








UNIVERSIDAD DE LAS PALMAS
DE GRAN CANARIA

Programación de Aplicaciones Móviles Nativas

Trabajo Final – DelegaApp

José Manuel Illera Rodríguez
&
Cristóbal José Jiménez Gómez
4 /12 /2022

Contenido

1.- Introducción	2
2.- Desarrollo	2
a. Fases de Desarrollo	2
Fase 1: Análisis del proyecto 	2
Fase 2: Diseño de la aplicación 	3
Fase 3: Documentación y estudio 	4
Fase 4: Desarrollo 	4
Fase 5: Pruebas y despliegue 	5
b. Temporización.....	7
c. Herramientas	9
d. Desarrollo/Ejecución	9
3.- Conclusiones	10
4.- Bibliografía	11
Algunos links de interés.....	12

1.- Introducción

Para este proyecto hemos decidido desarrollar una aplicación nativa para Android Studio[1], que solventa la necesidad de una vía de comunicación entre los estudiantes de la escuela de ingeniería informática y los eventos que la propia escuela organiza. También proporciona visibilidad a los delegados de la escuela, los cuales serán los que administren/publiquen las noticias de interés que consideren.

Habrán 3 roles de usuario:

- Invitados / No registrados
- Alumnos / Registrados
- Delegados / Administradores

Todos podrán ver las publicaciones, pero solo los usuarios registrados podrán comentar en ella y tener una lista de favoritos.

Los delegados / administradores serán los únicos que podrán:

- *Subir publicaciones* informativas, con foto y texto, o solo texto.
- *Editar y eliminar* las publicaciones
- *Eliminar cualquier comentario.*

Haciendo referencia a lo dado en la asignatura, intentaremos seguir las fases Devops, a la par que utilizaremos Metodologías de Desarrollo Ágil durante el desarrollo de la práctica.

2.- Desarrollo

a. Fases de Desarrollo

Fase I: Análisis del proyecto

Tras una investigación sobre el ámbito en el que se va a inspirar el proyecto y una previa experimentación del lenguaje que íbamos a emplear, analizamos mediante el uso de bocetos en digital las distintas funcionalidades y estética de la interfaz que podría llegar a tener la aplicación.

Para tratar de llevar una "Metodología de Desarrollo Ágil" decidimos utilizar las tecnologías "Figma"[2] en diseño y "Trello"[3] para la distribución de las historias de usuario y "Github"[4] como repositorio del proyecto.

Debido al poco tiempo para el desarrollo decidimos dividirnos en "Front-end" y "Back-end", para este utilizar Firebase[5], creando así historias de usuarios que corresponden a las funcionalidades que se planea tener en cada pantalla durante las dos primeras semanas, al mismo tiempo que se fue perfilando la idea del diseño de la aplicación.

Con las historias de usuarios hechas, nos agregamos según lo acordado y a partir de ahí realizamos reuniones semanales los jueves para ver como iba el proyecto, diciendo los

distintos problemas que hemos podido encontrar y corrigiendo los posibles errores al hacer los “merge”.

Fase 2: Diseño de la aplicación

Dado que pretendemos hacer una aplicación para la Delegación de Estudiantes de la Escuela de Ingeniería Informática de la UPGC, el diseño escogido trata de presentar coherencia y formalidad junto a la estética de la página oficial [\[6\]](#).



Figura 1: Comparativa de diseño entre la oficial y la aplicación



Figura 2: Colores basados en los de la UPGC

El diseño final es el que se presenta en nuestro repositorio de Figma [\[7\]](#). Para darle forma a los Frame en Figma se cogió un wireframe de un móvil, Samsung Galaxy Wireframe Mockup [\[8\]](#). También se añadieron los plugins Inconify [\[9\]](#), para poder añadir iconos vectoriales de forma más sencilla, Lore [\[10\]](#), para poder poner un texto de relleno. Wireframe [\[11\]](#), para la idea de algunos componentes de la aplicación, como puede ser la foto por defecto.

Fase 3: Documentación y estudio

Hemos recurrido a los Codelabs [\[12\]](#) oficiales de Android Studio y a la documentación oficial de Firestore-Android Studio [\[13\]](#)

Se irán colocando los Codelabs más interesantes para el desarrollo del trabajo, así como vídeos o páginas que nos sirvan de ayuda.

