

## Documentació

### Simulant fase Construcció - “Múscul” Projecte de curs

Plataformes en xarxa. Curs 2018 - 2019

Alumnes: Alvaro Ortega & Marc Viladegut

Data: 28 de maig de 2019

## Tècniques utilitzades en el projecte

Per l'aplicació de mòbil de taxis s'ha utilitzat el servei de **Firebase Authentication** per dur a terme el servei d'autenticació i administració d'usuaris. Per guardar les dades de l'aplicació i mantenir-les sincronitzades a temps real, s'ha utilitzat el servei de **Realtime Database** que ofereix Firebase. Per últim, s'ha utilitzat el servei de **Firebase Cloud Messaging (FCM)** per dur a terme les notificacions *push* de l'aplicació, així com també la utilització del **Firebase Cloud Functions** per executar el codi de backend de l'aplicació necessari per a poder-les enviar de forma automàtica.

## Firebase Authentication

El mètode d'autenticació que s'ha implantat en el projecte correspon a un dels proveïdors d'inici de sessió que ofereix Firebase Authentication. El mètode d'inici de sessió és per mitja de **correu electrònic i contrasenya**. Cada usuari tindrà l'opció de registrar-se i així crear una nova entrada d'autenticació en l'aplicació per mitjà del correu electrònic i la respectiva contrasenya en la fase de registre. Cada correu electrònic en Firebase Authentication és únic i per tant no es podrà repetir.

No s'ha implementat el mètode d'autenticació **anònim**, ja que és indispensable que tot usuari que entri en l'aplicació hagi iniciat sessió amb les credencials correctes per poder donar un servei més realista en l'aplicació.

## Realtime Database

S'ha decidit utilitzar aquest mètode d'emmagatzemament JSON perquè no és necessari guardar dades molt complexes, ja que únicament guardant diversos atributs, tant de l'usuari com de la reserva és suficient. Per tant, en aquest projecte s'ha prescindit l'ús del **Firestore Database**.

En la part dels **usuaris** s'ha guardat en la base de dades una referència anomenada “users” on s'emmagatzemaran tots els usuaris de l'aplicació segons el seu identificador (UID). Aquest identificador és únic i per tant no es podrà repetir, així que s'evitarà l'eliminació de les dades de forma indeguda. Cada UID formarà la parella clau/valor on el valor d'aquesta parella serà un client o bé, un taxista (veure figura 1b).

Per altra banda, les **reserves** estaran compostes de manera similar que els usuaris. Cada usuari tindrà les seves reserves i es guardarà en funció del UID de cada usuari que realitzi la reserva en cada moment. Tenint en compte que cada usuari pot tenir més d'una reserva, el UID de cada usuari estarà compost per la parella clau/valor essent la clau el identificador de la reserva i el valor l'objecte reserva (veure figura 1a). A més les reserves, tindran un atribut

```

reservations
├── J2aQ2jh7VCW86bbaYXb00b0mARW2
│   └── 0
│       ├── cost: 37.32€
│       ├── costToDollars: 41.800640000000€
│       ├── costToPounds: 31.723£
│       ├── dateTime: 155903176639
│       ├── dateToString: "28/05/2019"
│       ├── destinationLocality: "Lleida"
│       ├── distance: 2330€
│       ├── duration: "19 mins"
│       ├── idReservation: 0
│       ├── originLocality: "Torrebeses"
│       ├── timeToString: "10:22"
│       └── numReservation: 1

```

(a) Emmagatzemament d'una reserva en la BD

```

users
├── J2aQ2jh7VCW86bbaYXb00b0mARW2
│   ├── driver: false
│   ├── email: "mva8@alumnes.udl.ca"
│   ├── gender: 0
│   ├── name: "Marc"
│   ├── smoke: false
│   ├── surname: ""
│   └── token: "fs-A1ZL_1oI:APA91bGVwnNds

```

(b) Emmagatzemament d'un usuari en la BD

Figura 1: Emmagatzemament de la informació de l'aplicació

adicional (clau/valor) que serà el número de reserva que s'assignarà un cop s'efectua la reserva, ja que serà únic per tota l'aplicació i no es podrà repetir en cap usuari. Les dades de la reserva seran emmagatzemades de la mateixa forma tan en el client, com en el taxista assignat per aquella reserva, per tant en la base de dades hi haurà replicació de les dades.

Per acabar, respecte les **regles de la base de dades**, no s'efectuarà cap modificació en els permisos tant de lectura com d'escriptura. S'entén que cada usuari serà partícip de la gestió de les seves pròpies dades i per tant, un usuari mai tindrà accés a l'identificador únic d'un altre usuari, negant així tota probabilitat d'alteració de les mateixes.

## Firestore Cloud Messaging (FCM)

Per poder rebre notificacions en l'aplicació han estat implementats els serveis de Firebase Cloud Messaging per a poder rebre les notificacions pertinents que pugui enviar el backend (Firebase Cloud Messaging).

