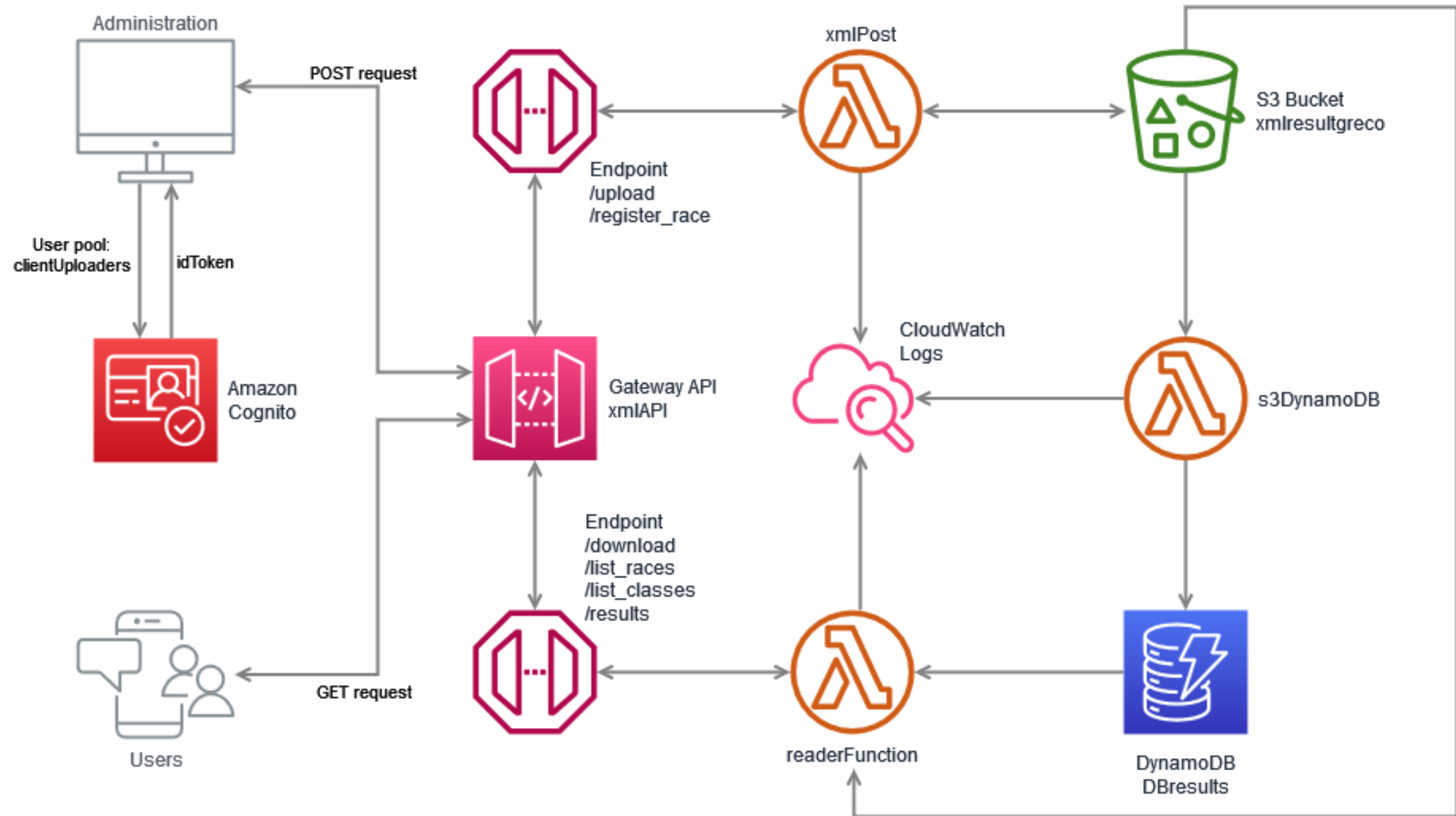


Infrastruttura cloud AWS

Componenti del gruppo:

Greco Daniele-1065570, Matias Negro-1065808

Infrastruttura generale



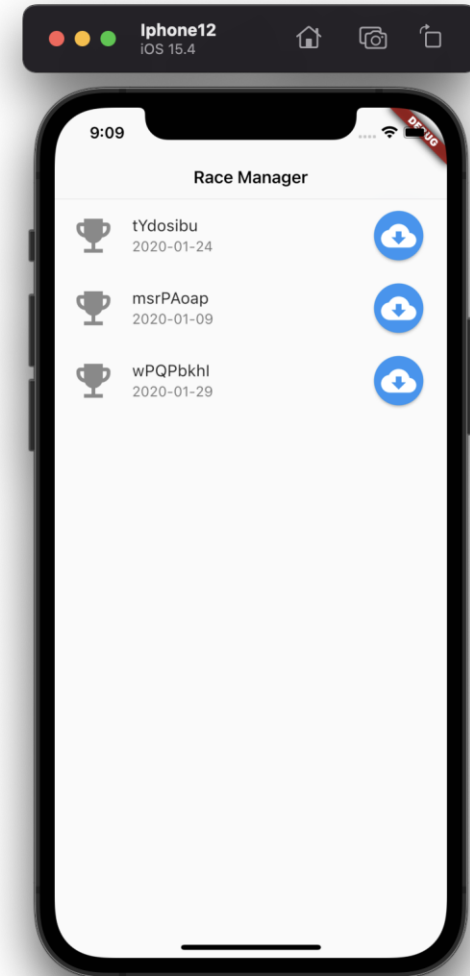
`/upload_start_list`
`/download_start_list?id=...&class=...`
`/split_time_jpeg?id=...&class=...`