- www.codigofacilito.com [8] -> Puede ayudar a la hora de hacer el login.
- Creación de layouts [9] -> Puede ayudar a la creación de las distintas publicaciones, comentarios y zonas donde tengamos que mostrar información.
- Creación de diálogos [10].

Fase 4: Desarrollo

Para el desarrollo de la aplicación se ha tratado de aplicar la “Guía de Arquitectura” de Android Studio [\[14\]](#). Con respecto a la arquitectura tratamos de hacer lo más parecido a MVC (Model View Controller), teniendo modelos como pueden ser los usuarios o las publicaciones, las vistas de la propia aplicación e interfaces como una especie de control.

Planteamiento de la Base de Datos

Como se ha planteado, se utilizará Firebase, en concreto el recurso Cloud Firestore para almacenar y sincronizar los datos de la app. El planteamiento para la primera versión sería tener dos colecciones -> "users" y "news".

La colección "users" almacenará todos los usuarios. Estos tendrán propiedades para poder identificarlos.

Propiedades de los "users" (El nombre de los documentos dentro de esta colección será el correo de los propios usuarios):

- **rol: Number** - Habrá 3 tipos de rol:
 - "0" -> Invitado
 - "1" -> Delegado
 - "2" -> Usuario General
- **description: String** - Descripción simple de quien es el usuario.
- **movil: String** - Número de móvil del usuario, se usará como vía de comunicación.
- **email: String** - Correo institucional del usuario, se usará para el login y como vía de comunicación.
- **discord: String** - Nombre de usuario en la plataforma Discord.
- **telegram: String** - Nombre de usuario en la plataforma Telegram.
- **instagram: String** - Nombre de usuario en la plataforma Instagram.
- **grade: String** - Grado al que perteneces.

La colección "news" almacenará todas las noticias y estas tendrán propiedades para poder identificarlas. Propiedades de las "news" (El nombre de los documentos dentro de esta colección será la dada por el propio Firestore):

- **titulo: String** - Nombre descriptivo de la noticia.
- **img: String** - Será una imagen guardada en CloudFirestore.
- **description: String** - Descripción de la publicación.
- **who: String** - Usuario Delegado que subió la publicación.
- **users: List of users** - Lista de Usuarios que habrán agregado a favoritos la propia publicación.
- **fecha: Timestamp** - Fecha con la que poder ordenar las publicaciones.

Fase 5: Pruebas y despliegue

Android Studio está compuesto por Android Emulator, que nos permite probar las APP en muchos dispositivos diferentes, en nuestro caso hemos decidido escoger el Pixel 6 Pro API 28 con Android 9.0.

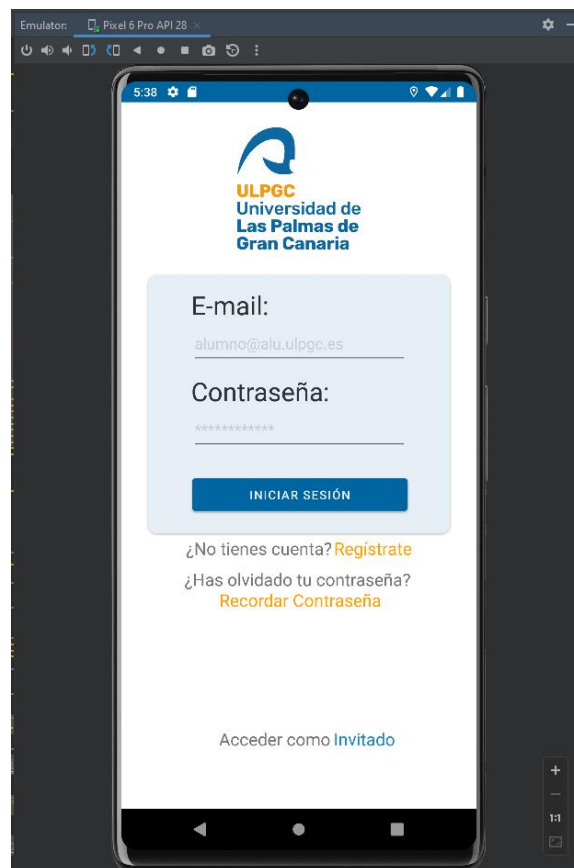


Figura 3: Android Emulator

Pruebas con Firebase-Firestore

A la hora de registrarse en la aplicación empleamos “createUserWithEmailAndPassword” el cual viene explicado en la documentación oficial de Firebase.



