



## **CICLO DE GRADO SUPERIOR EN DESARROLLO DE APLICACIONES MULTIPLATAFORMA**

Proyecto de desarrollo de aplicaciones multiplataforma

MY BOARD GAMES APP

Autor: Juan Manuel Barroso Fernández

Tutor: Miguel Ángel Martín Martínez

Año: 2022

## **Resumen:**

MyBoardGamesApp es una aplicación pensada para aficionados a los juegos de mesa.

A raíz del confinamiento que sufrimos en 2020 como consecuencia del COVID-19, muchas personas sobrellevaron dicho confinamiento jugando a juegos de mesa y la afición por dichos juegos creció mucho en España.

Esta aplicación está pensada tanto para estos nuevos aficionados, como para aquellos que ya eran aficionados a los juegos de mesa modernos desde hace más tiempo.

Al iniciar la aplicación podemos ver una sencilla pantalla de inicio, en la que podemos poner nuestros datos de registro para hacer login en la aplicación, o bien pulsar en el botón de registro para ir a un sencillo formulario donde podremos rellenar nuestros datos de registro para posteriormente hacer login en la pantalla de inicio. También incluye un botón para recuperar tu contraseña en el caso de haberla olvidado.

Tras hacer login, la aplicación nos muestra una sencilla interfaz de 4 botones, donde podemos explorar los siguientes menús.

- El icono de Mi Perfil, te lleva a una pantalla donde puedes cambiar tu contraseña y poner una imagen de perfil, haciendo una foto desde el móvil o seleccionando una imagen de la memoria del móvil.

- El icono de Mis Juegos, te lleva a una pantalla donde puedes añadir los juegos de mesa que tengas o que vayas comprando, mostrándolo como un listado de elementos, en los que aparece el nombre, un botón para borrar el juego, un botón + para añadir las partidas que se jueguen a dicho juego, y un botón menos para borrar la partida en caso de añadirlo por error.

- El icono Tutoriales, nos lleva a una pantalla donde se ofrecen un listado de tutoriales recomendados de algunos de los juegos más importantes.

- El icono de Contacto nos lleva a una pantalla donde aparece un pequeño comentario del autor y ofrece un correo de contacto para peticiones, sugerencias y reporte de errores, así como un botón que nos dirige al perfil de LinkedIn del autor.

## **Índice:**

1. Introducción
  - 1.1 Planteamiento del problema
  - 1.2 Objetivos
  - 1.3 Metodología
2. Descripción del problema
3. El proyecto Software
  - 3.1 AndroidStudio
  - 3.2 Java
  - 3.3 SQLite
  - 3.4 SQLiteStudio
  - 3.5 Genymotion
4. Solución
  - 4.1 Pantalla Inicio
  - 4.2 Pantalla Registro
  - 4.3 Pantalla Recuperar Contraseña
  - 4.4 Pantalla Nueva Contraseña
  - 4.5 Pantalla Menú Principal
  - 4.6 Pantalla Mis Juegos
  - 4.7 Pantalla Nuevo Juego
  - 4.8 Pantalla Mis Juegos (Con datos)
  - 4.9 Pantalla Tutoriales
  - 4.10 Pantalla Contacto
5. Evaluación
6. Conclusión

## **Índice de figuras:**

Figura 1.1 Gráfico de hitos del proyecto.

Figura 3.1 Android Studio

Figura 3.2: Ejemplo de archivos de proyecto en la vista de Android

Figura 3.3 Java

Figura 3.4 SQLite

Figura 3.5 SQLiteStudio

Figura 3.6 Genymotion

Figura 3.7 Estructura de carpetas

Figura 3.8 Estructura de carpetas (2)

Figura 4.1 Pantalla de inicio

Figura 4.2 Pantalla de registro

Figura 4.3 Pantalla Recuperar contraseña

Figura 4.4 Pantalla Nueva contraseña

Figura 4.5 Pantalla Menú principal

Figura 4.6 Pantalla Mi perfil

Figura 4.7 Pantalla Mis juegos

Figura 4.8 Pantalla Nuevo juego

Figura 4.9 Pantalla Mis juegos (con datos)

Figura 4.10 Pantalla Tutoriales

Figura 4.11 Pantalla Contacto

**Índice de tablas:**

Tabla 3.1 Tabla User

Tabla 3.2 Tabla Games

## **1. Introducción:**

### **1.1 Planteamiento del problema:**

La idea de desarrollar esta aplicación surge como consecuencia del aumento de aficionados a los juegos de mesa a raíz del confinamiento provocado por la pandemia del COVID-19.

Aunque los juegos de mesa modernos ya contaban con un importante número de seguidores en este país, durante la pandemia dicho número aumentó notablemente, ya que al no poder salir de casa muchas personas buscaron otras formas de ocio que supliesen a las actividades que hacemos en la calle.

Mucha gente empezó a comprar juegos de mesa para jugar en sus casas y descubrieron que actualmente existían muchos juegos nuevos más allá de los clásicos que jugábamos cuando éramos niños.

Esto provocó que aumentara el número de aficionados que empiezan a comprar juegos de mesa y a buscar información sobre esta afición.

Una vez entras en esta afición mucha gente empieza a querer tener control de los juegos que tienen y de las partidas que juegan y ahí es donde entra nuestra aplicación.

MyBoardGamesApp, pretende ser una ayuda para estos nuevos aficionados a los juegos de mesa, ofreciéndoles la posibilidad de ir agregando los juegos que van comprando a la aplicación, así como registrar las partidas que van jugando a cada juego, para así ver a que juegos han jugado más y a cuáles menos. De esta forma tienen una herramienta para controlar el número de juegos que tienen y poder valorar si ese juego de mesa al que no juegan casi nunca merecería más la pena venderlo para comprar otro al que den más uso.

