

# UniQuiz

**Luca Rossini, Lorenzo Raia, Ilaria Salami**

Anno Accademico 2024/2025



# Indice

|          |                                     |           |
|----------|-------------------------------------|-----------|
| <b>1</b> | <b>Introduzione</b>                 | <b>3</b>  |
| <b>2</b> | <b>Tecnologie utilizzate</b>        | <b>3</b>  |
| <b>3</b> | <b>Design</b>                       | <b>3</b>  |
| 3.1      | Progettazione e sketch UI . . . . . | 3         |
| 3.2      | Material Design 3 . . . . .         | 4         |
| <b>4</b> | <b>Architettura</b>                 | <b>5</b>  |
| <b>5</b> | <b>Funzionalità</b>                 | <b>6</b>  |
| 5.1      | Autenticazione . . . . .            | 6         |
| 5.1.1    | Login . . . . .                     | 6         |
| 5.1.2    | Registrazione . . . . .             | 7         |
| 5.2      | Home . . . . .                      | 7         |
| 5.3      | Visualizzazione corso . . . . .     | 8         |
| 5.4      | Esecuzione quiz . . . . .           | 9         |
| 5.5      | Creazione quiz . . . . .            | 10        |
| 5.6      | Progressi . . . . .                 | 11        |
| 5.7      | Riepilogo . . . . .                 | 12        |
| 5.8      | Profilo . . . . .                   | 13        |
| <b>6</b> | <b>Sviluppi futuri</b>              | <b>14</b> |
| <b>7</b> | <b>Considerazioni finali</b>        | <b>15</b> |

# 1 Introduzione

Il progetto ha come scopo la realizzazione di un'applicazione mobile Android dedicata agli studenti dell'Università degli Studi Milano - Bicocca che hanno la necessità di studiare e ripassare gli argomenti di un corso. Questa applicazione permette l'esecuzione di quiz creati dagli utenti stessi oppure generati direttamente dall'Intelligenza Artificiale, in maniera semplice e intuitiva. Garantisce inoltre la sincronizzazione dei propri progressi su tutti i propri dispositivi, e garantisce il suo funzionamento anche in modalità offline (con funzionalità limitate).

## 2 Tecnologie utilizzate

- **Android Studio:** IDE utilizzato per lo sviluppo di applicazioni Android.
- **GitHub:** Piattaforma online per la gestione e la condivisione del codice. Permette agli sviluppatori di collaborare, versionare e archiviare i loro progetti utilizzando Git, un sistema di controllo versione.
- **Firebase:** Piattaforma di sviluppo di Google che fornisce strumenti per creare app web e mobile in modo semplice e veloce. In particolare abbiamo utilizzato le seguenti funzionalità:
  - **Authentication**
  - **Firestore Database**
- **OpenAI API:** API esterna relativa al servizio ChatGPT utilizzata per la generazione automatica dei quiz, tramite il modello gpt-4o-mini.

## 3 Design

### 3.1 Progettazione e sketch UI

Per la progettazione iniziale dell'interfaccia utente abbiamo utilizzato un tool chiamato Figma, creando dei prototipi per ogni interfaccia dell'app.

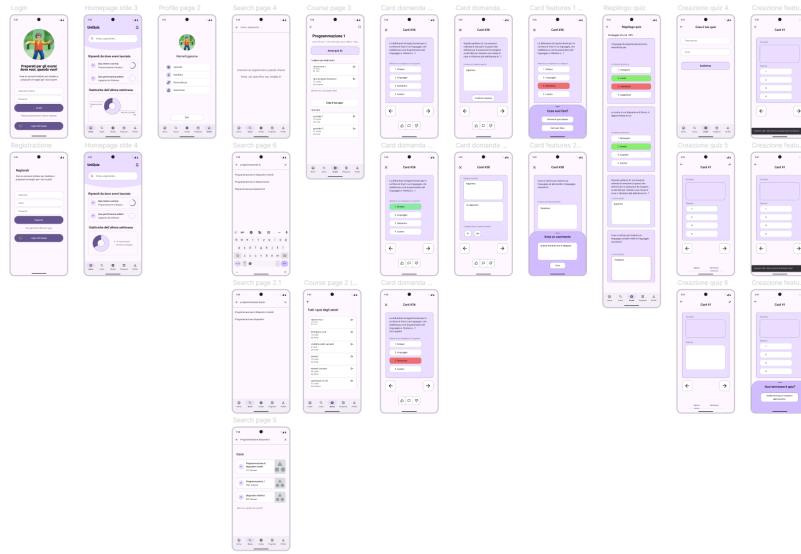


Figure 1: Prototipi su Figma

### 3.2 Material Design 3

Per la realizzazione dell'interfaccia grafica abbiamo seguito tutte le linee guida dello stile Material Design 3 utilizzando i componenti grafici disponibili. Inoltre, abbiamo implementato un tema generale realizzato con *Material Theme Builder* che permette anche la transizione da Light Mode e Dark mode.

