



UNIVERSITAT POLITÈCNICA DE CATALUNYA  
BARCELONATECH

Escola Politècnica Superior d'Enginyeria  
de Manresa



**Treball Final de Grau**

***APLICACIÓ  
“L’ESTUDIANTAT  
IMPORTA”***

**Grau en Enginyeria de Sistemes TIC  
Curs 2021/2022**

Autor: Carlos Martínez González

Directora: Marta Isabel Tarres Puerta

Data: 13/01/2022

Localitat: Manresa



## **RESUM DEL PROJECTE**

Aquest projecte es basa en el disseny i la implementació d'una aplicació Android, que permet omplir un qüestionari a l'estudiantat, amb l'objectiu de conèixer les causes d'abandonament dels estudis. Recollida d'informació i tractament de dades, amb resultats gràfics. En el projecte també hi ha la possibilitat de posar queixes.

Diferents tipus de rols; usuaris bàsics, professors, coordinadors i administradors, cadascun amb uns privilegis diferents. Els administradors tenen la possibilitat d'afegir rols i afegir o esborrar preguntes del qüestionari de forma permanent.

## **ABSTRACT**

This project is based on the design and implementation of an Android app that allows students to fill a questionnaire in order to find out the causes of dropping out of college. Information gathering and data processing with graphical results. There is also the possibility of making a complaint.

Different types of roles; basic users, professors, coordinators and administrators, each with different privileges. Administrators have the possibility to add roles and add or delete questionnaire questions permanently.

# ÍNDEX

<b>1. INTRODUCCIÓ</b>	<b>1</b>
1.1. Motivació	1
1.2. Objectius	1
1.3. Finalitat	2
1.4. Fases	3
<b>2. REQUISITS</b>	<b>3</b>
<b>3. ESTUDIS PREVIS</b>	<b>4</b>
3.1. Android Studio	4
3.1.1. Característiques	4
3.1.2. Plataformes i requisits del sistema	5
3.1.3. Emulador Android	6
3.1.4. Llenguatges de programació	7
3.2. Google Firebase	8
3.2.1. Authentication	9
3.2.2. Analytics	10
3.2.3. Firestore Database	12
3.2.4. Concurrencia de dades	16
3.2.5. Seguretat	17
<b>4. INTERFÍCIE D'USUARI</b>	<b>18</b>
4.1. Pantalla principal	19
4.2. Pantalla d'opcions	22
4.3. Pantalla qüestionari	24
4.4. Pantalla queixes/reclamacions	30
4.5. Pantalla gràfics	32
4.5.1. Pantalla gràfic pregunta 1	33
4.5.2. Pantalla gràfic pregunta 2	34
4.5.3. Pantalla gràfic pregunta 3	35
4.5.4. Pantalla gràfic pregunta 4	36
4.5.5. Pantalla gràfic pregunta 6	37
4.5.6. Pantalla gràfic pregunta 7	38
<b>5. CODI PRINCIPAL</b>	<b>39</b>
5.1. MainActivity.kt	39
5.2. opcions.kt	42
5.3. Cuestionario.kt	46
5.4. Queixes.kt	56
5.5. Grafiques.kt	58
5.6. Graphs.kt	60
5.7. Graphs2.kt fins a Graphs7.kt	63

<b>6. ROLS</b>	<b>63</b>
<b>7. GOOGLE PLAY STORE</b>	<b>65</b>
7.1. Requisites	65
7.2. Arxius APK	66
<b>8. ODS</b>	<b>67</b>
<b>9. CONCLUSIONS</b>	<b>68</b>
<b>10. BIBLIOGRAFIA</b>	<b>69</b>

# 1. INTRODUCCIÓ

## 1.1. Motivació

Bona part de l'estudiantat, per una raó o una altra, no pot completar el seu grau i, per les seves circumstàncies, ha d'abandonar la universitat. Els motius poden ser molt diferents, i a vegades no hi ha possibilitat de seguir en els estudis. No obstant, altres vegades el motiu d'abandonament pot ser un problema psicològic, un malentès, una mala relació amb el professorat o la resta de l'alumnat.

Amb aquesta aplicació es vol aconseguir que aquesta part de l'estudiantat se senti escoltada i, al menys, pugui expressar la seva opinió davant la universitat.

Hi ha casos en els que, en funció del problema que hagi fet abandonar el grau, existeix la possibilitat d'arreglar la situació i fer que continuï amb els estudis.

L'aplicació també incorpora l'opció de posar una queixa i explicar, amb tot tipus de detall, quin és el seu problema o la seva recomanació.

A més, hi ha un apartat reservat per als usuaris amb privilegis de l'app, que seran el professorat i el professorat encarregat de coordinar els diferents graus, on es poden visualitzar diferents gràfics amb els percentatges de les diferents respostes de qui hagi omplert el formulari.

Com a motivació especial, aquest treball de fi de grau forma part del projecte d'innovació docent '*Mentoria, tutorització i coordinació: Les claus per a una millora del rendiment i reducció de l'abandonament a la fase inicial*', facilitant molt la recollida de dades i la comunicació professorat - alumnat.

## 1.2. Objectius

L'objectiu del TFG ha estat el desenvolupament d'una app que permet agilitzar el procés de recollida d'informació de l'estudiantat amb l'objectiu de conèixer els principals motius d'abandonament dels estudis. A través d'un qüestionari amb preguntes de resposta oberta i d'altres on es permeten triar les opcions més adients, o bé d'una eina addicional de recollida de queixes i crítiques, el professorat podrà obtenir dades i estadístiques que permetran a la universitat fer els canvis necessaris en cada cas.

Aquestes dades, un cop tractades, seran mostrades en gràfics de tipus pastís, amb un disseny atractiu que permetrà visualitzar els percentatges de les respostes de tot l'alumnat que hagi omplert el qüestionari.

A més, cada vegada que s'omple un nou qüestionari, s'envien alertes cap al professorat encarregat de la coordinació dels diferents graus, per tal que puguin donar resposta a la informació recollida.

Per a aconseguir aquests objectius s'utilitzaran eines i llenguatges de programació innovadors amb l'objectiu de crear una app que sigui senzilla, robusta i eficient. També és important que tingui un disseny força atractiu per l'estudiantat i que sigui àgil, per no fer perdre massa temps.

Aquesta aplicació ha de complir uns certs requisits necessaris a qualsevol software:

- **Correcció:** ha de complir amb el seu objectiu.
- **Usabilitat:** ha de ser fàcil d'aprendre.
- **Seguretat:** ha de ser resistent a atacs externs.
- **Flexibilitat:** ha de poder ser modificada pels desenvolupadors.
- **Portabilitat:** ha de poder ser utilitzada en diversos equips.

### 1.3. Finalitat

Amb aquest projecte es vol aconseguir que l'alumnat disposi d'una plataforma, en aquest cas una aplicació mòbil, que els permeti sentir-se escoltats.

Quan algú abandona els estudis, normalment té una sensació de frustració, al no poder expressar la seva opinió de cara a la universitat.

És possible que aquesta persona tingui alguna queixa, una suggerència o que, simplement, vulgui expressar els seus motius d'abandonament del grau. És per això que amb l'aplicació "L'Estudiantat Importa", sentiran que tenen una eina que els permet donar la seva opinió i, sobretot, sentir-se escoltats.

A més, el professorat de la universitat podrà accedir a les respostes, gràfics de percentatges, etc. del l'alumnat, per tal de poder actuar en els casos que sigui necessari. Podran posar-se en contacte amb aquella persona per tal d'afegir un punt més d'empatia i, si és possible i les circumstàncies ho permeten, intentar persuadir-la per seguir cursant el grau.

Amb aquesta informació recollida es podran millorar processos de funcionament de la universitat, donant agilitat i facilitant la comunicació amb l'alumnat disposat a abandonar els estudis.



## 1.4. Fases

Per tal d'assolir els objectius del projecte, aquest es dividirà en les següents fases:

- **Disseny i implementació de l'aplicació mòbil:** s'ha fet una recerca, estudi i avaluació de les millors eines existents pel desenvolupament d'una aplicació. S'ha investigat el millor llenguatge de programació, amb una visió de cara al futur i pensant sempre en la comoditat dels usuaris.
- **Creació de BD i servidors per recollir les dades:** s'han buscat les millors opcions per poder crear BD i servidors de forma senzilla, en temps real i que suportessin altes concurrències de dades.
- **Compartició de l'aplicació per a la comprovació del seu bon funcionament:** per tal de comprovar el bon funcionament de l'app i el suport d'altres concurrències de dades, s'ha compartit l'app amb un alt número de persones de confiança, que l'han utilitzat i han donat feedback sobre aquesta.
- **Inscripció de l'aplicació a la Google Play Store i finalització del projecte:** s'ha decidit pujar l'aplicació a Google Play Store per tal de facilitar l'obtenció d'aquesta per part de l'estudiantat.

## 2. REQUISITS

Com a requisits, se li haurà de demanar a l'estudiantat:

1. **Quin grau vas començar?** (desplegable amb els diferents graus).
2. **Quan vas començar?** (desplegable amb les últimes possibles entrades: 02/2021, 07/2020, 02/2020, 07/2019, etc.).
3. **Motius d'abandonament del grau.** Opcions a triar:
  - Malaltia
  - No ho puc compaginar amb la feina
  - Ho trobo massa difícil
  - Cal treballar massa
  - No m'ha agradat
  - Altres (camp obert)
4. **Què faràs ara?** Opcions a triar:
  - Anar a treballar
  - Fer un CFGS
  - Anar a fer un altre grau del mateix àmbit
  - Anar a fer una enginyeria d'un altre àmbit
  - Anar a fer un grau que no té res a veure amb la tecnologia
  - Altres (camp obert)
5. **Explica'ns una mica més els motius d'abandonament i les previsions de futur** (camp obert)
6. **Tornaries a escollir l'EPSEM si fessin els estudis que desitges?** (sí o no)

7. **Tornaries a escollir la UPC si fessin els estudis que desitges?** (sí o no)
8. **Explica'ns una mica més els motius per escollir-nos un altre cop o no** (camp obert)
9. **Pregunta extra 1** (camp obert)
10. **Pregunta extra 2** (camp obert)

En funció de les respostes d'aquests es faran les gràfiques que després podran visualitzar els usuaris amb privilegis de l'aplicació.

## 3. ESTUDIS PREVIS

Abans de començar a dissenyar l'aplicació mòbil, són necessaris uns coneixements previs que ens permetin desenvolupar aquesta de forma senzilla i sense errors.

### 3.1. Android Studio

És l'entorn de desenvolupament integrat oficial per a la plataforma Android. Va ser anunciat el 16 de maig de 2013 a la conferència de *Google I/O*, i va reemplaçar a Eclipse com l'IDE oficial pel desenvolupament d'aplicacions Android. La primera versió estable va ser publicada al desembre de 2014.

Aquest entorn està basat en el software IntelliJ IDEA de *JetBrains* i ha estat publicat de forma gratuïta a través de la Llicència Apache 2.0. Està disponible per a les plataformes GNU/Linux, macOS, Microsoft Windows i Google Chrome OS. Ha estat dissenyat específicament per al desenvolupament d'Android.

S'ha utilitzat la darrera versió d'Android Studio, l'*Arctic Fox 2020.3.1 Patch 2*.

#### 3.1.1. Característiques

Android Studio, encara que està en constant creixement, i s'espera que es desenvolupin noves funcions amb cada nova versió, hi ha diverses característiques que proporciona actualment. Algunes d'elles són:

- Suport per a construcció basada en Gradle.
- Refactorització específica d'Android i arranjaments varis.
- Eines Lint per a detectar problemes de rendiment, usabilitat, compatibilitat de versions i altres problemes.
- Integració de ProGuard i funcions de signatura d'aplicacions.
- Plantilles per crear dissenys comuns d'Android i altres components.
- Un editor de disseny enriquit que permet als usuaris arrossegar i deixar anar components de la interfície d'usuari.
- Suport per programar aplicacions per a Android Wear.

- Suport integrat per a Google Cloud Platform, que permet la integració amb Firebase Cloud Messaging i Google App Engine.
- Un dispositiu virtual d'Android que s'utilitza per executar i provar aplicacions.
- Renderitzat en temps real.
- Consola de desenvolupament: consells d'optimització, ajuda per a la traducció, estadístiques d'ús, etc.

### 3.1.2. Plataformes i requisits del sistema

Està disponible per a Windows 2003, Vista, 7, 8 i 10 (tant per a plataformes de 32 com de 64bits), GNU/Linux, Linux amb GNOME o KDE i 2 GB de memòria RAM mínim, i macOS, a partir de la versió 10.8.5. Els requisits del sistema per a les tres plataformes són:

Versió 3.x:

	Windows	OS X/mac OS	Linux
<b>Versió OS</b>	Windows 10/8/7 (32- o 64-bit)	Mac OS X 10.10 (Yosemite) o superior, fins a 10.13 ( macOS High Sierra)	GNOME o KDE desktop

<b>RAM</b>	4GB RAM mínim, 8GB RAM recomanat més 1GB adicional per a l'emulador d'Android		
<b>Espai d'emmagatzematge</b>	2GB per a Android Studio, 4GB recomanats (500MB per a l'IDE i al menys 1.5GB per Android SDK, imatges de sistema d'emulador i memòria cau).		
<b>Versió de Java</b>	Java Development Kit (JDK) 8		
<b>Resolució de pantalla</b>	1280x800 mínim, 1440x900 recomanat		

*Il·lustració 1: Requisits per a la versió 3.x*  
 Font: <https://developer.android.com/studio>

\* Per tal de tenir una bona fluïdesa, es recomana instal·lar-ho en una SSD. A més, per evitar certs problemes de compatibilitat que arribin a reportar-se amb fabricants de processadors diferents d'*Intel*, es recomana al menys un processador *Intel i5 Quad Core*. Opcionalment, es pot optar per una tarjeta gràfica de 2GB *Nvidia 1050*.

### 3.1.3. Emulador Android

Android Emulator és una eina que simula dispositius Android en una computadora per permetre provar als usuaris les seves apps en diferents dispositius i nivell d'API d'Android sense necessitat de comptar amb els dispositius físics.

Aquest emulador proporciona gairebé totes les funcions d'un dispositiu Android real, permetent:

- Simular trucades i missatges de text.
- Especificar la ubicació del dispositiu.
- Utilitzar diferents velocitats de xarxa.
- Provar sensors de rotació.
- Etc.

En alguns casos, provar la teva app a l'emulador és més ràpid i més fàcil que fer-ho en un dispositiu físic, sobretot en apps més petites. Per exemple, pots transferir dades a major velocitat a l'emulador que a un dispositiu connectat mitjançant un cable USB.

L'emulador inclou configuracions predefinides per a varis telèfons i tablets Android, Wear OS i dispositius Android TV.

No obstant, aquest emulador té requisits addicionals més enllà dels requisits bàsics per a Android Studio, que són els següents:

- SDK Tools 26.1.1 o superior.
- Processador de 64 bits.
- Windows: CPU amb suport UG (invitat sense restriccions).
- HAXM 6.2.1 o posterior (es recomana HAXM 7.2.0 o posterior).

L'ús de l'acceleració de hardware té requisits addicionals en Windows i Linux:

- Processador Intel en Windows o Linux: processador Intel compatible amb Intel VT-x, Intel EM64T (Intel 64) i la funcionalitat d'Execute Disable (XD) Bit.
- Processador AMD en Linux: processador AMD amb suport per a AMD Virtualization (AMD-V) i Supplemental Streaming SIMD Extensions 3 (SSSE3).
- Processador AMD en Windows: Android Studio 3.2 o superior i Windows 10 d'abril de 2018 o superior per a la funcionalitat Windows Hypervisor Platform (WHPX).

Per poder utilitzar l'emulador d'Android, simplement creem un Dispositiu Virtual d'Android (AVD en anglès) i executem l'app.

### 3.1.4. Llenguatges de programació

Android Studio admet els mateixos llenguatges de programació que IntelliJ, que són:

- Java
- C++
- Go
- Kotlin

Les dues millors opcions per a aquest treball eren Java i Kotlin:

- Java: és el llenguatge de programació per defecte. Al ser un llenguatge multiplataforma, és molt senzill crear aplicacions per a una determinada plataforma, i amb molt pocs canvis (o cap), fer-lo funcionar en altres sistemes operatius. El problema d'aquest llenguatge és que funciona a través de la màquina virtual de Java i, a vegades, pot portar a problemes de rendiment.
- Kotlin: és un llenguatge de programació inspirat en Java, creat per Google, per incitar als seus usuaris a crear apps escrites en aquest llenguatge. Els avantatges d'utilitzar-lo són:
  - Menys codi combinat amb una millor llegibilitat.
  - Llenguatge i entorn madurs, amb eines robustes.
  - Suport a Android Jetpack i altres llibreries. Extensions KTX.
  - Interoperabilitat amb Java, sense necessitat de migració.
  - Suport per al desenvolupament de multiplataformes, també per a iOS i web.
  - Seguretat del codi. El compilador de Kotlin detecta els errors.
  - Fàcil aprenentatge, especialment per als desenvolupadors de Java.

**En els últims anys, Kotlin no ha parat de créixer en quant a popularitat i, a dia d'avui, més del 60% de les aplicacions d'Android estan fetes amb aquest llenguatge.**

Kotlin serà probablement el llenguatge predeterminat utilitzat en el futur, i és per això que s'ha optat per utilitzar-lo en aquest treball.

- Altres llenguatges: per descomptat, també podem utilitzar els llenguatges de programació web per crear tot tipus d'aplicacions per a mòbils. Llenguatges com HTML5 o CSS3 ens ajudaran a donar forma a les nostres apps. Aquests llenguatges són els utilitzats per crear les PWA, apps que estan guanyant una gran popularitat últimament i que tindran una gran importància en el futur.

## 3.2. Google Firebase

Firebase és una plataforma pel desenvolupament d'aplicacions web i mòbils llançada el 2011 i adquirida per Google el 2014.

És una plataforma única al núvol, integrada amb *Google Cloud Platform*, que utilitza un conjunt d'eines per a la creació i sincronització de projectes que seran dotats d'alta qualitat, fent possible el creixement del nombre d'usuaris i donant resultat també a l'obtenció d'una major monetització.

S'ha utilitzat l'última versió de la plataforma, la 29.0.3.

Els desenvolupadors tindran una sèrie d'avantatges al utilitzar aquesta plataforma:

- Sincronitzar fàcilment les dades dels seus projectes sense haver d'administrar connexions o escriure lògica de sincronització complexa.
- Utilitza un conjunt d'eines multiplataforma: s'integra fàcilment tant per a plataformes web com per a aplicacions mòbils. És compatible amb grans plataformes com iOS, Android, aplicacions web, Unity i C++.
- Utilitza la infraestructura de Google i escala automàticament per qualsevol tipus d'aplicació, des de les més petites fins les més potents.
- **Crea projectes sense necessitat d'un servidor:** les eines s'inclouen en els SDK per als dispositius mòbils i web i, per tant, no és necessària la creació d'un servidor per al projecte.

