

## Relazione del Progetto Android: Gestione Aziendale

### Informazioni:

- Repository GitHub(SSH):  
[git@gitlab.com:bergamasco\\_cattaneo/progetto\\_applicazioni\\_mobili.git](git@gitlab.com:bergamasco_cattaneo/progetto_applicazioni_mobili.git)
- Progetto IOS (google drive):  
[https://drive.google.com/file/d/1F8wPp2asQD908Vj8UFw25lezqRkZX8nL/view?usp=share\\_link](https://drive.google.com/file/d/1F8wPp2asQD908Vj8UFw25lezqRkZX8nL/view?usp=share_link)
- Utenti di Test
  - PM: [Andrea@merge.pm](mailto:Andrea@merge.pm)/ANDREAPM
  - PM: [Fabio@merge.pm](mailto:Fabio@merge.pm)/FABIOPM
  - PL: [Lorenzo@merge.pl](mailto:Lorenzo@merge.pl)/LORENZOPL
  - PL: [Luca@merge.pl](mailto:Luca@merge.pl)/LUCAPL
  - DEV: [Matteo@merge.dev](mailto:Matteo@merge.dev)/MATTEODEV
  - DEV: [Emma@merge.dev](mailto:Emma@merge.dev)/EMMADEV

**Funzionalità Implementate**

<b>Funzionalità per voto max 25/30</b>	<b>Sì/No/Parzialmente</b>
1) Registrazione	Si
2) Creare/modificare/cancellare un task e sottotask (solo nome, descrizione)	Si
3) Chat tra utenti	Si
4) Ricerca su progetti da parte del PM e sui task da parte del PL	Si
5) Ricezione notifiche	Si
6) Avanzamento task e progetto	Si

<b>Funzionalità per voto &gt;25/30</b>	<b>Sì/No/Parzialmente</b>
7) Voti e commenti sul lavoro svolto	No
8) Aggiunta foto/documenti relativo al task	No
9) Assegnazione task ad un altro developer	Si
10) Profilo utente con foto, elenco task/progetto a cui è assegnato, skills	No

11) Schermata statistiche: n. progetti/task conclusi/attivi/da iniziare. Tempo medio di completamento dei progetti.	No
---	----

#### Descrizione dell'applicazione

L'applicazione è stata sviluppata per gestire i task e i progetti all'interno di un'azienda, con ruoli distinti per Project Manager (PM), Project Leader (PL) e Developer (DEV).

L'applicazione è stata realizzata utilizzando **Android Studio** e **Firebase** come backend per la gestione dei dati e delle autenticazioni.

#### Scelte Implementative

##### 1. Registrazione e Autenticazione

- La registrazione e l'autenticazione degli utenti sono gestite tramite **Firebase Authentication**. Ogni utente viene registrato con un ruolo specifico (PM, PL, D) e i dati vengono memorizzati in **Firestore**.
- Durante la registrazione, viene generato un token FCM per le notifiche push.

##### 2. Gestione dei Task e Sottotask

- a. I task e i sottotask possono essere creati, modificati e cancellati. La struttura è gerarchica: un progetto contiene task, e ogni task contiene sottotask.
- b. L'avanzamento dei task e dei progetti è calcolato come la media dell'avanzamento dei sottotask.

##### 3. Chat tra Utenti

- a. La chat è stata implementata utilizzando **Firestore** per memorizzare i messaggi. Le chat sono possibili tra:
  - i. Dev e Dev dello stesso progetto
  - ii. PL e DEV
  - iii. PM e PL
- b. La chat è realizzata tramite una **RecyclerView** che mostra i messaggi in tempo reale

##### 4. Ricerca e Filtri

- a. Il PM può filtrare i progetti in base allo stato, al PL assegnato e alla scadenza.
- b. Il PL può filtrare i task in base allo stato, al developer assegnato, alla scadenza e alla priorità.