- **`/upload_start_list`**
 - L'Endpoint si collega alla funzione *lambda* omonima **`upload_start_list`** e consente di caricare le griglie di partenza relative alle singole categorie di una gara. Il file xml viene parsato e caricato in formato json in una tabella separata all'interno di DynamoDB, DBStartList.
- **`/download_start_list?id=...&class=...`**
 - L'Endpoint si interfaccia alla funzione *lambda* **`readerFunction`** e permette di estrarre la griglia di partenza relativa alla categoria di una gara in formato json.
- **`/split_time_jpeg?id=...&class=...`**
 - L'Endpoint si interfaccia alla funzione *lambda* **`readerFunction`** e restituisce il grafico contenente i checkpoint sull'asse delle x e lo scarto dal primo in classifica sull'asse y. La scelta di restituire l'immagine (codificata in base64) al posto dei dati «raw» è stata dettata dalla volontà di scaricare la totalità della computazione lato server (nonostante comporterebbe costi maggiori in un caso reale) perché non vi era necessità di rendere interattivo il grafico stesso.

I codici relativi sono consultabili dalla pagina di Github raggiungibile dall'ultima slide nella cartella «lambda».

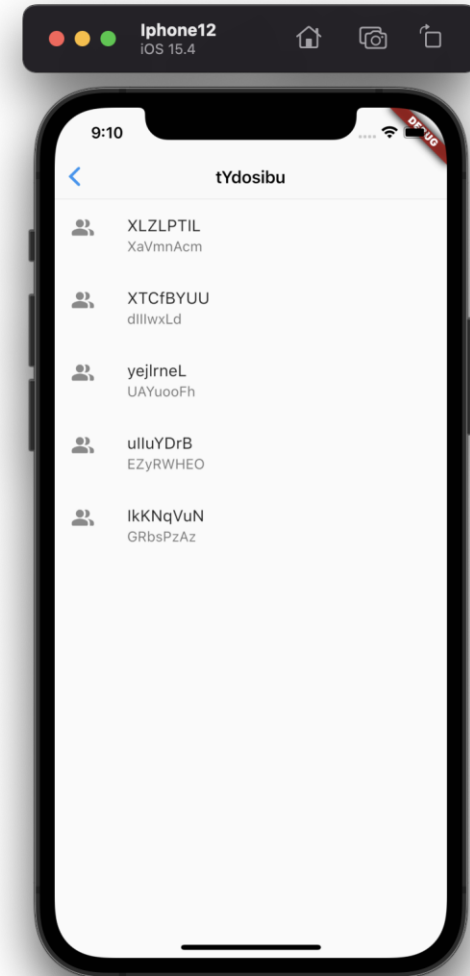
Flutter

L'applicazione è stata sviluppata seguendo lo stile grafico di IOS, il che ci ha portato ad utilizzare widget del package *Cupertino*. L'elemento chiave è stato l'utilizzo delle list-view create dinamicamente a seconda delle request. Per il refresh delle pagine abbiamo optato per l'utilizzo della gesture pull down la quale mostra un'animazione di ricaricamento mentre viene eseguita una nuova request.



Navigazione

La navigazione è stata impostata in main.dart dentro il quale sono definite tutte le route, di cui quella iniziale è uno splash screen che maschera parzialmente il caricamento iniziale. Abbiamo dato la possibilità di scaricare i file xml direttamente dall'applicazione a patto che venga data l'autorizzazione all'accesso alle directory del file system.