Por otro lado, a muchos de los aficionados les aburre o se les hace pesado leerse las reglas de algunos juegos de

mesa, con lo que muchos acuden a tutoriales en YouTube para ver cómo se juega. Aunque suelen existir muchos tutoriales disponibles en YouTube para aprender a jugar a un juego en concreto, muchos de dichos tutoriales tienen errores o no se cuentan todas las reglas, lo que provoca que juegues mal tu partida y la experiencia de juego no sea la que debiera ser. Para ayudar a estos nuevos aficionados se ofrece la sección de tutoriales, donde se ofrecen tutoriales recomendados y de los que el creador de la aplicación sabe que son correctos y cuentan todas las reglas de forma precisa, de ese modo el usuario de la aplicación no tiene que perder el tiempo en ver varios tutoriales para ver cual está mejor o arriesgarse a ver un tutorial de mala calidad.

## 1.2 Objetivos:

Como objetivos, la primera intención es conseguir una aplicación sencilla con una funcionalidad básica que con el tiempo vaya evolucionando y ampliándose con nuevas funcionalidades.

El público objetivo de esta aplicación es el de nuevos aficionados que están empezando en este hobby y que empiezan a comprar y buscar información sobre juegos de mesa, y pueden verse un poco abrumados por la cantidad y variedad de juegos que existen.

Esta aplicación pretende ayudarles a tener un registro de juegos y partidas para saber que juegos tienen y a cuáles juegan más y a cuáles menos.

Así mismo en la sección de tutoriales se ofrecen tutoriales a varios de los juegos de mesa más conocidos y valorados, de forma que estos nuevos aficionados, pueden usar esta herramienta, aparte de para aprender cómo se juega al juego en el caso de comprarlo, como guía de compra, al tener un listado de algunos de los mejores juegos de mesa con su correspondiente video para hacerse a la idea de cómo es y si les gusta.

Como la aplicación está pensada para aficionados nuevos que van a ir creciendo en la afición al mismo tiempo que la aplicación crecerá en un futuro, se ofrece un correo de contacto para sugerir nuevas funcionalidades y reportar

bugs. De esta manera si una funcionalidad concreta tiene mucha demanda, esto puede hacer que las prioridades del desarrollador se orienten primero a satisfacer dichas necesidades.

El principal objetivo para el futuro es añadir un foro, para ayudar a estos nuevos aficionados a conocer gente nueva para jugar a juegos de mesa.

### 1.3 Metodología:

Para el desarrollo de esta aplicación se ha optado por una metodología de desarrollo de software incremental, de forma que se han planificado diferentes hitos o incrementos para que la aplicación vaya creciendo.

El modelo incremental de gestión de proyectos tiene como objetivo un crecimiento progresivo de la funcionalidad. Es decir, el producto va evolucionando con cada una de las entregas previstas hasta que se amolda a lo requerido por el cliente o destinatario.

Este enfoque, que se usó inicialmente para proyectos de software, aunque más tarde se aplicó a otros sectores, establece entregas parciales mediante un calendario de plazos. En cada una de ellas, el producto debe mostrar una evolución con respecto a la fecha anterior; nunca puede ser igual.

La principal diferencia del modelo incremental con los modelos tradicionales es que las tareas están divididas en iteraciones, es decir, pequeños lapsos en los cuales se trabaja para conseguir objetivos específicos.

Sin embargo, no se trata de iteraciones independientes. Estas iteraciones están vinculadas de forma que cada una suponga un avance con respecto a la anterior. Otras características esenciales de este modelo son:

- Los incrementos son pequeños.
- Permite una fácil administración de las tareas en cada iteración.
- La inversión se materializa a corto plazo.



- Es un modelo propicio a cambios o modificaciones.
- Se adapta a las necesidades que surjan.

Para que esto sea posible, se debe tener en cuenta que las iteraciones no pueden ser demasiado rígidas y que no existan tareas simultáneas. El modelo incremental exige un encadenamiento progresivo de cada tarea.

En nuestro caso, y dada la particularidad de ser un proyecto desarrollado por una sola persona, la planificación de las diferentes etapas o hitos fue la siguiente:

-El primer hito que se planificó fue el de crear por completo los layouts que formarán la interfaz gráfica de la aplicación, donde el mayor reto fue la creación de ListViews personalizados tanto en la vista de Tutoriales, como en la de Mis juegos, ya que se pretendía que los diferentes elementos se mostraran de una forma concreta a la hora de añadirlos.

-El segundo hito que se planificó fue el de crear las distintas clases activity necesarias para los layouts que se crearon en la fase anterior, e implementar la correcta navegación entre las distintas vistas, añadiendo los eventos correspondientes a los botones de la aplicación y aprovechando el botón de "atrás" que ofrecen los teléfonos móviles Android.

-El tercer hito que se planteó fue el de programar la funcionalidad de la pantalla de registro, lo que conllevaba programar la creación de la base de datos y programar los diferentes condiciones de cada uno de los campos, como definir el patrón del campo destinado al correo electrónico, definir las condiciones que debe cumplir la contraseña, la cantidad máxima de caracteres permitidas para el nombre de usuario, etcétera.

También se planificó como parte de este tercer hito añadir la funcionalidad a la pantalla de inicio de la aplicación, ya que está íntimamente ligada a la pantalla de registro, para así comprobar que los datos que se han añadido en la pantalla de registro se cargan correctamente al hacer login

en la aplicación, además de programar los mensajes de error que puede leer el usuario en el caso de que no se haya registrado previamente o haya introducido mal los datos de usuario y/o contraseña.

También se planificó la creación del botón de recuperación de contraseña.

-El cuarto hito que se planificó fue el de implementar la funcionalidad de la pantalla de "Mi Perfil", donde el usuario podrá cambiar su contraseña y añadir una foto de perfil.

-El quinto hito planificado fue añadir la funcionalidad de la pantalla de "Tutoriales", donde se añadieron al ListView personalizado los diferentes ítems que se pretenden mostrar, con su imagen, nombre y botón para ver el tutorial correspondiente.

-El sexto hito que se planificó fue el de añadir la funcionalidad a la pantalla de "Mis Juegos", donde se irán agregando los diferentes juegos que vaya añadiendo el usuario, presentándolos en un ListView personalizado, mostrando su nombre, botón de borrar juego y los botones de + y - para añadir o restar partidas.

