

UNIVERSITÀ DEGLI STUDI DI FIRENZE
SCUOLA DI INGEGNERIA - DIPARTIMENTO DI INGEGNERIA
DELL'INFORMAZIONE

Tesi di Laurea Triennale in Ingegneria Informatica

**UN SISTEMA WEB PER LA PROGRAMMAZIONE DI
EVENTI SUL TERRITORIO ATTRAVERSO L'ANALISI
DI INFORMAZIONI GEOLOCALIZZATE DA TWITTER**

Candidato
Enrico Collini

Relatore
Alberto Del Bimbo

Correlatore
Andrea Ferracani

Anno Accademico 2017/2018

Indice

Introduzione	i
Ringraziamenti	iii
1 Stato dell'Arte	1
1.1 L'utilizzo di internet nel mondo, statistiche	1
1.2 Analisi dei dati di Twitter, Sentiment Analysis	2
2 Architettura Software	4
2.1 Descrizione	4
2.1.1 Database	4
2.1.2 Servers	5
2.1.3 User Interface	5
2.2 Deployment	5
3 Modulo Gestione Dati	6
3.1 MongoDB	6
3.2 Struttura del Database	7
4 Business Logic	9
4.1 Node.js	9
4.1.1 npm	10

4.2	Database Server	10
4.3	Application Server	12
4.3.1	SeatGeek	13
4.3.2	Twitter Developer	13
4.3.3	AYLIEN	14
4.4	Software Testing	16
5	Interfaccia Utente	17
5.1	Obiettivi predisposti	17
5.2	Design	18
5.3	Mappa	19
5.4	Layout	19
5.5	Schermate Applicazione	20
6	Deployment e link	28
6.1	Heroku	28
6.2	GitHub	28
7	Application Testing & Results	30
8	Conclusione	33
8.1	Possibili Sviluppi Futuri	33
9	Sitografia	35
9.1	Back End	35
9.2	Front End	35
9.3	Deployment	36
9.4	Altro	36
	Bibliografia	37

Introduzione

La pervasività dei Social Media nella vita quotidiana delle persone e la quantità di informazioni derivabili dall'interazione diretta tra individui sulla rete forniscono validi strumenti di analisi dei comportamenti e degli interessi sociali.

L'utilizzo di queste risorse, non solo offre un panorama applicativo voluminoso e variegato, ma costituisce il fondamento che ha motivato lo sviluppo di questo software.

Questa applicazione web si pone l'obiettivo di fornire a programmatori di eventi uno strumento innovativo per la ricerca di Locations in base ai propri requisiti e guidato attraverso il seguito sociale su Twitter.

Gli strumenti utilizzati per l'implementazione sono all'avanguardia ed in particolare usano criteri derivati dal libro *Designing the User Interface: Strategies for Effective Human-Computer Interaction* di Ben Shneiderman e Catherine Plaisant [1], per la creazione di un'interfaccia utente interattiva con l'obiettivo di ottimizzare l'usabilità del software.

Attraverso la definizione di questi parametri è stato scelto come elemento principale dell'applicazione una mappa interattiva. Questa permette di identificare zone attraverso poligoni più o meno evidenziati in base ai risultati ottenuti dal Sentiment Analysis dei Tweets sulle locations. Questi luoghi sono rappresentati da marker. Questi elementi sono selezionabili e contengono

informazioni di dettaglio. Tutto questo, integrato in un'interfaccia grafica che guida l'utente in base alle sue esigenze nella scelta di una location favorevole per la programmazione di un particolare evento.

La struttura di questo documento è composta da una parte introduttiva contenente lo stato dell'arte del contesto al quale appartiene l'applicazione, si evolve sui dettagli implementativi che vanno dalla descrizione dell'architettura del software e dei moduli funzionali alla presentazione dell'interfaccia utente con particolare attenzione ai requisiti sostanziali e termina con la descrizione del processo di dislocamento dell'intera applicazione con alcune osservazioni conclusive anche in riferimento ai possibili sviluppi futuri di questo software.

Ringraziamenti

Desidero ringraziare tutti coloro che mi hanno sostenuto in questo percorso che mi ha arricchito in maniera significativa, non solo a livello formativo, ma anche personale.

Prima di tutto vorrei ringraziare il Professore Alberto Del Bimbo e il Ricercatore Andrea Ferracani che mi hanno seguito nella stesura della tesi con suggerimenti, critiche ed osservazioni: a loro va la mia gratitudine, anche se a me spetta la responsabilità per ogni errore contenuto in questa tesi.

Vorrei ringraziare la mia famiglia che mi ha sostenuto e aiutato durante questo periodo, ma anche i miei amici che hanno reso questi tre anni di studio indimenticabili.

Ritengo questo risultato molto importante, averlo raggiunto rappresenta un ulteriore incentivo al conseguimento della laurea magistrale.

Capitolo 1

Stato dell'Arte

1.1 L'utilizzo di internet nel mondo, statistiche

Secondo uno studio di Hootsuite We Are Social sul totale della popolazione mondiale, stimata in 7,655 miliardi di individui il 55% di questi sono utenti di internet, il 44% sono attivi sui social network e il 42% utilizza dispositivi mobile per accedere alla rete.