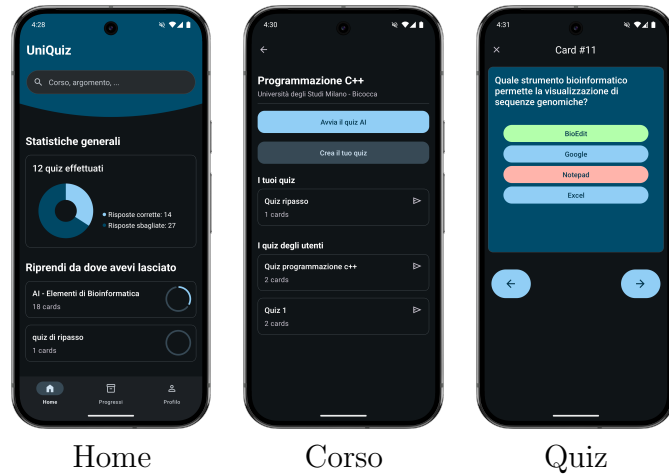


Figure 2: Interfaccia in Dark mode

## 4 Architettura

Per l'applicazione abbiamo scelto di utilizzare il pattern architetturale Model-View-Viewmodel (MVVM):

- **Model:** rappresenta i dati e la logica di business. Nell'applicazione è rappresentato da tutte le classi che gestiscono i dati veri e propri, e le classi utilizzate per l'ottenimento e la creazione dei dati stessi (Repository, Source, Service, Database, Synchronization)
- **View:** la parte dell'app che si occupa dell'interfaccia utente. Nell'applicazione è rappresentata dall'insieme di tutte i Fragment e Activity che visualizzano i dati e gestiscono l'interazione con l'utente.
- **Viewmodel:** la parte che gestisce la logica di presentazione, raccoglie i dati dal Model e li prepara per la visualizzazione nella View. Nell'applicazione è rappresentato dalle classi UserViewModel, QuizViewModel e CourseViewModel che comunicano attraverso i MutableLiveData.

Per l'implementazione dell'app inoltre abbiamo utilizzato diversi pattern:

- **Observer**
- **Factory**
- **Adapter**
- **Repository**

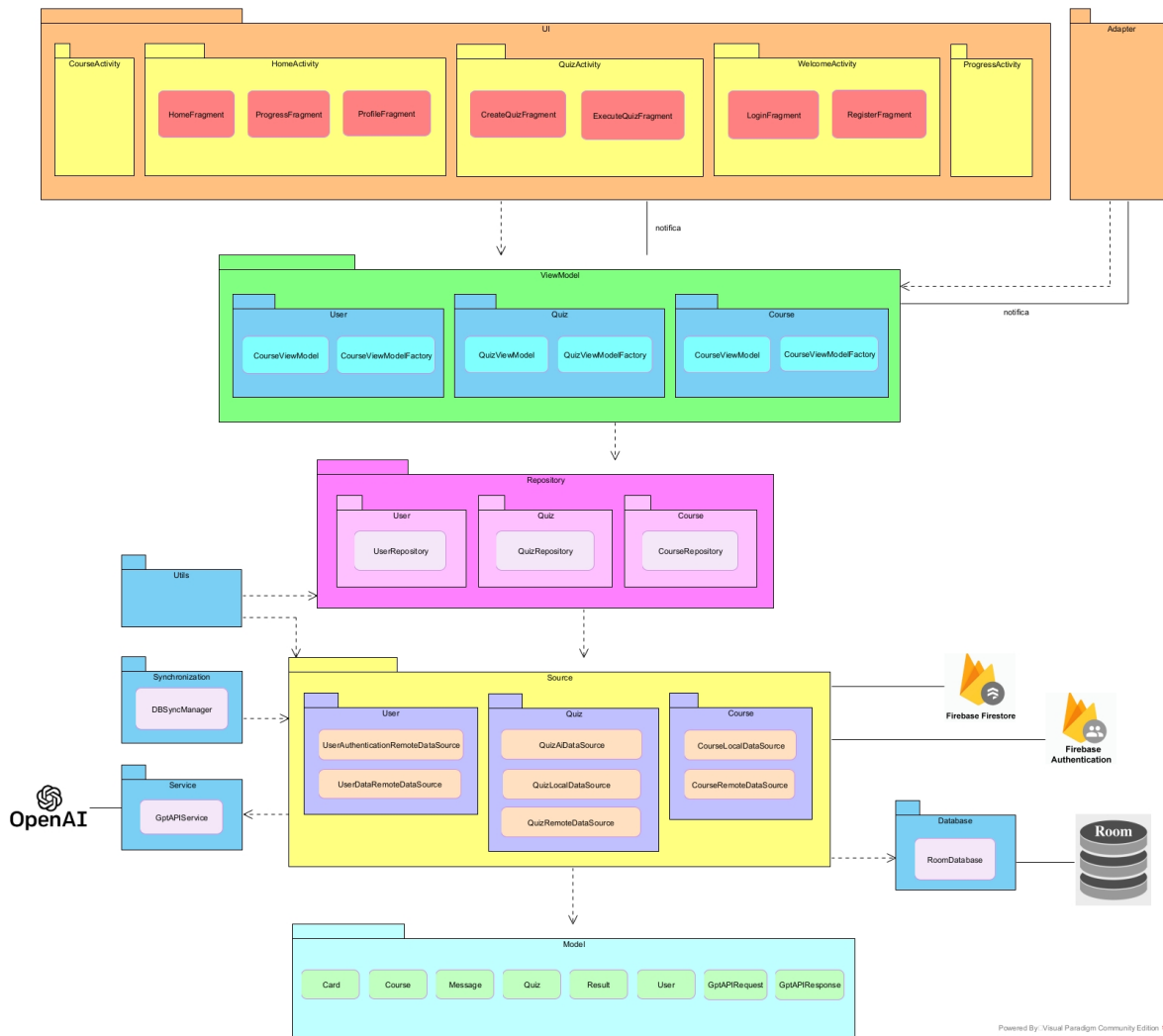


Figure 3: Diagramma architettura logica

## 5 Funzionalità

### 5.1 Autenticazione

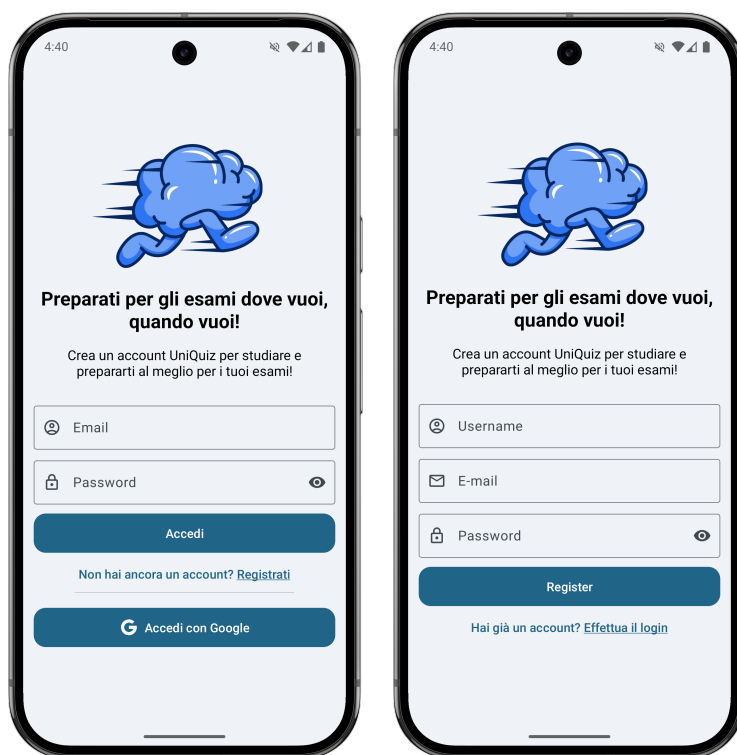
#### 5.1.1 Login

La schermata di login è la prima schermata che viene visualizzata nel momento in cui l'app viene aperta per la prima volta. In questa schermata l'utente è in grado di

effettuare il login tramite le credenziali di un account creato precedentemente oppure tramite un qualsiasi account Google (in questo caso le informazioni quali username - email verranno prese proprio da questo account).

