

**I.E.S LAS SALINAS**



**PROYECTO FINAL FIN DE GRADO**

**GAME CENTER**

**CICLO FORMATIVO GRADO SUPERIOR  
DESARROLLO DE APLICACIONES MULTIPLATAFORMA**

**Autor: Alejandro Méndez Sainz**

## **Resumen**

Hace 3 años el mundo del coleccionismo iba a cambiar debido a que algunos integrantes de ese mundo decidieron saltar a las distintas plataformas de stream, haciendo videos dedicados sobre algunos artículos exclusivos de distintos temas.

Con estos videos o stream de streamer o youtuber como fueron byVirusZ, Rubius Z, UniverseCard como otros, la gente empezó a interesarse en este mundillo debido a que sus youtuber favoritos lo estaban haciendo

Algunas de estos sectores en los que la gente esta mas interesada seria en los sobres de Pokémon llegando a costar un sobre, una caja o una carta miles de euros.

Actualmente este sector está en auge y esto ha llevado a crear más aplicaciones para crear, gestionar de colecciones y a su vez compras y subastas de artículos

El objetivo de este proyecto es que esas personas, que quieren saber sobre las colecciones tengan una App en la que puedan ver los tipos de artículos que tiene cada usuario.

## **Abstract**

Three 3 years ago the world of collecting was going to change because some members of that world decided to jump to the different streaming platforms, making dedicated videos about some exclusive items on different topics.

With these streamers or youtuber videos or streams such as byVirusZ, Rubius Z, UniverseCard and others, people became interested in this world because their favorite youtubers were doing it.

Some of these sectors in which people are most interested would be Pokémon envelopes, costing an envelope, a box or a letter thousands of euros.

Currently this sector is booming and this has led to the creation of more applications to create, manage collections and, in turn, purchase and auction items.

The objective of this project is that those people, who want to know about the collections, have an App in which they can see the types of articles that each user has.

## Índice

Resumen .....	2
Abstract.....	2
Índice.....	2
1. JUSTIFICACIÓN DEL PROYECTO .....	3
2. INTRODUCCIÓN.....	4
3. OBJETIVOS.....	5
- Objetivos generales:.....	5
- Objetivos específicos:.....	6
4. RECURSOS .....	6
5. DESARROLLO .....	7
5.1 FIREBASE .....	11
5.2 DIAGRAMAS .....	11
5.2.2. Diagrama de clases (UML) .....	13
5.2. INTERFACES GRAFICAS .....	14
5.3. Implementación .....	19
5.3.1. Código.....	19
6. CONCLUSIONES .....	20
7. BIBLIOGRAFIA.....	21
8. ANEXOS .....	21
Usuarios. ....	21

## 1. JUSTIFICACIÓN DEL PROYECTO

Esta idea de proyecto surge después de comprobar que hay pocas aplicaciones que sirvan para los coleccionistas de objetos, en específico los Pokémon (en general, cartas, muñecos, etc.), como a su vez a Funkos.

A este proyecto le podríamos dar varios puntos clave, entre los que destacan:

- Actividad Principal: Crear un registro de artículos, en los que se va a hacer un sistema donde es creado por diversos usuarios.
- A quien va dirigido: Este proyecto va dirigido a todas las personas que les gusta los colecciones.
- Nacimiento de la idea: Esta idea surge después de ver que no hay prácticamente aplicaciones parecidas en este sector de los coleccionistas. Se intenta realizar un diseño sencillo e intuitivo en el cual pretendemos que tenga una buena acogida entre la gente.
- Futuro: Esta aplicación no está terminada al 100% y puede ser bastante más grande añadiendo más opciones como la gestión de una tienda mediante un nuevo enfoque a la hora de poder desarrollar esta dicha tienda con nuevos apartados.

## **2. INTRODUCCIÓN**

La comunidad de los coleccionistas ha dado un gran salto en los últimos años sobre todo a raíz de diferentes entornos y personas que están dando a conocer al mundo.

Aunque hay gente que gracias a subir videos a YouTube o hacer stream a través Twitch este mundillo que poca gente conocía se esta dando a conocer más y mas debido a su interés sobre este tema.

¿Qué me motivo a realizarlo? La respuesta podría ser la oportunidad que vi en el sector, un sector en el que prácticamente no hay aplicaciones por el estilo. Mi objetivo con este proyecto es mejorar y adquirir nuevos conocimientos de programación, en este caso con Android Studio y el lenguaje Java, así como la utilización de Firebase como base de datos. Desde hace varios años he ido viendo como evolucionaba el mundo, cada vez más tecnológico y esta era la oportunidad para adentrarme en este mundo de la programación y el desarrollo tecnológico.

Decidí realizar este proyecto como dije anteriormente con Android Studio y Java después de su uso diario en las clases de Programación Multimedia y dispositivos móviles. Este software siempre me había llamado la atención, ver las distintas aplicaciones que existen para prácticamente cualquier cosa que necesites, y todo lo que se podía hacer me llevo a querer profundizar más en él.

Quería aprender a usarlo mejor, desarrollar nuevas interfaces, utilizar y crear listas, usar bases de datos no relacionales como es el caso de Firebase muy usado en el desarrollo Android... En definitiva, crecer como desarrollador Android.