Esta fase fue una de las más complejas, al tener que investigar como ir transmitiendo los datos de Id del usuario entre las distintas vistas, para que cuando el usuario agregase sus juegos, se guardasen correctamente en la base de datos con el Id del usuario que está logueado.

-El séptimo hito que se planeó fue el de dar funcionalidad a la pantalla de contacto.

-El octavo y último hito que se planificó fue el de realizar diferentes pruebas de integración, funcionales, desempeño y aceptación y corrección de detalles de la aplicación.

Se tuvo en cuenta que en esta fase podrían descubrirse fallos y poner de manifiesto pequeñas necesidades que no se había tenido en cuenta al principio del proyecto.

Durante cada uno de los hitos se fueron realizando pruebas para comprobar que todo lo que se añadía a la aplicación funcionaba correctamente.

Tras estas etapas se obtendría la aplicación final y de haber trabajado para un cliente, el cliente podría haber ido recibiendo pequeñas actualizaciones con funcionalidades nuevas en cada fase, donde podría expresar su opinión e ir adaptando el producto sus necesidades sobre la marcha, en vez de recibir un producto final al final del desarrollo y que el cliente vea que no es lo que necesitaba, bien porque nosotros no hemos sabido hacer lo que el cliente pedía, o bien por que el cliente no supo explicar o ni sabía muy concretamente lo que quería.

Con esta metodología aseguras aprovechar el tiempo de manera óptima, teniendo el feedback del cliente en etapas tempranas del desarrollo, evitando tener que rehacer una gran cantidad de trabajo en caso de que el cliente cambie de idea al final del desarrollo.

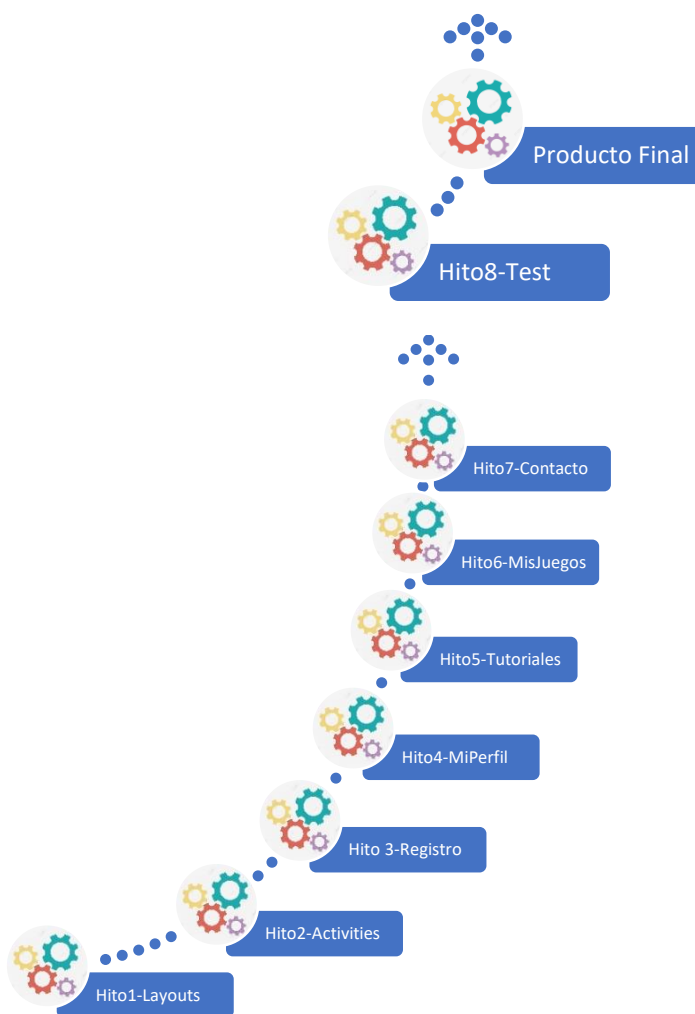


Figura 1.1 Gráfico de hitos del proyecto.

## 2. Descripción del problema:

Para la necesidad que se ha planteado en el punto 1.1, la solución que se ha pensado es realizar una aplicación sencilla que ayude a los nuevos aficionados a los juegos de mesa en sus primeros pasos en la afición y que vaya creciendo al mismo tiempo que ellos crecen en este hobby.

Ya que dar los primeros pasos en esta afición puede ser un poco abrumador al principio, esta aplicación va a ayudar a los nuevos aficionados a ir descubriendo nuevos juegos de calidad gracias a la sección de tutoriales, al mismo tiempo que sirva como herramienta para tener un registro de los juegos que poseen y de las partidas que han jugado a cada uno de ellos.

## 3. El proyecto Software:

Para realizar el proyecto se han utilizado las siguientes tecnologías:

**3.1 Android studio:** Android Studio es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de apps para Android y está basado en IntelliJ IDEA. Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece incluso más funciones que aumentan su productividad cuando desarrollas apps para Android, como las siguientes:

- A.** Un sistema de compilación flexible basado en Gradle.
- B.** Un emulador rápido y cargado de funciones.
- C.** Un entorno unificado donde puedes desarrollar para todos los dispositivos Android.
- D.** Aplicación de cambios para insertar cambios de código y recursos a la app en ejecución sin reiniciarla.
- E.** Integración con GitHub y plantillas de código para ayudarte a compilar funciones de apps comunes y también importar código de muestra.

- F.** Variedad de marcos de trabajo y herramientas de prueba.
- G.** Herramientas de Lint para identificar problemas de rendimiento, usabilidad y compatibilidad de versiones, entre otros.
- H.** Compatibilidad con C++ y NDK.
- I.** Compatibilidad integrada con Google Cloud Platform, que facilita la integración con Google Cloud Messaging y App Engine.