### 5.1.2 Registrazione

Nella schermata di registrazione è possibile registrare un nuovo account tramite l'inserimento di 3 informazioni (username, email e password). Dopo aver controllato la validità delle informazioni il nuovo account verrà creato.



Schermata login

Schermata registrazione

Figure 4: Interfacce di autenticazione

## 5.2 Home

La schermata di home rappresenta il centro dell'applicazione. Da qui è possibile visualizzare in maniera intuitiva le statistiche dei nostri quiz effettuati tramite un grafico a torta, ed inoltre è possibile proseguire i quiz avviati ma lasciati in sospeso. Un'altra

funzionalità principale della home è quella di poter cercare tramite l'apposita barra di ricerca dei corsi in modo da poter eseguire o creare nuovi quiz relativi a quel corso.



Figure 5: Schermata home

### 5.3 Visualizzazione corso

Una volta selezionato un corso dalla pagina di ricerca si apre la pagina con le informazioni dettagliate di un corso. In questa sezione è possibile vedere il nome del corso, i miei quiz e i quiz della community (creati dagli altri utenti) relativi al corso stesso. Sono presenti inoltre due pulsanti, con il primo si avvia la generazione di un quiz con l'AI, mentre il secondo pulsante permette la creazione di un quiz in maniera manuale. Se durante l'utilizzo dell'app non è presente la connessione ad internet verranno solo visualizzati i quiz utente, mentre quelli della community non saranno disponibili.

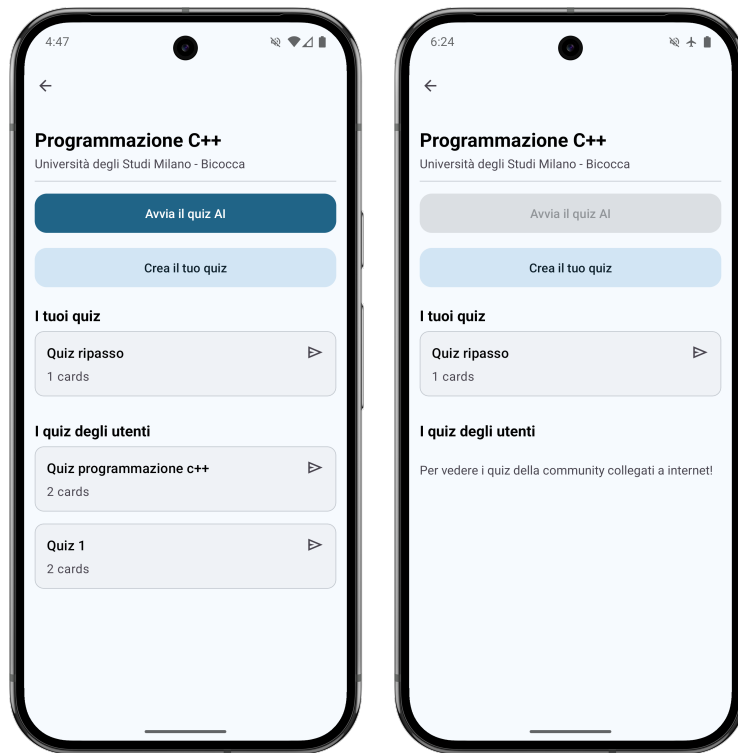


Figure 6: Schermata corso con e senza connessione a internet

## 5.4 Esecuzione quiz

Una volta selezionato un quiz (un modello oppure un'esecuzione salvata per dopo) si apre la schermata di esecuzione del quiz. In questa schermata l'utente potrà rispondere alle domande, che possono essere aperte o chiuse. Nel caso delle domande aperte bisogna inserire la propria risposta nell'apposita sezione e quando si è sicuri la si conferma. Una volta confermata verrà visualizzata anche la risposta effettivamente corretta alla domanda. Nel caso delle domande chiuse bisognerà premere il pulsante relativo alla risposta voluta, dopodiché verrà visualizzata in verde la risposta effettivamente corretta e in rosso la scelta dell'utente nel caso non fosse corretta. Una volta arrivato all'ultima card sarà presente un pulsante in cui è possibile terminare l'esecuzione del quiz. Durante tutta l'esecuzione del quiz sarà possibile terminare o salvare per dopo l'esecuzione del quiz premendo l'icona *X* in alto a sinistra.