Este proyecto se ha llevado a cabo de una manera "local" en la cual podremos realizar todas o la mayoría de acciones que se podrían realizar en una App que ya estuviese producida y lanzada en el mundo real.

Aunque no lo parezca estos tipos de torneos suelen generar cansancios mentalmente según el tipo de torneo que se realice y físicos dependiendo de la durabilidad de dicho torneo.

### **3.OBJETIVOS**

En este siguiente punto vamos a ver qué objetivos generales y específicos son a los que se quiere llegar, por lo tanto, podemos decir ¿Qué espero conseguir con esta elección?

#### **- Objetivos generales:**

- **Desarrollar una aplicación de gestión de fácil manejo y a medida para este sistema operativo en concreto.**

Android supone el 84% de la cuota de mercado de los sistemas operativos en smartphones, frente al 16% de iOS de Apple.

- **Dar un servicio a todos aquellos aficionados al mundo de los coleccionistas que existen alrededor de todo el mundo.**

Son cada vez más los aficionados que quieren probar este tipo de mundillo para entrar en este sector de los coleccionistas.

#### - **Objetivos específicos:**

- **Gestionar varios productos con sus especificaciones.**

Con este objetivo quiero centrarme en la gestión de los usuarios a entrar en nuestra aplicación para poder registrar sus objetos de colección.

- **Dar más visibilidad a los objetos únicos y raros**

Incluyendo los enlaces a su web, como a el entorno que hay específico de estos coleccionables.

- **Intentar publicar una app en las tiendas de apps**

Por último, se pretende en un futuro subir esta app a las distintas tiendas de apps como Play Store.

#### **4. RECURSOS**

A continuación, vamos a exponer los distintos recursos materiales y personales que tenemos disponibles para la realización de este proyecto.

Entre los recursos materiales podemos encontrar la documentación facilitada por el centro o la documentación oficial de Android y repositorios públicos como GitHub para consultar distintas fuentes.

En el plano personal he dispuesto de un ordenador portátil, un smartphone para realizar pruebas y corregir errores, cables USB para conectar ambos dispositivos, el Wifi para conectarme a internet.

Esto tiene a su vez un presupuesto económico, que desglosaremos de la siguiente manera:

- Ordenador de mesa → 1400€
- One Plus Nord2 → 400€
- Tarifa Internet → 50€
- Cable USB → 2€
- Total → 1850€

En lo que se refiere al software:

- Android Studio y Firebase Database → 0€
- Repositorio de GitHub → 0€

## **5. DESARROLLO**

En este punto vamos a exponer como hemos realizado nuestra aplicación, que realiza cada acción, fases, diagramas, casos de uso, clases UML, interfaces tanto del menú, Pokémon y Funkos. También veremos cómo hemos implementado el código y las bases de datos.

La primera imagen que se muestra debajo es un poco del código de inicio de sesión que es fundamental para nuestra aplicación, con esto nos ayudara a poder entrar en nuestra aplicación a través de un correo y una contraseña que nosotros pongamos.



También se puede ver la conexión con Firebase y las validaciones que tenemos puestas para cada campo utilizado.

```
public void addEntrenador(View view) {
    String email = String.valueOf(edt_correoRegistro.getText());
    String password = String.valueOf(edt_claveRegistro.getText());
    //----- Validacion -----
    if(email.isEmpty() || password.isEmpty()){
        Toast.makeText(context addEntrenador.this, text: "Rellene todos los campos", Toast.LENGTH_SHORT).show(); //mensaje que se muestra al usuario
        return;
    }
    if(email.isEmpty()){
        Toast.makeText(context addEntrenador.this, text: "Escriba un correo", Toast.LENGTH_SHORT).show(); //mensaje que se muestra al usuario
        return;
    }
    else if(password.isEmpty()){
        Toast.makeText(context addEntrenador.this, text: "Escriba una contraseña", Toast.LENGTH_SHORT).show(); //mensaje que se muestra al usuario
        return;
    }
    else if(password.length() < 6){
        Toast.makeText(context addEntrenador.this, text: "Escriba una contraseña de mas de 6 caracteres", Toast.LENGTH_SHORT).show(); //mensaje que se muestra al usuario
        return;
    }
}

mAuth.createUserWithEmailAndPassword(email, password)
    .addOnCompleteListener(activity: this, new OnCompleteListener<AuthResult>() {
        @Override
        public void onComplete(@NonNull Task<AuthResult> task) {
            if (task.isSuccessful()) {
                // Sign in success, update UI with the signed-in user's information
                Log.i(tag: "firebaseDb", msg: "createUserWithEmail:success");
                Toast.makeText(context addEntrenador.this, text: "¡Enorabuena! Ya te has registrado", Toast.LENGTH_SHORT).show();
                FirebaseUser user = mAuth.getCurrentUser();
                // updateUI(user);
                Intent intent = new Intent(context addEntrenador.this, Registro.class);
                startActivity(intent);
            } else {
                // If sign in fails, display a message to the user.
                Log.i(tag: "firebaseDb", msg: "createUserWithEmail:failure", task.getException());
                AlertDialog.Builder alerta = new AlertDialog.Builder(context addEntrenador.this);
                alerta.setMessage("Debe asegurarse de que la contraseña contenga mas de 6 caracteres ." +
                    " Si aun así todavía persiste el error, entonces compruebe si existe el correo.")
                    .setCancelable(true); //quita el mensaje si pulsas fuera del cuadro
            }
        }
    });
```

