A Single Web Page Touristic Guide

Sviluppo Web Avanzato

Stefano Cortellessa, *254260*

A.A. 2018/2019

Indice

1. Introduzione

VisitAq è una guida turistica pensata per tutte quelle persone che si trovano nella città di L'Aquila come turisti e che vogliono conoscere in dettaglio questa città.
Il capoluogo di regione è ricco di *beni culturali e paesaggistici* di rara bellezza, ma spesso questi non sono così facili da trovare e visitare se non si conosce bene il territorio. Grazie a questa guida è possibile trovarli con molta facilità. In dettaglio, per poter utilizzare l'applicazione è necessario essere registrati e quindi autenticati. Una volta che quest'ultima è avvenuta viene mostrata una pagina di scelta che permette all'utente di esplorare o tutte le **attrazioni** (musei, castelli, monumenti, …) o tutti gli **eventi** (culinari, di spettacolo, …) presenti in città.
Solo dopo aver scelto vengono mostrati tutti gli/le eventi/attrazioni e cliccando su quello interessato viene aperta una nuova pagina di dettaglio dove sono mostrati una descrizione, un'immagine, la località ed una mappa che mostra precisamente il punto di interesse in modo da raggiungerlo facilmente.
L'applicazione è stata implementata in Java con l'utilizzo di JAX-RS, per il Back-End, mentre il Front-End è stato implementato come single page application con javascript, HTML5 e CSS3.
Nelle sezioni seguenti è possibile vedere quelli che sono i servizi implementati lato server (sezione 2), quella che è la struttura del Database (sezione 3) e le viste lato client dell'applicazione.

2. Risorse Implementate

In questa sezione sono descritte tutte le risorse implementate unitamente a quelle che sono le chiamate REST (url e verbo). Inoltre, è possibile riconoscere i servizi collegati al client in quanto sono contrassegnati dal simbolo *****.

2.1 Account

loginUser *:
URL: http://localhost:8080/visitaq/api/users/login
Verbo HTTP: **POST**
Header: Content-Type Application JSON
Output: Token dell'utente con status **201** se è stato correttamente inserito nel DB e **403** se no in quanto l'autenticazione non ha effetto.
Descrizione: Effettua il login di un utente
Payload:

```
{
"email": "email_utente",
"password": "password_utente"
}
```

*insertUser *:*
URL: http://localhost:8080/visitaq/api/users
Verbo HTTP: POST
Header: Content-Type Application JSON
Output: Status Created (codice 201) se l'utente è stato correttamente inserito e 403 se non è stato creato.
Descrizione: Registra un utente.
Payload:

{
"name": "name _utente",
"surname": "surname _utente",
"email": "email_utente",
"password": "password_utente"
}

*logout *:*
URL: http://localhost:8080/visitaq/api/users/logout/{token_utente}
Verbo HTTP: DELETE
Output: No Content codice 204 se l'utente ha effettuato correttamente il logout, 200 se la richiesta è andata a buon fine ma il token dell'utente non è stato correttamente cancellato.
Descrizione: Effettua il logout utente

deleteUser:
URL: http://localhost:8080/visitaq/api/users/{token_utente}/{id_utente}
Verbo HTTP: DELETE
Output: No Content codice 204 se l'utente è stato correttamente cancellato, 200 se la richiesta è andata a buon fine ma il l'utente non è stato correttamente cancellato oppure 401 se l'utente non è autorizzato a fare tale operazione.
Descrizione: Cancella l'utente con id uguale a {id_utente}

updateUser:
URL: http://localhost:8080/visitaq/api/users/{token_utente}/{id_utente}
Verbo HTTP: PUT
Header: Content-Type Application JSON
Output: No Content, codice 204, se l'utente è stato correttamente aggiornato, 200 se la richiesta è andata a buon fine ma non ha aggiornato correttamente l'utente, 401 se l'utente non è autorizzato e 400 se la richiesta è mal formata.
Descrizione: Aggiorna i dati di un utente
Payload:

{"name": "new_name _utente",
"surname": "new_surname _utente",
"email": "new_email_utente",
"password": "new_password_utente"}

getAllUsers:
URL: http://localhost:8080/visitaq/api/users/{token_utente}
Verbo HTTP: GET
Output: Status OK (codice 200) se la richiesta è andata buon fine, 401 se l'utente non è autorizzato a fare la richiesta.
Descrizione: Restituisce tutti gli utenti registrati

getUserByEmail:
URL: http://localhost:8080/visitaq/api/users/{token_utente}/email/{email_utente}
Verbo HTTP: GET
Output: Status OK (codice 200) se la richiesta è andata buon fine, 401 se l'utente non è autorizzato a fare la richiesta.
Descrizione: Restituisce tutti i dettagli dell'utente con e-mail uguale a {email}

2.2 Evento

getAllEvents:*
URL: http://localhost:8080/visitaq/api/users/{token_utente}/events
Verbo HTTP: GET
Output: Status OK (codice 200) se la richiesta è andata buon fine, 401 se l'utente non è autorizzato.
Descrizione: Restituisce tutti gli eventi salvati nel DB

getEventById:*
URL: http://localhost:8080/visitaq/api/users/{token_utente}/events/{id_evento}
Verbo HTTP: GET
Output: Status OK (codice 200) se la richiesta è andata buon fine, 401 se l'utente non è autorizzato.
Descrizione: Restituisce tutti i dettagli dell'evento con id uguale a {id_evento}

getAllEventsByCreator:
URL: http://localhost:8080/visitaq/api/users/{token_utente}/events/idCreator/{id_creator}
Verbo HTTP: GET
Output: Status OK (codice 200) se la richiesta è andata buon fine, 401 se l'utente non è autorizzato.
Descrizione: Restituisce tutti gli eventi che sono stati creati dall'utente con id uguale a {id_creator}

getAllEventsByCategory:
URL: http://localhost:8080/visitaq/api/users/{token_utente}/events/idCategory/{id_categoria}
Verbo HTTP: GET
Output: Status OK (codice 200) se la richiesta è andata buon fine, 401 se l'utente non è autorizzato.
Descrizione: Restituisce tutti gli eventi che hanno l'id della categoria uguale a {id_categoria}

updateEvent:
URL: http://localhost:8080/visitaq/api/users/{token_utente}/events/{id_evento}
Verbo HTTP: PUT
Header: Content-Type Application JSON
Output: No Content, codice 204, se l'evento è stato correttamente aggiornato, 200 se la richiesta è andata a buon fine ma non ha aggiornato correttamente l'evento, 401 se l'utente non è autorizzato e 400 se la richiesta è mal formata.
Descrizione: Aggiorna l'evento con id uguale a {id_evento}

Payload:

```
{
“title”: “titolo_evento”,
“locality”: “località_evento”,
“startDate”: “data_nizio_evento”,
“endDate”: “data_fine_evento”,
“lat”: “latitudine_evento”,
“lng”: “longitudine_evento”,
“description”: “descrizione_evento”,
“category”:
        {
            “name”: “nome_categoria”,
            “id”: id_categoria
        }
“creator”:
        {
            “id”: id_utente
        }

}
```

insertEvent:

URL: http://localhost:8080/visitaq/api/users/{token_utente}/events
Verbo HTTP: POST
Header: Content-Type Application JSON
Output: Status OK (codice 200) se la richiesta è andata a buon fine, 201 unitamente all’uri generato se l’oggetto è stato creato, 401 se l’utente non è autorizzato.
Descrizione: Inserisce un evento nel DB
Payload:

```
{
“title”: “titolo_evento”,
“locality”: “località_evento”,
“startDate”: “data_nizio_evento”,
“endDate”: “data_fine_evento”,
“lat”: “latitudine_evento”,
“lng”: “longitudine_evento”,
“description”: “descrizione_evento”,
“category”:
        {
            “name”: “nome_categoria”,
            “id”: id_categoria
        }
“creator”:
        {
            “id”: id_utente
        }

}
```

deleteEvent:
URL: http://localhost:8080/visitaq/api/users/{token_utente}/events/{id_evento}
Verbo HTTP: DELETE
Output: No Content codice 204 se l'evento è stato correttamente cancellato, 200 se la richiesta è andata a buon fine ma il l'evento non è stato correttamente cancellato oppure 401 se l'utente non è autorizzato a fare tale operazione.
Descrizione: Cancella l'evento avente id uguale a {id_evento}
Payload:

```
{
        "creator":
        {
                "id": {id_utente}
        }
}
```