Altre statistiche sono riportate nei grafici sottostanti, da cui si ricava che per il social network Twitter, su cui si basa quest'applicazione, la nazione con il numero più alto di utilizzatori sono gli Stati Uniti con 49.350.000 utenti attivi. Gli utilizzatori di questa piattaforma in Italia sono invece 2.880.000 con il nostro paese al diciottesimo posto nella graduatoria.

Da questi dati traspare che l'80% degli utenti di internet sono anche attivi sui social network. Questo ribadisce il peso che questi strumenti hanno nella vita quotidiana come approfondito nel libro di Enrico Cheli "La realtà mediata: l'influenza dei mass media tra persuasione e costruzione sociale della realtà" [2].



Figura 1.1: Risultati statistici di Hootsuite We Are Social inerenti all'utilizzo di internet nel mondo



Figura 1.2: Risultati statistici di Hootsuite We Are Social inerenti all'utilizzo di Twitter nel mondo

1.2 Analisi dei dati di Twitter, Sentiment Analysis

Come riportato nell'articolo di Alexander Pak e Patrick Paroubek, "Twitter as a Corpus for Sentiment Analysis and Opinion Mining" [3], i microblog sono divenuti oggi uno strumento di comunicazione molto popolare tra gli utenti di internet. Milioni di user condividono opinioni su diversi aspetti della vita di ogni giorno. Queste rappresentano delle risorse preziose sulle quale effettuare Sentiment Analysis e Opinion Mining per ottenere risultati

interessanti nei vari settori dell'Economia o per studi sociali.

Per quanto riguarda l'applicazione elaborata per questa tesi, le informazioni derivate dal Sentiment Analysis dei Tweets sulle locations sono state utilizzate come oggetto di classificazione per la positività dei possibili luoghi in cui poter organizzare un evento.

Capitolo 2

Architettura Software

L'architettura della Web Application è definita dallo schema in 5.6

Figura 2.1: Schema Architettura Software dell'applicazione

2.1 Descrizione

L'architettura software implementata si basa sullo schema a tre livelli:

- modulo di gestione dati
- business logic
- interfaccia utente

2.1.1 Database

Per quanto riguarda il modulo di gestione dei dati è stato utilizzato come Database MongoDB, una base di dati non relazionale, ma basata su documenti.

2.1.2 Servers

Nella Business Logic distinguiamo un Server per il database ed un Server Applicativo. Entrambi implementati in Node.js Express.js, mentre per l'interfacciamento con il Database è stata utilizzata la libreria mongoose per Node.

2.1.3 User Interface

L'interfaccia utente è stata implementata con HTML CSS e Javascript senza l'utilizzo di alcun framework lato client per avere maggiore libertà di personalizzazione del software. Come Design lo stile adottato è il Flat Design e l'obiettivo è di massimizzare assieme a questo l'usabilità del software.

2.2 Deployment

Per effettuare il dispiegamento dell'applicazione web è stato utilizzato Heroku, una piattaforma Cloud che permette non solo l'hosting di applicazioni web, ma permette anche di ospitare server.

Capitolo 3

Modulo Gestione Dati

3.1 MongoDB

MongoDB è un DBMS (Database Management System) non relazionale orientato ai documenti. Appartiene alla categoria delle base di dati NoSQL. Tra le principali caratteristiche quindi troviamo:

- **Struttura Schemaless:** i dati possono essere strutturati senza alcuna regola. Possono quindi essere non rispettati i criteri definiti dalle forme normali nei database SQL.
- **Document Based:** La struttura della base di dati è composta da Collections di Documenti costruiti sulla base di uno Schema. I dati sono salvati in formato JSON.

La scelta di utilizzare un database non relazionale è stata presa perchè per lo sviluppo di questa applicazione web la struttura basata su documenti in formato JSON agevolava l'utilizzo e la creazione di questi. Un'altra motivazione per l'impiego di questo è la compatibilità con l'ambiente di sviluppo Node.js mediante la libreria Mongoose.

3.2 Struttura del Database

La base di dati è organizzata in una singola Collection Locations.

Lo schema dei luoghi è rappresentato come riportato.

```
let locationSchema = new mongoose.Schema({
  locName: String,
  locTax: String,
  locLat: Number,
  locLon: Number,
  locAddress: String,
  locUrl: String,
  locFollowersNum: Number,
  locDescription: String,
  locPercPos: Number,
  locPercNeu: Number,
  locPercNeg: Number
});
```

- locName contiene il nome del luogo.
- locTax identifica la tassonomia della location: musica, teatro, commedia.
- locLat locLon individuano la latitudine e la longitudine per inserire nella mappa interattiva il luogo.
- locAddress invece contiene l'indirizzo esplicito.
- locUrl permette di poter effettuare la connessione al sito ufficiale delle location.
- locFollowersNum conserva il numero totale di follower sulla pagina Twitter del luogo.

- locDescription invece conserva la descrizione della pagina Twitter.
- locPercPos, locPercNeu, locPercNeg salvano i risultati del Sentiment Analysis dei Tweet sulle locations.

Capitolo 4

Business Logic

4.1 Node.js

Node.js o semplicemente Node è una piattaforma software open-source che permette di sviluppare applicazioni di diverse tipologie: dallo sviluppo di software desktop cross-platform fino anche ad applicazioni per l' IoT (Internet of Things).

Il campo però in cui è maggiormente impiegato è sicuramente lo sviluppo web.