La otra parte del Login es el registro de nuestra aplicación, en la que pediremos primero un correo y una contraseña y después pediremos datos personales del usuario que se esta registrando como el DNI, nombre, apellidos, localidad, dirección, teléfono, con su respectiva validación para cada campo en específico.

```
public void record(View view) {
    boolean errores = false;
    FirebaseAuth currentUser = mAuth.getCurrentUser();
    String correo = String.valueOf(currentUser.getEmail());
    dni = String.valueOf(edt_dni.getText());
    nombre = String.valueOf(edt_nombre.getText());
    apellidos = String.valueOf(edt_apellidos.getText());
    localidad = String.valueOf(edt_localidad.getText());
    direccion = String.valueOf(edt_direccion.getText());
    try {
        telefono = Integer.valueOf(String.valueOf(edt_telefono.getText()));
    } catch (Exception e) {
        telefono = 0;
    }

    Usuario usu = new Usuario(dni,nombre,apellidos,localidad,direccion,correo,telefono);

    if (dni.isEmpty()) {
        edt_dni.setError("Debes de escribir un dni");
        errores = true;
    } else if(nombre.isEmpty()){
        edt_nombre.setError("Debes de escribir un nombre");
        errores = true;
    } else if (apellidos.isEmpty()){
        edt_apellidos.setError("Debes de escribir los apellidos");
        errores = true;
    } else if(localidad.isEmpty()){
        edt_localidad.setError("Debes de escribir una localidad");
        errores = true;
    } else if(direccion.isEmpty()){
        edt_direccion.setError("Debes de escribir una direccion");
        errores = true;
    } else if(telefono == 0){
        edt_telefono.setError("Debes de escribir un numero de telefono");
        errores = true;
    }

    double num_tel = Math.floor(Math.log10(Math.abs(telefono)) + 1);
    if(dni.length() != 9){
        edt_dni.setError("Debes de escribir el dni correctamente");
        errores = true;
    } else if(num_tel != 9){
        edt_telefono.setError("Debes de escribir el numero de telefono correctamente");
    }
}
```

Otra cosa importante que es uno de los pilares de la aplicación es la creación de objetos, como se ve en la imagen se tiene que insertar un id, un nombre, una categoría y opcional una imagen con su validación correspondiente utilizada para cada campo.

```
public void crearFunko(View view) {
    String idFunko = String.valueOf(idFunkoRegistro.getText());
    String nombreFunko = String.valueOf(nombreFunkoRegistro.getText());
    String categoriaFunko = String.valueOf(categoriaFunkoRegistro.getText());
    //----- Validacion-----
    if(idFunko.isEmpty()){
        Toast.makeText( context: addMFunkosActivitty.this, text: "Pon un id al Funko", Toast.LENGTH_SHORT).show();
        return;
    }
    else if(nombreFunko.isEmpty()){
        Toast.makeText( context: addMFunkosActivitty.this, text: "Pon un nombre al Funko", Toast.LENGTH_SHORT).show();
        return;
    }
    else if(categoriaFunko.isEmpty()){
        Toast.makeText( context: addMFunkosActivitty.this, text: "Pon una categoria al Funko", Toast.LENGTH_SHORT).show();
        return;
    }
}

if(img_insertar_foto != null)
{
    String email = mAuth.getCurrentUser().getEmail();
    ▲ Alejandro
    new ImageFirebase().subirFoto(new ImageFirebase.FotoStatus() {
        2 usages ▲ Alejandro
        @Override
        public void FotoIsDownload(byte[] bytes) {
        }
        ▲ Alejandro
        @Override
        public void FotoIsDelete() {
        }
        1 usage ▲ Alejandro
        @Override
        public void FotoIsUpload() {
            Toast.makeText( context: addMFunkosActivitty.this, text: "foto subida correcta", Toast.LENGTH_LONG).show();
        }
    },email, idFunko, img_insertar_foto);

    f = new Funko(idFunko,nombreFunko,categoriaFunko, foto: email+"/"+String.valueOf(idFunko)+".png");
    Intent intent = new Intent( packageContext: this, MenuFunkos.class);
    startActivity(intent);
}
```

## 5.1 FIREBASE

### ¿Qué es Firebase?

Firebase es una plataforma para el desarrollo de aplicaciones ubicada en la nube e integrada con Google Cloud Platform, fue lanzada en 2011 y adquirida por Google en 2014.

Para este proyecto he usado Firebase como backend con su base de datos **RealTime Database**, esta es una base de datos NoSQL, flexible y escalable, esta se encuentra en la nube y permite crear aplicaciones de calidad de una manera bastante rápida, está disponible para Android, iOS y Web.