Per crear un compte a Google Firebase, simplement ens haurem de registrar amb el nostre compte de Google, anar a la pàgina web de Firebase (<https://console.firebase.google.com>), i crear un projecte. Una vegada creat el nostre projecte, podrem anar incorporant tots els serveis que volguem dins dels que Firebase ens ofereix, que són molts.

En el nostre cas, Firebase estarà connectat amb l'app creada amb Android Studio. Per vincular-la, simplement haurem de prémer el botó "Agregar app", triar l'icona d'Android i seguir els passos que se'ns demanen.

**Gràcies a aquesta eina, podrem gaudir d'una BD al núvol sense la necessitat de crear un servidor ni un back-end complexe de forma totalment gratuïta sempre que no superem una quantitat exageradament gran de dades (més de 10.000 registres d'usuaris al mes o més de 10GB de dades a la BD), cosa que és molt improbable en aquest projecte.**

Tota la documentació en relació a Google Firebase es pot trobar a:

<https://firebase.google.com/docs>

No obstant, Google Firebase consta de molts més serveis que la seva base de dades. A continuació s'explicaran els serveis utilitzats en aquest projecte.

### 3.2.1. Authentication

La gran majoria d'apps necessiten identificar als usuaris. Conèixer la identitat d'un usuari permet que una app guardi les seves dades al núvol de forma segura i proporcioni la mateixa experiència personalitzada en tots els dispositius de l'usuari.

Firebase Authentication proporciona serveis de backend, SDK fàcils d'utilitzar i biblioteques de IU ja elaborades per autenticar als usuaris a la teva app.



Il·lustració 2: tipus d'autenticació a Google Firebase

Font: <https://console.firebase.google.com/u/0/project/estudiantat-importa-cmartinez/authentication/providers?hl=es>

S'admet autenticació mitjançant contrasenyes, números de telèfon, proveïdors d'identitat federada populars, com Google, Facebook, Twitter, etc., com es pot veure a la Il·lustració 2. Gràcies al servei d'autenticació ha estat possible fer els registres a l'app i els inicis de sessió directes i amb Google.

Per utilitzar aquests serveis al nostre projecte d'Android hem d'afegir les dependències als nostres arxius *build.gradle* i seguir els passos que se'ns indiquen a la documentació oficial.

Una vegada hem afegit autenticació al nostre projecte, a la pàgina web de Firebase podrem visualitzar i tractar els diferents usuaris registrats a la nostra app. Firebase ens mostrarà el correu de l'usuari i el tipus d'inici de sessió que ha utilitzat.

Identificador	Proveedores	Fecha de creación	Fecha de acceso	UID de usuario
teresa.escobet@upc.edu		30 nov. 2021	30 nov. 2021	HmorQtwcaDcPVwZvZcrHeCy71D...
albpont14@gmail.com		24 nov. 2021	24 nov. 2021	qKg6peTVFbcg3715f7iyvTm8nEy1
marta.isabel.tarres@upc.e...		17 nov. 2021	30 nov. 2021	YAYMf4bJEjhVLnlpsG5iB36QCY2
mp.gonzalezpla@gmail.com		10 nov. 2021	10 nov. 2021	OM2utsWktUPdfk9jUCGYbF54nHq1
cmartinezfuentes72@gmai...		2 nov. 2021	30 nov. 2021	nsCWfP0p1VVOqqYPlv7Vql09b3t2
carlos.mz.gz99@gmail.com		25 oct. 2021	17 nov. 2021	6bMSZWjuotNjtguWJE3LyRph2Po2
karlospont9@gmail.com		19 oct. 2021	24 nov. 2021	YKPxXK340XQ4OniHK29oVHiuM4...

Filas por página: 50 1 - 7 of 7

Il·lustració 3: comptes autenticats a Google Firebase

Font: <https://console.firebase.google.com/u/0/project/estudiantat-importa-cmartinez/authentication/users?hl=es>

És important mencionar que Google Firebase ens permetrà eliminar un usuari o restablir la seva contrasenya, però mai ens mostrarà la contrasenya de cap d'ells, degut a la seva política de privacitat.

### 3.2.2. Analytics

Google Analytics és una solució d'anàlisi il·limitada i gratuïta que ocupa un lloc central a Firebase. Analytics s'integra a diverses funcions de Firebase i proporciona una capacitat il·limitada de generar informes sobre un total de fins a 500 esdeveniments diferents que podem definir amb el SDK de Firebase.

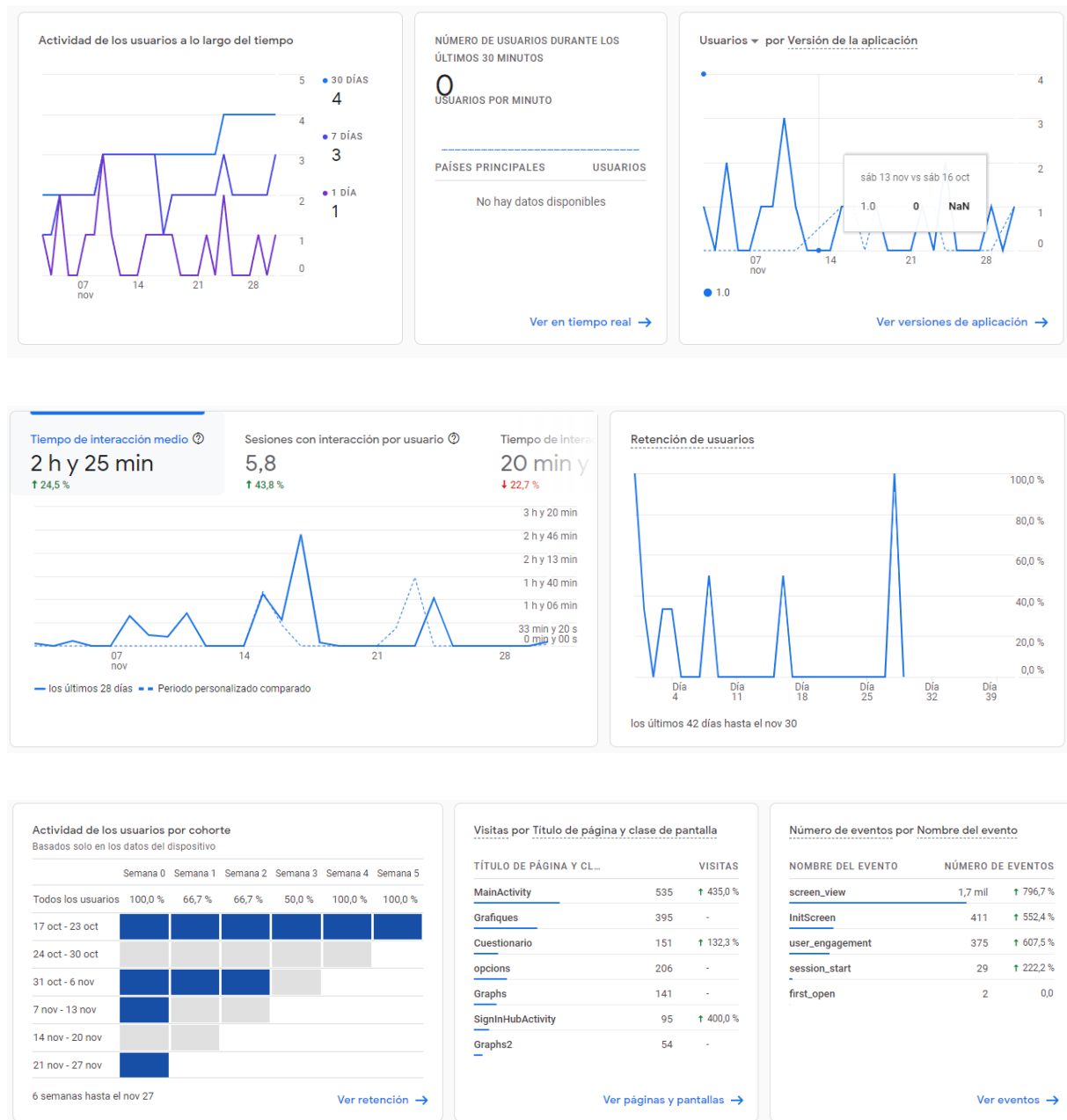
Els informes d'Analytics permeten entendre clarament com es comporten els nostres usuaris per poder prendre decisions fundamentals en relació amb el marketing de la nostra app i les optimitzacions del rendiment.

Aquesta eina ens ajuda a comprendre com les persones utilitzen la nostra app per a Apple, Android o Web. El SDK captura automàticament diversos esdeveniments i propietats de l'usuari, i també ens permet definir els nostres propis esdeveniments per tal de mesurar factors particularment importants. Una vegada capturades les dades, les podem visualitzar en un panell mitjançant Firebase console. Aquest panell proporciona estadístiques detallades sobre les dades; des de dades de resum, fins a informació més detallada, com els articles més comprats.

Analytics també s'integra amb altres funcions de Firebase i ens permet observar quins mètodes són els més eficaços per atraure usuaris a la nostra app.

Aquest servei ens permetrà visualitzar tot tipus de dades i gràfics sobre els usuaris que han utilitzat la nostra aplicació i sobre aquesta.





Il·lustració 4: analítiques de Google Firebase

Font:

<https://console.firebase.google.com/u/0/project/estudiantat-importa-cmartinez/analytics/app/android:com.martinezcarlos.estudiantatimporta/overview/~2F%3Ft%3D1641905961008&fpn%3D174842216329&swu%3D1&sgu%3D1&sus%3Dupgraded&cs%3Dapp.m.dashboard.overview&g%3D1?hl=es>

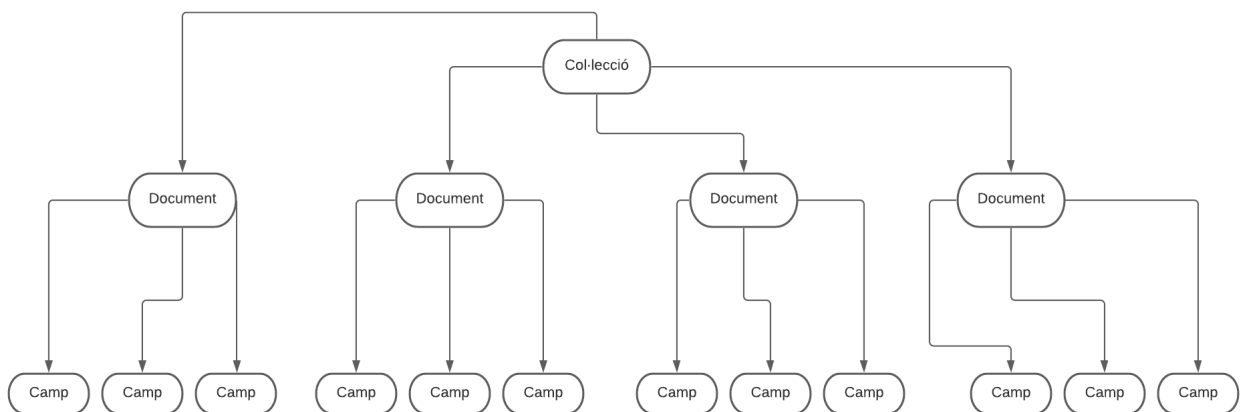
Això serà molt útil de cara a saber què podem millorar en la nostra aplicació, i quins són els seus punts forts i punts febles. A més, ens permet detectar errors que facin malbé el funcionament d'aquest per tal de solucionar-los el més aviat possible.

Per utilitzar aquests serveis al nostre projecte d'Android hem d'afegir les dependències als nostres arxius *build.gradle* i seguir els passos que se'ns indiquen a la documentació oficial.

### 3.2.3. Firestore Database

Aquest és el servei estrella de Google Firebase. Ens permet tenir una BD al núvol sense necessitat de crear servidors ni back-ends complexos i suporta altes quantitats de concurrència de dades de forma totalment gratuïta.

Aquesta és una BD NoSQL, a la qual poden accedir les nostres apps per a Apple, Android i web directament des dels SDK nadius. Cloud Firestore també està disponible als SDK nadius de Node.js, Java, Python, Unity, C++ i Go, a més de les API de REST i RPC.



Firestore Database es divideix en:

- **Documents:** són la unitat d'emmagatzematge de Firestore. Un document és un registre que utilitza pocs recursos i que conté camps amb valors assignats. Cada document s'identifica amb un nom. No hi poden haver dos documents amb el mateix nom.

\* Firestore és compatible amb diversos tipus de dades pels valors, com *booleans*, *nombres*, *strings*, *punts geogràfics*, *BLOB binaris* i *marques de temps*. A més, pots utilitzar *arrays* o *mapes*, per tal d'estructurar dades a dins d'un document.

Aquests documents són bàsicament JSON. Existeixen algunes diferències (per exemple, els documents admeten tipus de dades addicionals i el seu tamany es limita a 1MB), però en genera, es poden tractar com registres JSON.

- **Col·leccions:** els documents viuen en col·leccions, que simplement són contenidors de documents. Per exemple, podem tenir una col·lecció anomenada *users*, en la que tinguem un document que representi a cada un.  
Firestore no utilitza esquemes i, per tant, tenim llibertat total sobre els camps que posem a cada document i els tipus de dades que emmagatzemem en aquests. Els documents dins d'una mateixa col·lecció poden tenir diferents camps o emmagatzemar diferents tipus de dades en aquests. No obstant, es recomana utilitzar els mateixos camps i tipus de dades en diversos documents, per poder consultar-los de forma més senzilla.

Els noms dels documents dins d'una col·lecció són únics. Podem definir els seus ID o deixar que Firestore crei ID aleatoris de forma automàtica.

No és necessari crear ni esborrar les col·leccions. Quan es crea el primer document d'una col·lecció, aquesta passa a existir. Si borrem tots els documents d'una col·lecció, aquesta deixa d'existir.

- **Subcol·leccions:** també es poden crear subcol·leccions dins de documents i crear estructures de dades jeràrquiques que s'ajustin a escala a mesura que la BD creix. Una subcol·lecció és una col·lecció associada a un document específic. Les col·leccions i documents han de seguir sempre el mateix patró. No podem fer referència a una col·lecció dins d'una altra col·lecció ni a un document dins d'un altre document.

Les subcol·leccions ens permeten estructurar dades de forma jeràrquica, el que facilita l'accés a aquestes. Els documents de les subcol·leccions també poden contenir subcol·leccions, el que ens permet crear dades en més nivells. Podem crear dades fins a 100 nivells de profunditat.

El model de dades de Cloud Firestore admet qualsevol estructura de dades que funcioni millor amb l'aplicació. A més, les consultes de Cloud Firestore són expressives, eficients i flexibles. Es poden crear consultes superficials per recuperar dades al nivell del document, sense la necessitat de recuperar la col·lecció completa ni les subcol·leccions imbricades.

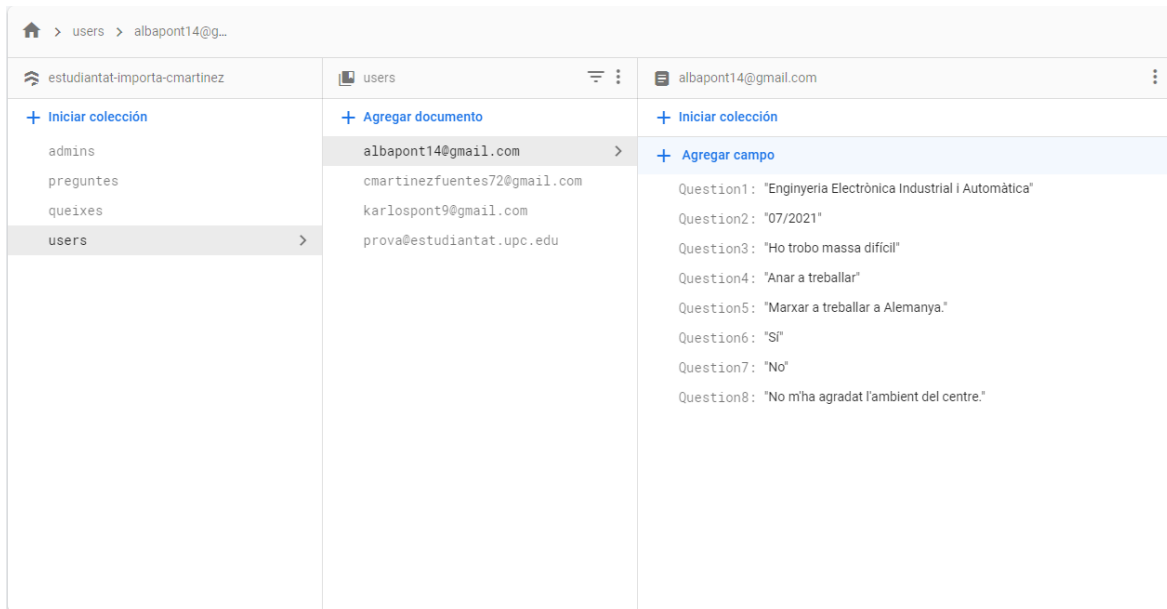
Capacitat d'afegir criteris d'ordre, filtres i límits a les consultes o cursors per paginar els resultats. Per mantenir actualitzades les dades de les aplicacions sense haver de recuperar tota la BD cada vegada que hi hagi una actualització, s'afegeixen agents d'escolta en temps real. Aquests notifiquen amb una instantània de les dades cada cop que les dades de les apps clients escolten per detectar canvis i recuperen només els canvis nous.

Una gran explicació de com funciona Cloud Firestore es pot veure al següent vídeo:

[https://www.youtube.com/watch?v=QcsAb2RR52c&list=PLI-K7zZEsYLmOF\\_07IayrTntevxtbUxDL&t=135s](https://www.youtube.com/watch?v=QcsAb2RR52c&list=PLI-K7zZEsYLmOF_07IayrTntevxtbUxDL&t=135s)

En aquest projecte s'han creat 4 col·leccions diferents:

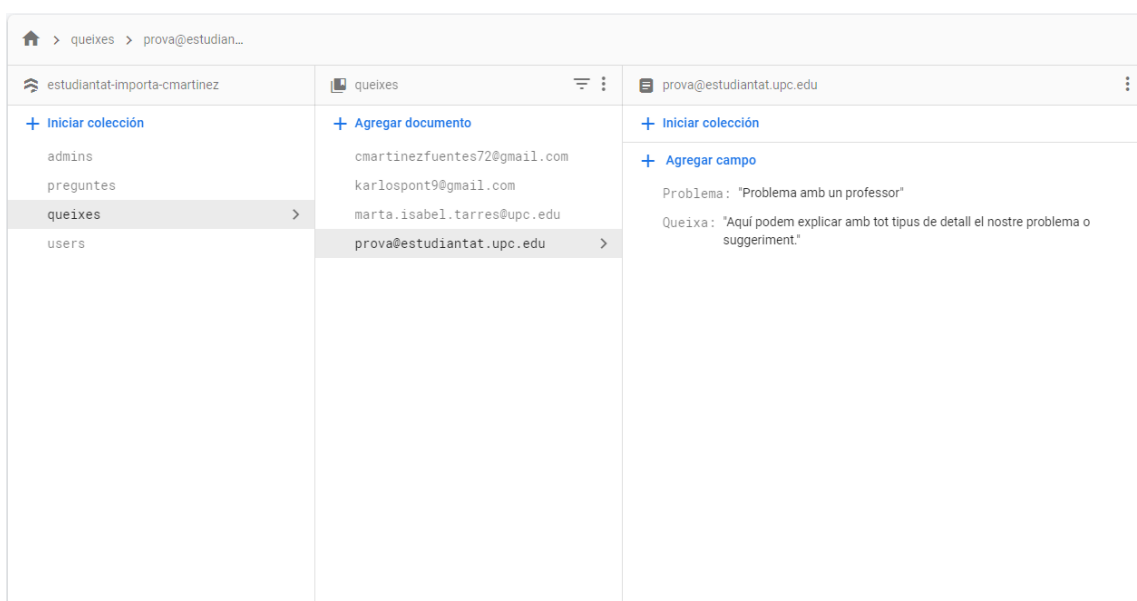
- **users:** és l'encarregada d'emmagatzemar les dades de les respostes dels qüestionaris de l'alumnat registrat a l'app, el compte dels quals estigui associat a un compte UPC.