Utilizza come motore il runtime Javascript V8 realizzato da Google, impiegato anche su Chrome. Questa caratteristica, assieme alla scelta implementativa di un modello orientato agli eventi (event-driven) ed un'architettura sul modello I/O non bloccante, lo rendono particolarmente adatto alla realizzazione di applicazioni web.

4.1.1 npm

Il Node Package Manager è sicuramente un elemento sostanziale di Node. Contiene pacchetti aggiuntivi che permettono di completare Node con strumenti adatti alle particolari esigenze applicative.

Per lo sviluppo di questa applicazione web, oltre ai moduli nativi sono stati aggiunti:

- Express.js.

Un framework Web che permette di eseguire una serie di compiti avanzati per le applicazioni web, in particolare per la gestione e la creazione di API (Application Programming Interface).

- Mongoose.

Mongoose invece è un modulo il cui compito è quello di gestire l'integrazione di un database MongoDB nell'ambiente Node.

Maggiori dettagli implementativi sono forniti nelle specifiche sezioni dedicate ai server.

4.2 Database Server

Questo modulo server è stato implementato utilizzando Node.js e messo in funzione attraverso la piattaforma Heroku.

La libreria aggiuntiva chiave per questo specifico componente è Mongoose.

Il compito principale è, infatti, quello di gestire la connessione con la base di dati per l'applicazione Web, ma per fare questo è stato utilizzato anche il framework Express.js per il colloquio con la parte client reso possibile attraverso API.

In particolare, i compiti implementati sono:

- Connessione con il Database
- Creazione dello Schema del Database
- Inserimento dei documenti nel Database
- Prelevare i dati dal Database
- Cancellare specifici documenti attraverso l'id caratteristico

Viene di seguito riportato il codice per la connessione con la base di dati.

```
mongoose.connect('mongodb+srv://enrico:' +  
process.env.MONGO_ATLAS_PW +  
'@cluster0-coj9j.mongodb.net/test?retryWrites=true',  
{ useNewUrlParser: true });
```

Ed il codice per prelevare i dati dal database.

```
router.get("/", (req, res, next) = {  
  LocationSchema.find()  
    .exec()  
    .then(docs = {  
      console.log(docs);  
      const response = {  
        count: docs.length,  
        locations: docs.map(doc = {  
          return {  
            _id: doc._id,  
            locName: doc.locName,  
            locTax: doc.locTax,  
            locLat: doc.locLat,  
            locLon: doc.locLon,
```

```
        locAddress: doc.locAddress,
        locUrl: doc.locUrl,
        locFollowersNum: doc.locFollowersNum,
        locDescription: doc.locDescription,
        locPercPos: doc.locPercPos,
        locPercNeu: doc.locPercNeu,
        locPercNeg: doc.locPercNeg
    }
    })
};
res.status(200).json(response);
})
.catch(err = {
    console.log(err);
    res.status(500).json({
        error: err
    });
});
});
```

4.3 Application Server

Questo modulo server è stato implementato utilizzando Node.js e messo in funzione attraverso la piattaforma Heroku.

La libreria aggiuntiva principale per questo componente è Express.js.

Il compito principale è quello di elaborare i dati necessari all'applicazione web. Questo processo comprende la gestione delle autenticazioni per l'utilizzo delle API di terze parti e la creazione di API per l'utilizzo delle informazioni ottenute lato client.

In particolare, sono stati implementati i seguenti sistemi gestionali per:

- SeatGeek API.

Per ottenere il Dataset degli upcoming eventi di New York.

- Twitter Developer API.

Per collegare le location con i propri account Twitter e ricercare i Tweet su queste.

- AYLIEN API.

Per effettuare il Sentiment Analysis dei Tweet sui luoghi per la programmazione di eventi.

4.3.1 SeatGeek

L'utilizzo di queste API necessita di una semplice Basic Auth Authentication e non richiede il passaggio di alcuni valori chiave per ottenere le informazioni contenute nel Dataset fornito.

La struttura di questa API è riportata nella tabella riepilogativa delle API.

4.3.2 Twitter Developer

La procedura per l'utilizzo delle API di Twitter risulta più complessa. Per ottenere le chiavi segrete ed i tokens per poter utilizzare le API è stata creata un'applicazione Twitter nel portale Twitter Developers.

Per evitare problemi CORS Cross-Origin Resource Sharing e poter effettuare l'autenticazione OAuth necessaria per utilizzare le API di Twitter è stato utilizzato il pacchetto per Node.js npm Twit.

Questo pacchetto è stato creato appositamente per la gestione delle chiamate API di Twitter, in particolare sono stati gestiti i servizi:

- Standard Search API.

Per poter ricercare Tweet secondo parole chiave passate come parametri nella chiamata API.

- Follow, search, and get users API.

Per ricercare gli account delle location in base al nome fornito dal Database Seatgeek ed estrarre informazioni su queste.

- Geo Search API.

Per ottenere informazioni inerenti alla geolocalizzazione.

Le API create per utilizzare le informazioni derivate dal social network sono riportate nella specifica tabella.

4.3.3 AYLIEN

Aylien è un'azienda specializzata in NLP Neuro Linguistic Programming che permette di usufruire dei loro strumenti attraverso l'integrazione mediante API gratuitamente fino a 1000 richieste giornaliere.

