utente.



Fig. 11. Popup

Nella parte superiore della mappa un insieme di pulsanti consentono agli utenti di eseguire una serie di operazioni fon-

damentali. Questi pulsanti offrono un facile accesso a funzionalità chiave, migliorando la versatilità e l'utilità dell'interfaccia utente. Ogni pulsante svolge un compito specifico sfruttando così appieno le potenzialità della mappa e personalizzare la propria esperienza secondo le proprie esigenze.

- Scegli GeoJSON: Cliccando su questo pulsante, si attiva un menu a tendina che offre agli utenti la possibilità di selezionare uno dei file GeoJSON precedentemente caricati. Una volta effettuata la scelta del file GeoJSON, la mappa reagisce mostrando esclusivamente le foto posizionate all'interno dell'area geografica definita da quel specifico GeoJSON. Inoltre, le regioni o le aree coperte dal GeoJSON selezionato vengono evidenziate di colore blu sulla mappa.

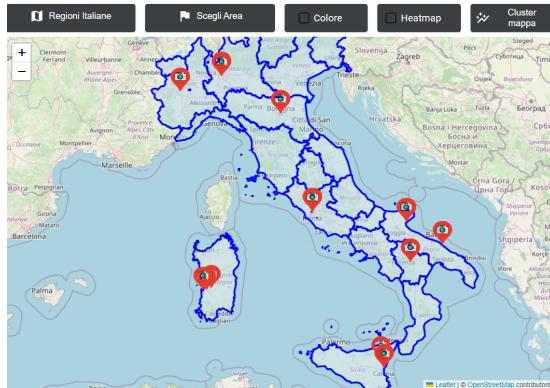


Fig. 12. Mappa con geojson caricato

- Scegli Area: Attraverso questo pulsante, gli utenti hanno la possibilità di definire una porzione più specifica all'interno del GeoJSON precedentemente selezionato. Questo permette loro di restringere ulteriormente l'area in cui desiderano visualizzare le immagini. Una volta delineata questa regione più specifica, la mappa si adatta, mostrando esclusivamente le immagini che si trovano all'interno di questa area limitata.

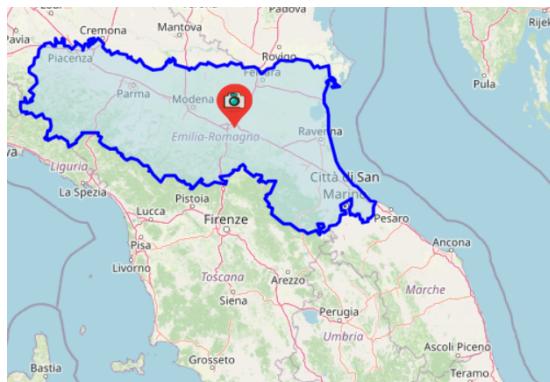


Fig. 13. Mappa con Emilia Romagna selezionata

- Colora: Una volta scelto un GeoJSON, viene attivata la funzione "Colora" tramite una checkbox. Quando questa

casella è selezionata, la mappa si colora in modo dinamico, utilizzando il GeoJSON selezionato come guida. La colorazione è basata sul numero di foto presenti all'interno delle diverse aree definite nel GeoJSON. Concretamente, questa funzione attribuisce colori distinti alle regioni in base alla quantità di immagini contenute al loro interno. Ad esempio, una regione con un numero maggiore di immagini verrà rappresentata con un colore differente rispetto a una regione che contiene un numero inferiore di immagini.



Fig. 14. Mappa colorata

- Heatmap: Questa checkbox offre agli utenti la possibilità di attivare una visualizzazione particolare all'interno della mappa, chiamata "heatmap." Questa funzione è potente perché consente di rappresentare dati di densità in modo chiaro e significativo. Se è selezionato un GeoJSON opera sulle foto presenti all'interno del GeoJSON, altrimenti opera su tutte le foto. L'heatmap è una rappresentazione grafica che mostra la concentrazione o densità di dati all'interno di una mappa. In sostanza, essa colora le aree con colori diversi in base alla quantità di dati presenti. Ad esempio, se ci sono molte foto in una zona specifica, quell'area verrà rappresentata con un colore intenso, mentre le zone con meno foto appariranno con colori più tenui.
- Cluster: Attraverso questo pulsante, gli utenti hanno la possibilità di svolgere un'operazione di clustering sulle foto caricate, indipendentemente che si tratti delle proprie foto o di quelle di un amico. Questa funzione lavora su tutto il pool di immagini disponibili. Cliccando sul pulsante, si apre un popup che consente agli utenti di scegliere tra due diversi algoritmi: K-Means o DBSCAN. Se l'utente sceglie l'algoritmo K-Means, ha la possibilità di optare per una configurazione automatica o manuale, con la conseguente scelta del numero di cluster desiderati.