Il·lustració 5: base de dades 'users'

Font: <https://console.firebase.google.com/u/0/project/estudiantat-importa-cmartinez/firestore/data/~2Fusers?hl=es>

- **queixes:** és l'encarregada d'emmagatzemar les dades de les respostes de les queixes de l'alumnat registrat a l'app, el compte dels quals estigui associat a un compte UPC.

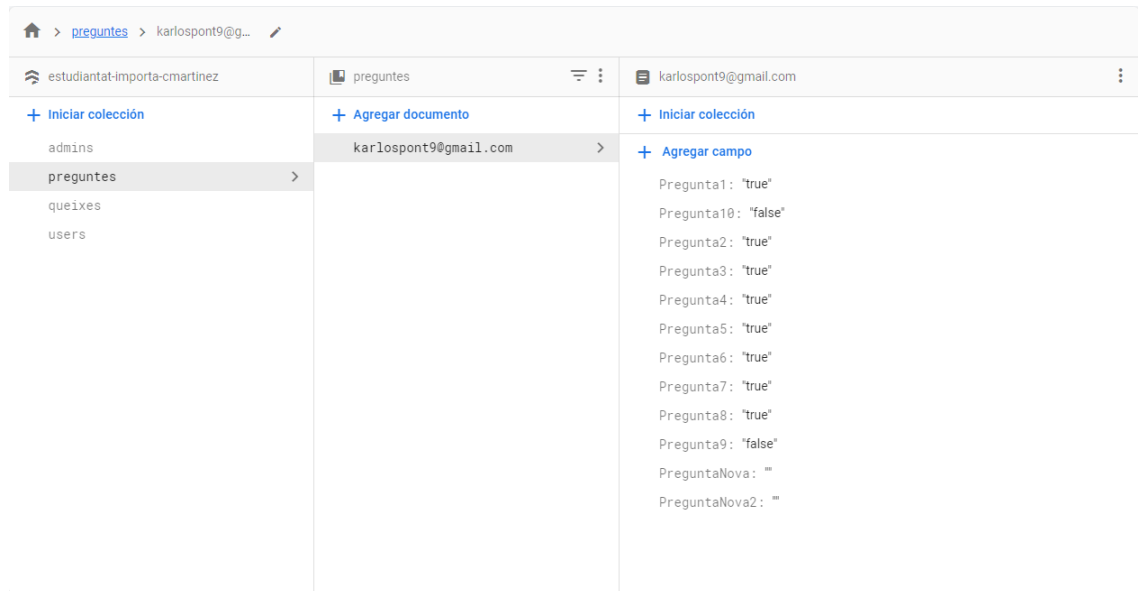


Il·lustració 6: base de dades 'queixes'

Font: <https://console.firebase.google.com/u/0/project/estudiantat-importa-cmartinez/firestore/data/~2Fqueixes?hl=es>

En ambdós casos, cada col·lecció està dividida en diferents documents (en funció del correu electrònic de l'usuari) i, a dins d'aquest, hi ha les respostes a cada pregunta.

- *preguntes*: aquesta col·lecció serveix per permetre als administradors de l'aplicació afegir i eliminar preguntes del qüestionari.



Il·lustració 7: base de dades 'preguntes'

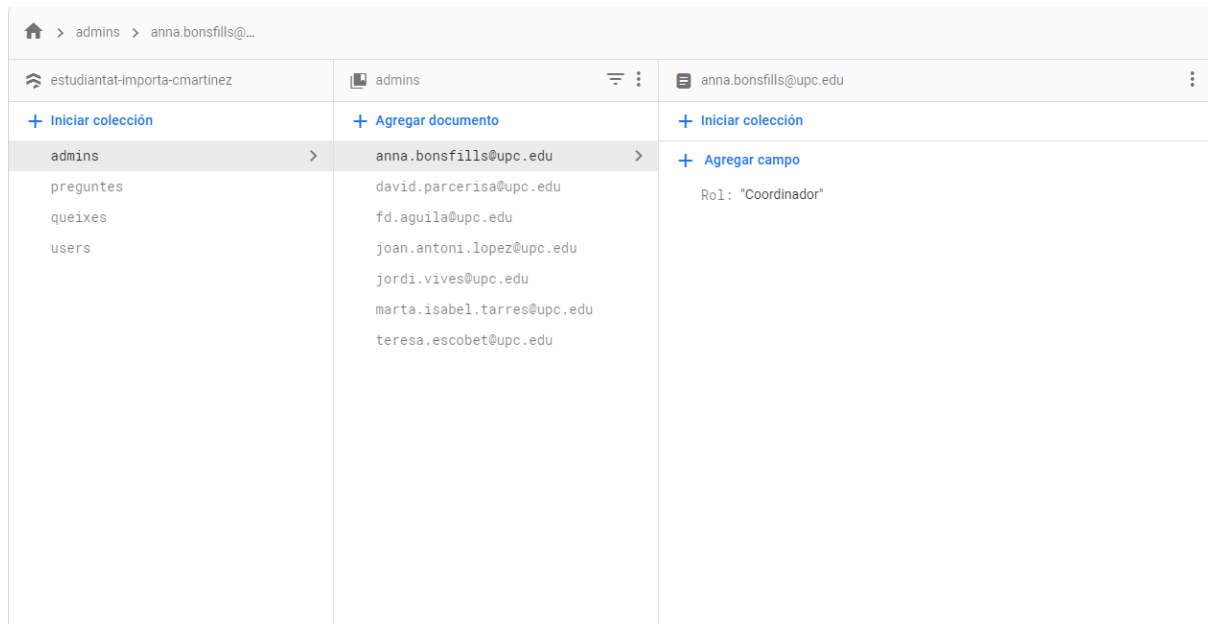
Font: <https://console.firebase.google.com/u/0/project/estudiantat-importa-cmartinez/firestore/data/~2Fpreguntes?hl=es>

En funció del valor d'aquests camps ('Pregunta1', 'Pregunta2', 'Pregunta10', etc.), en format 'Bool', les preguntes corresponents es mostraran o no en el formulari.

Aquest qüestionari consta de 8 preguntes fixes inicialment, amb la possibilitat de ser eliminades per un administrador. També es permet afegir 2 preguntes noves, les quals es veuran reflexades en els camps 'PreguntaNova' i 'PreguntaNova2'.

Aquestes modificacions seran realitzades idealment al començament de cada curs o quadrimestre, per tal de no anar modificant els gràfics, ja que el número de respostes serà diferent per a tots els alumnes, i serà molt difícil arribar a conclusions exactes.

- *admins*: en aquesta col·lecció estan registrats els professors/es i coordinadors/es de la universitat. Només l'administrador de l'app podrà afegir un rol.



Il·lustració 8: base de dades 'admins'

Font: <https://console.firebase.google.com/u/0/project/estudiantat-importa-cmartinez/firestore/data/~2Fadmins?hl=es>

Un punt important a tenir en compte és que aquestes dades són en temps real. És a dir, cada vegada que algú envia un nou formulari, la col·lecció (i el document corresponent) s'actualitza i, a la vegada, els gràfics a dins de l'aplicació també ho fan.

Per utilitzar aquests serveis al nostre projecte d'Android hem d'afegir les dependències als nostres arxius *build.gradle* i seguir els passos que se'ns indiquen a la documentació oficial.

Google Firebase ens permet eliminar en qualsevol moment qualsevol de les respostes d'alguns dels usuaris. Així, en cas d'haver-hi alguna resposta al qüestionari que no sigui adient, o el correu de la persona que l'ha respost no estigui vinculat a un compte UPC, aquesta simplement s'elimina i no es permet a aquest usuari tornar a accedir a l'aplicació.

### 3.2.4. Concurrència de dades

És important que la nostra app tingui una alta concurrència de dades, que permeti a molts usuaris accedir a l'hora i enviar formularis simultàniament sense que l'app deixi de funcionar ni el servidor caigui.

En el cas de Firebase, aquest ofereix un producte totalment gratuït amb les següents limitacions per a la seva base de dades (Firestore):

- 600.000 operacions d'escriptura de dades.
- 600.000 operacions d'esborrat de dades.
- 1.500.000 operacions de lectura de dades.

Com es pot intuir, aquestes dades són extremadament superiors a les operacions necessàries en un àmbit com una universitat, on el total d'alumnes/as no supera els 1000, i són molts menys els que valoren abandonar els estudis.

A més, Firebase té un escalat de recursos automàtic, la qual cosa optimitza al màxim la velocitat de la connexió i la utilitat dels serveis.

### **3.2.5. Seguretat**

Google ha assumit el compromís d'ajudar als seus clients a tenir èxit sota el compliment del GDPR (Reglament General de Protecció de Dades) i de la CCPA (Llei de Privacitat del Consumidor de Califòrnia), ja siguin grans empreses de software o desenvolupadors independents.

Per norma general, els clients de Firebase actuen com a "controladors de dades" (GDPR) o la "empresa" (CCPA) de les dades personals o la informació sobre els usuaris finals que proporcionen a Google en relació amb el seu ús de Firebase. Per la seva part, Google sol operar com un "processador de dades" (GDPR) o un "proveïdor de serveis" (CCPA).

Això significa que les dades estan sota el control del client. Els clients són els responsables de les seves obligacions, com complir amb els drets d'una persona física en relació amb les seves dades personals o informació.

Tots els serveis de Firebase van completar correctament els processors d'avaluació de ISO 27001, i SOC 1, SOC 2 i SOC 3. A més, alguns també van completar els processos de certificació d'ISO 27017 i d'ISO 27018.

Els serveis de Firebase encripten dades en trànsit amb HTTPS i dades de clients aïllats de manera lògica. A més, per tal de mantenir les dades personals protegides i reduir al mínim l'accés a elles, Firebase aplica mesures de seguretat exhaustives:

- Restringeix l'accés a una selecció d'empleats que tenen un fi comercial per accedir a les dades personals.
- Registra l'accés dels empleats als sistemes que contenen dades personals.
- Només permet que accedeixin a les dades personals els empleats que ho fan amb l'Accés amb Google i l'autenticació de 2 factors.

## 4. INTERFÍCIE D'USUARI

Aquesta aplicació creada amb Android Studio i Kotlin té un disseny molt agradable i estètic per a l'usuari, on algunes pantalles estan dissenyades per a tots els usuaris, i d'altres només per als administradors d'aquesta.

L'administrador de l'app serà el creador d'aquesta, jo mateix. A més, existiran els coordinadors/es i professors/es de la universitat, i la tutora del TFG. Així, els usuaris amb privilegis de l'app són:

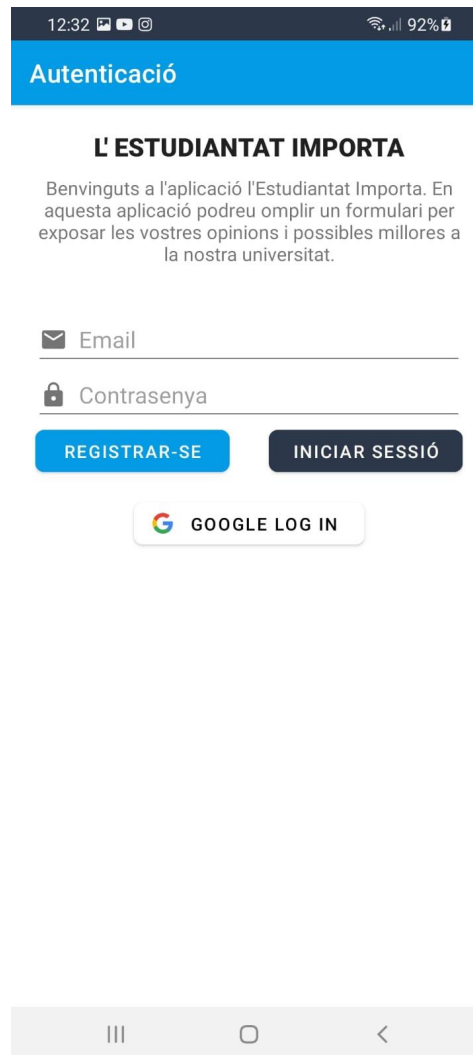
- Carlos Martínez González (administrador i propietari del TFG).
- Marta Isabel Tarres Puerta (tutora del TFG).
- Teresa Escobet Canal (coordinadora del grau en Eng. Electrònica Industrial i Automàtica).
- Joan Antoni López (coordinador del grau en Eng. Mecànica).
- Anna Bonsfills Pedrós (coordinadora del grau en Eng. Química).
- David Parcerisa Duocastella (coordinador del grau en Eng. de Recursos Minerals i el seu Reciclatge).
- Francisco del Àguila López (coordinador del grau en Eng. de Sistemes TIC).
- Jordi Vives Costa (coordinador del grau en Eng. d'Automoció).

A més, tot el professorat de la universitat serà afegit amb els seus privilegis pertinents, que es comentaran més endavant.



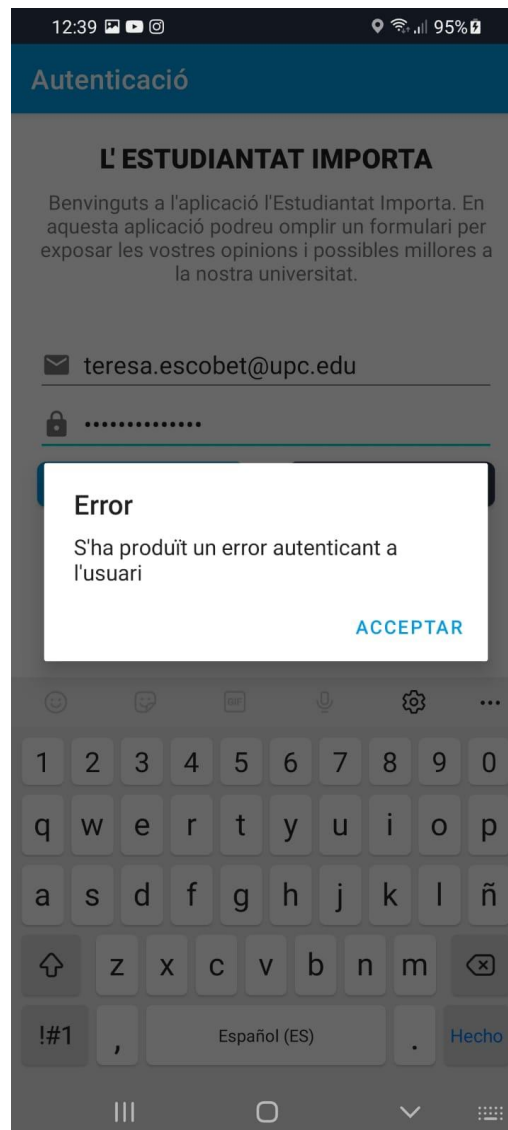
## 4.1. Pantalla principal

Una vegada instal·lem i iniciem l'aplicació, se'ns obre la pantalla principal, que té el següent disseny:



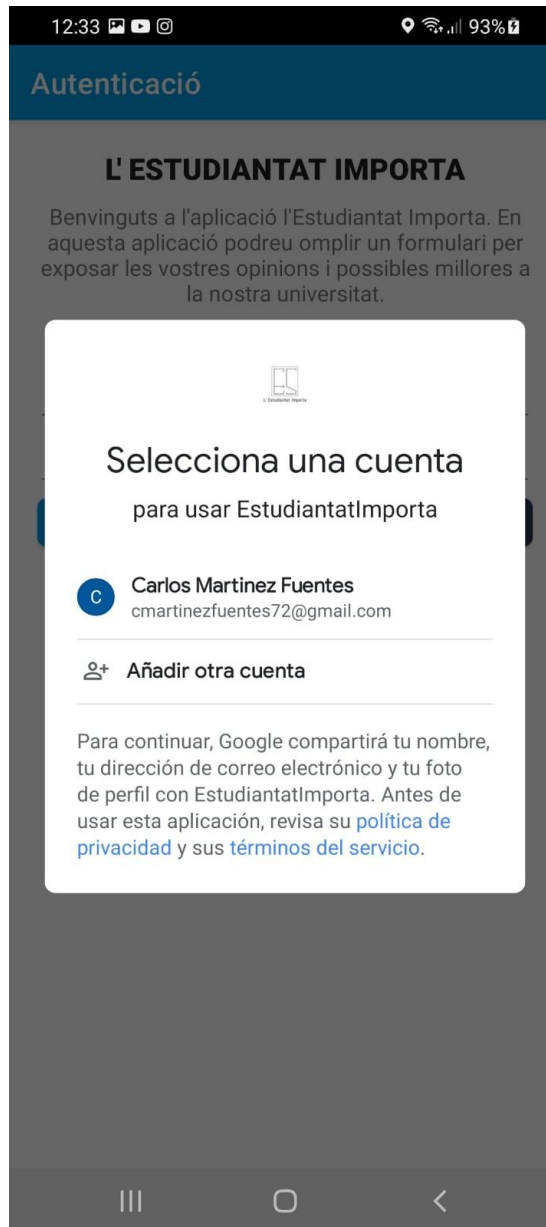
*Il·lustració 9: pantalla principal de l'aplicació*  
*Font: pròpia*

Com podem veure, aquesta pantalla consta de camps per registrar-nos a l'app amb el nostre correu electrònic i una contrasenya i per iniciar sessió una vegada ens hem registrat. Si intentem registrar un correu que ja existeix o iniciem sessió amb credencials errònies, se'ns mostrarà el següent missatge:



*Il·lustració 10: error a l'hora d'autenticar-se  
Font: pròpia*

A més, si no volem crear un compte a l'aplicació, podem iniciar sessió directament amb el nostre compte de Google lligat al nostre correu UPC:



*Il·lustració 11: inici de sessió amb un compte de Google  
Font: pròpia*

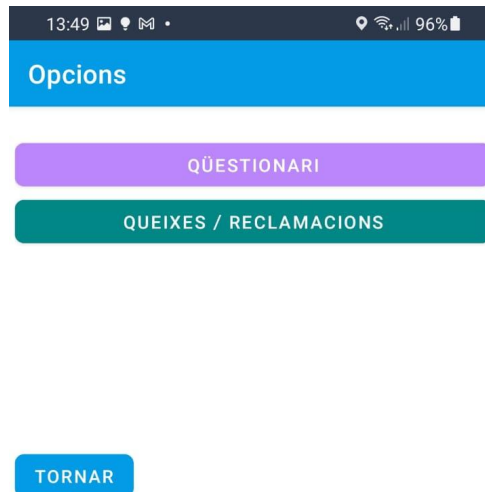
Una vegada ens registrem a l'aplicació, les dades romandran guardades i segures a Firebase, i ja podrem iniciar sessió sempre que vulguem.