Per risolvere problemi CORS ed effettuare l'autenticazione con le chiavi fornite al momento dell'iscrizione a questo servizio è stato utilizzato l'apposito modulo npm `aylien_textapi`. Gli strumenti di Text Analysis forniti da AYLIEN sono molteplici.

Quelli utilizzati ai fini dell'applicazione web sono i servizi specifici per effettuare il Sentiment Analysis, modificando la specializzazione per tweet e non per document (testi di grande dimensione).

Come per i servizi sopracitati, le informazioni delle application programming interface di AYLIEN sono riportate nella tabella descrittiva di tutte le API

fornite da questo server applicativo.

API	URL	Descrizione
GET	" <i>https://api.seatgeek.com/2/events?venue.state = NY&per_page = 2000&page = 1&client_id = ...</i> "	Prende i dati dal Dataset Di SeatGeek sulle locations
GET	"https://twittermapserver.herokuapp.com/tweets/:query"	Standard Search for Tweets passando una keyword nel parametro query
GET	"https://twittermapserver.herokuapp.com/search/:req"	User Search passando il nome del luogo nel parametro req
GET	"https://twittermapserver.herokuapp.com/lookup/:req"	User Lookup passando l'id dell'account nel parametro req
GET	"https://twittermapserver.herokuapp.com/ids/:req"	Get Followers ids passando l'id dell'account nel parametro req
GET	"https://twittermapserver.herokuapp.com/userTweets/:req"	Get Tweets di un utente tramite l'id passato nel parametro req
GET	"https://twittermapserver.herokuapp.com/sentiment/:query"	Sentiment Analysis passando il testo nel parametro query

4.4 Software Testing

Per effettuare il testing delle API fornite dall'applicazione e per controllare la persistenza dei dati è stato utilizzato Postman.

Postman è uno strumento e un software specifico per API Testing, ed è stato adoperato per testare sia l'Application Server che il DB Server e controllare i dati sul Database.

Capitolo 5

Interfaccia Utente

5.1 Obiettivi predisposti

Per costruire la User Interface per questa applicazione sono state seguite le linee guida definite in: 'Designing the User Interface: Strategies for Effective Human-Computer Interaction' di Ben Shneiderman 4th Ed, in particolare cercando di massimizzare l'usabilità dell'applicazione web.

Non sono stati utilizzati framework lato Client, ma è stata sviluppata una interfaccia completamente custom per avere una completa personalizzazione sugli elementi fondamentali del software, primo fra tutti, la mappa. L'applicazione web è classificabile nella categoria definita da Ben Shneiderman come

Office, home, and entertainment applications

Pag 17-19

Le caratteristiche che rientrano nei requisiti di questa categoria sono un' interfaccia semplice, intuitiva, interattiva, attrattiva con procedure facili da memorizzare, adattabile a qualsiasi tipologia di dispositivo. La base della

user interface dovrebbe essere layered based, con una piacevole evoluzione durante l'utilizzo che guidi l'utente verso l'obiettivo dell'applicazione.

5.2 Design

Dopo aver definito i parametri secondo i quali implementare la user interface lo step successivo è la scelta di uno stile per la nostra applicazione. Per ottimizzare la semplicità e l'intuitività del software, la scelta è ricaduta sul Flat Design.

Questo stile si focalizza su una tendenza minimalista per i dettagli degli elementi stilistici, ma preserva uno stile chiaro e tridimensionale utilizzando elementi decorativi semplici come ombre composte da una linea semplice grigia, come i bottoni scelti.

La versione dell'applicazione corrente è stata costruita utilizzando come città nucleo New York. Questa scelta è stata effettuata tenendo presente l'impatto particolare che il seguito Sociale di Twitter ha sulla vita quotidiana della persone di questa città, ma anche perché a New York vivono individui eterogenei con culture molto differenti.

Questa particolarità ha reso più difficile la scelta di una palette di colori che ottimizzasse la componente attrattiva del software, però a vantaggio di una possibile estensione del software anche a livello globale. Questo problema è stato risolto utilizzando i colori derivati da Google ed Amazon che offrono quindi un panorama vario ed equilibrato per qualsiasi utente.

5.3 Mappa

Componente chiave dell'interfaccia dell'applicazione è la mappa. Come scelta è stata utilizzata Leaflet di OpenStreetMap open source.

Questa componente permette di far fronte alle caratteristiche di interattività del software e anche sulla portabilità dell'applicazione per la caratteristica mobile-friendly di Leaflet assieme ad una strategia di sviluppo Responsive.

Gli elementi custom utilizzati nella mappa sono:

- Poligoni.

Questi permettono di identificare delle zone sulla mappa composte da un numero di marker. Se selezionati, i poligoni mostrano un pop-up con delle informazioni specifiche della zona e sono colorati con un'intensità direttamente proporzionale con la percentuale di positività (calcolata dal Sentiment Analysis dei Tweets).

- Marker.

Questi componenti rappresentano i luoghi in cui è possibile organizzare un evento. Se selezionati mostrano anche questi un pop-up con le informazioni dettagliate sulla location sito ufficiale incluso, così come i Tweets più recenti su questa. I Marker si suddividono in 3 categorie; Musica, Teatro e Commedia. Ad ogni tipologia è associato un colore specifico.

5.4 Layout

Per guidare l'utente verso la scelta di una specifica location, oltre all'intensità delle varie zone in base alla positività e alla lista delle locations con indicazioni inerenti alla qualità sopra citate, è stata studiata un'interfaccia

specifica.

