

1 Indice

1. Introduzione
2. Funzionalità implementate
3. Scelte UI
4. Scelte di implementazione

2 Introduzione

2.1 Scelta del progetto

Abbiamo scelto il progetto dei Pokémon in quanto conoscitori dell'ambiente e possibili utilizzatori di un'eventuale applicazione terminata.

2.2 Metodologia di lavoro

Essendo lontani gli uni dagli altri, abbiamo sfruttato largamente strumenti come la chat vocale, la condivisione dello schermo e GitHub. Abbiamo collaborato molto sia nella scrittura del codice, sia nelle scelte grafiche, e molto raramente una parte dell'applicazione è stata sviluppata da un singolo elemento.

3 Funzionalità implementate

Abbiamo deciso di strutturare il progetto come una nostra implementazione del Pokédex.

Alcune delle nostre scelte implementative sono quindi state:

- All'apertura viene mostrata una schermata con la **lista di tutti i Pokémon**, con informazioni basilari come nome, ID e tipo del Pokémon. *[Per “tutti i Pokémon” abbiamo inteso i Pokémon presenti nell'API nel momento dello sviluppo della nostra applicazione, quindi tutti i Pokémon precedenti a Pokémon Spada e Scudo, escluse le forme alternative, come le forme di altre regioni e le mega evoluzioni.]*
- Si possono **cercare i Pokémon tramite nome**, anche non completo, e **tramite ID**. Abbiamo anche inserito un **filtro per tipi**, e la possibilità di muoversi velocemente attraverso la lista, tramite **una barra di scorrimento laterale**, visto l'alto numero di elementi nella lista.
- Gli **elementi della lista sono espandibili e mostrano informazioni approfondite** sul Pokémon in questione. In particolare abbiamo deciso di mostrare:
 - Delle **informazioni generali** sul Pokémon, ad esempio:
 - * *Flavor-text*, ovvero un piccolo testo descrittivo sul Pokémon
 - * Tasso di cattura e di genere
 - * Habitat
 - Le **statistiche** del Pokémon, come HP e attacco
 - Le **abilità** che il Pokémon può avere, a loro volta espandibili
 - Le **mosse** che il Pokémon può imparare, a loro volta espandibili
 - La sua **catena evolutiva**
- I Pokémon possono essere **aggiunti ai preferiti**, e salvati in locale, per essere più facilmente raggiungibili in un'apposita schermata. L'aggiunta ai preferiti si può effettuare sia dalla lista del Pokédex, sia dalla pagina dei suoi dettagli.
- È presente un minigioco liberamente ispirato al famoso intermezzo delle puntate del cartone animato dei Pokémon **“Who's that Pokémon?”**. Questo consiste nell'indovinare il nome del Pokémon data la sua sagoma oscurata.

4 Scelte UI

Abbiamo sfruttato largamente ciò che avevamo appreso con l'esonero di Material Design, utilizzando ad esempio:

- Le CardView per i Pokémon nella vista del Pokédex
- I Floating Button per la ricerca e per i filtri
- I Bottom Sheet Dialog per mostrare informazioni
- Il supporto per il tema scuro
- Icone di Material Design

Abbiamo inoltre curato molto la scelta dei colori dei vari elementi della UI. Una scelta su cui abbiamo investito molto è stata quella di rendere la CardView nella vista del Pokédex del colore medio del Pokémon, calcolato a partire dalla sua immagine e in seguito desaturato. Le carte sono inoltre disposte diversamente in base all'orientamento dello schermo, per migliorare la leggibilità (ciò viene fatto tramite un GridLayout).

Inoltre abbiamo cercato di utilizzare il più possibile elementi grafici per rappresentare i dati, come le barre di progressione per le statistiche del Pokémon.

Per migliorare la navigabilità, nella activity di dettaglio è stato aggiunto anche un bottone per tornare alla home della applicazione, questo perchè l'utente può aprire un gran numero di activity di dettaglio dalla linea evolutiva.

5 Scelte implementative

Di seguito riportiamo le più significative scelte implementative che abbiamo fatto.