Una vegada iniciem sessió a l'app, anirem a parar a la pantalla d'opcions.

## 4.2. Pantalla d'opcions

Aquesta pantalla és diferent a l'anterior, en sentit que tindrà un aspecte o un altre en funció de si som usuaris bàsics o administradors de l'app.

Els usuaris bàsics visualitzaran la pantalla de la forma següent:

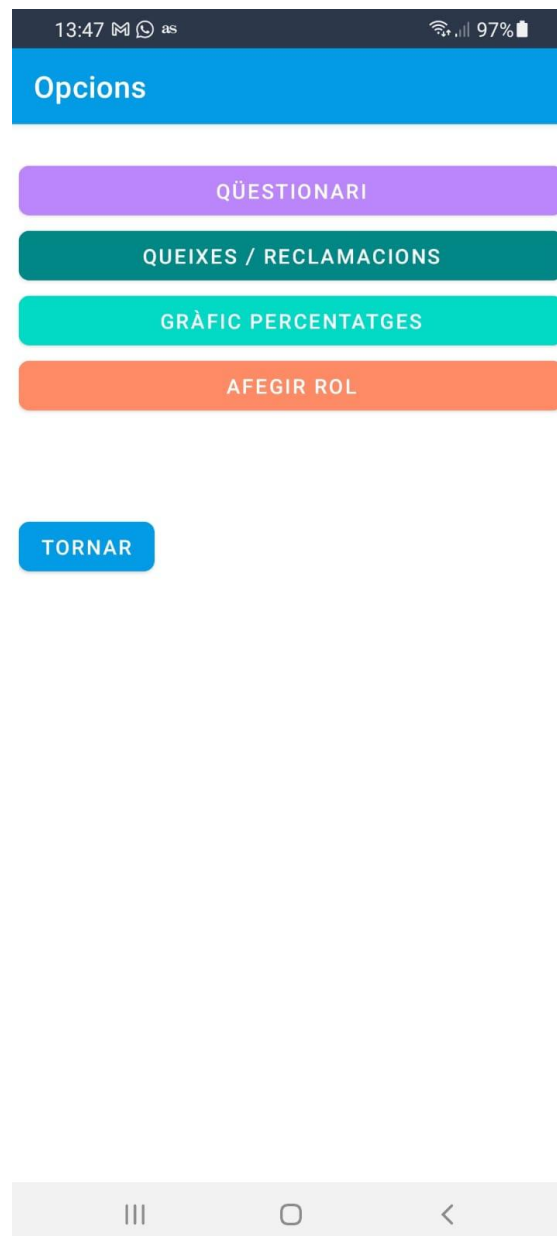


*Il·lustració 12: pantalla 'opcions' per a usuaris bàsics  
Font: pròpia*

Com podem veure, es poden visualitzar tres botons:

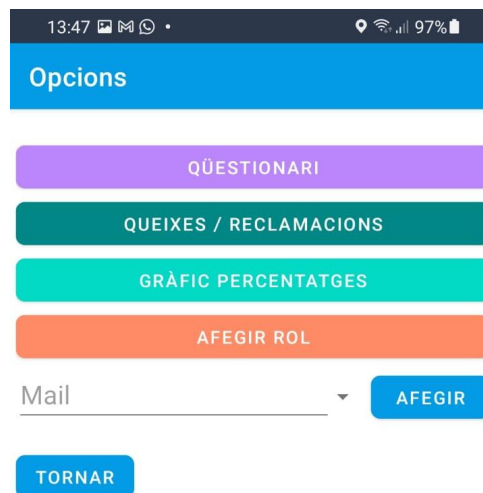
- Un botó "Començar Qüestionari", que ens porta directament a la pantalla per omplir el qüestionari d'abandonament d'estudis.
- Un botó "Queixes/Reclamacions", que ens porta directament a la pantalla per posar queixes, reclamacions o qualsevol tipus d'aportacions o consells per a la universitat, la qual tindrà totalment en compte de cara al futur.
- Un botó per tornar enrere i tancar sessió.

Els professors i coordinadors, a més d'aquests tres botons, tindran un botó extra per poder visualitzar els diferents gràfics amb els percentatges dels resultats obtinguts amb les respostes dels alumnes en el qüestionari. Per últim, els administradors de l'app tindran un botó extra per afegir rols, com es pot veure a continuació:



Il·lustració 13: pantalla 'opcions' per a usuaris administradors  
Font: pròpia

Una vegada premem el botó “Afegir rol”, ens apareixerà el següent:



*Il·lustració 14: opció d'afegir rols (només per a admins)  
Font: pròpia*

Aquí podem afegir el correu al que volem afegir-li un rol, i escollir de quin tipus, professor o coordinador.

### 4.3. Pantalla qüestionari

Aquesta pantalla és probablement la més important de l'aplicació, ja que és on l'alumnat pot omplir el qüestionari que s'enviarà a la universitat de cara a millorar tot el possible.

The image shows a mobile application interface for a questionnaire. At the top, the status bar displays the time 12:33, signal strength, Wi-Fi, and 93% battery. The app title 'Qüestionari' is in a blue header. Below it, the user's email 'cmartinezfuentes72@gmail.com' is shown. The form contains several questions with dropdown menus:

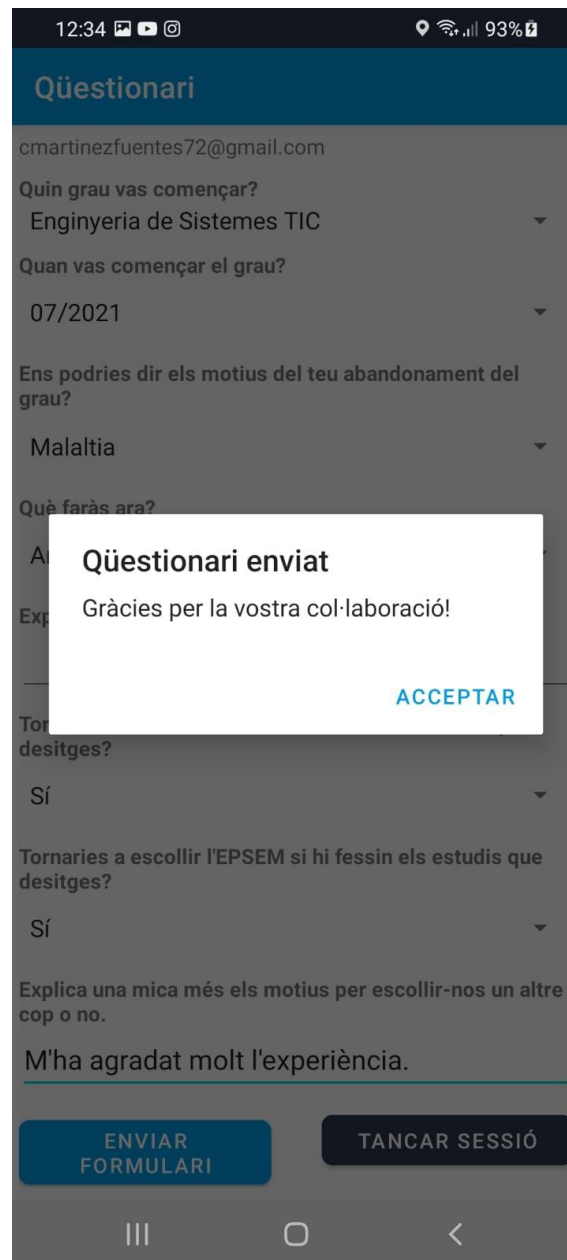
- 'Quin grau vas començar?' with the selected option 'Enginyeria de Sistemes TIC'.
- 'Quan vas començar el grau?' with the selected option '07/2021'.
- 'Ens podries dir els motius del teu abandonament del grau?' with the selected option 'Malaltia'.
- 'Què faràs ara?' with the selected option 'Anar a treballar'.
- 'Tornaries a escollir la UPC si hi fessin els estudis que desitges?' with the selected option 'Sí'.
- 'Tornaries a escollir l'EPSEM si hi fessin els estudis que desitges?' with the selected option 'Sí'.

There are two text input fields: 'Explica una mica millor què voldries fer ara.' and 'Explica una mica més els motius per escollir-nos un altre cop o no.' At the bottom, there are two buttons: 'ENVIAR FORMULARI' (blue) and 'TANCAR SESSIÓ' (dark grey). The Android navigation bar is visible at the very bottom.

Il·lustració 15: pantalla 'qüestionari' per a usuaris bàsics  
Font: pròpia

Com es pot veure, es tracta d'un qüestionari senzill però que ens indica els principals motius d'abandonament del grau i el futur d'aquests alumnes. A més, hi ha dues preguntes (5 i 8) que són de resposta oberta i permeten als estudiants explicar amb tot tipus de detall quins són els seus plans de futur i per què tornarien a escollir o no l'EPSEM com a universitat. També s'incorpora la possibilitat d'arribar a tenir fins a un màxim de 10 preguntes al qüestionari, el qual dependrà de la voluntat dels administradors.

Una vegada premem el botó “Enviar Formulari”, les dades s’enviaran directament a la BD de Firebase i se’ns mostrarà un missatge com el següent:



Il·lustració 16: missatge de qüestionari enviat  
Font: pròpia



En el cas dels administradors, aquesta pantalla es visualitza de forma diferent:

13:48 13:48 96%

## Qüestionari

Explica una mica millor què voldries fer ara.

Tornaries a escollir la UPC si hi fessin els estudis que desitges?

Sí

Tornaries a escollir l'EPSEM si hi fessin els estudis que desitges?

Sí

Explica una mica més els motius per escollir-nos un altre cop o no.

TANCAR SESSIÓ

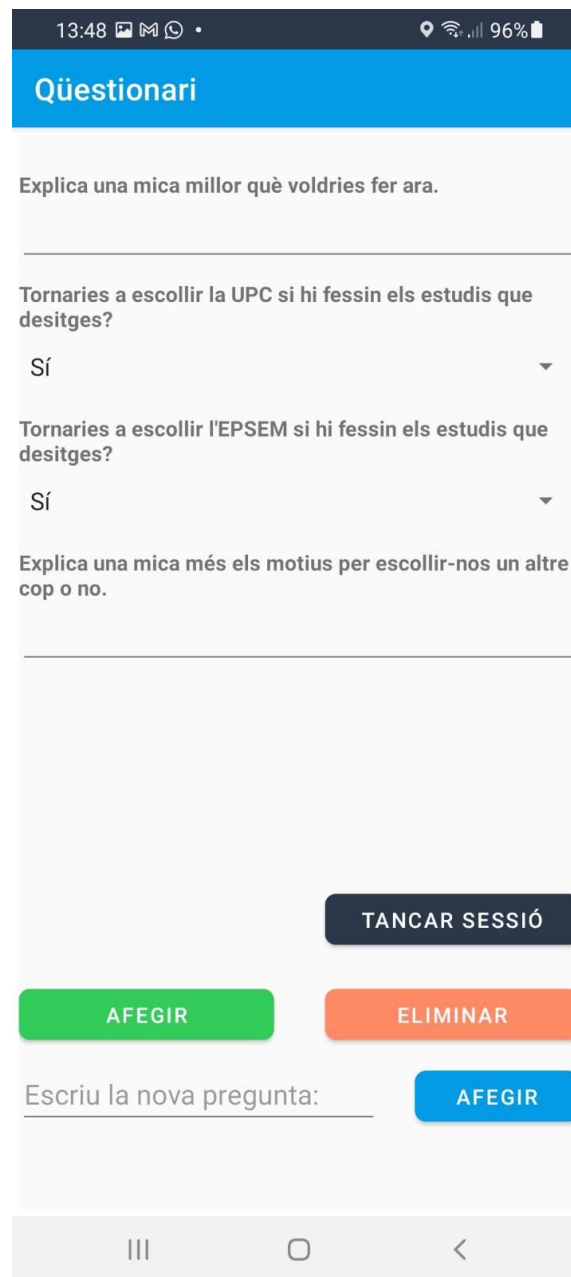
AFEGIR

ELIMINAR

*Il·lustració 17: pantalla 'qüestionari' per a administradors  
Font: pròpia*

Com es pot veure, els administradors no tenen la possibilitat d'enviar un formulari, però poden tant afegir com eliminar preguntes del qüestionari, les quals es quedaran de forma permanent.

Quan premem el botó “Afegir”, apareix el següent:

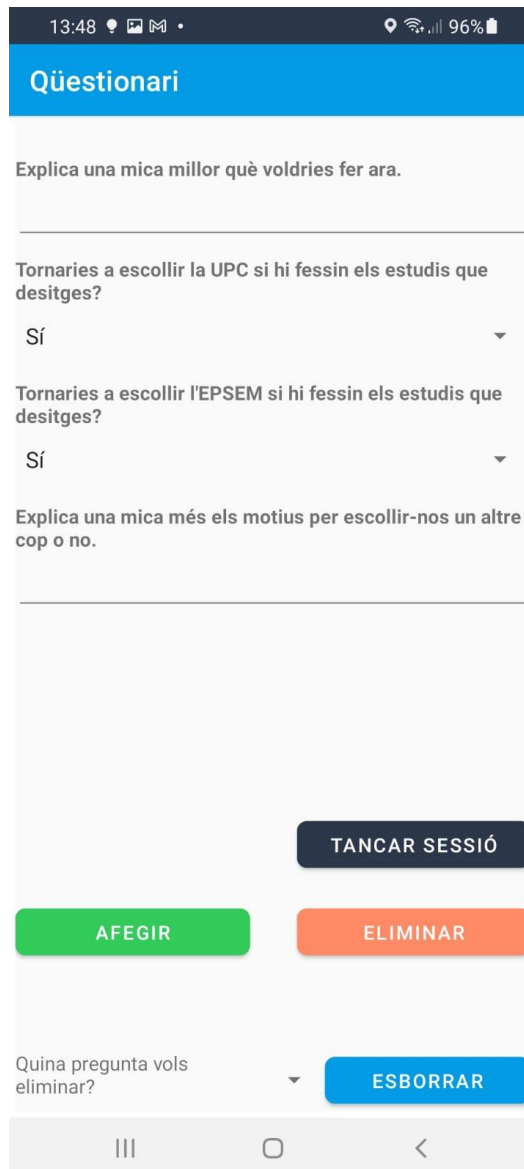


*Il·lustració 18: opció d'afegir preguntes  
Font: pròpia*

Simplement escrivim la pregunta que volem afegir i premem el botó “Afegir”. Aquesta aplicació permet esborrar cada una de les preguntes del formulari, i afegir fins a 2 preguntes noves, fins a un màxim de 10.

Si quan premem el botó “Afegir” el nombre de preguntes ja és de 10, no se'ns permetrà afegir aquesta última.

En el cas del botó “Eliminar”, quan el premem apareixerà el següent:



13:48 • 96%

### Qüestionari

Explica una mica millor què voldries fer ara.

Tornaries a escollir la UPC si hi fessin els estudis que desitges?

Sí

Tornaries a escollir l'EPSEM si hi fessin els estudis que desitges?

Sí

Explica una mica més els motius per escollir-nos un altre cop o no.

TANCAR SESSIÓ

AFEGIR ELIMINAR

Quina pregunta vols eliminar?

ESBORRAR

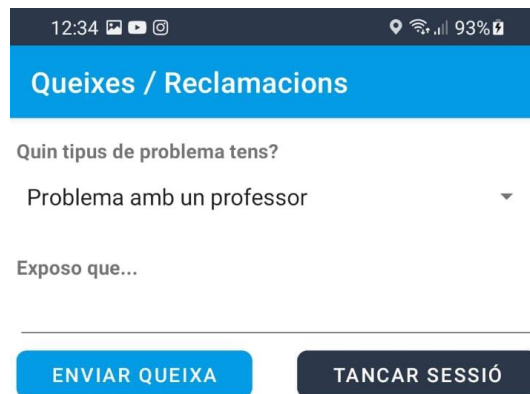
*Il·lustració 19: opció d'esborrar preguntes  
Font: pròpia*

Ens apareixerà un desplegable amb les opcions de les diferents preguntes. Simplement triem quina volem esborrar i premem el botó “Esborrar”. Aquesta pregunta desapareixerà del qüestionari, i no serà visible més.

## 4.4. Pantalla queixes/reclamacions

Aquesta és l'altre pantalla més important de l'aplicació, ja que és aquí on els usuaris poden posar queixes en relació amb el professorat, l'alumnat, la universitat, etc.

A més, també es poden enviar recomanacions i consells de forma oberta, per tal de permetre a la universitat millorar en un futur i prendre mesures quan sigui necessari.



The screenshot shows a mobile application interface for reporting issues. At the top, there is a status bar with the time 12:34, signal strength, Wi-Fi, and 93% battery. Below this is a blue header with the text 'Queixes / Reclamacions'. The main content area has a question 'Quin tipus de problema tens?' followed by a dropdown menu currently showing 'Problema amb un professor'. Below the dropdown is a text input field labeled 'Exposo que...'. At the bottom, there are two buttons: a blue button labeled 'ENVIAR QUEIXA' and a dark grey button labeled 'TANCAR SESSIÓ'.

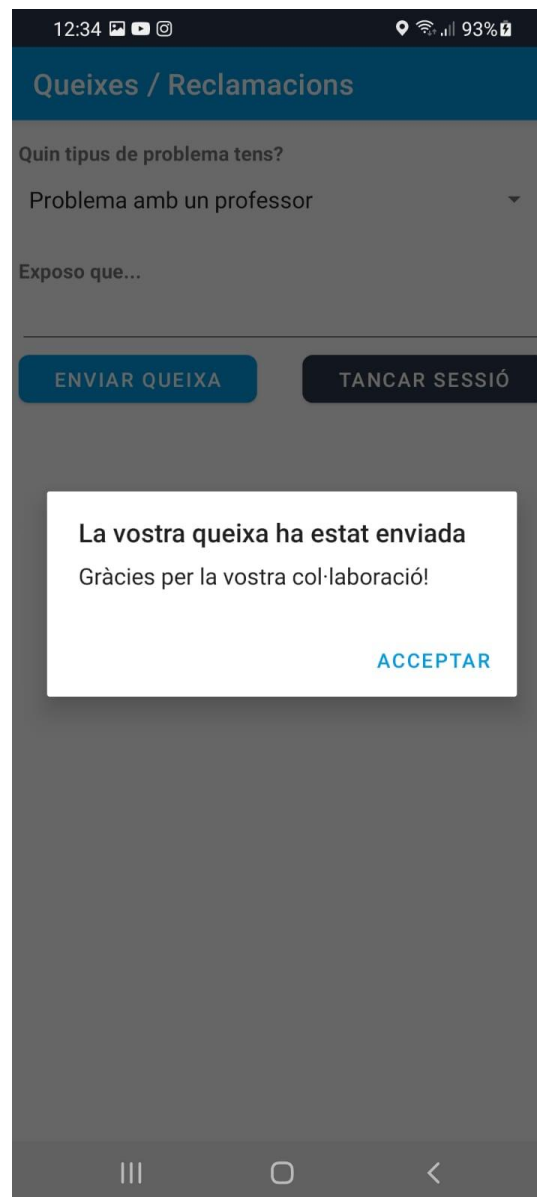


*Il·lustració 20: pantalla 'queixes' per a usuaris bàsics  
Font: pròpia*

Les diferents opcions a triar són:

- Problema amb un professor
- Problema amb una assignatura
- Problema personal
- Queixa general
- Problema amb les instal·lacions del centre
- Possible millora a la universitat
- Altres

Una vegada premem el botó “Enviar Queixa”, les dades s’enviaran directament a la BD de Firebase i se’ns mostrarà un missatge com el següent:



*Il·lustració 21: missatge de queixa enviada*  
*Font: pròpia*

En el cas dels administradors, el botó “Enviar Queixa” no estarà disponible, i no es podrà enviar una queixa.