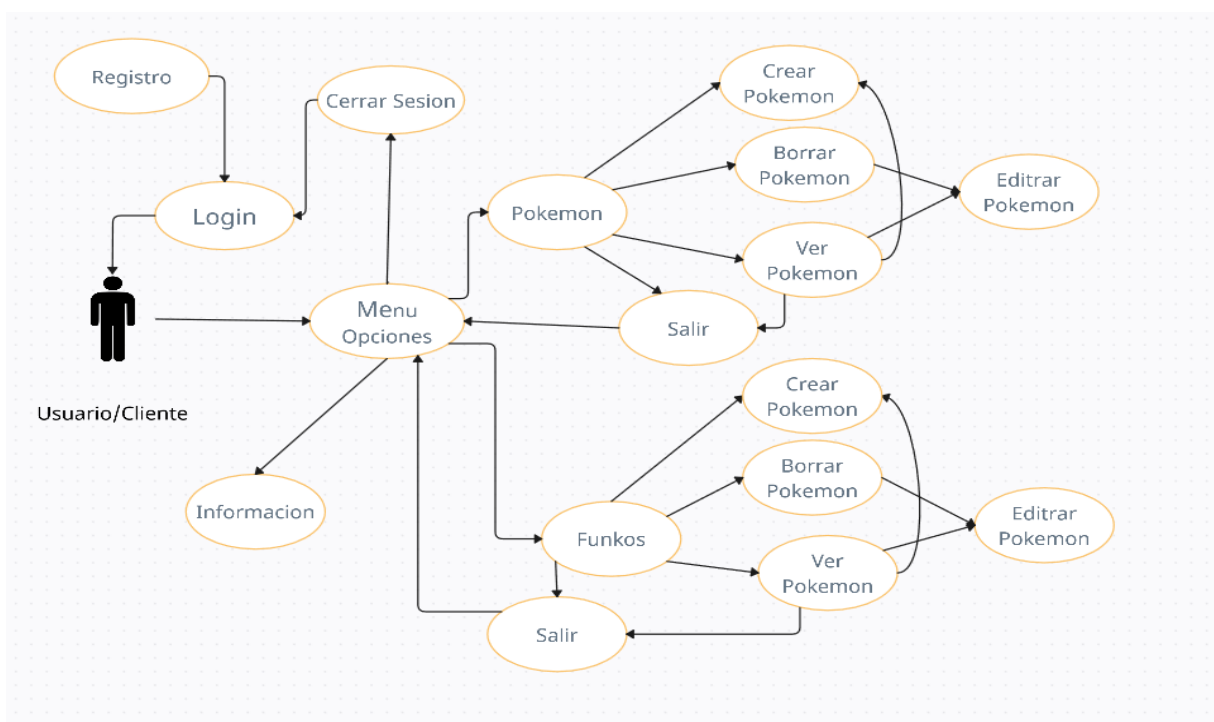
Además, usaré **Autenticación** para la gestión y registro de usuarios, guardar sus datos, como el teléfono, nombre, correo o inicio de sesión con Google o Facebook.

Por último, también usaré **Storage** que es un sistema de almacenamiento de archivos. Estos pueden cargarse asíncronamente, es decir, que si se interrumpe la conexión se subirá una vez volvamos a tener conexión. Utilizaremos esto para subir y gestionar nuestras fotos.

## 5.2 DIAGRAMAS

### 5.2.1 Casos de uso

Explicación del diagrama usuario:



## Usuario:

**Inicio de Sesión** → Si el usuario está registrado, es decir, existe en el sistema, rellena los campos con sus credenciales y pulsa acceder para acceder a la página principal.

**Crear Usuario** → Si el usuario no está registrado, rellena los campos con sus credenciales, y pulsa registrarse.

**Registro** → Rellena los datos del formulario sobre el usuario.

**Cerrar Sesión** → El usuario pulsa el botón de cerrar sesión y le manda a la página de login para volver a iniciar sesión.

**Botón Crear Pokémon** → En la parte superior de la pantalla pulsamos el botón morado, envía al usuario a la pantalla de crear nuevo Pokémon.

**Nuevo Pokémon** → Una vez en la pantalla nuevo campeonato, rellenamos todos los datos para crear un Pokémon, guardarlo y mostrarlo en un RecyclerView en la pantalla usuario.

**Imagen** → Al pulsar en el botón podremos cambiar la imagen.

**Eliminar Pokémon** → Aquí vamos a poder eliminar el Pokémon creado.

**Lista Pokémon** → Pulsamos el botón y podremos ver todos los Pokémon que tenemos.

**Editar Pokémon** → Al pulsar en uno de nuestros Pokémon podremos editarlo.

**Detalles Pokémon** → Podremos ver todos los atributos de nuestro Pokémon.

**Botón Crear Funko** → En la parte superior de la pantalla pulsamos el botón morado, envía al usuario a la pantalla de crear nuevo Funko.

**Nuevo Funko** → Una vez en la pantalla nuevo Funko, rellenamos todos los datos para crear un Funko nuevo, guardarlo y mostrarlo en un RecyclerView en la pantalla usuario.