Per dirigere l'utente all'utilizzo dell'applicazione, al momento dell'apertura dell'applicazione appare un modal che guida l'utilizzatore alla scelta della categoria di eventi che vorrebbe organizzare.

La possibilità di cambiare categoria è comunque applicabile seguendo il pannello di controllo con i bottoni inerenti alle locations.

L'interfaccia è stata studiata per essere il più Responsive possibile. Infatti è possibile utilizzarla anche su dispositivi mobile oltre che attraverso l'utilizzo di Computer.

5.5 Schermate Applicazione

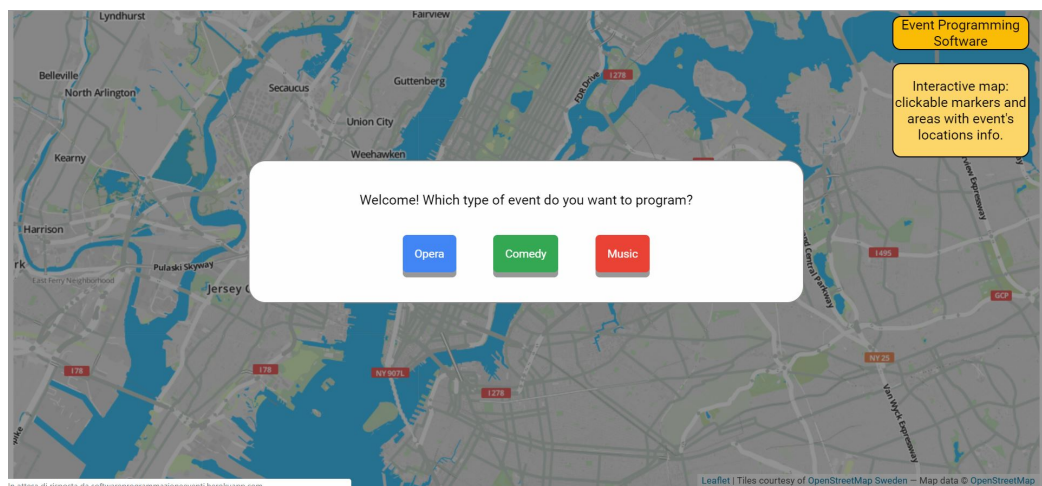


Figura 5.1: Schermata iniziale. Interfaccia introduttiva che guida l'utente alla scelta di una categoria in base all'evento che deve programmare.

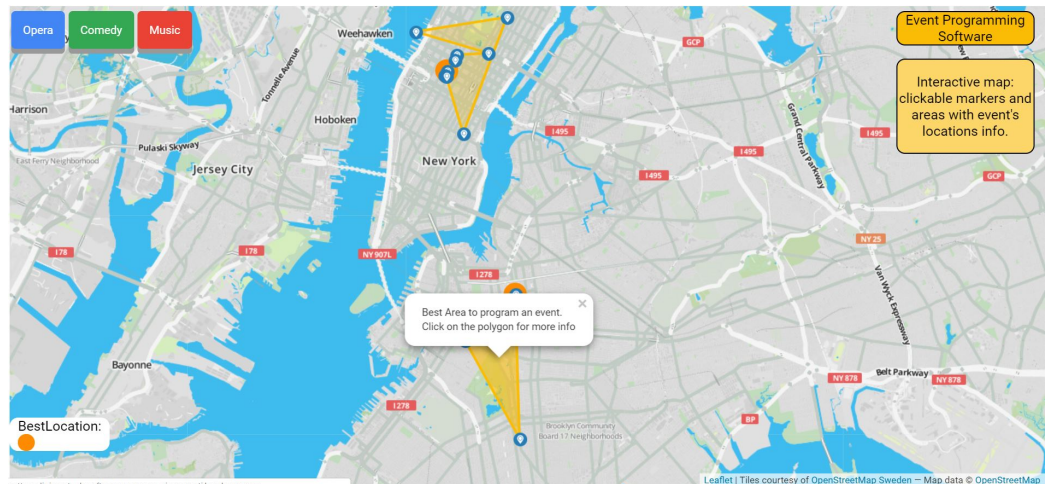


Figura 5.2: Zone evidenziate delimitate dai poligoni costituiti da markers rappresentanti luoghi. L'intensità è proporzionale alla positività media calcolata in base al Sentiment Analysis dei Tweet sulle locations.