## 4.5. Pantalla gràfics

Aquesta pantalla només serà visible per als usuaris amb privilegis de l'app, que com s'ha comentat amb anterioritat, seran els coordinadors/es i professors/es dels diferents graus cursats a l'EPSEM, a part de mi i la tutora del TFG.



*Il·lustració 22: pantalla 'gràfics' per a usuaris amb privilegis  
Font: pròpia*

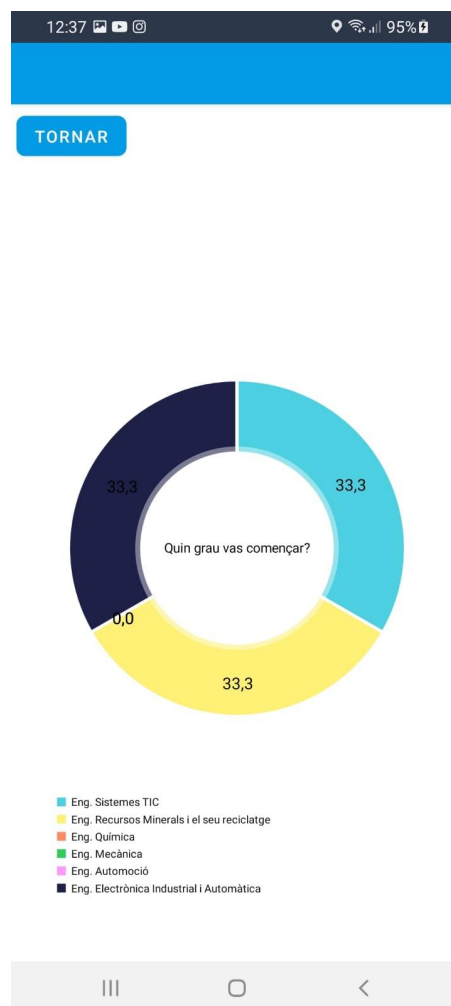
La pantalla consta de 6 botons diferents, els quals mostren un gràfic per a cada una de les 6 preguntes de resposta predefinida que hi ha al qüestionari. Aquests mostren els percentatges de respostes que ha obtingut cada possible resposta en diferents colors, els quals s'especifiquen a la llegenda.

Aquests gràfics són actualitzats en temps real i suporten concurrències de dades petites i mitjanes, suficientment grans per a aquest projecte.

#### 4.5.1. Pantalla gràfic pregunta 1

Aquesta pantalla mostra el gràfic dels diferents percentatges de les respostes obtingudes a la pregunta 1 del qüestionari, la qual era:

**“Quin grau has escollit?”**



Il·lustració 23: pantalla 'gràfic pregunta 1' per a usuaris amb privilegis

Font: pròpia

Les possibles respostes a aquesta pregunta són els 6 diferents graus que es poden cursar a l'EPSEM, que són:

- Grau en Enginyeria de Sistemes TIC
- Grau en Enginyeria de Recursos Mineral i el seu Reciclatge
- Grau en Enginyeria Química
- Grau en Enginyeria Mecànica
- Grau en Enginyeria d'Automoció
- Grau en Enginyeria Electrònica Industrial i Automàtica

### 4.5.2. Pantalla gràfic pregunta 2

Aquesta pantalla mostra el gràfic dels diferents percentatges de les respostes obtingudes a la pregunta 2 del qüestionari, la qual era:

**“Quan vas començar el grau?”**



Il·lustració 24: pantalla 'gràfic pregunta 2' per a usuaris amb privilegis  
Font: pròpia

Les possibles respostes són les dades entre el juliol de 2016 i el juliol de 2021:

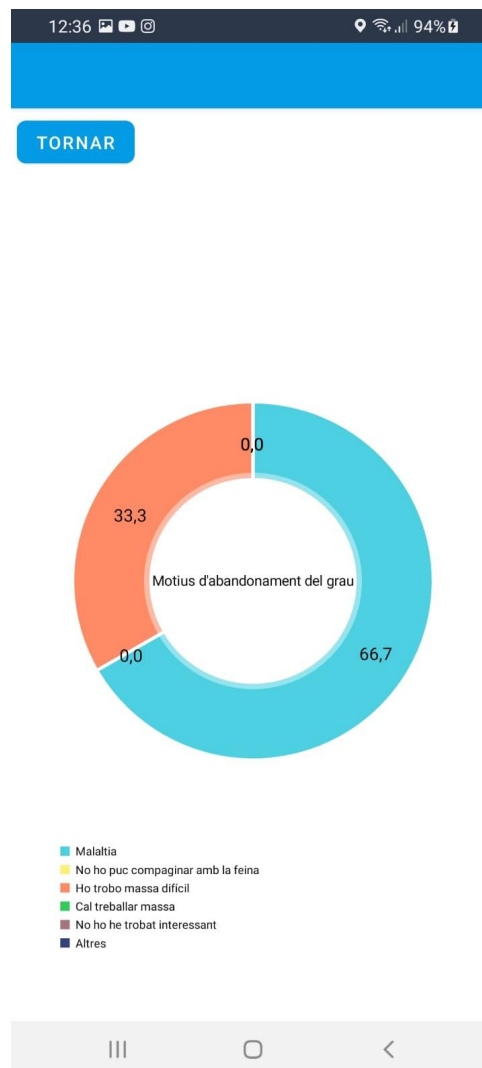
- 07/2021
- 02/2021
- 07/2020
- 02/2020
- 07/2019
- 02/2019
- 07/2018
- 02/2018
- 07/2017
- 02/2017
- 07/2016



### 4.5.3. Pantalla gràfic pregunta 3

Aquesta pantalla mostra el gràfic dels diferents percentatges de les respostes obtingudes a la pregunta 3 del qüestionari, la qual era:

#### “Motius d'abandonament del grau”



Il·lustració 25: pantalla 'gràfic pregunta 3' per a usuaris amb privilegis  
Font: pròpia

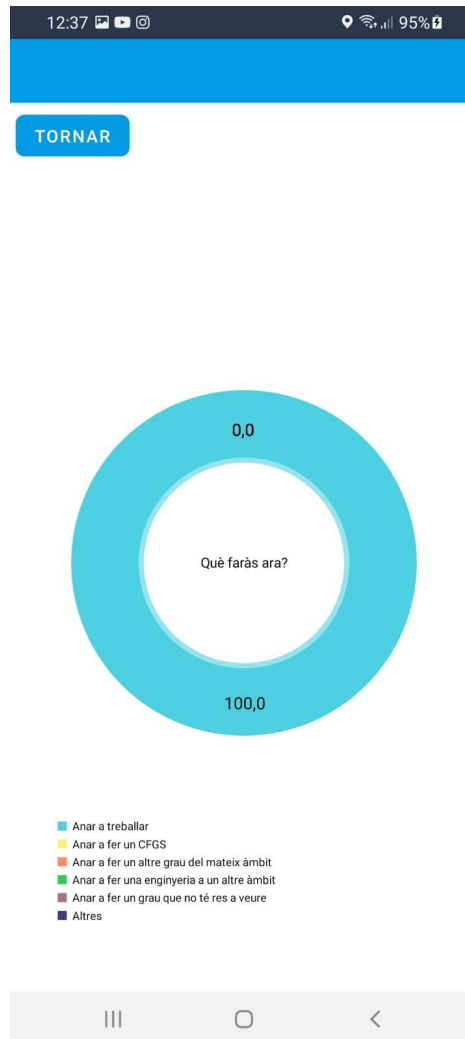
Les possibles respostes són les següents:

- Malaltia
- No ho puc compaginar amb la feina
- Ho trobo massa difícil
- Cal treballar massa
- No ho he trobat interessant
- Altres

#### 4.5.4. Pantalla gràfic pregunta 4

Aquesta pantalla mostra el gràfic dels diferents percentatges de les respostes obtingudes a la pregunta 4 del qüestionari, la qual era:

“Què faràs ara?”



Il·lustració 26: pantalla 'gràfic pregunta 4' per a usuaris amb privilegis  
Font: pròpia

Les possibles respostes són:

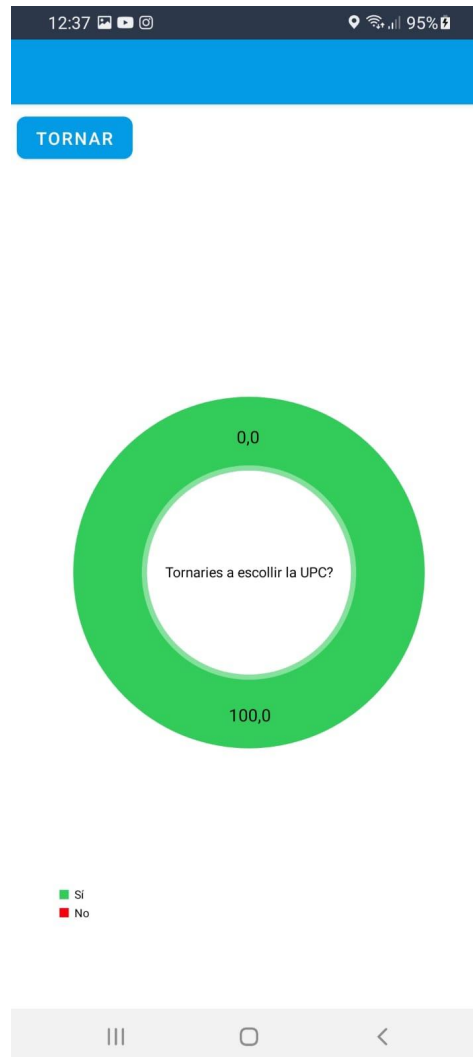
- Anar a treballar
- Anar a fer un CFGS
- Anar a fer un altre grau del mateix àmbit
- Anar a fer una enginyeria a un altre àmbit
- Anar a fer un grau que no té res a veure
- Altres

Una resposta més detallada i lliure es pot donar a la pregunta 5 del qüestionari.

### 4.5.5. Pantalla gràfic pregunta 6

Aquesta pantalla mostra el gràfic dels diferents percentatges de les respostes obtingudes a la pregunta 6 del qüestionari, la qual era:

**“Tornaries a escollir la UPC?”**



*Il·lustració 27: pantalla 'gràfic pregunta 6' per a usuaris amb privilegis  
Font: pròpia*

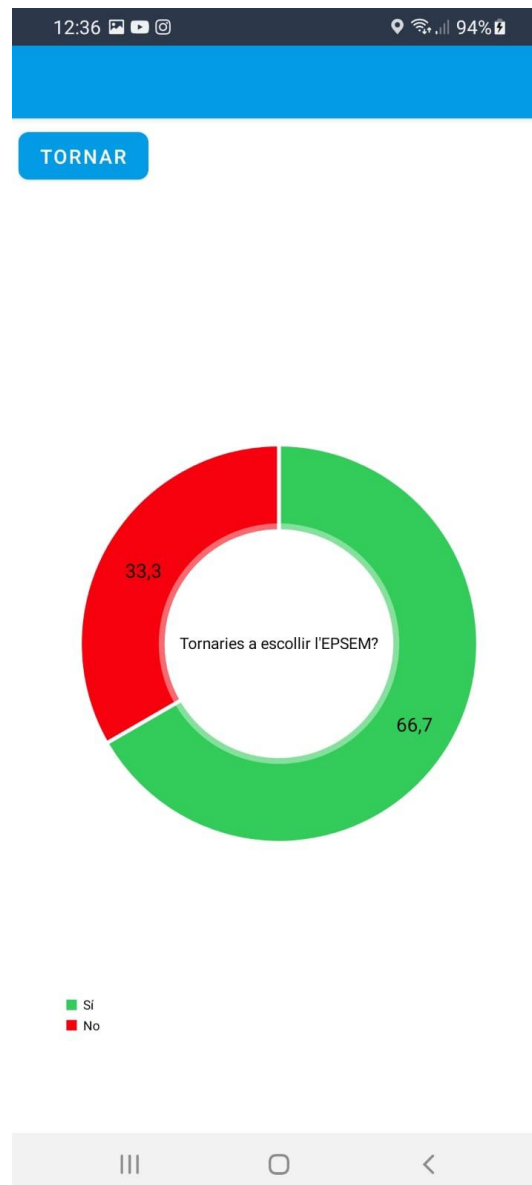
En aquest cas, aquesta pregunta només pot ser contestada amb:

- Sí
- No

#### 4.5.6. Pantalla gràfic pregunta 7

Aquesta pantalla mostra el gràfic dels diferents percentatges de les respostes obtingudes a la pregunta 7 del qüestionari, la qual era:

**“Tornaries a escollir l'EPSEM?”**



Il·lustració 28: pantalla 'gràfic pregunta 7' per a usuaris amb privilegis  
Font: pròpia

Igual que a la pregunta anterior, aquesta només pot ser contestada amb:

- Sí
- No

Una resposta més detallada i lliure es pot donar a la pregunta 8 del qüestionari.

## 5. CODI PRINCIPAL

Aquesta aplicació no hauria pogut ser desenvolupada sense un codi robust que li doni forma i sentit. En aquest punt es mostraran i s'explicaran les línies de codi més importants del projecte, ja que el total de línies és molt elevat i algunes són molt trivials.

Aquestes línies de codi estaran dividides en funció dels arxius on estiguin escrites, els quals són arxius Kotlin (.kt), ja que és aquí on es fan totes les declaracions i funcions del projecte.

En una aplicació d'Android Studio, no obstant, també existeixen els arxius .xml o layouts, que és on es dóna forma visual a les diferents pantalles. Aquest arxius, com ja s'han mostrat els dissenys de les diferents pantalles, no seran mostrats en aquest projecte, ja que manquen de valor per al funcionament d'aquest.

### 5.1. MainActivity.kt

En aquest arxiu Kotlin estan escrites les línies de codi que permeten als usuaris registrar-se i iniciar sessió en l'aplicació.

```
class MainActivity : AppCompatActivity() {  
  
    private lateinit var firebaseAnalytics: FirebaseAnalytics  
    private lateinit var auth: FirebaseAuth  
    private val GOOGLE_SIGN_IN = 100  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        //Thread.sleep(2000)  
        setTheme(R.style.Theme_EstudiantatImporta)  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
  
        //Analytics Events  
        firebaseAnalytics = Firebase.analytics  
        val bundle = Bundle()  
  
        bundle.putString("message", "Integració de Firebase completa")  
        firebaseAnalytics.logEvent("name: "InitScreen", bundle)  
        //Authentication  
        auth = FirebaseAuth  
  
        //Setup  
        setup()  
        session()  
    }  
  
    private fun session() {  
        val prefs = getSharedPreferences("com.martinezcarlos.estudiantatimporta.PREFERENCE_FILE_KEY", Context.MODE_PRIVATE)  
        val email = prefs.getString(key: "email", defValue: null)  
        val provider = prefs.getString(key: "provider", defValue: null)  
        //val authLayout = findViewById<Layout>(R.id.authLayout)  
        if (email != null && provider != null) {  
            showHome(email, ProviderType.valueOf(provider))  
        }  
    }  
}
```

Il·lustració 29: codi pantalla principal 1

Font: pròpia

Com es pot veure, ha estat necessari incloure dependències de *Google Firebase* per poder utilitzar tant les *analítiques* com *l'inici de sessió amb Google*.

En la funció *session()* definim els valors d'*email* i *provider* que seran els que ens permetran diferenciar entre un inici de sessió normal (amb un compte creat a l'app) o un de Google.

```
private fun setup() {
    title="Autenticació"

    var emailEditText = findViewById<EditText>(R.id.emailEditText)
    var passwordEditText = findViewById<EditText>(R.id.passwordEditText)
    var botonIniciarSessio = findViewById<Button>(R.id.botonIniciarSessio)
    var botonRegistrar = findViewById<Button>(R.id.botonRegistrar)
    var googleButton = findViewById<Button>(R.id.googleButton)

    botonRegistrar.setOnClickListener { @View()
        if (emailEditText.text.isNotEmpty() && passwordEditText.text.isNotEmpty()){
            FirebaseAuth.getInstance().createUserWithEmailAndPassword(emailEditText.text.toString(), passwordEditText.text.toString()).addOnCompleteListener { @Task<AuthResult>()
                if (it.isSuccessful){
                    showHome( email = it.result?.user?.email ?: "", ProviderType.BASIC)
                } else{
                    showAlert()
                }
            }
        }
    }

    botonIniciarSessio.setOnClickListener { @View()
        if (emailEditText.text.isNotEmpty() && passwordEditText.text.isNotEmpty()){
            FirebaseAuth.getInstance().signInWithEmailAndPassword(emailEditText.text.toString(), passwordEditText.text.toString()).addOnCompleteListener { @Task<AuthResult>()
                if (it.isSuccessful){
                    showHome( email = it.result?.user?.email ?: "", ProviderType.BASIC)
                } else{
                    showAlert()
                }
            }
        }
    }

    googleButton.setOnClickListener { @View()
        //Configuració
        val googleConf = GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN).requestIdToken("174842216329-gpp11boo194n7c1e7gju1nf7uugqskp.apps.googleusercontent.com").requestEmail().build()
        val googleClient = GoogleSignIn.getClient( activity this, googleConf)
        googleClient.signOut()

        startActivityForResult(googleClient.signInIntent, GOOGLE_SIGN_IN )
    }
}
```

Il·lustració 30: codi pantalla principal 2

Font: pròpia

En la funció *setup()* és on donem sentit als botons de registre d'usuaris i inici de sessió normal o amb el compte de Google.

```

private fun showAlert() {
    val builder = AlertDialog.Builder(context: this)
    builder.setTitle("Error")
    builder.setMessage("S'ha produït un error autenticant a l'usuari")
    builder.setPositiveButton(text: "Acceptar", listener: null)
    val dialog: AlertDialog = builder.create()
    dialog.show()
}

private fun showHome(email: String, provider: ProviderType) {
    val homeIntent = Intent(packageContext: this, /* Cuestionario */ options::class.java).apply { this: Intent
        putExtra(name: "email", email)
        putExtra(name: "provider", provider.name)
    }
    startActivity(homeIntent)
}

override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
    super.onActivityResult(requestCode, resultCode, data)
    if (requestCode == GOOGLE_SIGN_IN){
        val task = GoogleSignIn.getSignedInAccountFromIntent(data)

        try {
            val account = task.getResult(ApiException::class.java)

            if(account != null){
                val credential = GoogleAuthProvider.getCredential(account.idToken, accessToken: null)
                FirebaseAuth.getInstance().signInWithCredential(credential).addOnCompleteListener { it: Task<AuthResult>
                    if (it.isSuccessful){
                        showHome(email: account.email ?: "", ProviderType.GOOGLE)
                    } else{
                        showAlert()
                    }
                }
            }
        } catch (e: ApiException){
            showAlert()
        }
    }
}
}

```

Il·lustració 31: codi pantalla principal 3  
Font: pròpia

Per últim tenim la funció *showAlert()*, la qual s'encarregarà de mostrar el missatge d'error en cas necessari a l'hora d'identificar-se i la funció *showHome()* que, en cas de no haver-hi cap error, ens enviarà cap a la pantalla d'*opcions*. La funció *onActivityResult()* s'encarrega d'organitzar-ho tot.