**Imagen** → Al pulsar en el botón podremos cambiar la imagen.

**Lista Funko** → Pulsamos el botón y podremos ver todos los Funkos que tenemos.

**Editar Funko** → Al pulsar en uno de nuestros Funkos podremos editarlo.

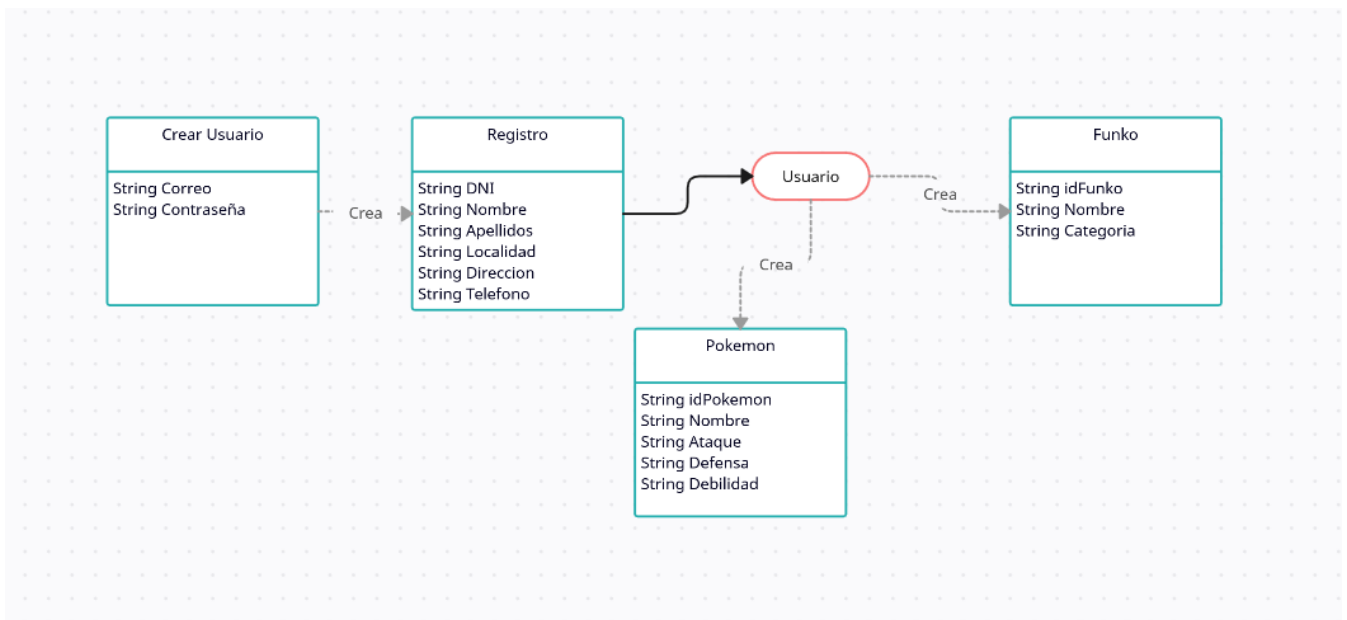
**Detalles Funko** → Podremos ver todos los atributos de nuestro Funkos.

**Eliminar Funko** → Aquí vamos a poder eliminar el Funko creado.

**Información** → Veremos toda la información de nuestra aplicación con un mapa de nuestra localización.

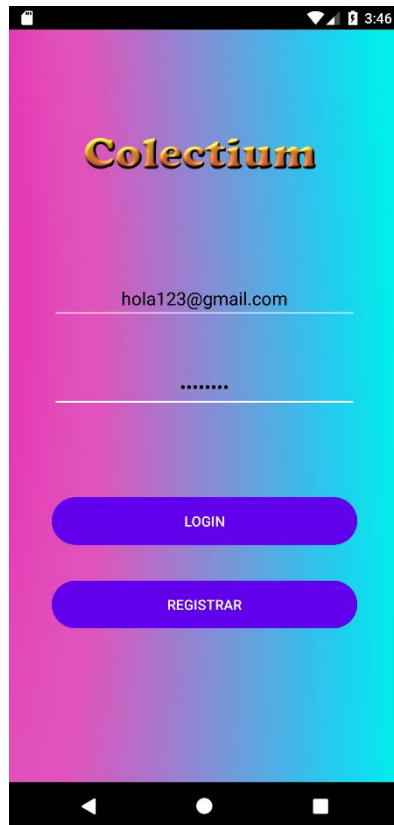
### 5.2.2. Diagrama de clases (UML)

Se trata del lenguaje de diseño y modelado de sistemas más usado y conocido en la actualidad. A continuación, se presenta el que se ha usado para este proyecto:

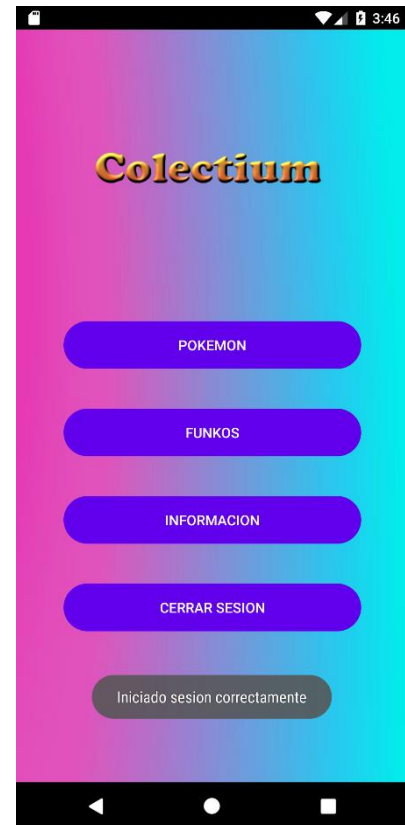


## 5.2. INTERFACES GRAFICAS

Aquí vamos a mostrar algunas capturas de pantalla de los distintos actividades que nos encontramos en la aplicación.