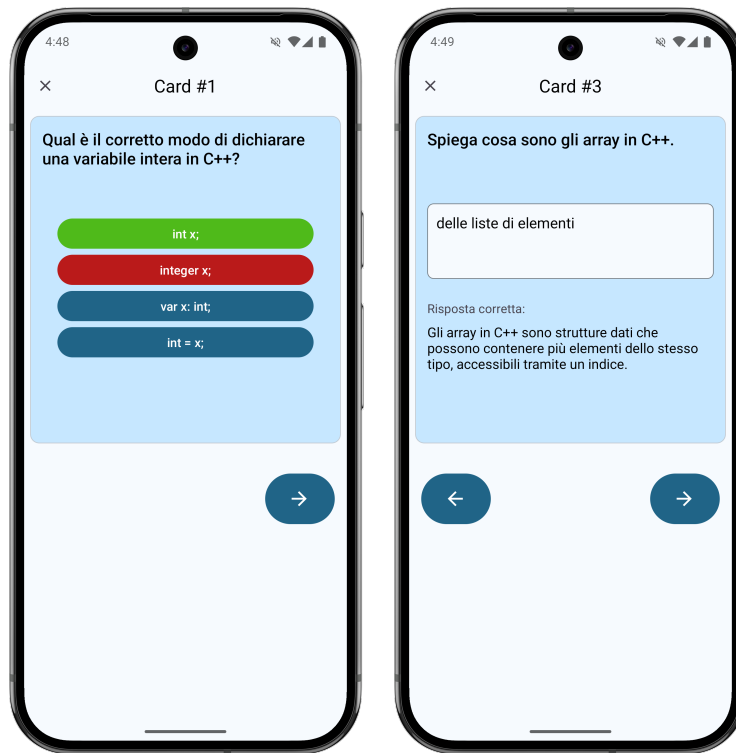


Figure 7: Schermate esecuzione quiz

## 5.5 Creazione quiz

Questa è la sezione in cui l'utente può creare un nuovo quiz relativo ad un corso. Inizialmente verrà mostrata una schermata in cui viene chiesto all'utente di inserire il nome del quiz e il numero di domande di cui è composto. Una volta confermate queste informazioni verrà visualizzata un'interfaccia simile a quella dell'esecuzione, in questo caso però bisogna inserire tutte le informazioni ossia:

- domanda
- risposta corretta (nel caso di domanda aperta)
- risposte possibili (nel caso di domanda chiusa, inserendo anche quale delle 4 è la corretta)

Dopo aver inserito i dati bisogna confermarli per poter passare alla card successiva. Durante tutta la creazione sarà possibile annullare la creazione del quiz premendo l'icona *X* in alto a sinistra.