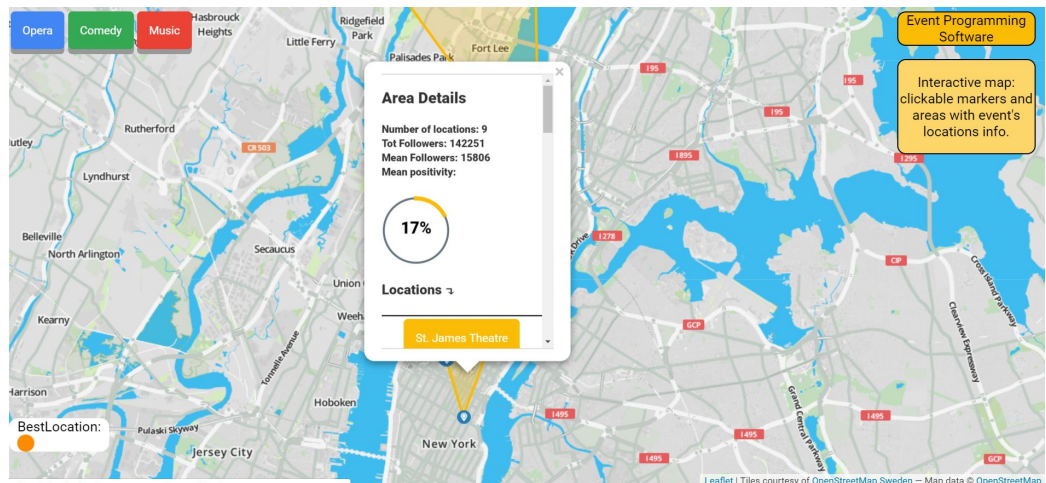


Figura 5.3: Pop-up di una zona. Contiene informazioni dettagliate e mostra la percentuale di positività media calcolata in base al Sentiment Analysis dei Tweet sulle locations. Vengono mostrate inoltre i luoghi che sono inclusi nello specifico poligono.

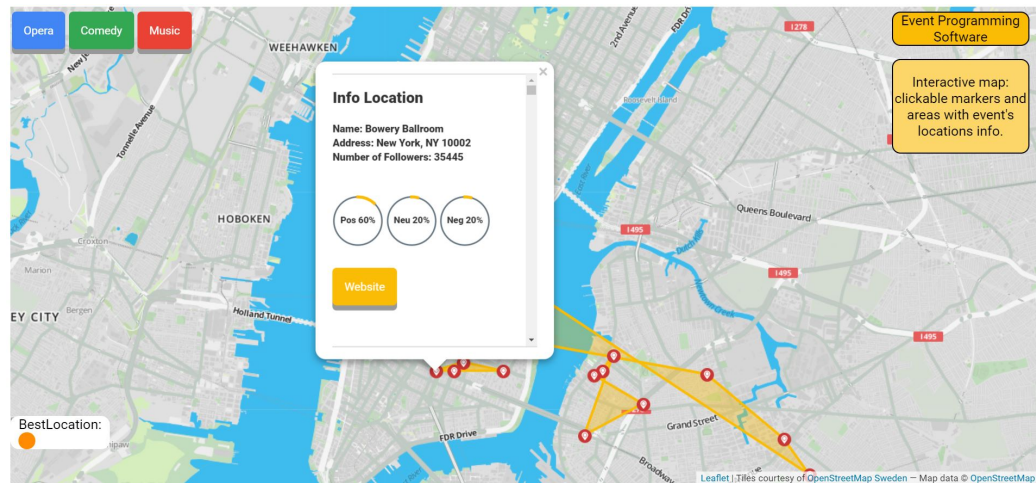


Figura 5.4: Pop-up di un marker rappresentante una location con determinate informazioni incluso il sito ufficiale.

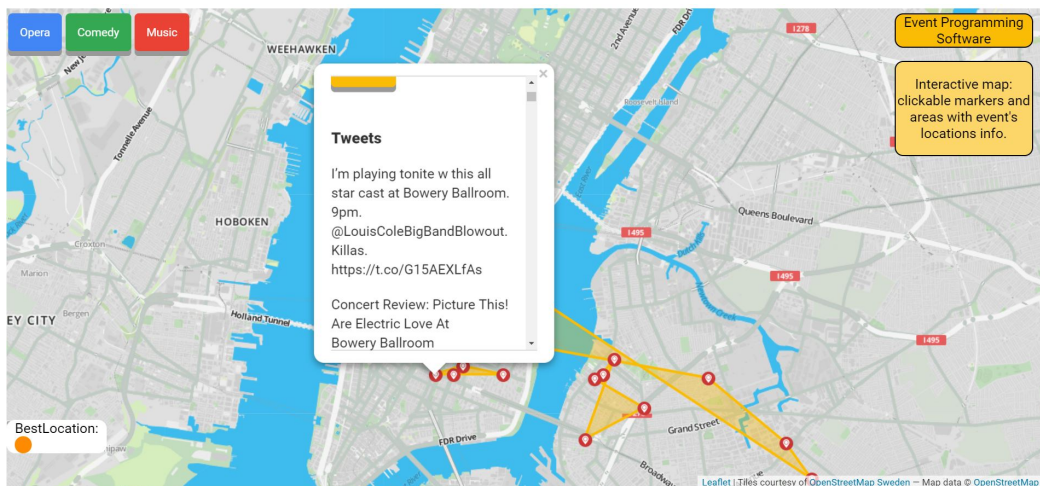


Figura 5.5: Pop-up del marker dinamico dopo il caricamento dei Tweets più recenti sulla location.

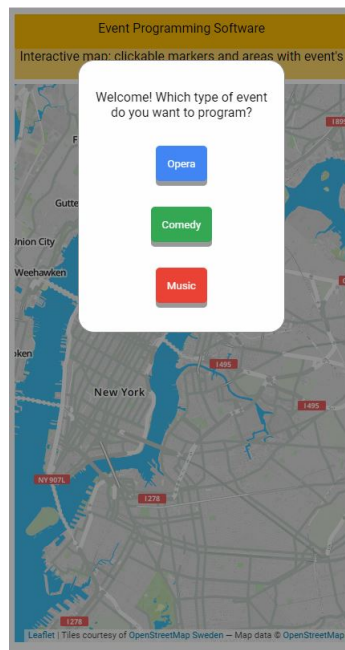


Figura 5.6: Schermata iniziale. Interfaccia introduttiva che guida l'utente alla scelta di una categoria in base all'evento che deve programmare. Mobile.