Identificador	Proveedores	Fecha de creación	↓	Fecha de acceso	UID de usuario
jose.illera101@alu.ulpgc.es		1 dic 2022		1 dic 2022	YWCLoNysfnWq4Sbjgx1Lo0xFidw1
invitado@invitado.invitado		1 dic 2022		1 dic 2022	FQ1rJur9ySSGFKNYSuBzl4Ztc0g2
cristobal.jimenez101@alu....		14 nov 2022		1 dic 2022	p5ldRKY1Ccba365dXrg9UDSeypY2

Figura 4: Prueba de que pueden los usuarios registrarse

Una vez registrado, se tendrá que iniciar sesión y añadir unos datos más de información del usuario que se inicia por primera vez (nombre, descripción, . . .), en este caso empleamos el método “set” y esta operación se realiza sobre Cloud Firestore.

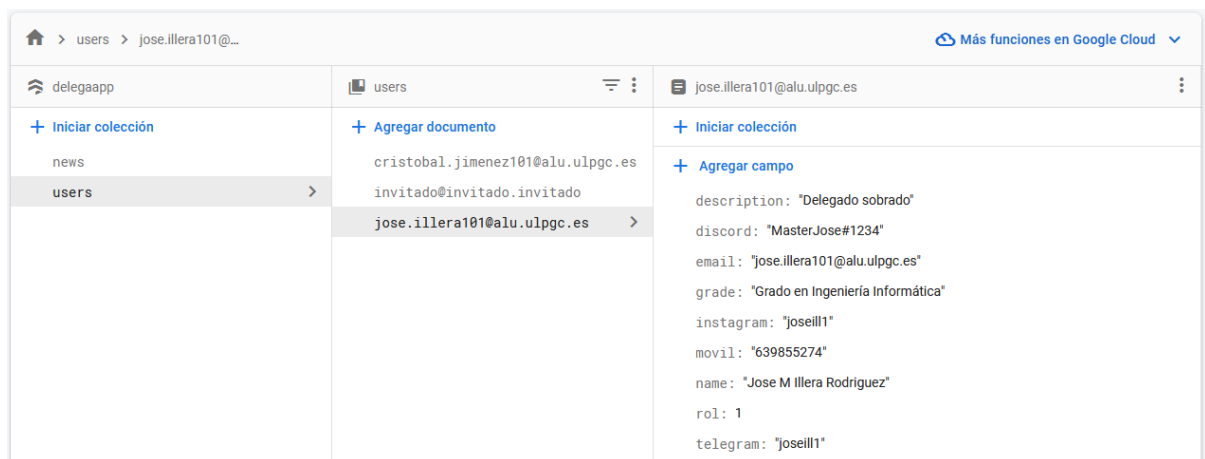


Figura 5: Se añade información a Firestore

Todas las llamadas a Firestore se hacen de forma asíncrona, lo que nos dio muchos problemas a la hora de leer los datos por estar desarrollando una arquitectura MVC, tuvimos que implementar interfaces como control para que estos pudiesen devolver los datos una vez finalizada la llamada.