5.1 API

L'API fornita è molto completa e ricca di informazioni, anche se difficile da localizzare in altre lingue. Allo stesso tempo però ritorna spesso oggetti molto complessi, come ad esempio le catene evolutive, o incompleti, e che necessitano quindi di più richieste per ottenere tutte le informazioni. Inoltre è presente una limitazione a 100 richieste/minuto che ci ha imposto di trovare soluzioni alternative e di utilizzare ampiamente la persistenza. Le informazioni dall'API sono state ottenute tramite le Volley, e durante lo sviluppo abbiamo sempre cercato di diminuire il numero delle richieste.

5.2 Inizializzazione dell'applicazione

Vista la limitazione del numero delle chiamate dell'API e dell'alto numero dei Pokèmon da ottenere, al momento dello sviluppo 807, abbiamo dovuto trovare alcune soluzioni alternative.

Volevamo infatti che già al primo avvio dell'applicazione, il Pokédex fosse consultabile per intero. Ciò ci imponeva di avere quanto meno le informazioni basilari per rappresentare un Pokémon, quindi nome, ID, immagine e in seguito abbiamo anche aggiunto il tipo.

Per il download di tutte le immagini dei Pokémon abbiamo deciso di hostare sulla nostra repository di GitHub un file .zip delle dimensioni circa di circa 15MB, da noi prodotto e contenente tutte le immagini, così da poter in qualsiasi momento cambiare il set di immagini e di ridurre le richieste di download da più di 800 ad 1. Ciò è stato fatto non tanto per la limitazione delle chiamate delle API, ma piuttosto per diminuire il numero di download da gestire in contemporanea.

Per ottenere invece il nome e l'ID del Pokémon abbiamo sfruttato una chiamata dell'API che ritorna appunto questi due dati per tutti i Pokémon presenti.

Infine per ottenere il tipo, o i tipi, del Pokémon abbiamo fatto una chiamata per ogni tipo presente, quindi 18 chiamate. Con queste chiamate abbiamo anche ottenuto l'intera lista delle mosse e il loro relativo tipo, ottimizzando il numero di richieste. Per completezza abbiamo fatto ulteriori 3 chiamate sul tipo di danno della mossa, avendo così un modo per rappresentarle facilmente ed in maniera completa.

Tutte queste informazioni vengono poi inserite nel nostro database e permettono un completo e fluido utilizzo della vista del Pokédex, già dal primo avvio.

5.3 Rappresentazione dettagli del Pokémon

Alcuni elementi nel dettaglio del Pokémon, come le statistiche o il flavor-text, sono stati abbastanza elementari da rappresentare.

A differenza di questi, le abilità, le mosse e la catena evolutiva hanno richiesto una maggiore attenzione.

Le abilità, visto che un Pokémon ne presenta fino ad un massimo di 3, sono riportate direttamente nel dettaglio e sono cliccabili. Il click comporta la comparsa di un Bottom Sheet Dialog che ne riporta i dettagli, i quali vengono scaricati tramite chiamata all'API.

Per le mosse, visto che un Pokémon ne può presentare nell'ordine delle molte decine, sono state raccolte in una View, espandibile tramite il click di un bottone, e sono rappresentate come una lista di CardView. Al tap la CardView si apre e mostra i dettagli della mossa, che vengono ottenuti tramite chiamata all'API.

La catena evolutiva è stata invece decisamente più difficile da gestire, vista la grande variabilità dei casi possibili. Infatti alcuni Pokémon presentano catene evolutive abbastanza complesse. *[Si passa infatti da Pokémon che non presentano nessuna evoluzione, come Mew, a Pokémon con multiple evoluzioni o diramazioni, come Eevee e Ralts, passando per Pokémon con catene evolutive semplici, come Bulbasaur].*

Allo stesso tempo, il meccanismo di evoluzione dei Pokémon, può variare molto in complessità. *[Si passa da Pokémon che si evolvono per il semplice aumento di livello, come Ivysaur, a Pokémon che richiedono multiple condizioni, come Sylveon].*

Un'ulteriore difficoltà era dovuta al difficile parsing delle informazioni ritornate dall'API, e al fatto che per ottenere le informazioni andava fatta una chiamata aggiuntiva. Il parsing, molto complesso, è stato fatto in maniera da poter ottenere le informazioni a noi utili ed inserirle in una classe che ci permettesse di utilizzarle in maniera semplice.