Figura 3.1 Android Studio

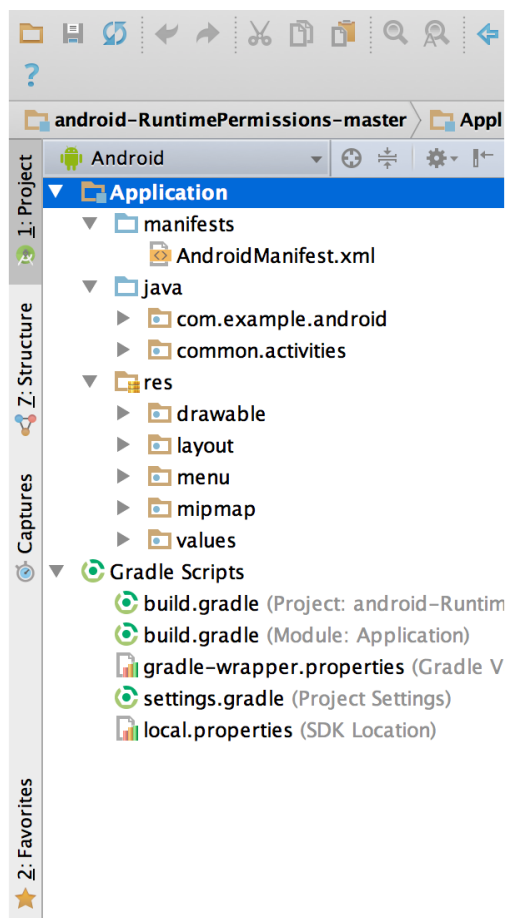


Figura 3.2: Ejemplo de archivos de proyecto en la vista de Android

Cada proyecto de Android Studio incluye uno o más módulos con archivos de código fuente y archivos de recursos. Entre los tipos de módulos se incluyen los siguientes:

- Módulos de apps para Android
- Módulos de biblioteca
- Módulos de Google App Engine

De manera predeterminada, Android Studio muestra los archivos de tu proyecto en la vista de proyecto de Android, como se ve en la figura 3.2. Esta vista está organizada en módulos para que puedas acceder rápidamente a los archivos fuente clave de tu proyecto.

Puedes ver todos los archivos de compilación en el nivel superior de **Secuencias de comando de Gradle** y cada módulo de app contiene las siguientes carpetas:

**manifests:** contiene el archivo AndroidManifest.xml.

**java:** contiene los archivos de código fuente Java, incluido el código de prueba de JUnit.

**res:** contiene todos los recursos sin código, como diseños XML, strings de IU e imágenes de mapa de bits.

**3.2 Lenguaje Java con open JDK 17:** Java es un lenguaje de programación y una plataforma informática que fue comercializada por primera vez en 1995 por Sun Microsystems... Java es rápido, seguro y fiable. Si es ejecutado en una plataforma no tiene que ser recompilado para correr en otra. Java es, a partir de 2012, uno de los lenguajes de programación más populares en uso, particularmente para aplicaciones de cliente-servidor de web, con unos diez millones de usuarios reportados. Java fue desarrollado originalmente por James Gosling, de Sun Microsystems y publicado en 1995 como un componente fundamental de la plataforma Java de Sun Microsystems. Su sintaxis deriva en gran medida de C y C++, pero tiene menos utilidades de bajo nivel que cualquiera de ellos. Las aplicaciones de Java son compiladas a bytecode (clase Java), que puede ejecutarse en cualquier máquina virtual Java (JVM) sin importar la arquitectura de la computadora subyacente.



Figura 3.3 Java

Por su parte Open JDK 17 trajo consigo las siguientes nuevas características:

- A.** Restaurar la semántica estricta de coma flotante.
- B.** Generadores de números pseudo-aleatorios mejorado.
- C.** Nuevo pipeline de renderizado para macOS.
- D.** Portado a macOS/AArch64.
- E.** Marcado como obsoleto para eliminación la Applet API.
- F.** Encapsulación fuerte de las clases internas del JDK.
- G.** Eliminación de RMI Activation.
- H.** Clases sealed.
- I.** Eliminar el compilador experimental AOT y JIT.
- J.** Filtros de deserialización específicos para cada contexto.

### **3.3 SQLite:**



Figura 3.4 SQLite

SQLite, creada por Richard Hipp, es un sistema de gestión de bases de datos relacional compatible con ACID, contenida en una relativamente pequeña biblioteca escrita en C.

A diferencia de los sistemas de gestión de bases de datos cliente-servidor, el motor de SQLite no es un proceso independiente con el que el programa principal se comunica. En lugar de eso, la biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo. El programa utiliza la funcionalidad de SQLite a través de llamadas simples a subrutinas y funciones. Esto reduce la latencia en el acceso a la base de datos, debido a que las llamadas a funciones son más eficientes que la comunicación entre procesos. El conjunto de la base de datos (definiciones, tablas, índices, y los propios datos), son guardados como un solo fichero estándar en la máquina host. Este diseño simple se logra bloqueando todo el fichero de base de datos al principio de cada transacción.

### **3.4 SQLite Studio:**



Figura 3.5 SQLiteStudio

SQLiteStudio es una interfaz gráfica para bases de datos SQLite, potente, ligero, rápido e intuitivo, que he utilizado para ayudar a trabajar visualmente con todo lo referente a la base de datos de la aplicación. Es una aplicación multiplataforma, disponible para los principales sistemas operativos Windows, MacOSX y Linux.

Principales Características:

- A. No hay que instalarlo, tan solo desempaquetar y ejecutar.
- B. Podemos exportar a diferentes formatos. Desde instrucciones SQL a CSV, HTML, XML, PDF ó JSON.
- C. Se pueden importar datos desde CSV
- D. Está soportado unicode
- E. Se puede configurar el aspecto, colores, fuentes y atajos.



- F. Es de código abierto y gratuito.
- G. Permite el uso de complementos.

SQLite Studio presenta diferentes complementos, algunos que vienen por defecto, y otros que se pueden instalar. De la misma forma, existen tanto complementos oficiales, como desarrollados por terceros.