## 5.2. opcions.kt

Una vegada hem iniciat sessió a l'aplicació, passem a la pantalla d'*opcions*. En aquesta, com s'ha vist amb anterioritat, hi ha les diferents opcions de *Començar un Qüestionari* i *Posar una Queixa* pels usuaris bàsics, *Mostrar els gràfics* per a professors/es, coordinadors/es i admins i *Afegir un rol* (només per als admins).

```
class opcions : AppCompatActivity() {  
  
    private val db = FirebaseFirestore.getInstance()  
    private val TAG = "MyActivity"  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_opcions)  
  
        title = "Opcions"  
  
        val rolSpinner = findViewById<Spinner>(R.id.rolSpinner)  
        val list = resources.getStringArray(R.array.rol)  
        val adapter = ArrayAdapter<Context, this, android.R.layout.simple_spinner_item, list>  
        rolSpinner.adapter = adapter  
        rolSpinner.setVisibility(View.INVISIBLE)  
  
        val bundle = intent.extras  
        val email = bundle?.getString( key: "email")  
        val provider = bundle?.getString( key: "provider")  
        setup( email: email ?: "", provider: provider ?: "" )  
  
        //Guardat de dades  
        val prefs = getSharedPreferences("com.martinezcarlos.estudiantatimporta.PREFERENCE_FILE_KEY", Context.MODE_PRIVATE).edit()  
        prefs.putString("email",email)  
        //prefs.putString("provider",provider)  
        prefs.apply()  
  
        tipusRol( email: email ?: "" )  
    }  
}
```

*Il·lustració 32: codi pantalla 'opcions' 1  
Font: pròpia*

El primer que tenim en aquesta pantalla és el codi que ens permet diferenciar entre els 3 tipus de rols: usuaris bàsics, professors/es i coordinadors/es. Simplement consultem en la BD de Firebase quin rol té l'usuari autènticat i, en funció del rol, es mostren uns botons o uns altres.



```
private fun setup(email:String, provider: String){

    //Botons
    val btnQuestionari = findViewById<Button>(R.id.btnQuestionari)
    val btnQueixes = findViewById<Button>(R.id.btnQueixes)
    val btnGraphs = findViewById<Button>(R.id.btnGraphs)
    val btnAddAdmins = findViewById<Button>(R.id.btnAddAdmins)
    var adminEditText = findViewById<EditText>(R.id.adminEditText)
    val btnAfegir = findViewById<Button>(R.id.btnAfegir)
    val btnBack = findViewById<Button>(R.id.btnBack8)
    //Admins
    var llistaAdmins: MutableList<String> = mutableListOf("karlospont9@gmail.com")
    val rolSpinner = findViewById<Spinner>(R.id.rolSpinner)

    //Visibilitat
    btnGraphs.setVisibility(View.INVISIBLE)
    btnAddAdmins.setVisibility(View.INVISIBLE)
    adminEditText.setVisibility(View.INVISIBLE)
    btnAfegir.setVisibility(View.INVISIBLE)

    //Admin global
    if (email in llistaAdmins) {
        btnGraphs.setVisibility(View.VISIBLE)
        btnAddAdmins.setVisibility(View.VISIBLE)
    }
}
```

*Il·lustració 33: codi pantalla 'opcions' 2*

*Font: pròpia*

En el cas de l'admin de l'app, simplement hi ha escrit el meu correu, però es poden afegir nous admins en el futur, afegint més correus a la *llistaAdmins*. A continuació tenim les definicions i la visibilitat dels diferents botons, en funció del rol.

```
//Botons

btnBack.setOnClickListener { it: View!
    onBackPressed()
}

btnQuestionari.setOnClickListener { it: View!
    showHome( email: email ?: "", ProviderType.GOOGLE)
}

btnGraphs.setOnClickListener { it: View!
    val intent = Intent( packageContext: this, Grafiques::class.java)
    startActivity(intent)
}

btnQueixes.setOnClickListener { it: View!
    showHome2( email: email ?: "", ProviderType.GOOGLE)
}

btnAddAdmins.setOnClickListener { it: View!
    adminEditText.setVisibility(View.VISIBLE)
    btnAfegir.setVisibility(View.VISIBLE)
    rolSpinner.setVisibility(View.VISIBLE)
}

btnAfegir.setOnClickListener { it: View!
    db.collection( collectionPath: "admins").document(adminEditText.text.toString()).set(
        hashMapOf("Rol" to rolSpinner.selectedItem.toString())
    )
    adminEditText.setText("")
    showAlert()
}
}
```

Il·lustració 34: codi pantalla 'opcions' 3

Font: pròpia

Tot seguit tenim la funcionalitat dels diferent botons. En el cas dels botons “Enrere”, “Questionar” i “Queixes”, simplement redirigim a la pantalla pertinent. El botó “Afegir Rol”, afegeix a la BD anomenada *admins* el correu de l'usuari i el seu rol afegit (professor o coordinador) i esborra el text, enviant un missatge de *rol afegit*.

```
private fun tipusRol(email: String){
    val btnGraphs = findViewById<Button>(R.id.btnGraphs)
    var rol = ""
    db.collection( collectionPath: "admins").document(email).get().addOnSuccessListener{ documents ->
        if (documents.exists()){
            //Log.d(TAG, "${documents.id} => ${documents.data}")
            rol = documents.data?.get("Rol").toString()
            if (rol == "Professor"){
                btnGraphs.setVisibility(View.VISIBLE)
            }
            if (rol == "Coordinador"){
                btnGraphs.setVisibility(View.VISIBLE)
            }
        }
    }
    .addOnFailureListener { exception ->
        Log.w(TAG, msg: "Error getting documents: ", exception)
    }
}
}
```

Il·lustració 35: codi pantalla 'opcions' 4  
Font: pròpia

L'última funció important és la de comprovació de rol. En aquesta consultem la BD anomenada *admins* que havíem utilitzat anteriorment, i comprovem el rol de l'usuari. En cas de ser *professor* o *coordinador*, mostrem el botó per accedir als *Gràfics de percentatges*.

### 5.3. Cuestionario.kt

En aquest arxiu Kotlin tindrem, com el seu propi nom indica, tot el codi encarregat de crear el qüestionari que després enviarem cap a la BD de Firebase.

```
enum class ProviderType {
    BASIC,
    GOOGLE
}
```

Il·lustració 36: codi tipus de provider  
Font: pròpia

El primer que necessitem comprovar és quin tipus d'inici de sessió s'ha utilitzat. El *Provider* de tipus *BASIC* serà un inici de sessió amb un compte de la mateixa app, i un de tipus *GOOGLE* serà un inici amb un compte de Google.

També serà necessari connectar-nos a la BD de la forma següent:

```
private val db = FirebaseFirestore.getInstance()
```

Il·lustració 37: codi connexió amb la base de dades  
Font: pròpia

Una vegada tenim el tipus d'inici de sessió i ens hem connectat a la BD, passem a crear els diferents objectes que ens permeten crear el qüestionari:

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_cuestionario)

    val spinner1 = findViewById<Spinner>(R.id.Question1)
    val lista1 = resources.getStringArray(R.array.Question1)
    val adaptador1 = ArrayAdapter<String>(context: this, android.R.layout.simple_spinner_item, lista1)
    spinner1.adapter = adaptador1

    val spinner2 = findViewById<Spinner>(R.id.Question2)
    val lista2 = resources.getStringArray(R.array.Question2)
    val adaptador2 = ArrayAdapter<String>(context: this, android.R.layout.simple_spinner_item, lista2)
    spinner2.adapter = adaptador2

    val spinner3 = findViewById<Spinner>(R.id.Question3)
    val lista3 = resources.getStringArray(R.array.Question3)
    val adaptador3 = ArrayAdapter<String>(context: this, android.R.layout.simple_spinner_item, lista3)
    spinner3.adapter = adaptador3

    val spinner4 = findViewById<Spinner>(R.id.Question4)
    val lista4 = resources.getStringArray(R.array.Question4)
    val adaptador4 = ArrayAdapter<String>(context: this, android.R.layout.simple_spinner_item, lista4)
    spinner4.adapter = adaptador4

    val spinner6 = findViewById<Spinner>(R.id.Question6)
    val lista6 = resources.getStringArray(R.array.bool)
    val adaptador6 = ArrayAdapter<String>(context: this, android.R.layout.simple_spinner_item, lista6)
    spinner6.adapter = adaptador6

    val spinner7 = findViewById<Spinner>(R.id.Question7)
    val lista7 = resources.getStringArray(R.array.bool)
    val adaptador7 = ArrayAdapter<String>(context: this, android.R.layout.simple_spinner_item, lista7)
    spinner7.adapter = adaptador7

    val spnBorrar = findViewById<Spinner>(R.id.spnBorrar)
    val lista8 = resources.getStringArray(R.array.eliminar)
    val adaptador8 = ArrayAdapter<String>(context: this, android.R.layout.simple_spinner_item, lista8)
    spnBorrar.adapter = adaptador8
}
```

Il·lustració 38: codi pantalla 'qüestionari' 1

Font: pròpia

Per a les preguntes de resposta múltiple, s'han utilitzat spinners, que aniran assignats a les diferents llistes d'opcions definides a l'arxiu *string.xml*. Així, per a cada pregunta tindrem una llista amb les diferents opcions possibles:

```
<string-array name="Question1">
  <item>Enginyeria de Sistemes TIC</item>
  <item>Enginyeria de Recursos Minerals i el seu Reciclatge</item>
  <item>Enginyeria Química</item>
  <item>Enginyeria Mecànica</item>
  <item>Enginyeria en Automoció</item>
  <item>Enginyeria Electrònica Industrial i Automàtica</item>
</string-array>

<string-array name="Question2">
  <item>07/2021</item>
  <item>02/2021</item>
  <item>07/2020</item>
  <item>02/2020</item>
  <item>07/2019</item>
  <item>02/2019</item>
  <item>07/2018</item>
  <item>02/2018</item>
  <item>07/2017</item>
  <item>02/2017</item>
  <item>07/2016</item>
</string-array>

<string-array name="Question3">
  <item>Malaltia</item>
  <item>No ho puc compaginar amb la feina</item>
  <item>Ho trobo massa difícil</item>
  <item>Cal treballar massa</item>
  <item>No ho he trobat interessant</item>
  <item>Altres</item>
</string-array>
```

Il·lustració 39: codi arxiu 'strings.xml' 1  
Font: pròpia

El primer que hem de fer és comprovar si l'usuari és un administrador:

```
if (email in llistaAdmins){
    btnEnviar.setVisibility(View.INVISIBLE)
    btnAfegirQuest.setVisibility(View.VISIBLE)
    btnEliminarQuest.setVisibility(View.VISIBLE)
}
```

Il·lustració 40: codi pantalla 'qüestionari' 2  
Font: pròpia

En aquest cas, mostrem els botons “Afegir pregunta” i “Esborrar pregunta” i amaguem el botó “Enviar Qüestionari”.

```

var pregunta1 = ""
var pregunta2 = ""
var pregunta3 = ""
var pregunta4 = ""
var pregunta5 = ""
var pregunta6 = ""
var pregunta7 = ""
var pregunta8 = ""
var pregunta9 = ""
var pregunta10 = ""
var novaPregunta = ""
var novaPregunta2 = ""

db.collection( collectionPath: "preguntes").document( documentPath: "karlospont9@gmail.com").get().addOnSuccessListener{ documents -
    if (documents.exists()){
        pregunta1 = documents.data?.get("Pregunta1").toString()
        pregunta2 = documents.data?.get("Pregunta2").toString()
        pregunta3 = documents.data?.get("Pregunta3").toString()
        pregunta4 = documents.data?.get("Pregunta4").toString()
        pregunta5 = documents.data?.get("Pregunta5").toString()
        pregunta6 = documents.data?.get("Pregunta6").toString()
        pregunta7 = documents.data?.get("Pregunta7").toString()
        pregunta8 = documents.data?.get("Pregunta8").toString()
        pregunta9 = documents.data?.get("Pregunta9").toString()
        pregunta10 = documents.data?.get("Pregunta10").toString()

        if (pregunta1 == "true"){
            textQuest1.setVisibility(View.VISIBLE)
            Question1.setVisibility(View.VISIBLE)
        }
        if (pregunta1 == "false"){
            textQuest1.setVisibility(View.INVISIBLE)
            Question1.setVisibility(View.INVISIBLE)
        }
    }
}

```

Il·lustració 41: codi pantalla 'qüestionari' 3

Font: pròpia

Una vegada hem fet la comprovació, passem a obtenir les dades de les 10 preguntes (com a màxim) del qüestionari. Comprovem en la BD anomenada *preguntes* el valor de cada pregunta. En cas de “true”, agafem el seu valor per mostrar-lo en el qüestionari. En cas contrari, significa que un administrador ha esborrat aquesta pregunta i, per tant, no agafem el seu valor, sinó un valor predeterminat (“”).

Això ens permet esborrar preguntes del qüestionari sense fer malbé cap gràfic de percentatges.

En cas de “false”, aquesta pregunta tampoc es mostra en el formulari.

```
btnEnviar.setOnClickListener { it: View!  
  
    var enviar1 = ""  
    var enviar2 = ""  
    var enviar3 = ""  
    var enviar4 = ""  
    var enviar5 = ""  
    var enviar6 = ""  
    var enviar7 = ""  
    var enviar8 = ""  
    var enviar9 = ""  
    var enviar10 = ""  
  
    if (pregunta1 == "true") {  
        enviar1 = Question1.selectedItem.toString()  
    }  
    if (pregunta2 == "true") {  
        enviar2 = Question2.selectedItem.toString()  
    }  
}
```

Il·lustració 42: codi pantalla 'qüestionari' 4  
Font: pròpia

```
if (pregunta9 == "true") {  
    enviar9 = Question9.text.toString()  
}  
if (pregunta10 == "true") {  
    enviar10 = Question10.text.toString()  
}  
  
db.collection( collectionPath: "users").document(email).set(  
    hashMapOf(  
        "Question1" to enviar1,  
        "Question2" to enviar2,  
        "Question3" to enviar3,  
        "Question4" to enviar4,  
        "Question5" to enviar5,  
        "Question6" to enviar6,  
        "Question7" to enviar7,  
        "Question8" to enviar8,  
        "Question9" to enviar9,  
        "Question10" to enviar10  
    )  
)
```

Il·lustració 43: codi pantalla 'qüestionari' 5  
Font: pròpia

En el cas del botó “btnEnviar” (Enviar Qüestionari), amb els valors que acabem d’agafar de la BD anomenada *preguntes*, si aquests són “true”, agafarem el valor de la resposta de l’usuari a cada pregunta. En cas contrari, vol dir que aquesta pregunta no ha de ser contestada i, per tant, s’enviarà un valor predeterminat (“”). Una vegada hem comprovat el valor de les 10 preguntes, les enviarem cap a la BD anomenada *users*, que és d’on agafarem les dades per fer els gràfics de percentatges.

```
//Correu TIC
if (Question1.selectedItem.toString() == "Enginyeria de Sistemes TIC") {
    if (pregunta9 == "false" && pregunta10 == "false") {
        val mail = SendMail(
            correu,
            contrasenya,
            correu,
            subject: "L' Estudiantat Importa - Nou formulari - ENG. SISTEMES TIC",
            message: "L'usuari " + "$email" + " ha omplert un nou formulari amb les dades següents:\n\n" +
                "\n" + "Si us plau, poseu-vos en contacte amb l'alumne/a."
        )
        val mailTIC = SendMail(
            correu,
            contrasenya,
            email: "fd.aguila@upc.edu",
            subject: "L' Estudiantat Importa - Nou formulari - ENG. SISTEMES TIC",
            message: "L'usuari " + "$email" + " ha omplert un nou formulari amb les dades següents:\n\n" +
                "\n" + "Si us plau, poseu-vos en contacte amb l'alumne/a."
        )
        mail.execute()
        mailTIC.execute()
    } else if (pregunta9 == "true" && pregunta10 == "false") {
        val mail = SendMail(
            correu,
            contrasenya,
            correu,
            subject: "L' Estudiantat Importa - Nou formulari - ENG. SISTEMES TIC",
            message: "L'usuari " + "$email" + " ha omplert un nou formulari amb les dades següents:\n\n" +
                "\n" + "Si us plau, poseu-vos en contacte amb l'alumne/a."
        )
        val mailTIC = SendMail(
            correu,
            contrasenya,
            email: "fd.aguila@upc.edu",
            subject: "L' Estudiantat Importa - Nou formulari - ENG. SISTEMES TIC",
            message: "L'usuari " + "$email" + " ha omplert un nou formulari amb les dades següents:\n\n" +
                "\n" + "Si us plau, poseu-vos en contacte amb l'alumne/a."
        )
        mail.execute()
        mailTIC.execute()
    }
}
```

Il·lustració 44: codi pantalla ‘qüestionari’ 6

Font: pròpia



```

} else if (pregunta9 == "false" && pregunta10 == "true") {
    val mail = SendMail(
        correu,
        contrasenya,
        correu,
        subject: "L' Estudiantat Importa - Nou formulari - ENG. SISTEMES TIC",
        message: "L'usuari " + "$email" + " ha omplert un nou formulari amb les dades següents:\n\n" +
            "\n" + "Si us plau, poseu-vos en contacte amb l'alumne/a."
    )
    val mailTIC = SendMail(
        correu,
        contrasenya,
        email: "fd.aguila@upc.edu",
        subject: "L' Estudiantat Importa - Nou formulari - ENG. SISTEMES TIC",
        message: "L'usuari " + "$email" + " ha omplert un nou formulari amb les dades següents:\n\n" +
            "\n" + "Si us plau, poseu-vos en contacte amb l'alumne/a."
    )
    mail.execute()
    mailTIC.execute()
} else if (pregunta9 == "true" && pregunta10 == "true") {
    val mail = SendMail(
        correu,
        contrasenya,
        correu,
        subject: "L' Estudiantat Importa - Nou formulari - ENG. SISTEMES TIC",
        message: "L'usuari " + "$email" + " ha omplert un nou formulari amb les dades següents:\n\n" +
            "\n" + "Si us plau, poseu-vos en contacte amb l'alumne/a."
    )
    val mailTIC = SendMail(
        correu,
        contrasenya,
        email: "fd.aguila@upc.edu",
        subject: "L' Estudiantat Importa - Nou formulari - ENG. SISTEMES TIC",
        message: "L'usuari " + "$email" + " ha omplert un nou formulari amb les dades següents:\n\n" +
            "\n" + "Si us plau, poseu-vos en contacte amb l'alumne/a."
    )
    mail.execute()
    mailTIC.execute()
}
}

```

Il·lustració 45: codi pantalla 'qüestionari' 7

Font: pròpia

Una funcionalitat afegida a l'app és que, una vegada s'envia un nou formulari, s'enviarà un correu al coordinador del grau, el qual s'ha triat en la *Pregunta 1*. És a dir, si un alumne/a que està cursant "Enginyeria de Sistemes TIC" envia un formulari, s'enviarà un correu al coordinador/a d'aquest grau, en aquest cas, a "[fd.aguila@upc.edu](mailto:fd.aguila@upc.edu)", amb el següent format:

L' Estudiantat Importa - Nou formulari - ENG. SISTEMES TIC Safata d'entrada x

estudiantatimporta@gmail.com

per a mi ▾

L'usuari [prova@estudiantat.upc.edu](mailto:prova@estudiantat.upc.edu) ha omplert un nou formulari amb les dades següents:

Titulació: Enginyeria de Sistemes TIC.

Data començament: 02/2019.

Motiu d'abandonament: No ho puc compaginar amb la feina.

Què faràs ara? Anar a treballar. Anar a treballar a Alemanya.

Tomaria a escollir la UPC? Sí.

Tomaria a escollir l'EPSEM? Sí. Per què? M'he sentit molt còmode.

Quants anys tens? 21.

Tens carnet de conduir? Sí.

Si us plau, poseu-vos en contacte amb l'alumne/a.



Il·lustració 46: format correu cap als coordinadors dels diferents graus 1

Font: pròpia

Així, el coordinador del grau pot veure les respostes d'aquest alumne/a i posar-se directament en contacte per intentar que reconsideri la seva estada a la universitat o, almenys, comprovar quin ha estat el motiu del seu abandonament per poder canviar o solucionar el que sigui necessari.

En cas que una pregunta hagi estat esborrada del qüestionari, aquesta no serà enviada.