##### 5. Notifiche

- a. Le notifiche push sono gestite tramite **Firestore Cloud Messaging (FCM)**. Le notifiche vengono inviate in caso di:
  - i. Completamento di una task (notifica al PL)
  - ii. Completamento di un progetto (notifica al PM)
  - iii. Solleciti da parte del PL ai DEV o da parte dei PM ai PL

## 6. Assegnazione task

- a. Il PL può assegnare task ai developer. L'assegnazione è gestita tramite un'interfaccia utente intuitiva che permette di selezionare il developer da un elenco.

## 7. Incremento delle Sottotask

- a. L'incremento è gestito tramite l'API **Firestore Firestore**. Quando il Developer modifica lo stato di avanzamento, viene aggiornato il valore nel database e vengono ricalcolati i progressi del task e del progetto.

## Parti Complesse e Soluzioni Adottate

### 1. Gestione delle notifiche:

- a. La parte più complessa è stata la gestione delle notifiche a cascata. Per risolvere questo problema, è stato implementato un sistema di callback che verifica lo stato di avanzamento dei task e dei progetti e invia notifiche solo quando vengono raggiunte determinate condizioni (es. completamento di un task o di un progetto).

### 2. Sincronizzazione in Tempo Reale:

- a. La sincronizzazione in tempo reale dei dati tra l'app e **Firestore** è stata realizzata utilizzando i listener di **Firestore**. Questo permette di aggiornare l'interfaccia utente in tempo reale senza bisogno di ricaricare manualmente i dati.

### 3. Gestione delle Chat:

- a. La gestione delle chat è stata complicata dalla necessità di mantenere l'ordine cronologico dei messaggi e di gestire correttamente i messaggi inviati e ricevuti. È stato utilizzato un **FirestoreRecyclerAdapter** per gestire i messaggi in modo efficiente.

### 4. Gestione dell'incremento delle Sottotask:

- a. L'incremento delle sottotask aggiorna correttamente lo stato del task e del progetto. E' stato implementato un sistema di callback che aggiorna i progressi in cascata:
  - i. Aggiorna la sottotask

- ii. Ricalcola l'avanzamento del task
  - iii. Ricalcola l'avanzamento del progetto
- b. Questo sistema garantisce che i dati siano sempre sincronizzati e coerenti

## Descrizione Dettagliata dell'Interfaccia Utente e del Flusso di Navigazione

### Schermata di Accesso:

- All'avvio dell'applicazione, viene mostrata la schermata di accesso per gli utenti già registrati. Se l'utente preme il pulsante "Sign up", la schermata viene modificata per permettere la registrazione.
- Dopo il login, l'applicazione verifica il ruolo dell'utente (PM, PL, D) e reindirizza alla schermata di Home.

### Schermata Home:

- La schermata di Home è composta da una **RecyclerView** che mostra progetti, task o sottotask in base al ruolo dell'utente e alle preferenze impostate tramite un **ToggleButton** situato in alto a destra.
- Il pulsante dei filtri, situato in alto al centro, apre un **BottomSheet** con i filtri specifici per il ruolo dell'utente.

### Ruolo PM:

- Il PM vede una lista di progetti non cliccabili, con un pulsante "warning" per inviare solleciti al PL incaricato.
- Il PM può aggiungere nuovi progetti tramite un pulsante dedicato.
- Il PM riceve una notifica quando un progetto raggiunge il 100%.

### Ruolo PL:

- Il PL vede una lista di progetti o task cliccabili, a seconda della preferenza impostata nel ToggleButton.
- Cliccando su un progetto/task, si accede a una pagina dettagliata con una **GridView** che mostra le task/sottotask relative.
- Il PL può aggiungere, modificare o eliminare task/sottotask, e inviare solleciti ai developer.
- Il PL riceve una notifica quando una sottotask raggiunge il 100%.
- Può visualizzare lo stato di avanzamento dei task e dei progetti nella schermata principale.

**Ruolo DEV:**

- Il DEV vede una schermata simile a quella del PL, con la possibilità di aggiungere sottotask alle task assegnate.
- Cliccando su una sottotask, può incrementare o decrementare il progresso tramite i pulsanti +25% e -25%.
- Se una sottotask raggiunge il 100%, viene visualizzata una notifica e il progresso del task viene aggiornato automaticamente.