Entre los complementos oficiales, pero que no vienen instalados por defecto cabe destacar DbAndroid. Este complemento de SQLite Studio te permite acceder directamente a bases de datos en un dispositivo Android.

### 3.5 Genymotion:



Figura 3.6 Genymotion

Como emulador para realizar pruebas y ver el funcionamiento de la aplicación he usado Genymotion.

Genymotion es un emulador de Android que aprovecha la arquitectura x86 para ejecutar de forma fluida y rápida distintos dispositivos Android. A diferencia de la lentitud de carga del emulador nativo de Android, podemos ejecutar todo tipo de aplicaciones y juegos en nuestros sistemas operativos Windows, Mac o Linux.

Uno de los principales usos de Genymotion es facilitar el desarrollo de aplicaciones Android. Se integran perfectamente

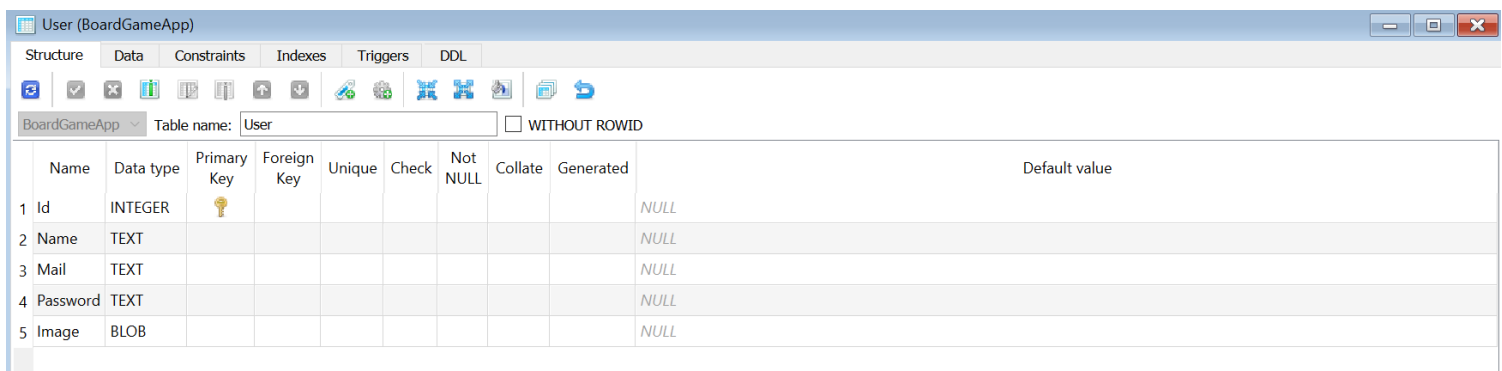
con Android Studio. Podemos lanzar, en apenas un par de segundos, las aplicaciones conectadas al ADT desde el IDE para testearla.

Genymotion está basado en el uso de máquinas virtuales x86 optimizadas para correr sobre Virtualbox (\*). Tiene una interfaz simple capaz de soportar distintas funcionalidades accesibles a cualquier usuario, sin olvidar a los desarrolladores que es su público objetivo.

GenyMotion permite realizar la instalación de máquinas virtuales emulando diferentes dispositivos Android como por ejemplo Nexus 4, Nexus 5, Nexus 7, Galaxy Nexus, HTC One, Moto X, Galaxy S4, Xperia Z etc. Además, se puede seleccionar la versión de sistema operativo Android que se desea utilizar e inclusive modificar la resolución de pantalla.

- Las tablas que se usan para la aplicación son las siguientes:

#### Tabla User:




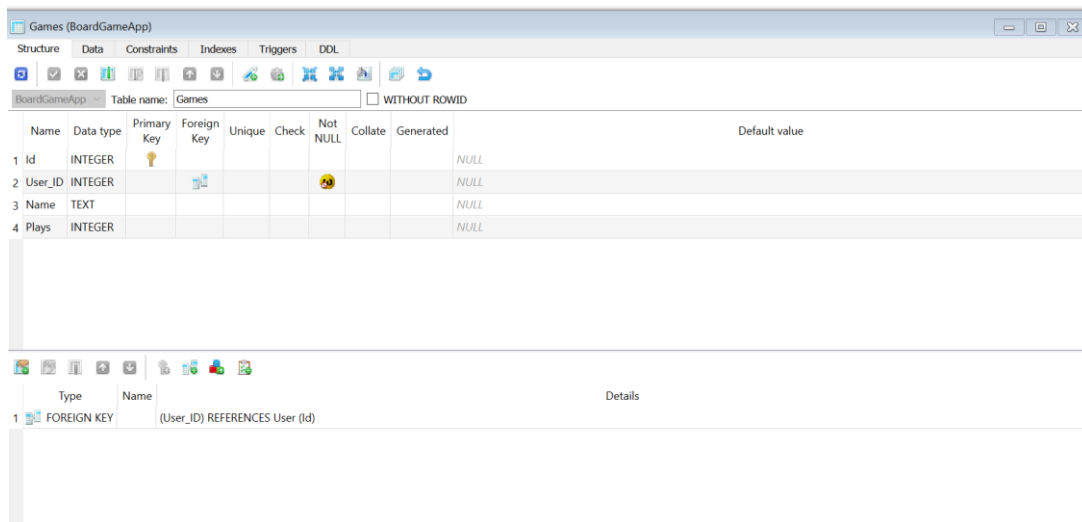
	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated	Default value
1	Id	INTEGER								NULL
2	Name	TEXT								NULL
3	Mail	TEXT								NULL
4	Password	TEXT								NULL
5	Image	BLOB								NULL

Tabla 3.1 Tabla User

## Tabla Games:



	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated	Default value
1	Id	INTEGER	Yes							NULL
2	User_ID	INTEGER		Yes			Yes			NULL
3	Name	TEXT								NULL
4	Plays	INTEGER								NULL

Type	Name	Details
FOREIGN KEY	(User_ID) REFERENCES User (Id)	

Tabla 3.2 Tabla Games

La estructura de carpetas que se ha utilizado para realizar el proyecto es la siguiente:

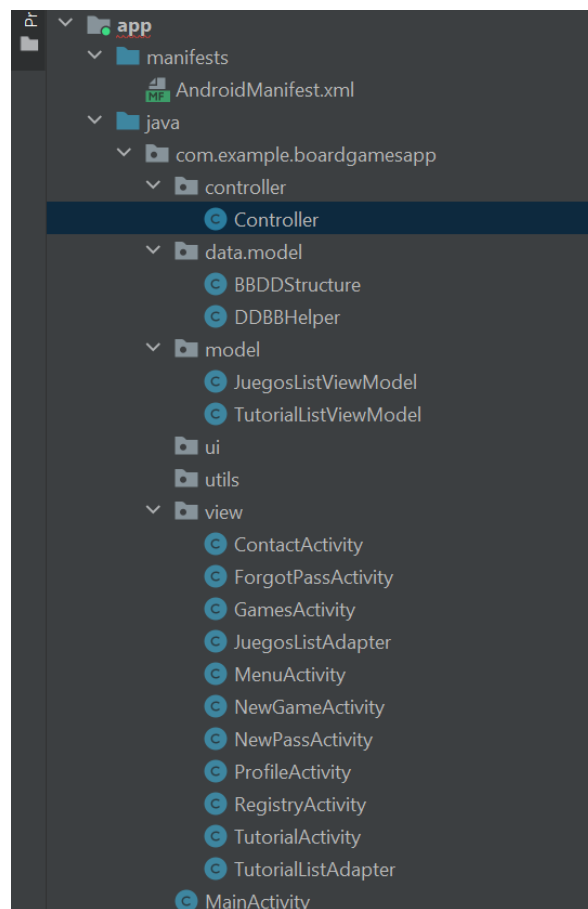


Figura 3.7 Estructura de carpetas

- Por un lado, Tenemos una carpeta Controller, donde tenemos la clase Controller donde albergar diferentes eventos.
- Por otro lado en la carpeta data.model, tenemos 2 clases, una para la estructura de la base de datos y otra para el helper.
- En la carpeta model, tenemos 2 clases necesarias para la creación de los ListView personalizados.
- En View, tenemos las diferentes clases necesarias para asociar a los diferentes layouts del proyecto.
- En la carpeta drawable tenemos todas las imágenes utilizadas en el proyecto.
- Por último, en la carpeta layout tenemos los diferentes layouts con los que formamos el aspecto gráfico de la aplicación.

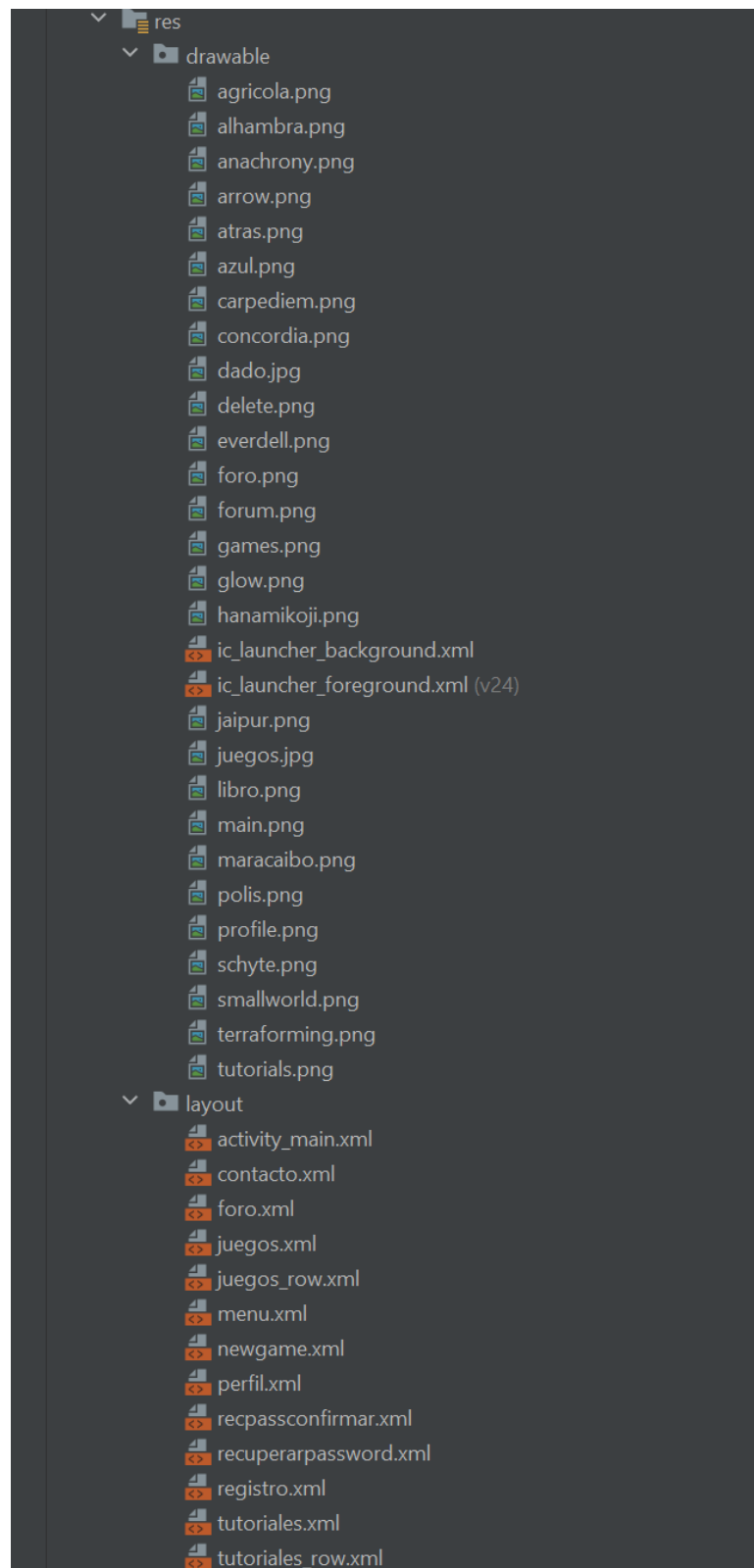


Figura 3.8 Estructura de carpetas (2)

## 4. Solución:

En esta sección presentamos las diferentes pantallas que dan forma a la aplicación, así como una breve descripción de las funcionalidades que tiene cada pantalla.