2.3 Attrazione

getAllAttractions:*
URL: http://localhost:8080/visitaq/api/users/{token_utente}/attractions
Verbo HTTP: GET
Output: Status OK (codice 200) se la richiesta è andata buon fine, 401 se l'utente non è autorizzato.
Descrizione: Restituisce tutte le attrazioni salvate nel DB

getAttractiontById:*
URL: http://localhost:8080/visitaq/api/users/{token_utente}/attractions/{id_attrazione}
Verbo HTTP: GET
Output: Status OK (codice 200) se la richiesta è andata buon fine, 401 se l'utente non è autorizzato.
Descrizione: Restituisce tutti i dettagli dell'attrazione con id uguale a {id_attrazione}

getAllAttractionsByCreator:
URL: http://localhost:8080/visitaq/api/users/{token_utente}/attractions/idCreator/{id_utente}
Verbo HTTP: GET
Output: Status OK (codice 200) se la richiesta è andata buon fine, 401 se l'utente non è autorizzato.
Descrizione: Restituisce tutte le attrazioni che sono stati creati dall'utente con id uguale a {id_utente}

getAllAttractionsByCategory:
URL: http://localhost:8080/visitaq/api/users/{token_utente}/attractions/idCategory/{id_categoria}
Verbo HTTP: GET
Output: Status OK (codice 200) se la richiesta è andata buon fine, 401 se l'utente non è autorizzato.
Descrizione: Restituisce tutte le attrazioni che hanno l'id della categoria uguale a {id_categoria}

updateAttraction:
URL: http://localhost:8080/visitaq/api/users/{token_utente}/attractions/{id_attrazione}
Verbo HTTP: PUT
Header: Content-Type Application JSON

Output: No Content, codice 204, se l'attrazione è stato correttamente aggiornato, 200 se la richiesta è andata a buon fine ma non ha aggiornato correttamente l'attrazione, 401 se l'utente non è autorizzato e 400 se la richiesta è mal formata.
Descrizione: Aggiorna l'attrazione con id uguale a {id_attrazione}
Payload:

```
{
"title": "titolo_attrazione",
"locality": "località_attrazione",
"startDate": "data_inizio_ attrazione",
"endDate": "data_fine_ attrazione",
"lat": "latitudine_ attrazione",
"lng": "longitudine_ attrazione",
"description": "descrizione_ attrazione",
"image": "immagine_ attrazione",
"category":
        {
            "name": "nome_categoria",
            "id": id_categoria
        }
"creator":
        {
            "id": id_utente
        }

}
```

insertAttraction:
URL: http://localhost:8080/visitaq/api/users/{token_utente}/attractions
Verbo HTTP: POST
Header: Content-Type Application JSON
Output: Status OK (codice 200) se la richiesta è andata buon fine, 201 con l'uri generato se l'oggetto è stato creato e 401 se l'utente non è autorizzato ad eseguire l'operazione.
Descrizione: Inserisce un'attrazione nel DB
Payload:

```
{
"name": "nome_attrazione",
"locality": "località_ attrazione",
"startDate": "data_inizio_ attrazione",
"endDate": "data_fine_ attrazione",
"lat": "latitudine_ attrazione",
"lng": "longitudine_ attrazione",
"description": "descrizione_ attrazione",
"image": "immagine_ attrazione",
"category":
        {
            "name": "nome_categoria",
            "id": id_categoria
        }
```

```
"creator":
        {
            "id": id_utente
        }

}
```

deleteAttraction:
URL: http://localhost:8080/visitaq/api/users/{token_utente}/attractions/{id_attrazione}
Verbo HTTP: DELETE
Output: No Content codice 204 se l'attrazione è stata correttamente cancellato, 200 se la richiesta è andata a buon fine ma il l'attrazione non è stata correttamente cancellato oppure 401 se l'utente non è autorizzato a fare tale operazione.
Descrizione: Cancella l'attrazione avente id uguale a {id_attrazione}
Payload:

```
{
        "creator":
        {
                "id":{id_utente}
        }
}
```