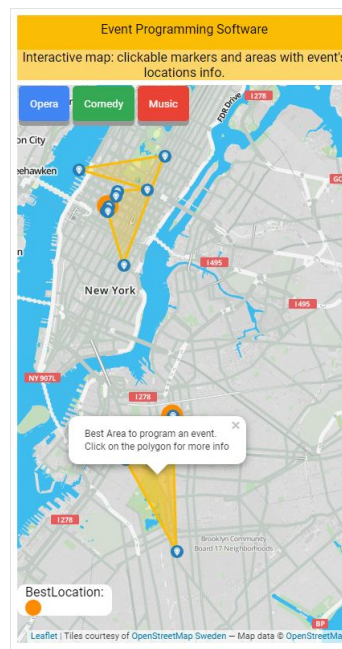


Figura 5.7: Zone evidenziate delimitate dai poligoni costituiti da markers rappresentanti luoghi. L'intensità è proporzionale alla positività media calcolata in base al Sentiment Analysis dei Tweet sulle locations. Mobile.

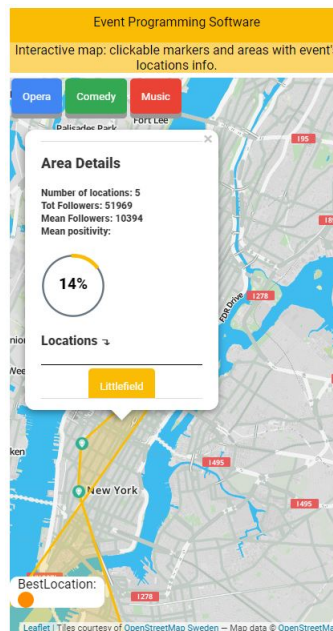


Figura 5.8: Pop-up di una zona. Contiene informazioni dettagliate e mostra la percentuale di positività media calcolata in base al Sentiment Analysis dei Tweet sulle locations. Vengono mostrate inoltre i luoghi che sono inclusi nello specifico poligono.

Mobile.

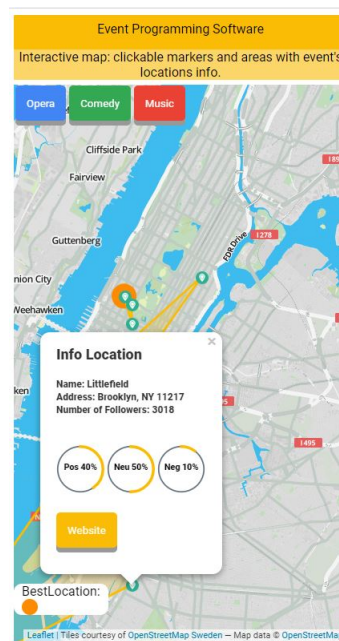


Figura 5.9: Pop-up di un marker rappresentante una location con determinate informazioni incluso il sito ufficiale,
Mobile

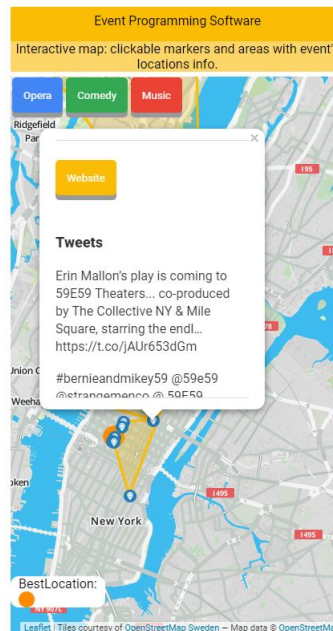


Figura 5.10: Pop-up del marker dinamico dopo il caricamento dei Tweets più recenti sulla location.
Mobile.

Capitolo 6

Deployment e link

6.1 Heroku

Heroku è una piattaforma Cloud che permette l'hosting sia di applicazioni web che server.

Sono riportati di seguito i link dell'applicazione web:

- Database Server
<https://locationdatabase.herokuapp.com/>
- Application Server
<https://twittermapserver.herokuapp.com/>
- Applicazione <https://softwareprogrammazioneeventi.herokuapp.com/>

6.2 GitHub

Il codice sorgente dei server e dell'applicazione è disponibile al seguente link:

- Applicazione <https://github.com/EnricoCollini/Software-Programmazione-Eventi>
- Application Server <https://github.com/EnricoCollini/Application-ServerTesi>
- Database Server <https://github.com/EnricoCollini/Database-ServerTesi>

Capitolo 7

Application Testing & Results

Il testing dell'applicazione è stato eseguito assegnando a determinati utenti il compito di scegliere un luogo in cui poter programmare un evento di una categoria specifica.

Nella prima fase è stata consegnata una lista di luoghi della categoria selezionata, questi attraverso una ricerca manuale hanno scelto una location per organizzare un evento della tassonomia scelta.

Nella seconda fase gli utilizzatori attraverso l'applicazione web hanno scelto un luogo adatto sempre per la categoria selezionata.