### 4.1 Pantalla de inicio:

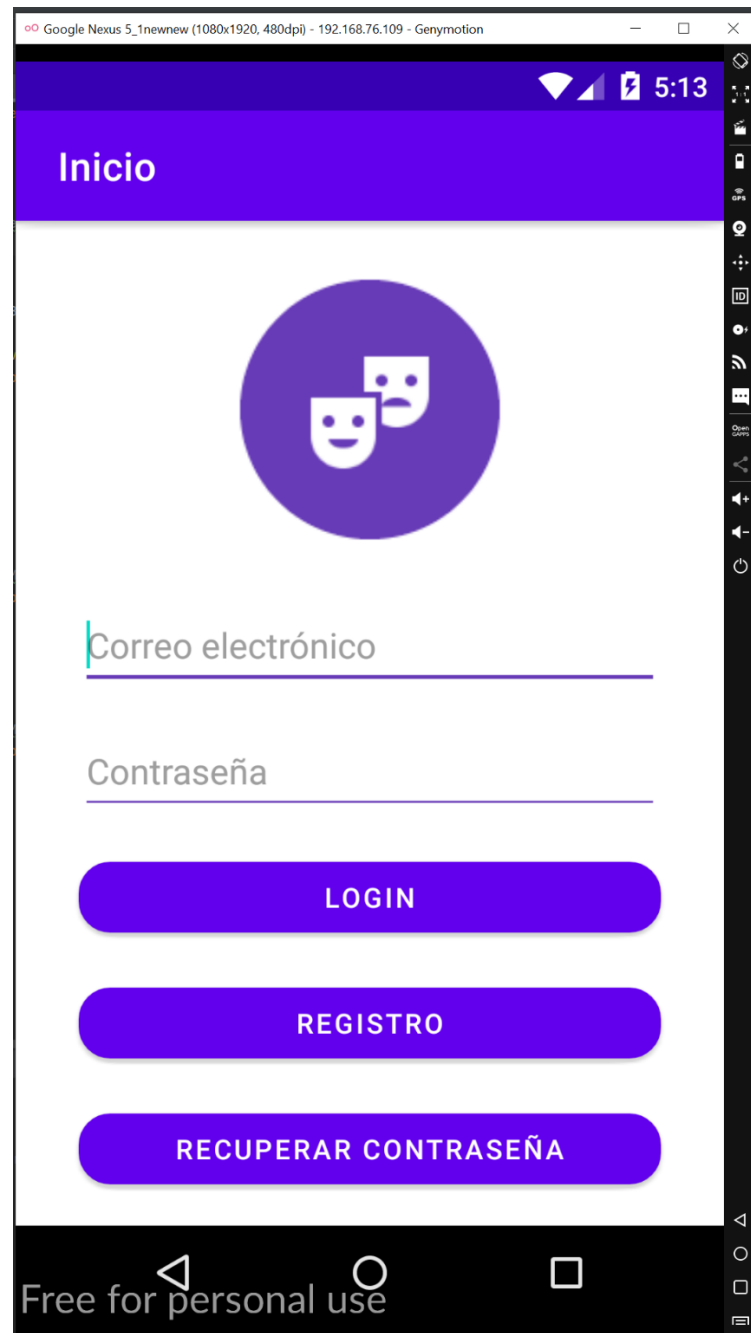


Figura 4.1 Pantalla de inicio

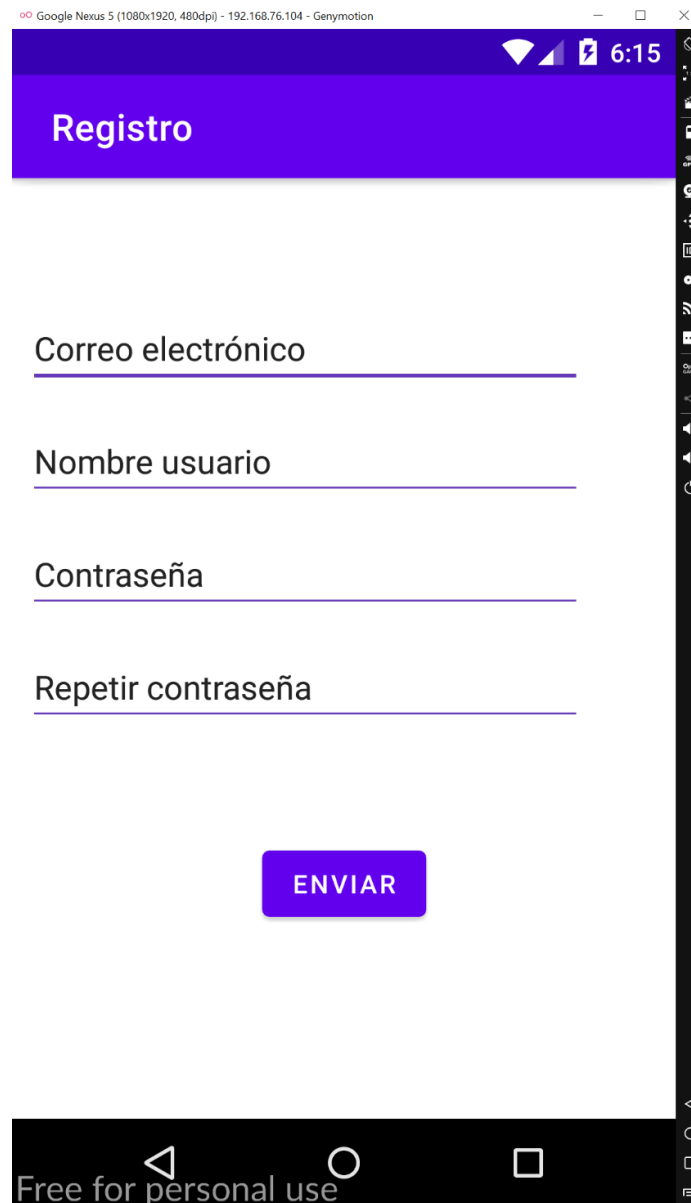
Esta es la primera pantalla que se ve al iniciar la aplicación.