Per dur-ho a terme, s'ha creat un nou servei a l'aplicació anomenat MyService i que s'encarrega de gestionar tant els tokens que rep cada usuari, com de mostrar les notificacions que rep de la plataforma Firebase Cloud Messaging. La forma de visualització de les notificacions serà per mitjà de les notificacions *push* on s'encarregarà el propi servei tant de crear-les com de llençar-les (veure figura 2).

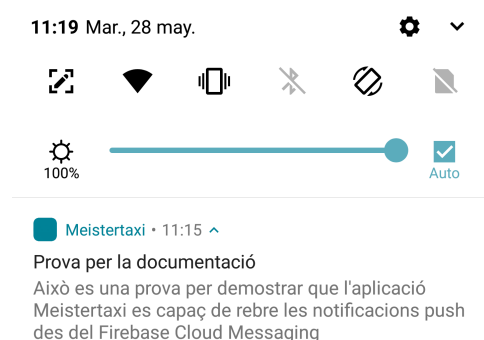


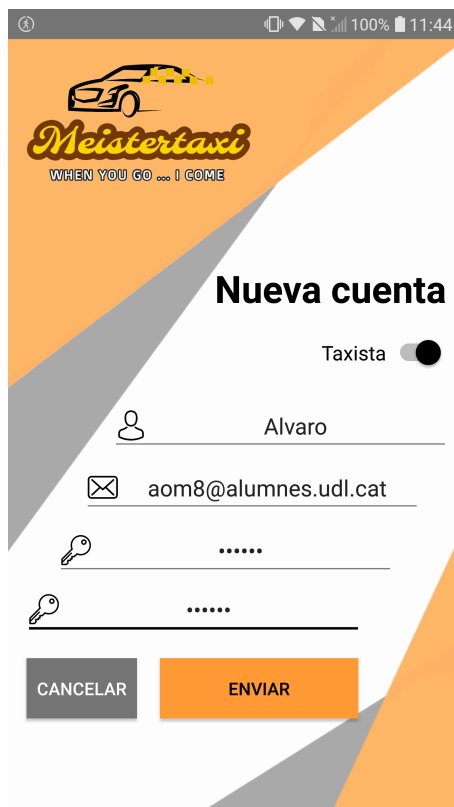
Figura 2: Exemple de notificació *push*

## Firestore Cloud Functions

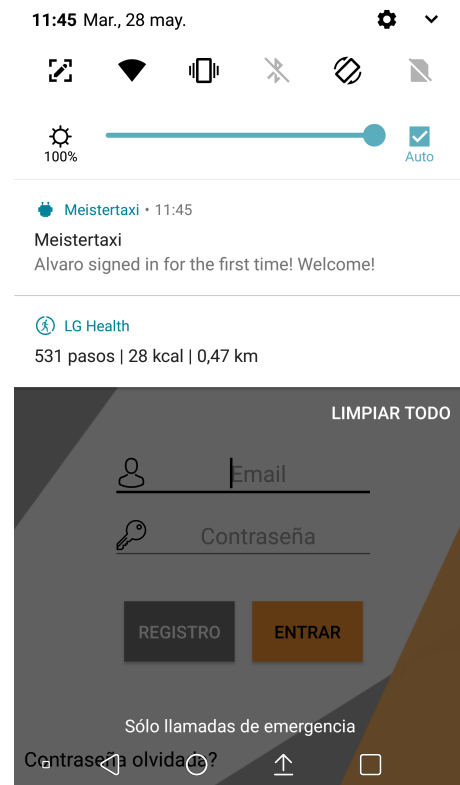
En aquesta aplicació, les funcions de Firestore Cloud Functions s'han introduït per enviar notificacions *push* a aquells usuaris que es donen d'alta el primer cop en l'aplicació. S'envia una notificació donant la benvinguda a l'usuari al mateix instant de temps que aquest es registra en l'aplicació.

Com s'ha observa en la figura 1b, cada usuari disposa d'un token únic necessari per poder enviar-li les notificacions des del backend. Quan es registra un usuari nou en la base de dades, l'aplicació assigna el token a l'usuari on aquest és enviat des de Firestore Cloud Messaging i on posteriorment, es guarda en la Realtime Database (veure figura 3a).

Aprofitant la generació d'aquest nou usuari en la Realtime Database, salta el disparador que conté la funció que està programada per a cada usuari que es generi i es crea un missatge de benvinguda. Posteriorment s'agafa el token de l'usuari en qüestió i s'envia el missatge des de Firestore Cloud Messaging, això sí, de forma automàtica (veure figura 3b).



(a) Fase de registre d'un usuari nou



(b) Notificació de benvinguda a l'aplicació

Figura 3: Exemple de la notificació *push* de benvinguda