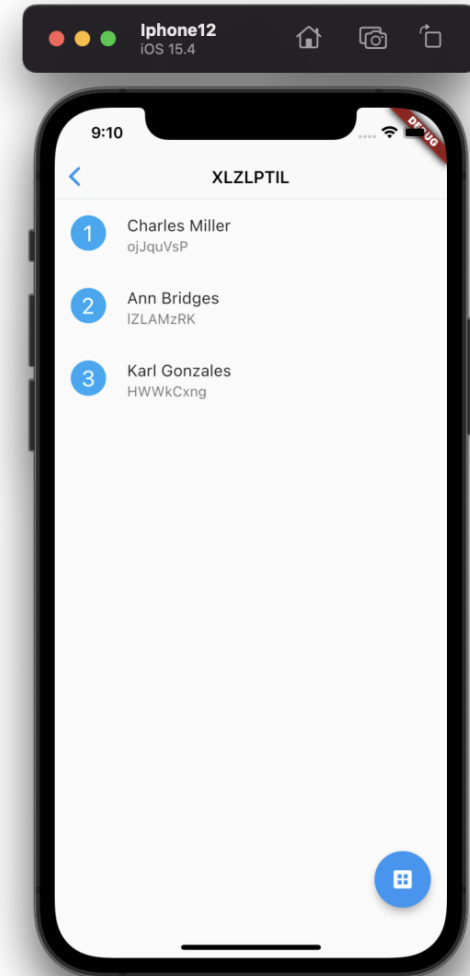


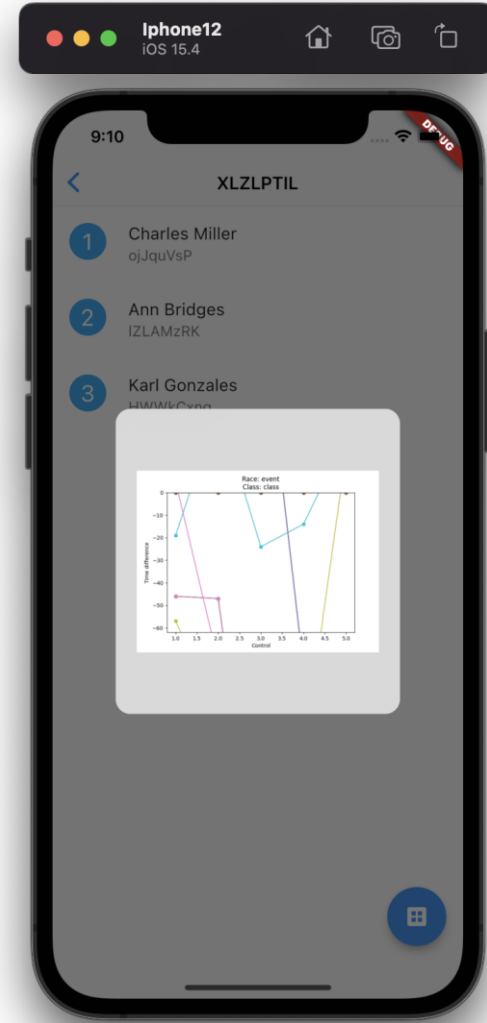
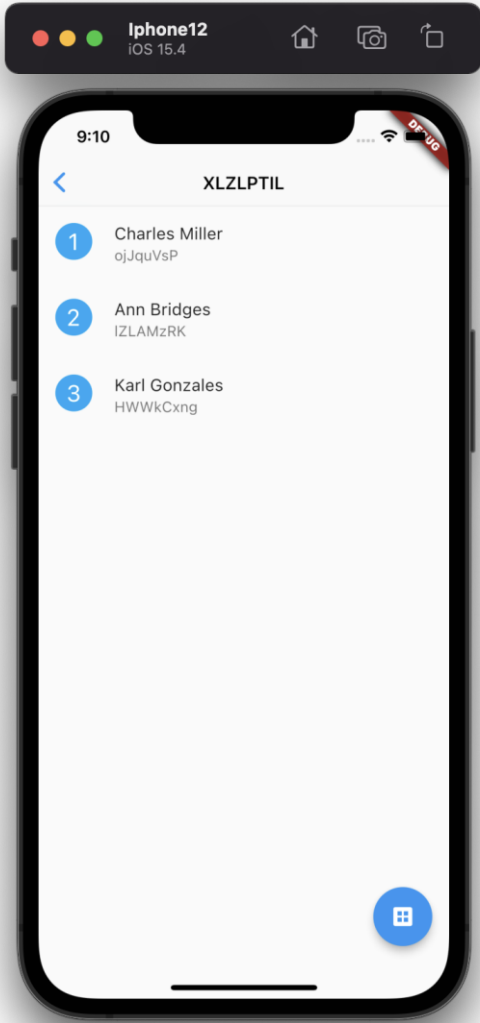
Start list & Result list

Visualizzazione grafico

Nel momento in cui si visualizzano le classi la scelta tra start list e result list è data attraverso un Cupertino action sheet che si occupa di indirizzare verso la route selezionata. Mentre nella result list viene posto un place holder, icon.person, in sostituzione di un eventuale foto dell'atleta, in result list viene mostrata la classifica numerata insieme ai dati degli atleti.

Per la visualizzazione del grafico avviene attraverso un Cupertino alert dialog contenente l'immagine.





GitHub

Link alla repository con le *applicazioni di test*, l'applicazione in **Flutter**, le funzioni **lambda** e gli **screenshot** in formato .png visibili pubblicamente:

https://github.com/MatiasNegro/tcm_homeworks_and_project