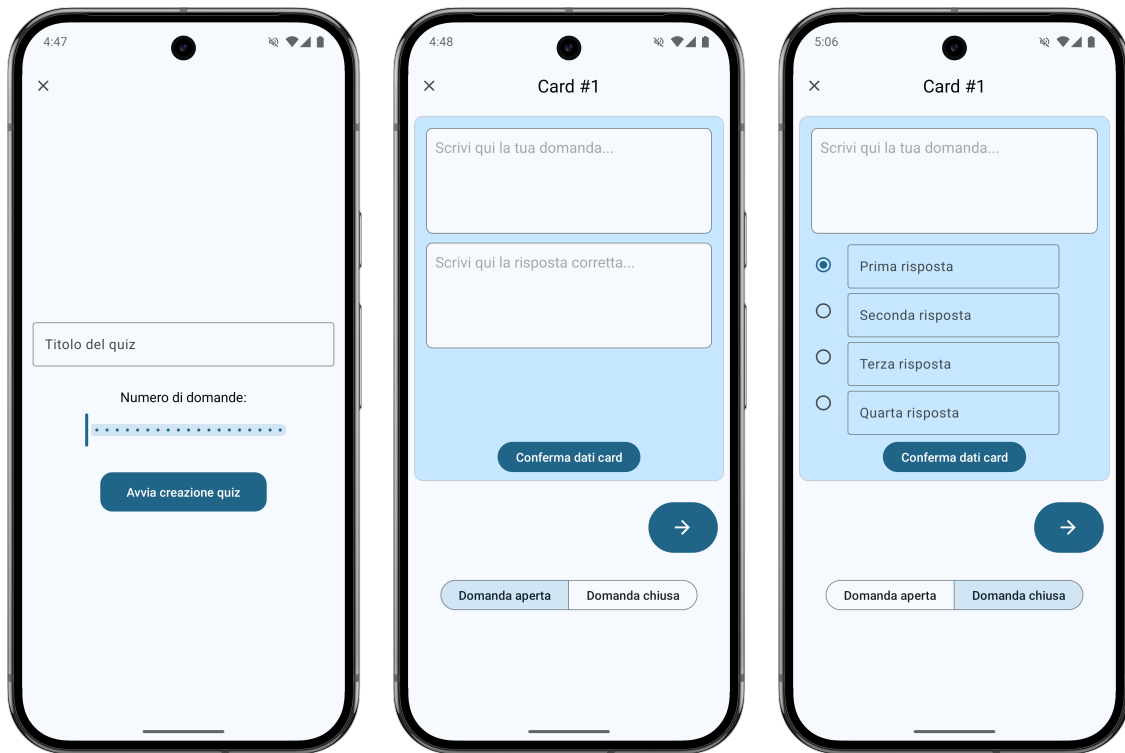


Figure 8: Schermate creazione quiz

## 5.6 Progressi

In questa schermata è possibile visualizzare tutti i progressi dell'utente, ossia tutti i quiz terminati e le esecuzioni lasciate per dopo, mostrando le informazioni più importanti. Da qui sarà possibile proseguire l'esecuzione di quelli lasciate per dopo, oppure vedere il riepilogo delle esecuzioni terminate.

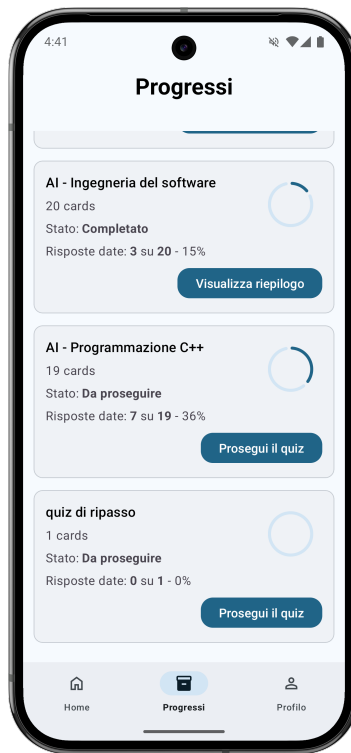


Figure 9: Schermata progressi

## 5.7 Riepilogo

Una volta premuto il pulsante '*Visualizza riepilogo*' di un'esecuzione terminata, verrà visualizzato il riepilogo, dove verranno mostrate tutte le card e le eventuali risposte fornite dall'utente. Per le domande chiuse, nel caso in cui l'utente non abbia fornito una risposta la risposta corretta, verrà colorata in arancione.



Figure 10: Schermata riepilogo esecuzione quiz

## 5.8 Profilo

In questa schermata è possibile visualizzare tutte le informazioni principali del profilo. E' possibile inoltre reimpostare la password, scegliere la lingua preferita dell'applicazione ed effettuare il logout.



Figure 11: Schermata profilo

## 6 Sviluppi futuri

Nello stato attuale l'applicazione fornisce tutte le funzionalità essenziali (ma comunque efficaci) per lo studio e il ripasso. In futuro è possibile aggiungere ulteriori funzionalità che posso migliorare la qualità e la completezza del sistema:

- **Sistema di controllo e verifica:** dare la possibilità agli utenti di segnalare dei quiz o card specifiche, introducendo un team di operatori dedicati alla risoluzione di queste segnalazioni.
- **Supporto a diversi atenei:** al momento nell'applicazione è possibile ricercare unicamente i corsi dell'Università di Milano Bicocca. Una valida aggiunta può essere quella di permettere l'inserimento di corsi relativi a diversi atenei, in modo da ampliare notevolmente la community dell'app.

## 7 Considerazioni finali

Il progetto funziona nella sua interezza ed è stata un'ottima esperienza aver avuto la possibilità di creare un'applicazione Android.

Sappiamo che nella gestione dei branches su GitHub il modo migliore era quello di creare almeno un branch di *develop* e uno di *master*, noi purtroppo ci siamo resi conto di questa situazione troppo tardi dopo che avevamo effettuato molte commit nel *master*, e quindi abbiamo deciso di non crearne uno nuovo per non avere discrepanza nel numero di commit nei branches.

Dato che nella repository di GitHub non si possono mettere file contenenti chiavi di servizi, qui c'è il link di Google Drive per avere i file che servono per debuggare correttamente il progetto: [link](#).

Dentro la cartella ci saranno due file: il primo in cui sono inserite le chiavi da mettere dentro il file `local.properties`; il secondo sarà da mettere dentro la directory `app`