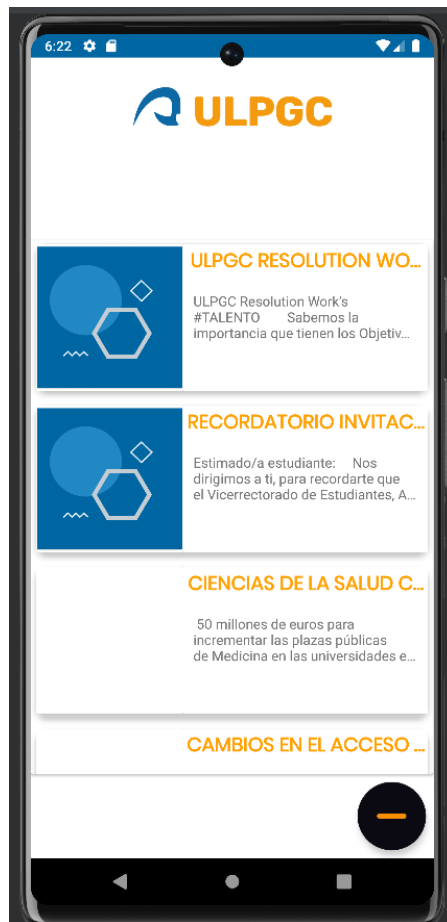


Figura 6: Lectura de datos de Cloud Firestore

b. Temporización

Durante las 2 primeras semanas fuimos alternando entre proponer ideas para el Trello e ir diseñando la aplicación:

- Dentro de Trello hemos decidido tener 10 calles:
 - **Historias épicas:** Historias demasiado grandes y costosas, pero que si a futuro seguimos con la aplicación podrían llegar a ser viables.
 - **Figma:** Historias de usuario de creación de mockups en Figma.
 - **Historia de usuario(product backlog):** Historias de usuarios reales y relacionadas con la programación de la aplicación.
 - **En proceso:** Historias que se están desarrollando en el momento actual.
 - **Estancado:** Historias que por diversos motivos no pueden continuar o no se pueden dar por terminadas.
 - **Pendiente de revisión:** Historias ya acabadas, pendientes de que la otra parte del grupo o algún usuario externo compruebe su correcto funcionamiento (en caso de cambio se pondría otra vez en *Historia de usuario* o en *En proceso*).
 - **Actividades terminadas:** Actividades/Vistas de la aplicación terminadas 100%.

- **Mockups terminados:** Mockups diseñados en Figma que se han aprobado a nivel estético y únicamente como referencia.
- **Funciones terminadas:** Funcionalidades acabadas de forma que son seguras para proseguir con la siguiente actividad.

Decidimos no agregar horas a las historias de usuarios debido a que estamos empleando un lenguaje nuevo y no sabemos con aproximación el tiempo que nos llevaría. Lo que hicimos fue agregar etiquetas de prioridad a aquellas historias que eran totalmente indispensables para la primera versión de la aplicación.

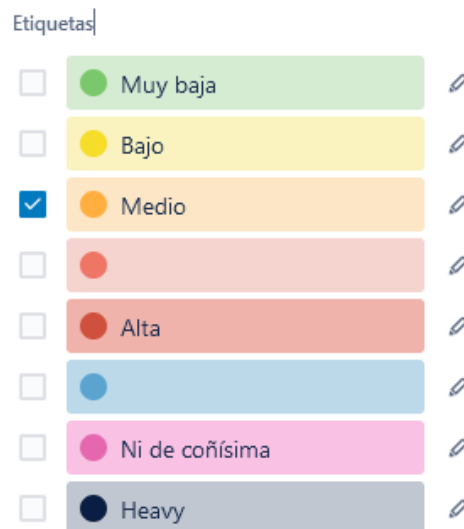


Figura 7: Etiquetas de prioridad

Una vez pasaron esas 2 semanas de organización, empezamos a usar Android Studio para desarrollar las primeras actividades y layouts (Iniciar sesión y registro) y para conectar con la Cloud Firestore, de forma correcta y sin errores.

Cada jueves, en la clase de práctica, comentamos los problemas que fuimos encontrando durante la semana de forma que ese mismo día pudieran estar resueltos para poder proseguir con la aplicación.

Además, diariamente se hablaba un poco de lo que habíamos hecho y los avances que íbamos teniendo.

A 5 semanas de la entrega del trabajo nos encontramos con que no éramos capaces de mostrar los datos recogidos de Firestore, como se explicó antes, las llamadas son asíncronas y el hilo principal se ejecutaba antes de recibir los datos, lo que nos llevó a los dos integrantes tratar de resolver este problema que nos llevó su tiempo.

Una vez esto fue solventado, solo quedaba crear un par de vistas y recoger los 2 tipos de datos para poder usarlos en toda la aplicación.

c. Herramientas

GitKraken [\[15\]](#) -> Para el control de versiones.

Trello [\[16\]](#) -> Monitorizar las tareas y el trabajo que se va realizando.