Login



Menú Principal

Colectium

hola12345@gmail.com

\*\*\*\*\*

SIGUIENTE

Creación Usuario

Colectium

DNI Nombre

Apellidos Localidad

Direccion Telefono

REGISTRAR

Registro de Datos

Pokémon

CREAR POKEMON

BORRAR POKEMON

MOSTRAR POKEMON

SALIR

Menú Pokémon

Pokémon GO

Id Pokemon

Nombre Pokemon

Ataque Pokemon

Defensa Pokemon

Debilidad Pokemon

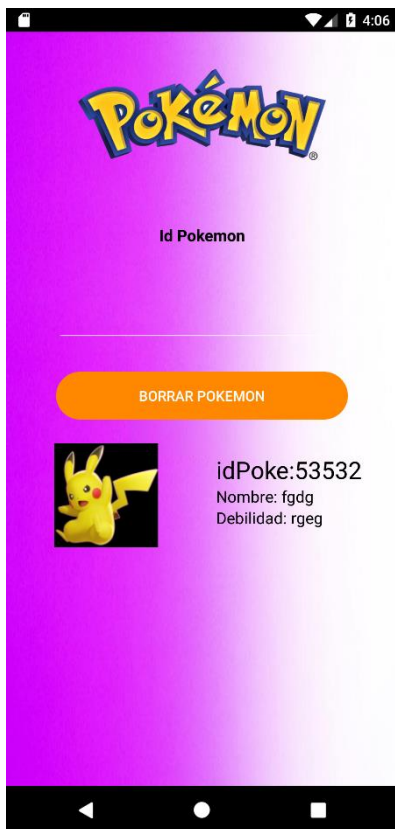
Pokémon MASTERS EX

IMAGEN

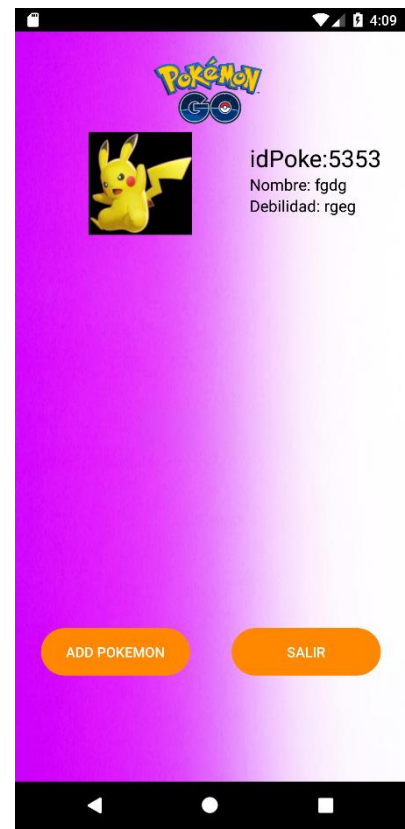
AÑADIR POKEMON

Crear Pokémon

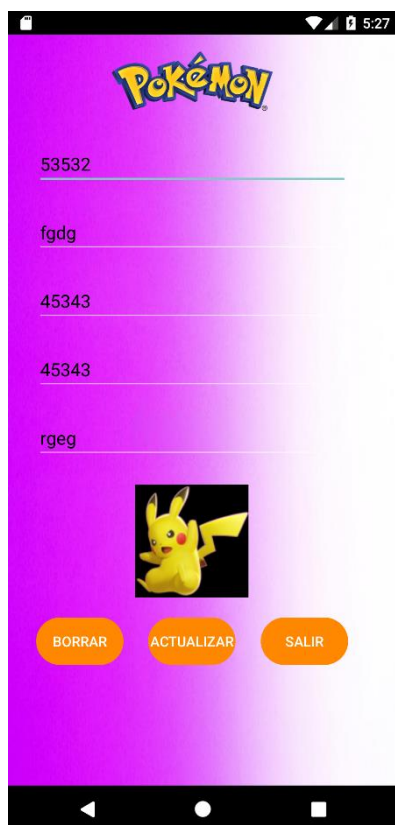




Borrar Pokémon



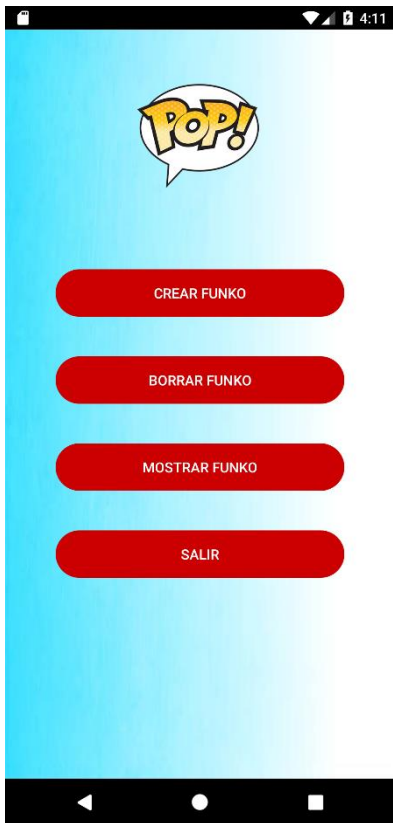
Mostrar Pokémon



Editar Pokémon



Detalle Pokémon



Menú Funko



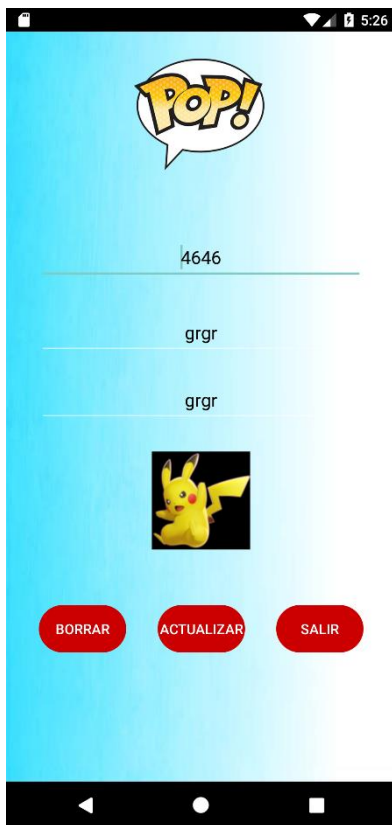
Crear Funko



Borrar Funko



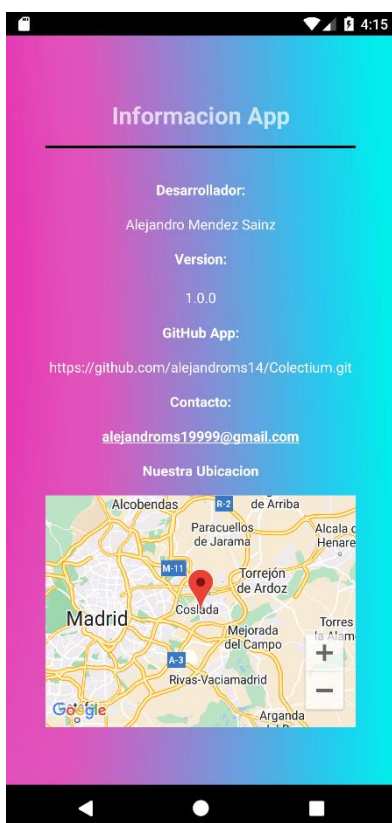
Mostrar Funko



Editar Funko



Detalle Funko

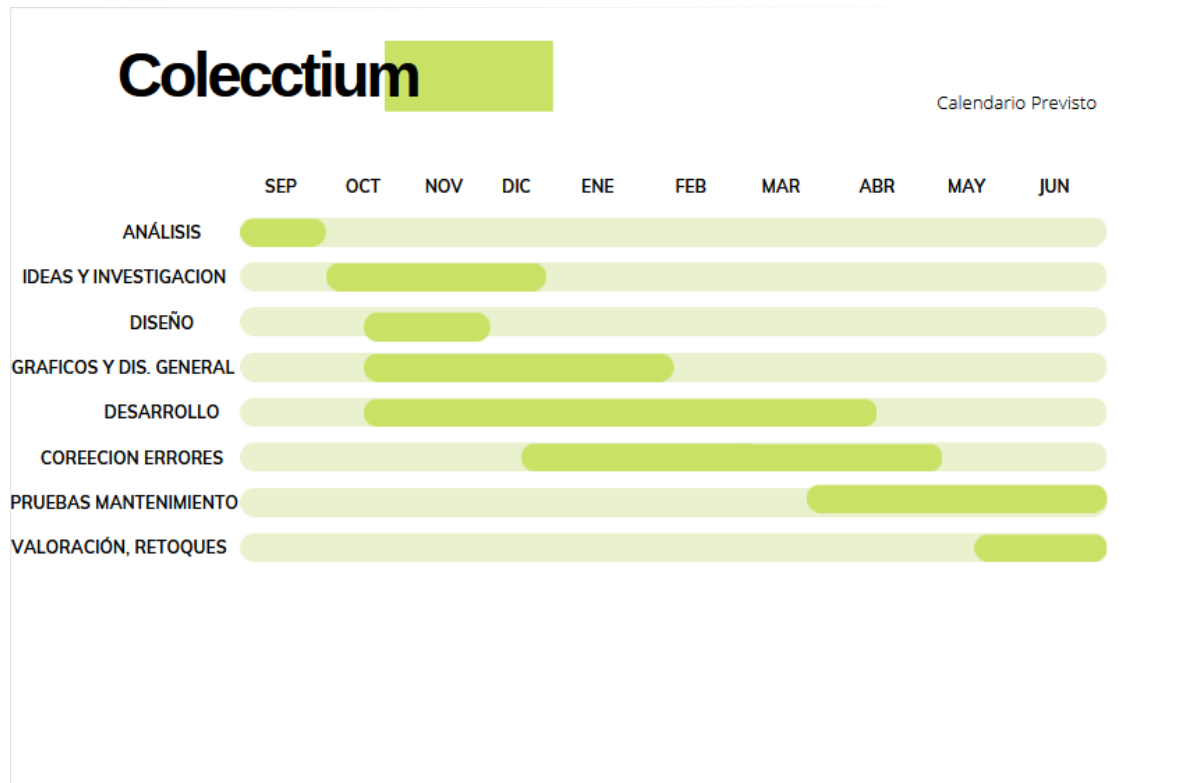


Informacion App

## 5.3. Implementación

### 5.3.1. Código

Este proyecto empezó en septiembre y se llevó a cabo durante todo el curso. Esto se muestra en el siguiente diagrama de Gantt.



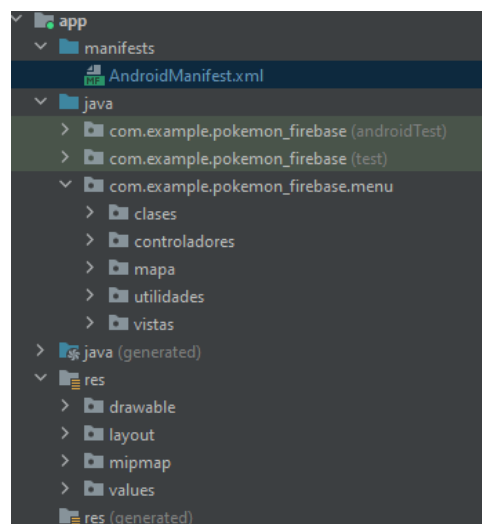
Todo este desarrollo se ha llevado a cabo con Android, un sistema de pila de código abierto basado en Linux, está creado para trabajar con una amplia gama de dispositivos. A continuación, se muestra la estructura de este software.



Dentro de esta estructuranos centraremos en el marco de trabajo de la API de Java. Esta es la base que vamos a necesitar para poder crear y desarrollar nuestras aplicaciones, simplificando la reutilización de componentes del sistema. Podemos destacar:

- Administrador de recursos: Aquí nos vamos a encontrar los recursos mas importantes del software que son los Layouts, son archivos xml que nos permiten crear la interfaz del proyecto.
- Administrador de actividades (Activity): Estos sirven como punto para interactuar con nuestra App y son fundamentales para navegar por ella. También administra el ciclo de vida de nuestras aplicaciones.

Este proyecto se ha estructurado de la siguiente manera:



- Clases: Aquí encontramos las clases o modelo.
- Controladores: Aquí están las opciones de la base de datos (Firebase).
- Mapa: Aquí está la configuración del mapa.
- Vistas: Aquí están los activitys que hacen la aplicación.
- Utilidades: Tenemos métodos para subir y descargar las imágenes en el Storage.

## 6. CONCLUSIONES

Las conclusiones que puedo sacar de este proyecto es que al principio ha sido un poco complicado el empezar a organizar todo, por la elección de usar una base de datos NoSQL como es Firebase y las dificultades que me he ido encontrando, pero gracias a ir preguntando a compañeros, foros, profesores... Lo he podido sacar adelante.

Algunas de las dificultades que me he encontrado han sido a la hora de estructurar los datos en documentos y colecciones y realizar las consultas a la base de datos para recuperar los datos que necesitaba, pero al final leyendo la documentación y consultando foros como Stackoverflow y los distintos canales en YouTube me fueron solucionando las dudas.