Ottenute queste informazioni la linea evolutiva viene rappresentata come insieme di CardView dei Pokémon, con relativo metodo di evoluzione e cliccabili per mostrarne i dettagli.

5.4 Database

Il database viene riempito delle informazioni fondamentali per rappresentare il Pokédex nella fase di inizializzazione al primo avvio dell'applicazione.

Quando poi per un Pokémon si apre la scheda di dettaglio o viene aggiunto ai preferiti, se tutti i suoi dati di dettaglio non sono presenti nel nostro database, viene fatta una richiesta all'API e vengono salvati i dati nel database. In questo modo si usa la persistenza per diminuire il numero di chiamate all'API e contemporaneamente migliorare le prestazioni dell'applicazione, in quanto accedere ai dati dal database è più veloce che aspettare la risposta

dell'API.

Vengono salvate tutte le informazioni per poter rappresentare il dettaglio del Pokémon, compresa la lista delle sue mosse e delle abilità, ma non i dettagli di queste ultime, in quanto richiederebbero ulteriori richieste all'API.

5.5 Gestione dei preferiti

Avendo deciso di implementare in maniera così estensiva la persistenza, per implementare la funzionalità di aggiungere i Pokémon ai preferiti è bastato aggiungere un campo che fungeva da flag per indicare se il Pokémon era o meno tra i preferiti.

Un Pokémon può essere aggiunto o rimosso dai preferiti sia nella vista Pokédex, sia nella vista del dettaglio del Pokémon, tramite l'apposita icona a forma di stella. È possibile anche aggiungere più Pokémon alla volta utilizzando un sistema di scelta multipla, che si attiva con un long click nella vista Pokédex. Abbiamo però deciso di limitare il numero di Pokémon selezionabili in una singola volta, per non creare situazioni in cui l'utente selezioni un numero di Pokémon troppo elevato che esaurisca le chiamate all'API.

È presente anche un fragment apposita per la visualizzazione dei Pokémon preferiti, facilmente accessibile dal menu a comparsa laterale.

5.6 Who's that Pokémon?

Abbiamo deciso di implementare questa funzionalità aggiuntiva come tributo a l'intermezzo del cartone animato e per offrire la possibilità di utilizzare l'applicazione in maniera più ludica.

Per come è stata implementato il database, questa funzionalità non richiede nessuna chiamata aggiuntiva all'API e inoltre non richiede il salvataggio della sagoma del Pokémon, in quanto questa viene prodotta a runtime partendo dall'immagine che scarichiamo nella fase di inizializzazione.

Abbiamo deciso anche di implementare la parte audio di questa funzionalità, in quanto elemento cardine e molto facilmente riconoscibile.

5.7 Fragments e LiveData

Sia la lista visualizzata nel Pokédex che quella dei Pokémon preferiti si trovano in appositi ViewModel, al fine di sopravvivere ad eventi catastrofici come la rotazione del dispositivo. Queste liste vengono modificate dalla classe che gestisce l'accesso al database e sono osservate dai Fragments corrispondenti.

L'utilizzo dei ViewModel permette inoltre ai Fragments di comunicare tra

loro.

Avendo optato per l'utilizzo di un `NavigationDrawer`, la maggior parte delle funzionalità offerte dall'applicazione sono situate all'interno di appositi `Fragments`. La `PokedexActivity` ha il compito di coordinare i diversi `Fragments`, di sostituirli e di salvare i loro stati. Il salvataggio, in particolare, viene portato a termine con l'aiuto di una classe aggiuntiva: la `FragmentManager`. In questa, gli stati dei `Fragments` sono archiviati in una `HashMap` che ha come chiave l'id del `Fragment` corrispondente. La stessa `PokedexActivity` tiene traccia dei `Fragments` da lei creati con una propria `HashMap` dal comportamento simile. Prima di sostituire un `Fragment`, il suo stato attuale viene salvato, per poi essere ripristinato quando dovrà essere visualizzato nuovamente. I vari stati possono essere salvati in un `Bundle` per sopravvivere alla rotazione della `PokedexActivity`.