Figma [\[7\]](#) -> Bocetos y diseño de la aplicación.

Android Studio [\[1\]](#) -> Para el desarrollo de la Aplicación.

Firebase [\[5\]](#) -> Para cosas de Login y BBDD.

d. Desarrollo/Ejecución

Hemos desarrollado 6 actividades con sus correspondientes layouts en Android Studio:

- Login/Register:
 - Donde encontraremos las vistas para iniciar sesión y registrarse en la aplicación
- Publications:
 - Donde encontraremos la vista general de publicaciones, esta contiene un menú, desde el que se podrá ir moviendo por la aplicación al Inicio, al perfil, a la lista de delegados, o se podrá también cerrar sesión.
 - Así mismo, también podremos encontrar la pantalla de Añadir publicación, a la que solo se tendrá acceso siendo Delegado (Rol=2).
- Profile:
 - Donde encontraremos la vista donde se muestra la información particular del usuario registrado y la vista donde podremos cambiar los datos de nuestro usuario.

3.- Conclusiones

En resumen, el proyecto se podría decir que aspira a ser muy grande y complejo, para el tiempo que teníamos. Quizás nos deberíamos haber centrado más en hacer un estudio previo de la arquitectura a emplear, pero se han solventado los errores que nos han surgido.

Consideramos que nos ha venido bastante bien este proyecto para aprender a programar en kotlin y su implementación en aplicaciones para el móvil.

4.- Bibliografía

[1] Página oficial de Android Studio

<https://developer.android.com/studio>

[2] Página oficial de Figma.

<https://www.figma.com/>

[3] Página oficial de Trello.

<https://trello.com/>

[4] Página oficial de Github.

<https://github.com/>

[5] Página oficial de Firebase

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

[6] Página oficial de ULPGC

<https://www.ulpgc.es/>

[7] Planificación del diseño en Figma.

<https://www.figma.com/file/G9zPaIo3azyYjJuFsx6RSS/COSA-PAMN?node-id=0%3A1>

[8] Wireframe de un móvil android en Figma. (Visto a día 31 de Octubre)

[https://www.figma.com/file/YxhF8ALi0VuQqARKTiDSUe/samsung-galaxy-wireframe-mockup-\(Community\)?node-id=0%3A1](https://www.figma.com/file/YxhF8ALi0VuQqARKTiDSUe/samsung-galaxy-wireframe-mockup-(Community)?node-id=0%3A1)

[9] Iconify. Iconify and Vjacheslav Trushkin

<https://www.figma.com/community/plugin/735098390272716381>

[10] Lore (for Lorem Ipsum). Robert Coyle

<https://www.figma.com/community/plugin/984557085378252054>

[11] Wireframe. Hexorial Studio and others

<https://www.figma.com/community/plugin/742764242781786818>

[12] Codelabs de Conceptos básicos para desarrolladores de Android

<https://developer.android.com/courses/fundamentals-training/toc-v2>

[13] Documentación de Firebase

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

[14] Guía de arquitectura de apps

https://developer.android.com/jetpack/guide?gclid=Cj0KCQiA4aacBhCUARIsAI55maFqNCZvOpj9QH6NYHotGqUF1vP0ilaJWg84oqFgITzmXC7xLC-oQNkaAvN2EALw_wcB&gclidsrc=aw.ds

[15] Página oficial de GitKraken

<https://www.gitkraken.com/>

[16] Historias de usuario Trello

<https://trello.com/invite/b/5WkKy91/ATTIdccddb7496ef1092f2d1b9f742d40e2635290E99/pamndelega>

Algunos links de interés

[-] Gabinete de comunicación. Universidad de Las Palmas de Gran Canaria. INFORME Logosímbolo de la Escuela de Ingeniería Informática.

https://www.ulpgc.es/sites/default/files/ArchivosULPGC/identidad-corporativa/logosimbolos%20unidades/Centros%20docentes/informe_logosimbolo_esc_ingenieria_informatica.pdf

[-] Youtube. BeeDevelopers;

<https://www.youtube.com/watch?v=5jXy2S-qdVQ&t=2729s>

[-] Android Developers Documentación. Cuadros de diálogo.

<https://developer.android.com/guide/topics/ui/dialogs?hl=es-419>