3. Struttura Database

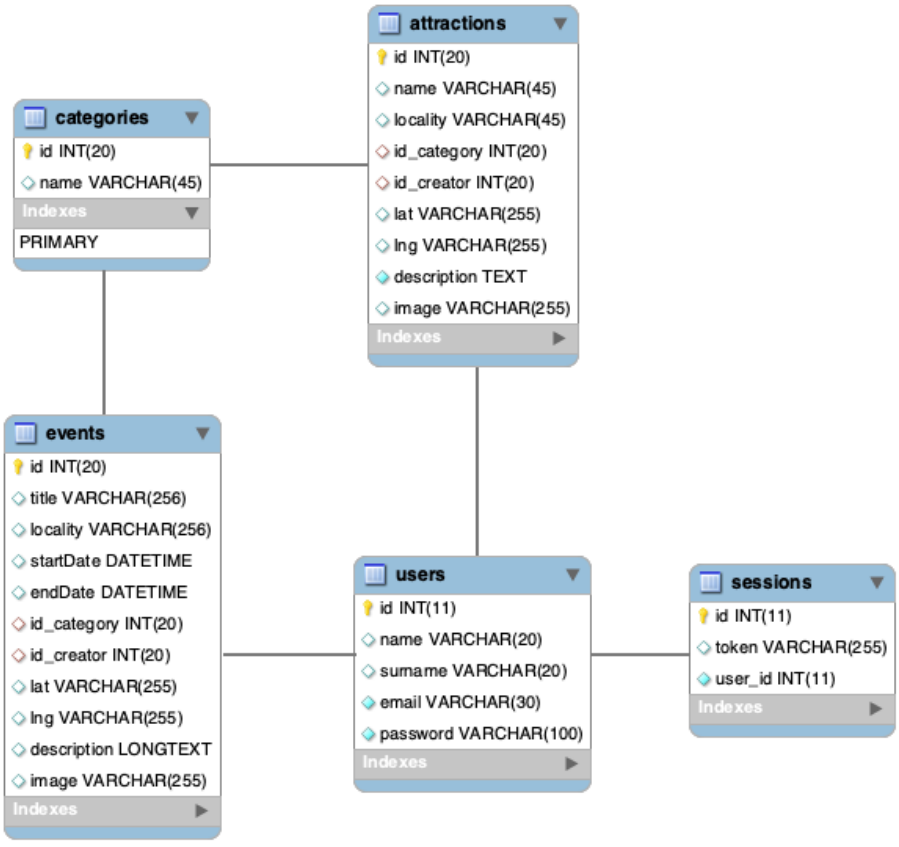


Figura 1: Database Schema

4. Vista

In questa sezione sono mostrate le varie viste implementate lato client dalle quali l'utente può effettuare le operazioni descritte precedentemente.