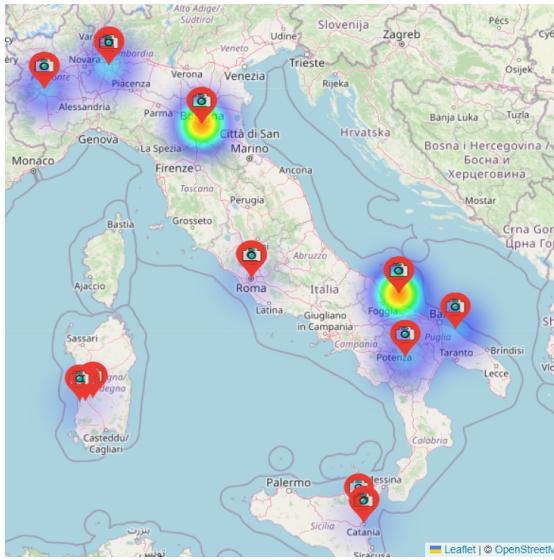


Fig. 15. Mappa con heatmap

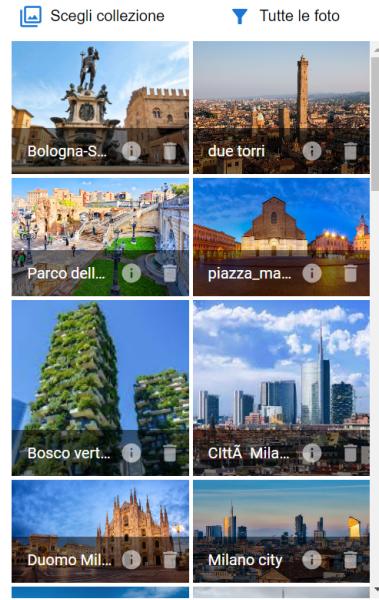


Fig. 17. Galleria

Se, invece, l’utente sceglie l’algoritmo DBSCAN, sarà necessario impostare i valori di epsilon e minpoints. Una volta configurato il clustering, l’utente può avviare il processo e visualizzarlo sulla mappa. Questa funzione offre agli utenti anche la possibilità di rimuovere una clusterizzazione precedentemente effettuata.

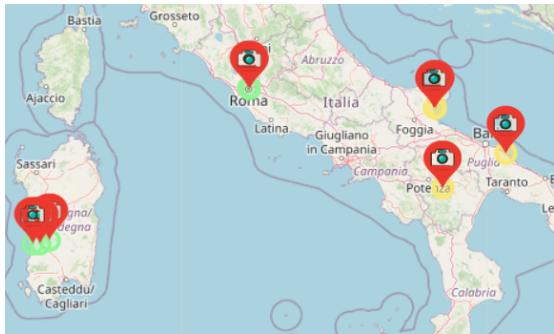


Fig. 16. Mappa con cluster

Galleria

La galleria è un hub visivo delle immagini, e permette di sfogliarle con facilità offrendo anche funzionalità avanzate.

Una delle opzioni chiave è la capacità di filtrare le immagini in base alle collezioni. È possibile quindi visualizzare solo le immagini all’interno di una collezione specifica. Su ogni immagine è possibile aprire un piccolo popup che fornisce dettagli aggiuntivi, tra cui informazioni sull’indirizzo associato alla foto e altri dati pertinenti. Questa funzione aggiunge valore alle tue immagini, fornendo ulteriori contesti e informazioni. Se si desidera rimuovere una foto specifica, basta fare clic sull’icona del cestino associata a quella foto.

IV. RISULTATI

Riteniamo che il nostro progetto abbia raggiunto e superato con successo gli obiettivi stabiliti in fase di progettazione, rispettando appieno le richieste della traccia del progetto. Abbiamo sviluppato una piattaforma software che gestisce in modo efficiente file multimediali georeferenziati, fornendo agli utenti un’esperienza completa e immersiva. Il nostro sistema si compone di tre componenti chiave: un client mobile, un back-end e un front-end Web. Il client mobile offre agli utenti la possibilità di catturare foto con georeferenziazione, creare collezioni personalizzate e condividere le proprie esperienze in modo semplice e intuitivo. Il back-end si occupa di memorizzare e gestire i dati, inclusi file multimediali e geo-tag, e fornisce funzionalità di query geo-spaziali per accedere alle informazioni in modo efficiente. Il front-end Web offre un’interfaccia utente avanzata, consentendo agli utenti di visualizzare e manipolare i dati in varie modalità. Riteniamo di aver implementato le specifiche richieste dalla traccia del progetto ma anche di aver implementato funzionalità aggiuntive che arricchiscono l’esperienza dell’utente. La creazione di un sistema di login e registrazione, la gestione degli utenti e la possibilità di condividere contenuti con altri utenti sono solo alcune delle funzionalità aggiuntive che abbiamo incluso. Abbiamo inoltre implementato con successo funzionalità avanzate, come la clusterizzazione delle varie immagini utilizzando gli algoritmi K-Means e DBSCAN, la visualizzazione delle foto in base a vari criteri di selezione e la creazione di heatmap per visualizzare i dati in modo intuitivo. Il nostro sistema offre una soluzione scalabile e attraente e siamo fiduciosi che il nostro progetto possa essere un valido strumento per la gestione di file multimediali geo-referenziati.

REFERENCES

- [1] React, La libreria React, Facebook, 2018.
- [2] Node.js, Documentazione *JavaScript runtime*, OpenJS Foundation, 2023.
- [3] Flutter, *Google's UI toolkit*, Google, 2023. Documentazione
- [4] Leaflet, *a JavaScript library for interactive maps*, Leaflet, 2023. Documentazione
- [5] Google Maps API, *Google Maps Platform*, Google, 2023. Documentazione
- [6] PostgreSQL: *The world's most advanced open source database*, PostgreSQL Global Development Group, 2023. Documentazione
- [7] PostGIS - *Geographic Objects for PostgreSQL*, PostGIS Project, 2023. Documentazione
- [8] pgAdmin - *The PostgreSQL Management Tool*, The pgAdmin Development Team, 2023. Documentazione
- [9] Android Studio, IDE Android Studio, Katherine Chou, 2013.