

# AEVENTI

Progetto di “Progettazione di Applicazioni Web e Mobili”

**Questa breve relazione è composta da:**

- Breve introduzione sull'idea del progetto e quali frameworks sono stati usati
- Descrizione di come è stato implementato il DB
- Descrizione di come è stato implementato il BackEnd
- Descrizione di come è stato implementato il FrontEnd
- Altre informazioni che aiutano a comprendere come è stato svolto il progetto.

LA PRIMA APP DEDICATA AGLI  
EVENTI

## Relazione Finale

**Matricola: 105328**

**Autore: Fabio Lori**

**Mail: fabio.lori@studenti.unicam.it**



**Scarica il progetto**



---

# Introduzione

AEventi è un'applicazione creata per condividere e visionare gli eventi come: Feste, Concerti, Inaugurazioni e Sagre.

L'obiettivo di AEventi è quello di dare l'opportunità di pubblicizzare gli eventi ad un bacino più ampio di quello che può essere la ristretta cerchia di amici su Facebook o su altri social.

Al momento chiunque può aggiungere un evento compilando un breve form dove verranno chieste le informazioni basilari come: Nome Evento, Breve Descrizione, Tipo e Data. Per ora è opportuno specificare la location dell'evento nella descrizione.

AEventi è stata sviluppata con le seguenti tecnologie: MySQL, SpringBoot, Ionic (usando Angular) di seguito descritte nel dettaglio.

## Database (MySQL)

Il database è stato implementato utilizzando MySQL: per il momento è composto dalle seguenti tabelle:

[Accounts](#): per Accounts si intendono gli utenti che si sono registrati con successo, un Account è composto dai seguenti parametri: id (chiave primaria auto-incrementante), email, password, name, surname.

[Events](#): per Events si intendono gli eventi che sono stati aggiunti con successo, un Event è composto dai seguenti parametri: id (chiave primaria auto-incrementante), name, description, time, publisher (chiave esterna collegata all'id della tabella [Accounts](#)), type.

**Note:** nel DB è presente una tabella non usata chiamata [EventType](#): in una versione provvisoria questa tabella veniva usata per contenere le tipologie degli eventi disponibili così da permettere ad un eventuale amministratore di aggiungere tipologie su richiesta senza andare a fare modifiche troppo invadenti e rispettando il principio Solid Open - Closed; questa tabella non è stata rimossa per permettere eventuali implementazioni future.

---

## Backend (SpringBoot)

Il Backend è stato implementato utilizzando Java con i framework SpringBoot e Maven.

Le classi sono divise in base alla loro funzione: infatti abbiamo 4 sotto-packages:

- [Models](#): composto dai modelli degli oggetti che vengono utilizzati, come ad esempio gli Account e gli Event
- [Repo](#): composto dalle interfacce che si occupano di dare la persistenza agli oggetti che verranno poi salvati sul Database.
- [Services](#): sono i servizi che si occupano di gestire le operazioni richieste dai Controllers.
- [Controllers](#): si occupano di “accettare” le richieste REST inviate dal Frontend e delegano i servizi dedicati.

**Note:** come nel DB sono presenti una serie di classi dedicate alla gestione degli oggetti [EventType](#).

## FrontEnd (Ionic 5- AngularTs)

Il FrontEnd è stato implementato utilizzando TypeScript – Html – Css con i framework Ionic e Angular.

I files sono divisi in packages in base alla loro funzione, in particolare prestiamo attenzione al package “pages”, all’interno di questo package si trovano le pagine accessibili in AEventi:

- [Feed](#): è la pagina in cui vengono visualizzati gli eventi, quest’ultimi vengono presi dal Backend nel metodo OnInit() -> (che a sua volta richiama il [feed-service](#) che si occupa di fare una richiesta get al Backend) e viene rieseguito ogni volta che la pagina viene ricaricata. Inoltre è presente un bottone fluttuante che permette di aggiungere un evento spostandosi quindi nella pagina [new-event](#).
- [Homepage](#): è la pagina di benvenuto che viene mostrata appena si apre l’app.
- [Login](#): è la pagina che permette agli utenti iscritti di accedere al loro account, il login vero e proprio viene effettuato nel [login-service](#).
- [New-Event](#): è la pagina che permette di aggiungere un evento, l’aggiunta dell’evento avviene tramite il [new-event-service](#) che invia i dati al Backend tramite una richiesta post.
- [Notifications](#): è la pagina accessibile da qualunque altra pagina dell’app che permetterà agli utenti loggati di vedere le notifiche ricevute.

- 
- [Register](#): è la pagina che permette di creare un account, l'aggiunta dell'account avviene tramite il [register-service](#) che invia i dati al Backend tramite una richiesta post.
  - [Settings](#): è la pagina che permetterà di modificare le impostazioni dell'app, al momento si limita a permettere la creazione di un account o di fare il login.
  - [Success](#): è la pagina che viene visualizzata dopo la creazione di un account, composta da 3 slide dà il benvenuto ad ogni utente appena registrato.

**Note:** i nomi delle cartelle in cui sono situate le pagine sono differenti dal titolo visualizzato nell'interfaccia web a causa della traduzione in italiano effettuata nell'ultimo periodo: pertanto per rendere il codice più chiaro in quest'ultimo ho mantenuto la lingua inglese.

#### Probabili implementazioni:

- Si possono creare varie tipologie di Account, così da separare i "creatori di eventi" da coloro che invece sono interessati solo a partecipare.
- Si possono implementare le notifiche push.
- Si può aggiungere l'autenticazione tramite qualche servizio apposito più sicuro.
- Si può aggiungere la possibilità di selezionare la location dell'evento sulla mappa e di visualizzare quest'ultimo solo agli utenti nelle vicinanze.