4.1 Login e Registrazione

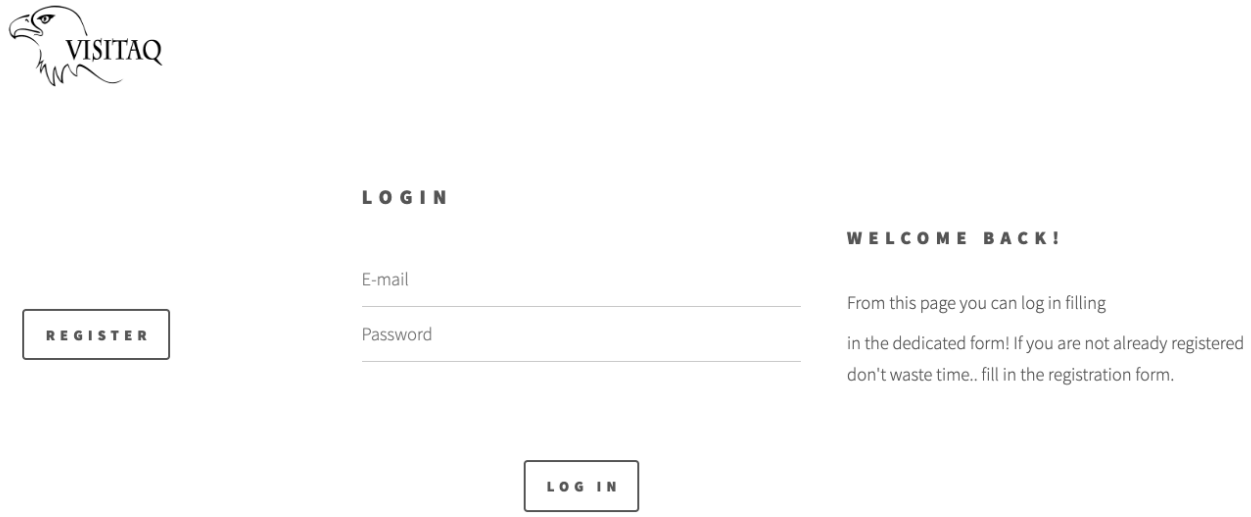


Figura 2: Form di Login

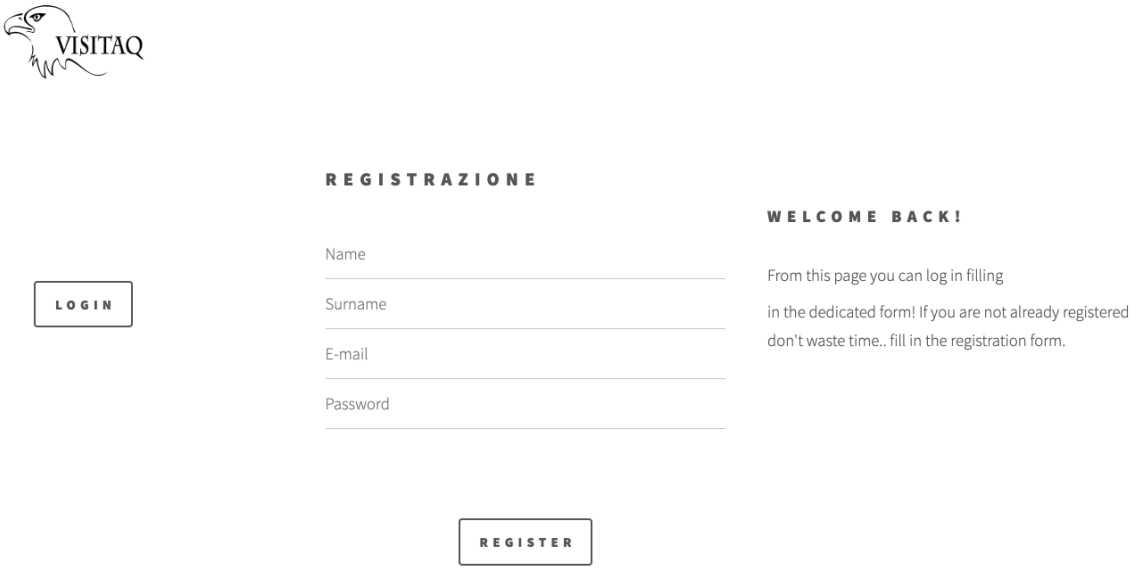


Figura 3: Form di Registrazione

4.2 Pagina di scelta

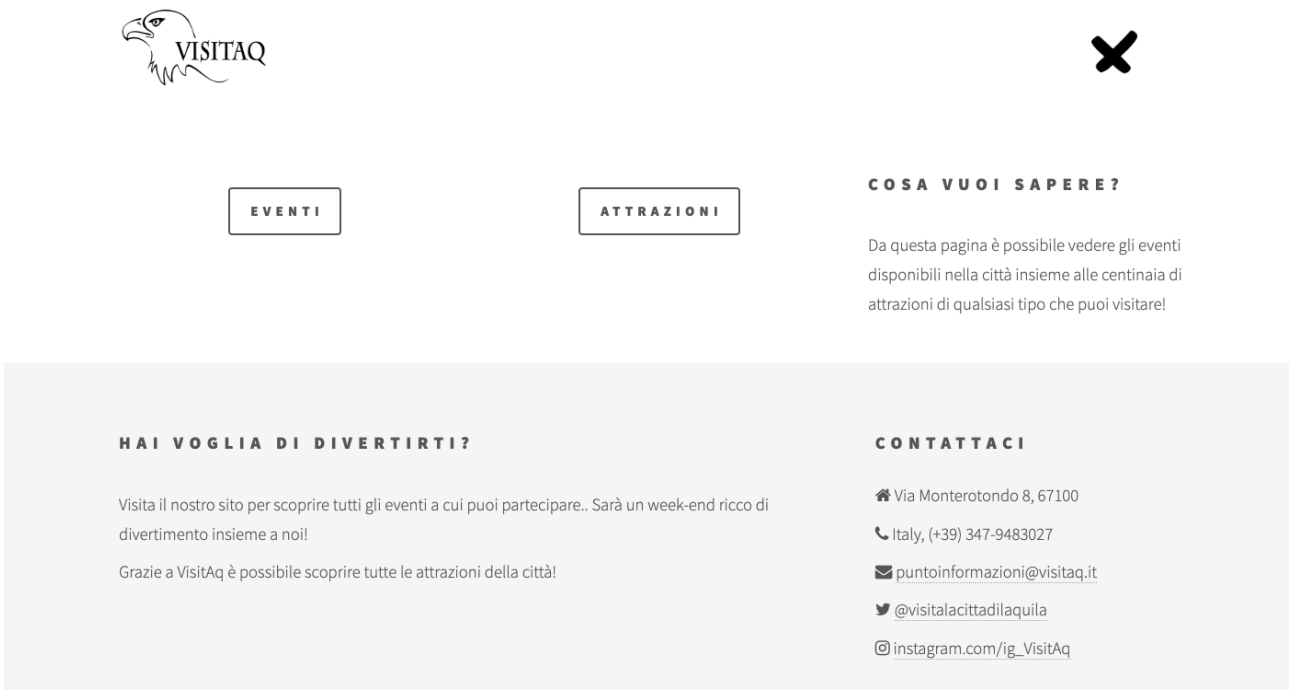
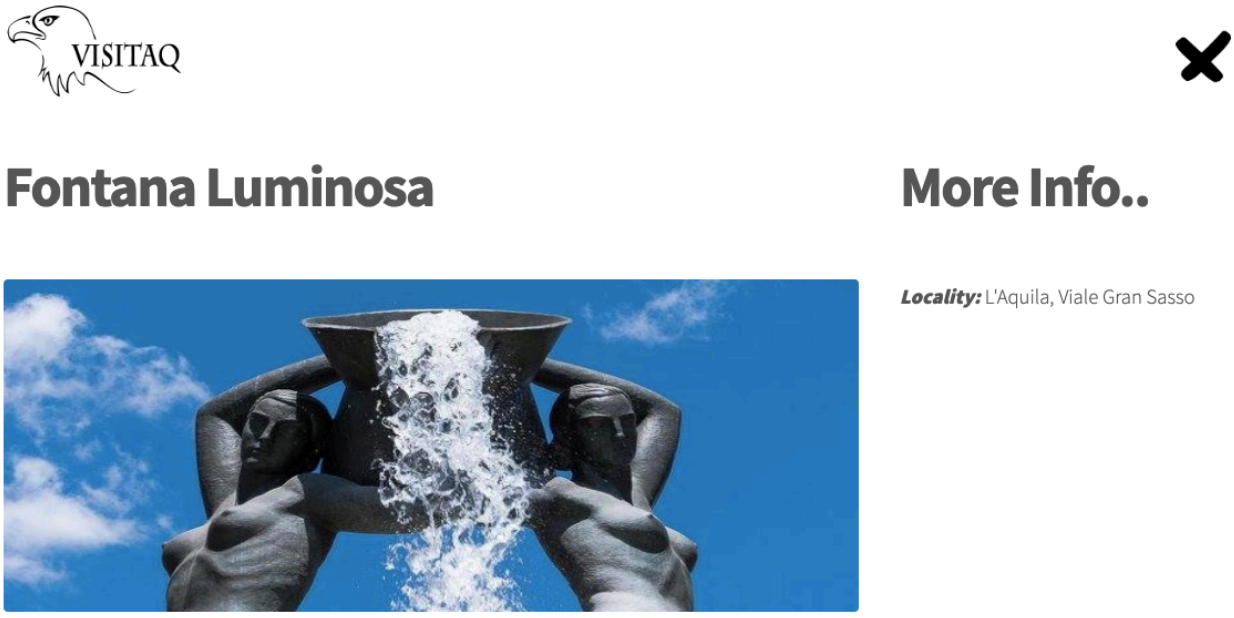


Figura 4: View dalla quale è possibile scegliere se vedere gli eventi disponibili o le attrazioni presenti nella città

4.3 Lista di Eventi/Attrazioni

Figura 5: View che mostra tutti gli eventi o le attrazioni disponibili

4.4 Pagina Dettagliata dell'Evento/Attrazione



HAI VOGLIA DI DIVERTIRTI?

Visita il nostro sito per scoprire tutti gli eventi a cui puoi partecipare.. Sarà un week-end ricco di divertimento insieme a noi!

Grazie a VisitAq è possibile scoprire tutte le attrazioni della città!

CONTATTACI

Via Monterotondo 8, 67100

Italy, (+39) 347-9483027

puntoinformazioni@visitaq.it

@visitalacittadilaquila

instagram.com/ig_VisitAq

Figura 6: View di una singola attrazione o un singolo evento