```

btnEsborrar.setOnClickListener { it:View!
    if (spnBorrar.selectedItem.toString() == "Pregunta 1"){
        Question1.setVisibility(View.INVISIBLE)
        textQuest1.setVisibility(View.INVISIBLE)
        db.collection( collectionPath: "preguntes").document( documentPath: "karlospont9@gmail.com").update( field: "Pregunta1", value: "false")
        showAlert("Pregunta 1 eliminada")
    }
    if (spnBorrar.selectedItem.toString() == "Pregunta 2"){
        Question2.setVisibility(View.INVISIBLE)
        textQuest2.setVisibility(View.INVISIBLE)
        db.collection( collectionPath: "preguntes").document( documentPath: "karlospont9@gmail.com").update( field: "Pregunta2", value: "false")
        showAlert("Pregunta 2 eliminada")
    }
    if (spnBorrar.selectedItem.toString() == "Pregunta 3"){
        Question3.setVisibility(View.INVISIBLE)
        textQuest3.setVisibility(View.INVISIBLE)
        db.collection( collectionPath: "preguntes").document( documentPath: "karlospont9@gmail.com").update( field: "Pregunta3", value: "false")
        showAlert("Pregunta 3 eliminada")
    }
}

```

Il·lustració 47: codi pantalla 'qüestionari' 8

Font: pròpia

En el cas del botó “Esborrar pregunta”, es modificarà el valor de la pregunta seleccionada a la BD anomenada *preguntes*, que és la que es comprovava anteriorment per mostrar les preguntes al qüestionari i enviar-els cap a la BD de *users*, i es posarà en “false”.

```
btnAddQuest.setOnClickListener { it: View!
    if (pregunta9 == "false") {
        db.collection( collectionPath: "preguntes").document( documentPath: "karlospont9@gmail.com").set(
            hashMapOf(
                "Pregunta1" to pregunta1,
                "Pregunta2" to pregunta2,
                "Pregunta3" to pregunta3,
                "Pregunta4" to pregunta4,
                "Pregunta5" to pregunta5,
                "Pregunta6" to pregunta6,
                "Pregunta7" to pregunta7,
                "Pregunta8" to pregunta8,
                "Pregunta9" to "true",
                "Pregunta10" to pregunta10,
                "PreguntaNova" to addQuestEditText.text.toString(),
                "PreguntaNova2" to novaPregunta2
            )
        )
        textQuest9.text = addQuestEditText.text.toString()
        novaPregunta = addQuestEditText.text.toString()
        textQuest9.setVisibility(View.VISIBLE)
        Question9.setVisibility(View.VISIBLE)
        addQuestEditText.setText("")
        showAlert("Nova pregunta afegida")
        pregunta9 = "true"
    }
}
```

Il·lustració 48: codi pantalla 'qüestionari' 9  
Font: pròpia

```

else{
  if (pregunta10 == "false"){
    db.collection( collectionPath: "preguntes").document( documentPath: "karlospont9@gmail.com").set(
      hashMapOf(
        "Pregunta1" to pregunta1,
        "Pregunta2" to pregunta2,
        "Pregunta3" to pregunta3,
        "Pregunta4" to pregunta4,
        "Pregunta5" to pregunta5,
        "Pregunta6" to pregunta6,
        "Pregunta7" to pregunta7,
        "Pregunta8" to pregunta8,
        "Pregunta9" to pregunta9,
        "Pregunta10" to "true",
        "PreguntaNova" to novaPregunta,
        "PreguntaNova2" to addQuestEditText.text.toString()
      )
    )
    textQuest10.text = addQuestEditText.text.toString()
    textQuest10.setVisibility(View.VISIBLE)
    Question10.setVisibility(View.VISIBLE)
    novaPregunta2 = addQuestEditText.text.toString()
    addQuestEditText.setText("")
    showAlert("Nova pregunta afegida")
    pregunta9 = "true"
    pregunta10 = "true"
  }
  else{
    addQuestEditText.setText("")
    showAlert2("Nombre màxim de preguntes!")
    pregunta9 = "true"
    pregunta10 = "true"
  }
}
}

```

Il·lustració 49: codi pantalla 'qüestionari' 10  
Font: pròpia

Per últim, el botó "Afegir pregunta" comprovarà si ja existeix alguna pregunta extra (9 i/o 10) i, en cas de no superar el límit (10), canviarà el valor de la que quedi disponible a "true", enviant el valor de la pregunta cap a la variable *PreguntaNova* o *PreguntaNova2*. En cas d'haver arribat al límit de preguntes, s'enviarà un missatge dient que no es poden afegir més preguntes, i aquesta no es tindrà en compte.

## 5.4. Queixes.kt

Aquest és l'arxiu Kotlin encarregat de la pantalla de *Queixes / Reclamacions* que s'enviaran cap a la BD de Firebase.

El primer que hem de fer, com a la pantalla de *Qüestionari*, és connectar-nos a la BD de la forma següent:

```
private val db = FirebaseFirestore.getInstance()
```

*Il·lustració 50: codi connexió amb la base de dades*

*Font: pròpia*

Una vegada estem connectats, passem a crear els diferents objectes que donaran utilitat:

```
private fun setup(email: String, provider: String) {  
    //var emailTextView = findViewById<TextView>(R.id.emailTextView)  
    var spnProblema = findViewById<Spinner>(R.id.spnProblema)  
    val btnEnviarQueixa = findViewById<Button>(R.id.btnEnviarQueixa)  
    val btnTancarSessio = findViewById<Button>(R.id.btnTancarSessio)  
    var queixaEditText = findViewById<EditText>(R.id.queixaEditText)
```

*Il·lustració 51: codi pantalla 'queixes' 1*

*Font: pròpia*

Com hem vist abans, les preguntes amb opcions predefinides venen en un objecte de tipus *spinner*, i les diferents respostes estan definides a l'arxiu *strings.xml*:

```
<string-array name="problema">  
    <item>Problema amb un professor</item>  
    <item>Problema amb una assignatura</item>  
    <item>Problema personal</item>  
    <item>Queixa general</item>  
    <item>Problema amb les instal·lacions del centre</item>  
    <item>Possible millora a la universitat</item>  
    <item>Altres</item>  
</string-array>
```

*Il·lustració 52: codi arxiu 'strings.xml' 2*

*Font: pròpia*

Un cop tenim definits els botons, passem a donar funcionalitat:

```
btnTancarSessio.setOnClickListener { it: View!
    //Esborrat de dades
    val prefs = getSharedPreferences("com.martinezcarlos.estudiantatimporta.PREFERENCE_FILE_KEY",Context.MODE_PRIVATE).edit()
    prefs.clear()
    prefs.apply()

    FirebaseAuth.getInstance().signOut()
    onBackPressed()
}
```

*Il·lustració 53: codi pantalla 'queixes' 2*  
Font: pròpia

En el cas del botó "btnTancarSessió" (Tancar Sessió) simplement tancarem la sessió de l'usuari i tornarem a la pantalla d'opcions.

```
btnEnviarQueixa.setOnClickListener { it: View!
    db.collection( collectionPath: "queixes").document(email).set(
        hashMapOf("Problema" to spnProblema.selectedItem.toString(),
            "Queixa" to queixaEditText.text.toString())
    )
    showAlert()
}
```

*Il·lustració 54: codi pantalla 'queixes' 3*  
Font: pròpia

En el cas del botó "btnEnviarQueixa" (Enviar Queixa), simplement enviem les diferents respostes a la BD de Firebase.

```
private fun showAlert(){
    val builder = AlertDialog.Builder( context: this)
    builder.setTitle("La vostra queixa ha estat enviada")
    builder.setMessage("Gràcies per la vostra col·laboració!")
    builder.setPositiveButton( text: "Acceptar", listener: null)
    val dialog: AlertDialog = builder.create()
    dialog.show()
}
```

*Il·lustració 55: codi pantalla 'queixes' 4*  
Font: pròpia

Per últim, una vegada s'ha enviat la queixa, apareixerà un missatge de queixa enviada. En el cas dels usuaris amb privilegis, el botó per enviar queixes no serà visible.

## 5.5. Grafiques.kt

Aquest és l'arxiu Kotlin encarregat de la pantalla de visualització dels gràfics amb els percentatges de les diferents respostes a les preguntes del qüestionari. Cal recordar que aquesta pantalla només estarà disponible per als usuaris amb privilegis de l'aplicació.

Aquest arxiu simplement tindrà els 6 botons que enviaran a les 6 diferents pantalles, cada una amb el gràfic d'una de les preguntes de resposta tancada del qüestionari.

Així, l'únic que fem en aquest arxiu és definir els botons i redirigir cap a les 6 pantalles:

```
val btnQ1 = findViewById<Button>(R.id.btnQ1)
val btnQ2 = findViewById<Button>(R.id.btnQ2)
val btnQ3 = findViewById<Button>(R.id.btnQ3)
val btnQ4 = findViewById<Button>(R.id.btnQ4)
val btnQ6 = findViewById<Button>(R.id.btnQ6)
val btnQ7 = findViewById<Button>(R.id.btnQ7)
val btnGoBack = findViewById<Button>(R.id.btnBack8)
```

*Il·lustració 56: codi pantalla 'gràfiques' 1*

*Font: pròpia*

```
btnQ1.setOnClickListener { it: View!
    val intent = Intent( packageContext: this, Graphs::class.java)
    startActivity(intent)
}

btnQ2.setOnClickListener { it: View!
    val intent2 = Intent( packageContext: this, Graphs2::class.java)
    startActivity(intent2)
}

btnQ3.setOnClickListener { it: View!
    val intent3 = Intent( packageContext: this, Graphs3::class.java)
    startActivity(intent3)
}

btnQ4.setOnClickListener { it: View!
    val intent4 = Intent( packageContext: this, Graphs4::class.java)
    startActivity(intent4)
}

btnQ6.setOnClickListener { it: View!
    val intent6 = Intent( packageContext: this, Graphs6::class.java)
    startActivity(intent6)
}

btnQ7.setOnClickListener { it: View!
    val intent7 = Intent( packageContext: this, Graphs7::class.java)
    startActivity(intent7)
}

btnGoBack.setOnClickListener { it: View!
    onBackPressed()
}
```

*Il·lustració 57: codi pantalla 'gràfiques' 2  
Font: pròpia*

En el cas del botó “btnGoBack” (Tornar), simplement tornarem a la pantalla d'opcions.



## 5.6. Graphs.kt

Aquests són els arxius amb els gràfics de les respostes del qüestionari.

Per poder fer els gràfics del tipus *PieChart* s'ha utilitzat una llibreria agafada del següent enllaç:

<https://github.com/PhilJay/MPAndroidChart>

El primer que hem de fer és definir els diferents valors del gràfic:

```
private fun initGraph(){  
  
    pieChart.setUsePercentValues(true)  
    pieChart.description.text = ""  
    //Hollow pie chart  
    pieChart.isDrawHoleEnabled = false  
    pieChart.setTouchEnabled(false)  
    pieChart.setDrawEntryLabels(false)  
    //Adding padding  
    pieChart.setExtraOffsets( left: 20f, top: 0f, right: 20f, bottom: 20f)  
    pieChart.setUsePercentValues(true)  
    pieChart.isRotationEnabled = false  
    pieChart.setDrawEntryLabels(false)  
    pieChart.legend.orientation = Legend.LegendOrientation.VERTICAL  
    pieChart.legend.isWordWrapEnabled = true  
}
```

*Il·lustració 58: codi pantalla 'graphs' 1*

*Font: pròpia*

Simplement definim coses com la descripció, els marges, els tipus de dades, etc.

```

private fun setDataToChart(){
    pieChart.setUsePercentValues(true)
    val dataEntries = ArrayList<PieEntry>()
    dataEntries.add(PieEntry(opcio1.toFloat(), label: "Eng. Sistemes TIC"))
    dataEntries.add(PieEntry(opcio2.toFloat(), label: "Eng. Recursos Minerals i el seu reciclatge"))
    dataEntries.add(PieEntry(opcio3.toFloat(), label: "Eng. Química"))
    dataEntries.add(PieEntry(opcio4.toFloat(), label: "Eng. Mecànica"))
    dataEntries.add(PieEntry(opcio5.toFloat(), label: "Eng. Automoció"))
    dataEntries.add(PieEntry(opcio6.toFloat(), label: "Eng. Electrònica Industrial i Automàtica"))

    val colors: ArrayList<Int> = ArrayList()
    colors.add(Color.parseColor(colorString: "#4DD0E1"))
    colors.add(Color.parseColor(colorString: "#FFF176"))
    colors.add(Color.parseColor(colorString: "#FF8A65"))
    colors.add(Color.parseColor(colorString: "#33CC5A"))
    colors.add(Color.parseColor(colorString: "#FF99FF"))
    colors.add(Color.parseColor(colorString: "#1E2048"))

    val dataSet = PieDataSet(dataEntries, label: "")
    val data = PieData(dataSet)

    //In Percentage
    data.setValueFormatter(PercentFormatter())
    dataSet.sliceSpace = 3f
    dataSet.colors = colors
    pieChart.data = data
    data.setValueTextSize(15f)
    pieChart.setExtraOffsets(left: 5f, top: 10f, right: 5f, bottom: 20f)
    pieChart.animateY(durationMillis: 1400, Easing.EaseInOutQuad)

    //Forat al centre
    pieChart.holeRadius = 58f
    pieChart.transparentCircleRadius = 61f
    pieChart.isDrawHoleEnabled = true
    pieChart.setHoleColor(Color.WHITE)

    //Afegir text al centre
    pieChart.setDrawCenterText(true)
    pieChart.centerText = "Quin grau vas començar?"

    pieChart.invalidate()
}

```

Il·lustració 59: codi pantalla 'graphs' 2

Font: pròpia

Una vegada tenim tot definit passem a enviar les dades cap al gràfic. Definim les diferents respostes, els colors de cada una d'aquestes i li donem paràmetres varis, com el tamany del text o del forat del centre, el text del centre o el color del fons.

```
private fun obtenirDatosQ1(){
    db.collection( collectionPath: "users").whereEqualTo( field: "Question1", value: "Enginyeria de Sistemes TIC").get().addOnSuccessListener { documents ->
        for (document in documents) {
            Log.d(TAG, msg: "${document.id} => ${document.data}")
            opcio1 += 1
            initGraph()
            setDataToChart()
        }
    }.addOnFailureListener { exception ->
        Log.w(TAG, msg: "Error getting documents: ", exception)
    }
}
```

Il·lustració 60: codi pantalla 'graphs' 3

Font: pròpia

Per últim hem d'agafar les dades de cada resposta de la BD per poder fer els percentatges. Per fer això, simplement ens connectem a BD i mirem els usuaris que tenen ficada la resposta a la pregunta en cada cas. Per exemple, a la primera pregunta tenim 6 possibles solucions. Creem 6 funcions per obtenir dades, cada una amb una de les respostes possibles.

Així, la funció *obtenirDatosQ1* comprovarà si l'usuari ha respost "Enginyeria de Sistemes TIC", "Enginyeria Química", etc. i augmentarem en 1 el comptador d'aquesta opció. Una vegada ha comprovat les respostes de tots els usuaris de la BD, es faran els percentatges de respostes.

