



Overview

# Pok, Esame di Progettazione Applicazioni Web e Mobili

Filippo Biondi

Dicembre 2020

## Contenuti

<b>1</b>	<b>Introduzione</b>	<b>1</b>
<b>2</b>	<b>Descrizione applicativo</b>	<b>1</b>
<b>3</b>	<b>Dettagli di Implementazione</b>	<b>1</b>
3.1	Frontend . . . . .	2
3.1.1	Interfaccia utente . . . . .	2
3.2	Autenticazione . . . . .	3
3.3	Backend . . . . .	3
3.4	Infrastruttura . . . . .	3

## 1 Introduzione

Questa relazione analizza gli aspetti di implementazione dell'applicazione **Pok** realizzata per l'esame di **Progettazione Applicazioni Web e Mobili**

## 2 Descrizione applicativo

Pok è un'applicazione mobile che permette di creare **Post** anonimi legati alle coordinate geografiche dell'utente, i Post sono poi accessibili e commentabili anonimamente solo da utenti nei pressi delle coordinate in cui è stato creato il Post

## 3 Dettagli di Implementazione

Questa sezione descrive i pattern e le tecnologie adottate per l'implementazione dell'applicativo

### 3.1 Frontend

Il frontend di Pok è stato realizzato con l'sdk **Flutter**, utilizzando il pattern **BLoC** (**B**usiness **L**ogic **C**omponent) che permette di gestire lo stato della UI attraverso dei componenti middleware chiamati BLoC che si pongono tra l'interfaccia utente e il codice per le chiamate all'API, essi comunicano i cambiamenti di stato alla UI attraverso uno **Stream** di eventi.

#### 3.1.1 Interfaccia utente

L'interfaccia utente presenta 2 viste principali oltre a quella per l'autenticazione, una per la ricerca di post nelle vicinanze e una per la creazione di nuovi post

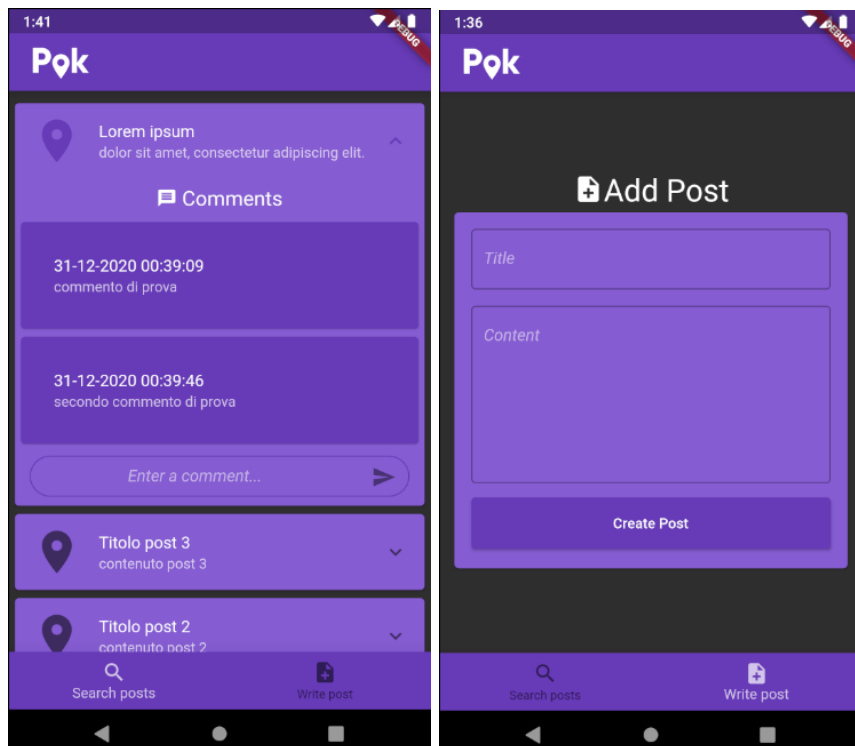


Figure 1: (a) Ricerca post nelle vicinanze e commenti (b) Creazione nuovi post

### 3.2 Autenticazione

L'autenticazione all'api è stata gestita utilizzando il servizio **Firebase**, con supporto all'autenticazione tramite account google.

### 3.3 Backend

Il server di backend è stato realizzato in **Rust** utilizzando il framework **Actix web** per la creazione della Rest API HTTP e la libreria **Diesel** per interfacciarsi al database **Postgresql**. Non esistendo una release della libreria **Firebase Admin SDK** per il linguaggio Rust la verifica dei Token ID di Firebase per l'autenticazione è stata effettuata seguendo le linee guida di Google [1] utilizzando la libreria **jsonwebtoken** per la validazione di token jwt.

### 3.4 Infrastruttura

L'infrastruttura di Pok si compone di un server **VPS** che ospita una sottorete di container **Docker**, essa comprende un container per il server postgresql e un container per il server di backend, un proxy **NGINX** agisce da gateway e rende accessibile l'api del backend esternamente tramite **HTTPS**

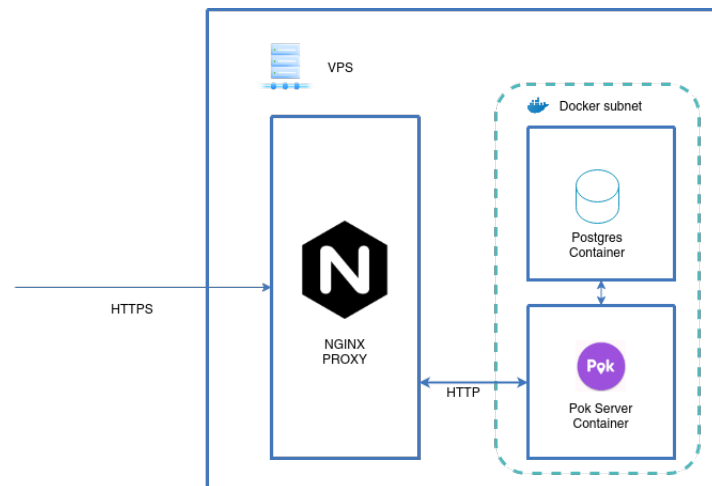


Figure 2: Rappresentazione grafica dell'infrastruttura di Pok

## References

- [1] Firebase: Verify ID Tokens  
<https://firebase.google.com/docs/auth/admin/verify-id-tokens>