I risultati ottenuti hanno evidenziato due aspetti principali che sono stati presi in considerazione quando è stata utilizzata l'applicazione:

- La caratteristica di geolocalizzazione dei luoghi. In alcuni casi non è stata scelta una location con percentuale di positività massima perchè in una posizione geografica non favorevole. Questo risultato è più difficile da analizzare avendo a disposizione una lista di luoghi anziché una mappa interattiva.
- La possibilità di visualizzare una percentuale di positività, basata sul

seguito sociale di Twitter dei luoghi. Nella prima fase di testing gli utilizzatori non hanno potuto verificare direttamente come fosse il seguito sociale dei luoghi candidati in maniera intuitiva e diretta, mentre l'interfaccia del software ha permesso di considerare anche questo fattore.

Riportiamo i valori medi di positività ottenuti nella prima fase di testing, rispetto alla seconda.

	Test1 Perc Media Positività	Test2 Perc Media Positività
Opera	25.2%	67%
Music	29.2%	89%
Comedy	12%	70%

Dopo aver utilizzato l'applicazione gli utenti hanno controllato la loro prima scelta, e nel caso in cui la positività dell'applicazione è risultata bassa, hanno analizzato i tweet più recenti sul luogo ed il 73% degli interessati hanno così modificato la scelta iniziale.

User	Categoria	A: lista luoghi	B: Application	Scelta
1	M	Iridium Jazz Club	Blue Note Jazz Club	B
2	M	Music Hall of Williamsburg	Nublu	A
3	M	Tralf Music Hall	Blue Note Jazz Club	B
4	M	SOB's	Blue Note Jazz club	B
5	M	Brooklyn Steel	Bearsville Theater	B
6	O	Apollo Theater	Brooklin Comedy Festival	A
7	O	Lyric	St James Theater	B
8	O	Kings Theater	Brooklin Comedy Festiva	A
9	O	Cafe Carlyle	St James Theater	B
10	O	Towne Crier Cafe	St James Theater	B
11	C	St.George Theater	Alice Tully Hall	B
12	C	SVA Theater	The Paramount	A
13	C	Patchogue Theater	Alice Tully Hall	B
14	C	LittleField	Alice Tully Hall	B
15	C	Levity Live	Alice Tully Hall	B

Capitolo 8

Conclusione

L'obiettivo di questo elaborato di Tesi è la creazione di un'applicazione web fondata sull'utilizzo delle informazioni derivabili dall'analisi dei social network, nel nostro caso Twitter, e sui requisiti costituenti per la creazione di un'interfaccia che ottimizzasse l'usabilità del software.

L'interfaccia grafica intuitiva, interattiva e portatile consente di guidare l'utente verso lo scopo applicativo del software implementato: la scelta di un luogo favorevole ed ottimale in cui programmare un evento.

I criteri di scelta derivano, oltre che dalle specifiche inerenti alla tassonomia dell'evento, anche dall'analisi delle informazioni derivate da Twitter attraverso strumenti di Intelligenza Artificiale per effettuare il Sentiment Analysis di Tweet sulle Locations di eventi.

8.1 Possibili Sviluppi Futuri

Le possibilità di sviluppo future sono molteplici, in primis l'opportunità di estendere l'area geografica di applicazione del software.

Il contesto applicativo offre numerose possibilità di specializzazione per quanto riguarda le informazioni analizzabili dai social network per una possibile integrazione supplementare, come lo studio della distribuzione degli utenti interessati a particolari Locations.

Un altro possibile miglioramento consiste nell'interfacciamento con altre piattaforme social al fine di estrarre le informazioni derivabili da questi attraverso specifici strumenti di Intelligenza Artificiale.

Capitolo 9

Sitografia

9.1 Back End

- Node.js <https://nodejs.org/it/>
- Express.js <https://expressjs.com/it/>
- Twitter Developer <https://developer.twitter.com/en/docs.html>
- SeatGeek <https://seatgeek.com/>
- Aylien <https://aylien.com/>
- MongoDB <https://www.mongodb.com/>
- Mongoose <https://mongoosejs.com/>
- Postman <https://www.getpostman.com/>

9.2 Front End

- Leaflet <https://leafletjs.com/>

- IconFinder <https://www.iconfinder.com/>
- Dribbble <https://dribbble.com/>
- Flat Buttons inspired by <https://codepen.io/boelmanj/>
- Donut Chart inspired by <https://codepen.io/jdnierth/>

9.3 Deployment

- Heroku <https://www.heroku.com/>

9.4 Altro

- W3C Schools <https://www.w3schools.com/>
- StackOverflow <https://stackoverflow.com/>
- Wikipedia <https://it.wikipedia.org/wiki/Wikipedia>
- FreeCodeCamp <https://medium.freecodecamp.org/>
- Hootsuite We Are Social <https://digitalreport.wearesocial.com/>

Bibliografia

- [1] Ben Shneiderman, Catherine Plaisant. Designing the User Interface: Strategies for Effective Human-Computer Interaction. *Pearson Addison Wesley*, 2004
- [2] Enrico Cheli. La realtà mediata: l'influenza dei mass media tra persuasione e costruzione sociale della realtà. *FrancoAngeli*, 1997
- [3] Pak, Alexander; Paroubek, Patrick. Twitter as a Corpus for Sentiment Analysis and Opinion Mining. *Proceedings of LREC*, 2010