```
private fun obtenirDatosQ3(){
    db.collection( collectionPath: "users").whereEqualTo( field: "Question1", value: "Enginyeria Quimica").get().addOnSuccessListener { documents ->
        for (document in documents) {
            Log.d(TAG, msg: "${document.id} => ${document.data}")
            opcio3 += 1
            initGraph()
            setDataToChart()
        }
    }.addOnFailureListener { exception ->
        Log.w(TAG, msg: "Error getting documents: ", exception)
    }
}

private fun obtenirDatosQ4(){
    db.collection( collectionPath: "users").whereEqualTo( field: "Question1", value: "Enginyeria Mecànica").get().addOnSuccessListener { documents ->
        for (document in documents) {
            Log.d(TAG, msg: "${document.id} => ${document.data}")
            opcio4 += 1
            initGraph()
            setDataToChart()
        }
    }.addOnFailureListener { exception ->
        Log.w(TAG, msg: "Error getting documents: ", exception)
    }
}
```

Il·lustració 61: codi pantalla 'graphs' 4

Font: pròpia

## 5.7. Graphs2.kt fins a Graphs7.kt

Aquests arxius Kotlin tenen exactament la mateixa utilitat i forma que l'arxiu anterior. L'únic que variarà seran les possibles opcions a les preguntes i els colors del gràfic. La resta romandrà exactament igual.

La pregunta 5 i la 8 no tindran pantalla de gràfics ja que, al ser de resposta oberta, no és possible agrupar-les entre elles. No obstant, aquestes també s'enviaran cap a la BD i al correu cap al coordinador del grau.

## 6. ROLS

Com s'ha comentat amb anterioritat en aquest projecte, existeixen diferents rols dins de l'aplicació que donen diferents privilegis i permeten accedir a funcionalitats. Aquests són:

- *Usuari bàsic*: aquests usuaris són els alumnes de la universitat. Aquests només poden enviar un formulari i una queixa o suggeriment. No tenen cap possibilitat de visualitzar gràfiques, modificar preguntes ni afegir cap rol.
- *Professors/es*: aquests usuaris són tots els professors/es de la universitat. Aquest no podran omplir ni qüestionaris ni posar queixes/suggeriments. No obstant, podran visualitzar els gràfics dels percentatges de respostes.
- *Coordinadors/es*: aquests usuaris són els coordinador/es dels 7 graus que es poden cursar a la universitat. Tenen els mateixos privilegis que els professors/es, amb una diferència: cada vegada que un alumne/a enviï un formulari, se li enviarà un correu al coordinador/a del seu grau, amb les dades següents:

L' Estudiantat Importa - Nou formulari - ENG. SISTEMES TIC Safata d'entrada x**estudiantatimporta@gmail.com**

per a mi ▾

L'usuari [prova@estudiantat.upc.edu](mailto:prova@estudiantat.upc.edu) ha omplert un nou formulari amb les dades següents:

Titulació: Enginyeria de Sistemes TIC.

Data començament: 02/2019.

Motius d'abandonament: No ho puc compaginar amb la feina.

Què faràs ara? Anar a treballar. Anar a treballar a Alemanya.

Tornaria a escollir la UPC? Sí.

Tornaria a escollir l'EPSEM? Sí. Per què? M'he sentit molt còmode.

Quants anys tens? 21.

Tens carnet de conduir? Sí.

Si us plau, poseu-vos en contacte amb l'alumne/a.



*Il·lustració 62: codi format correu cap als coordinadors dels diferents graus 2*  
*Font: pròpia*

- **Administradors:** aquests són els usuaris amb tots els privilegis dins de l'app. Tenen els mateixos privilegis que els coordinadors/es dels graus i, a més, poden afegir rols i afegir o esborrar preguntes del formulari de forma permanent. Inicialment, l'únic administrador de l'aplicació serà el creador d'aquesta, jo mateix.

Els usuaris amb rols de professor/a i coordinador/a canviaran cada inici de curs. És a dir, abans de començar un curs, s'hauran d'assignar els 7 professors/es que tindran rols de coordinador/es, i la resta dins de la universitat es mantindran amb el de professor/a.

Aquests canvis els faran els administradors de l'aplicació en la pantalla 'opcions'. Si a un usuari que tenia el rol de professor/a li assignem el de coordinador/a (o viceversa), simplement s'actualitzarà la BD amb l'últim valor assignat.

## 7. GOOGLE PLAY STORE

S'ha decidit pujar l'aplicació a la plataforma de distribució digital d'aplicacions mòbils *Google Play Store*. D'aquesta manera, l'alumnat podrà accedir fàcilment a l'app a través del seu telèfon mòbil, com a la resta d'aplicacions.

### 7.1. Requisites

Per tal de pujar una aplicació a la Google Play Store, aquesta ha de passar una sèrie de filtres i complir amb els requisits de Google Play. D'acord amb Google, les polítiques del Programa per a Desenvolupadors i l'Acord de Distribució per a Desenvolupadors pretenen garantir l'oferta de les aplicacions més fiables i innovadores del món i fer de la Play Store una plataforma segura i respectuosa.

Per poder aconseguir això, s'han definit uns estàndards per tal de definir què és nociu i inadequat pels usuaris. Aquests són:

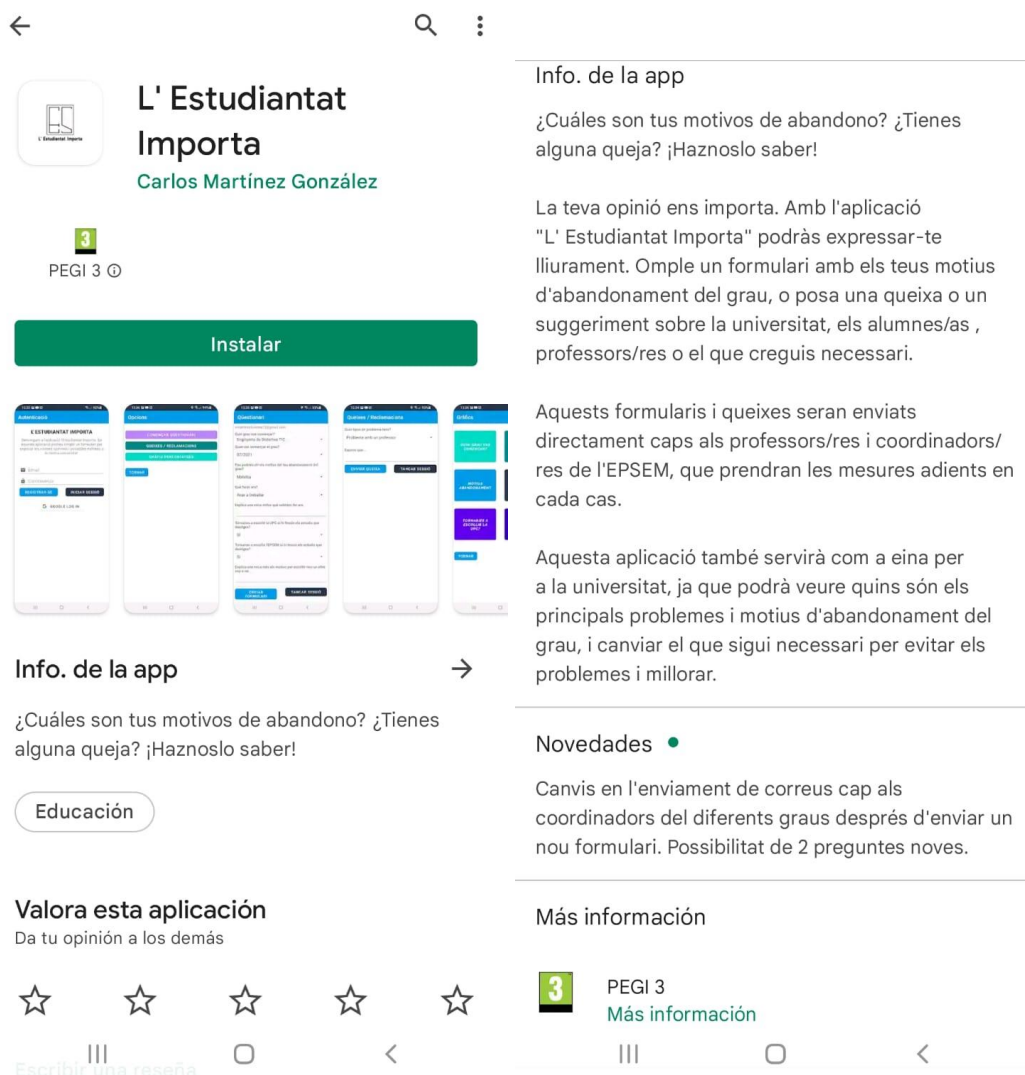
- Contingut sexual explícit
- Promoció de l'odi
- Violència
- Esdeveniments de caràcter delicat
- Assetjament
- Productos perillosos

A més, les aplicacions que permetin la generació de contingut per part de l'usuari han de tenir mecanismes senzills de denúncia i retirada.

Si l'aplicació no compleix cap dels estàndards anteriors, podem pujar-la seguint uns passos de manera molt senzilla. Necessitarem:

- El paquet APK de l'app
- Una breu descripció on expliquem de què tracta l'app (màxim de 80 caràcters)
- Una descripció més detallada (màxim de 4000 caràcters)
- Icona de l'app en alta resolució (512x512px en format PNG)
- Captures de pantalla de l'app (8 diferents en format PNG, màxim de 3500px)
- Enllaç d'un vídeo sobre el funcionament de l'app (opcional)
- Seleccionar el tipus d'app, la categoria, els detalls de contacte i una URL de política de privacitat de l'app
- Detalls sobre el preu
- Seleccionar els països on volem que estigui disponible i els llenguatges disponibles

Per últim, hem de pagar una quota única de 25\$ i esperar que Google ens verifiqui l'app.



Il·lustració 63: aplicació 'L'Estudiantat Importa' a la Play Store  
Font: pròpia

## 7.2. Arxius APK

Totes les aplicacions que es vulguin pujar a Google Play Store han d'estar en format APK d'Android. Aquests arxius són executables dissenyats per a Android. Cada sistema operatiu té els seus propis formats d'arxiu executable, com poden ser els arxius .exe de Windows o els .deb de GNU/Linux basats en Debian. En el cas d'Android, són els *Android Application Package*. Aquest format és una variant del format JAR de Java, i s'utilitza per a distribuir i instal·lar aplicacions i jocs de dispositius que utilitzin el sistema operatiu de Google, tant smartphones com tablettes, sistemes de SmartTV o qualsevol altre que utilitzi Android.

Per a executar un arxiu .APK no és necessari res, serà suficient obrir-lo perquè comenci a funcionar i no es necessitarà cap aplicació especial per a ells. Aquests arxius serveixen quasi sempre per a instal·lar aplicacions sense necessitat de passar per Google Play, i encara que la majoria de casos contenen totes les dades necessàries per fer funcionar l'app, hi ha vegades que potser faci falta descarregar alguns arxius extres.

**Si dona la casualitat que estàs instal·lant un APK que necessita baixar-te més dades però no tens connexió, existeix un altre tipus d'arxiu derivat anomenat XAPK. Aquests contenen tant el propi APK com un arxiu OBB amb les dades addicionals que necessitaràs.**

Els arxius .APK es comporten com qualsevol altre tipus d'arxiu, i es poden compartir lliurement entre els teus dispositius. Pots enviar-los per correu, missatgeria instantània o el que vulguis, i una vegada els descarreguis en el teu Android, només hauràs de prémer sobre ells per executar-los.

No obstant, els arxius .APK també poden ser la porta d'entrada per a diversos virus i diferents tipus d'amenaçes. S'ha de tenir en compte que no han passat els filtres de seguretat de la botiga d'aplicacions oficials de Google, i per això és important que ens assegurem que la seva procedència és segura en el cas que vulguem instal·lar un.

**Per tal d'evitar aquesta amenaça, la majoria de mòbils Android tenen bloquejada l'utilització d'arxius APK extern que no s'instal·lin directament des de Google Play. Si es vol instal·lar un arxiu APK fora de la botiga oficial d'apps, primer s'ha de desactivar aquesta opció.**

Aquest tipus d'arxius han estat molt útils de cara a provar l'aplicació en dispositius mòbils durant tot el temps de producció de l'app sense haver de crear diferents versions per a cada petit canvi. A més, els telèfons mòbils tenen molta més capacitat que no els AVD Manager d'Android Studio que, una vegada tenim un nombre gran d'informació, tarden molt en carregar i moltes vegades es queden penjats.

## 8. ODS

Els **Objectius de Desenvolupament Sostenible** (ODS) són els 17 punts que van adoptar els líders mundials al 2015 per tal d'eradicar la pobresa, protegir el planeta i assegurar la posteritat per a tothom com a part de la nova agenda de desenvolupament sostenible.

Cada objectiu té metes específiques que s'han d'assolir en els pròxims 15 anys. Per tal d'assolir aquestes metes, tothom ha de fer la seva part: els governs, el sector privat i la societat civil.

Dins d'aquest projecte s'ha volgut format part d'aquest objectiu, donant efusivitat en el compliment del punt número 4 (*educació de qualitat*) i el número 10 (*reducció de les desigualtats*) dels ODS.



Això es vol aconseguir amb una eina útil i totalment gratuïta per a tothom, que permet als alumnes/as sentir-se escoltats/des, com ja diu el propi nom de l'aplicació, 'L'Estudiantat Importa'.

## 9. CONCLUSIONS

L'objectiu inicial d'aquest treball de fi de grau era fer una aplicació utilitzant l'eina Android Studio i el llenguatge de programació Kotlin, que permetés a l'alumnat omplir un qüestionari on poguessin expressar les causes d'abandonament dels seus estudis.

Aquest objectiu era força ambiciós, ja que Kotlin és un llenguatge de programació pràcticament nou i, a més, es volien utilitzar les dades per fer gràfics a temps real que permetessin a la universitat veure quins eren els principals motius d'abandonament.

Una vegada finalitzat aquesta aplicació, aquests objectius no només han estat assolits, sinó que han estat superats dràsticament. S'ha aconseguit crear uns formularis editables, amb un màxim de fins a 10 preguntes, i una eina per posar queixes o suggeriments i explicar de forma detallada qualsevol problema. També s'ha aconseguit un control de verificació de cara al mal ús de l'aplicació (només es tindran en compte els qüestionaris omplerts amb comptes assignats a comptes UPC).

A més, s'ha afegit l'eina d'enviament de correus electrònics directament cap als coordinadors/es dels graus cada vegada que s'omple un nou formulari.

Per últim, s'han establert diferents rols, que donen privilegis diferents a cadascun d'ells i que poden ser modificats també per l'administrador de l'aplicació.

Ha estat un privilegi poder formar part del projecte d'innovació docent '*Mentoria, tutorització i coordinació: Les claus per a una millora del rendiment i reducció de l'abandonament a la fase inicial*', el qual crec que tindrà una gran repercussió en el futur i ajudarà molt tant a l'estudiantat com al professorat.

## 10. BIBLIOGRAFIA

- [1] Vaati, E., 2022. *Android Fundamentos: Creando un Formulario de Usuario Sencillo*. [en línea] Disponible a: <<https://code.tutsplus.com/es/tutorials/android-essentials-creating-simple-user-forms--mobile-1758>> [Últim accés: 20 octubre 2021].
- [2] APK de Android, qué son estos archivos y cómo se instalan [en línea] Disponible a: <<https://www.xataka.com/basics/apk-android-que-estos-archivos-como-se-instalan>> [Últim accés: 20 octubre 2021].
- [3] Youtube. Como crear una app Android con Kotlin. [en línea] Disponible a: <<https://www.youtube.com/watch?v=B1jGFvFzJNg>> [Últim accés: 20 octubre 2021].
- [4] Youtube. KOTLIN: Curso ANDROID desde CERO para PRINCIPIANTES. [en línea] Disponible a: <<https://www.youtube.com/watch?v=ebQphhLpJG0>> [Últim accés: 20 octubre 2021].
- [5] Youtube. El EMULADOR Android: Conócelo A FONDO. [en línea] Disponible a: <<https://www.youtube.com/watch?v=8WfQQ0BRDY8>> [Últim accés: 20 octubre 2021].
- [6] Android Developers. *Introducción a Android Studio | Desarrolladores de Android | Android Developers*. [en línea] Disponible a: <<https://developer.android.com/studio/intro?hl=es>> [Últim accés: 20 octubre 2021].
- [7] es.wikipedia.org. *Android Studio - Wikipedia, la enciclopedia libre*. [en línea] Disponible a: <[https://es.wikipedia.org/wiki/Android\\_Studio](https://es.wikipedia.org/wiki/Android_Studio)> [Últim accés: 20 octubre 2021].
- [8] Kotlin. *Kotlin Programming Language*. [en línea] Disponible a: <<https://kotlinlang.org/>> [Últim accés: 20 octubre 2021].
- [9] ¿Qué es Java y por qué lo necesito?. [en línea] Disponible a: <[https://www.java.com/es/download/help/whatis\\_java.html](https://www.java.com/es/download/help/whatis_java.html)> [Últim accés: 20 octubre 2021].
- [10] Youtube. *Introducing Firebase*. [en línea] Disponible a: <[https://www.youtube.com/watch?v=QcsAb2RR52c&list=PLI-K7zZEsYlMOf\\_07IayrTntevxtbUxDL&t=135s](https://www.youtube.com/watch?v=QcsAb2RR52c&list=PLI-K7zZEsYlMOf_07IayrTntevxtbUxDL&t=135s)> [Últim accés: 20 octubre 2021].
- [11] es.wikipedia.org. *Firebase - Wikipedia, la enciclopedia libre*. [en línea] Disponible a: <<https://es.wikipedia.org/wiki/Firebase>> [Últim accés: 27 octubre 2021].
- [12] Firebase. *Firebase Pricing*. [en línea] Disponible a: <<https://firebase.google.com/pricing?authuser=0>> [Últim accés: 2 novembre 2021].
- [13] Android Developers. *Cómo ejecutar apps en Android Emulator*. [en línea] Disponible a: <<https://developer.android.com/studio/run/emulator?hl=es-419>> [Últim accés: 2 novembre 2021].
- [14] Youtube. FIRESTORE Android. [en línea] Disponible a: <<https://www.youtube.com/watch?v=t5yyc1XfQrs>> [Últim accés: 10 novembre 2021].

[15] Firebase. *Cloud Firestore | Firebase Documentation*. [en línea] Disponible a: <<https://firebase.google.com/docs/firestore?hl=es>> [Últim accés: 10 novembre 2021].

[16] Youtube. GOOGLE PLAY STORE. Cómo PUBLICAR tu primera APP Android [en línea] Disponible a: <<https://www.youtube.com/watch?v=vBHmJHp8Ldo>> [Últim accés: 25 novembre 2021].

[17] Medium. *Kotlin Line Chart*. [en línea] Disponible a: <<https://medium.com/@yilmazvolkan/kotlinlinecharts-c2a730226ff1>> [Últim accés: 26 novembre 2021].

[18] Javapapers. *Android Chart using MPAndroidChart*. [en línea] Disponible a: <<https://javapapers.com/android/android-chart-example-app-using-mpandroidchart/>> [Últim accés: 26 novembre 2021].

[19] intensecoder.com. *PieChart tutorial using MpAndroidChart in Kotlin*. [en línea] Disponible a: <<https://intensecoder.com/piechart-tutorial-using-mpandroidchart-in-kotlin/>> [Últim accés: 27 novembre 2021].

[20] GitHub Charts. [en línea] Disponible a: <<https://github.com/PhilJay/MPAndroidChart/tree/master/MPChartExample/src/main/res/layout>> [Últim accés: 27 novembre 2021].

[21] Youtube. OBTENER DATOS DE CLOUD FIRESTORE. [en línea] Disponible a: <<https://www.youtube.com/watch?v=K5BY7suCp24>> [Últim accés: 2 desembre 2021].

[22] Nieto, J., *Requisitos app Google Play*. [en línea] Disponible a: <<https://www.xatakandroid.com/aplicaciones-android/estos-requisitos-que-app-debe-cumplir-quiере-estar-google-play>> [Últim accés: 20 desembre 2021].

[23] epsem.upc.edu. *Memòria Curs 2016-2017*. [en línea] Disponible a: <<https://www.epsem.upc.edu/ca/escola/memories/memoria-curs-2016-2017-epsem.pdf>> [Últim accés: 4 gener 2022].

[24] Firebase. *Privacidad y seguridad en Firebase*. [en línea] Disponible a: <<https://firebase.google.com/support/privacy?hl=es-419>> [Últim accés: 10 gener 2022].

[25] FIDO Alliance. *Google Security Keys Work*. [en línea] Disponible a: <<https://fidoalliance.org/case-study-series-google-security-keys-work/>> [Últim accés: 10 gener 2022].