También me gustaría recalcar que esta aplicación no creo que este completa al 100%, ya que puede tener varios puntos a mejorar. Primero mejorando el diseño, al no haber visto prácticamente nada de diseño de Apps, también creo que se podría adaptar a otros dispositivos móviles como tabletas, añadir algún mensaje de notificación al usuario para cuando un organizador cree un nuevo campeonato, este, en el rol de piloto pueda inscribirse.

Para terminar con las conclusiones sabemos que es imposible desarrollar un proyecto así de manera perfecta desde el inicio, nos encontramos con que no podemos mantener actualizada la App constantemente y seguro nos vamos a quedar un poco desfasados.

Contando con un equipo pequeño de programadores esto podría realizarse de una manera más rápida y simple, garantizando su constante evolución, la seguridad e integridad de los datos de nuestros clientes y la resolución de problemas que estos pudiesen tener.

## **7. BIBLIOGRAFIA**

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

<https://www.developer.android.com/guide?hl=es>

<https://www.youtube.com/channel/UCxPD7bsocoAMq8Dj18kmGyQ>

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

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

Apuntes y recursos de la asignatura PMDM

## **8. ANEXOS**

Junto con esta memoria se adjunta igual el archivo APK para instalar la aplicación en el dispositivo móvil elegido.

Además de un archivo pdf con los pasos a seguir para su instalación y 1 (correo y contraseña para el login) para realizar las pruebas que el tribunal crea necesarias.

### **Usuarios.**

Correo: [hola123@gmail.com](mailto:hola123@gmail.com)

Contraseña: 12345678