La interfaz muestra dos EditText donde debemos poner el usuario y contraseña. Dichos EditText presentan mensajes de error que el usuario puede ver en el caso de tratar de hacer Login con unos datos de usuario y/o contraseña que no existen.

También incluye tres botones:

- El botón de login para que nos lleve a la pantalla de menú, en el caso de que dicha combinación de usuario y contraseña existan.
- El botón registro, para que nos lleve a la pantalla de registro en el caso de que sea la primera vez que iniciamos la aplicación y no tengamos un usuario creado, o bien para crear un usuario nuevo.
- El botón recuperar contraseña, que nos lleva a la pantalla de recuperar contraseña en el caso de que hallamos olvidado nuestra contraseña y queramos poner una nueva.

## 4.2 Pantalla de registro:



The screenshot shows a mobile application interface for registration. At the top, there is a purple header bar with the word "Registro" in white. Below the header, there are four text input fields with purple underlines, labeled "Correo electrónico", "Nombre usuario", "Contraseña", and "Repetir contraseña". Below these fields is a purple button with the text "ENVIAR" in white. The bottom of the screen shows a black navigation bar with three icons: a back arrow, a circle, and a square. The text "Free for personal use" is visible in the bottom left corner of the screen.

Figura 4.2 Pantalla de registro

Esta pantalla presenta cuatro EditText donde tendremos que poner nuestro correo electrónico, nombre de usuario, contraseña o confirmación de contraseña.

Dichos EditText tienen diferentes validaciones que presentan mensajes de error en el EditText afectado, en el caso de incumplir alguno de los requisitos.



### 4.3 Pantalla Recuperar contraseña:

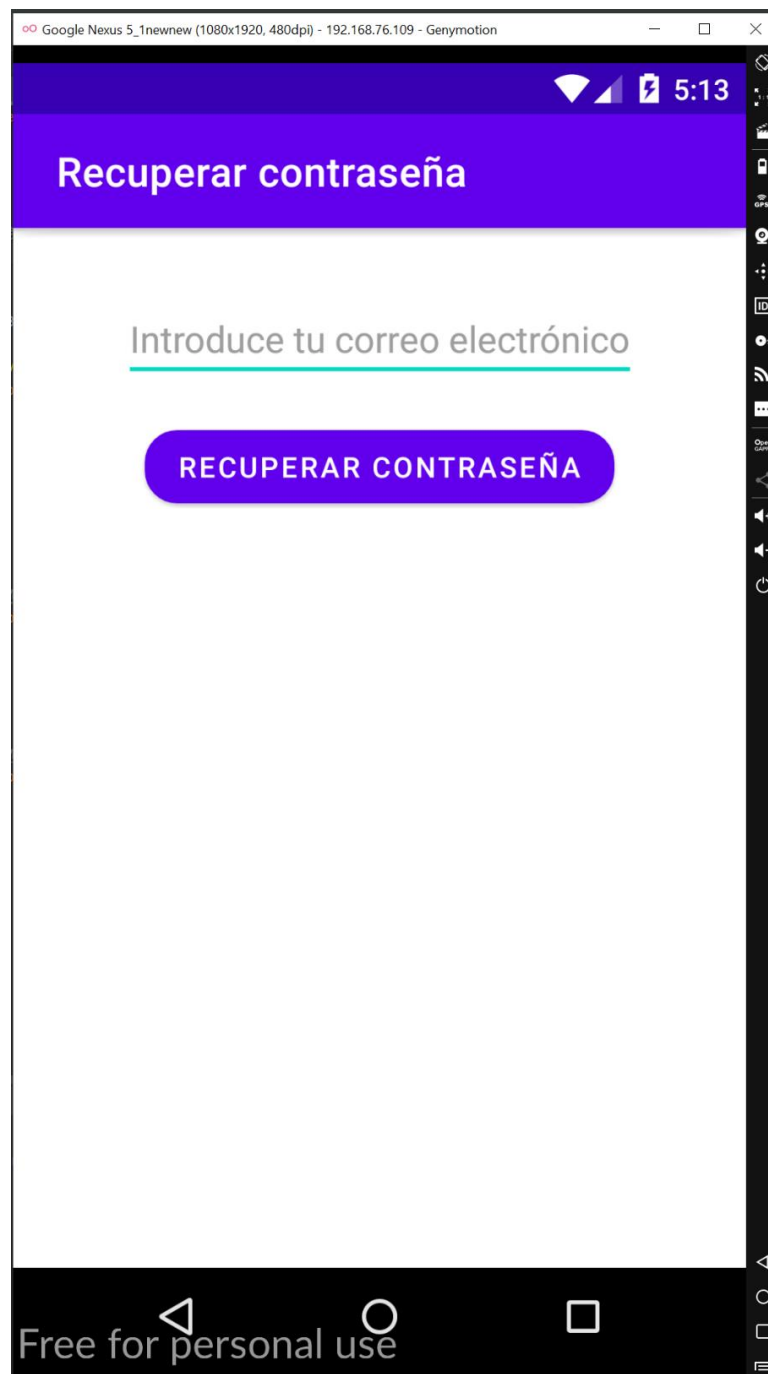


Figura 4.3 Pantalla Recuperar contraseña

En esta pantalla podemos introducir nuestro correo electrónico con el que nos hemos registrado y pulsar el botón de recuperar contraseña, para que en el caso de que dicho correo electrónico exista, nos lleve a otra pantalla donde poder cambiar nuestra contraseña.

#### 4.4 Pantalla Nueva contraseña:

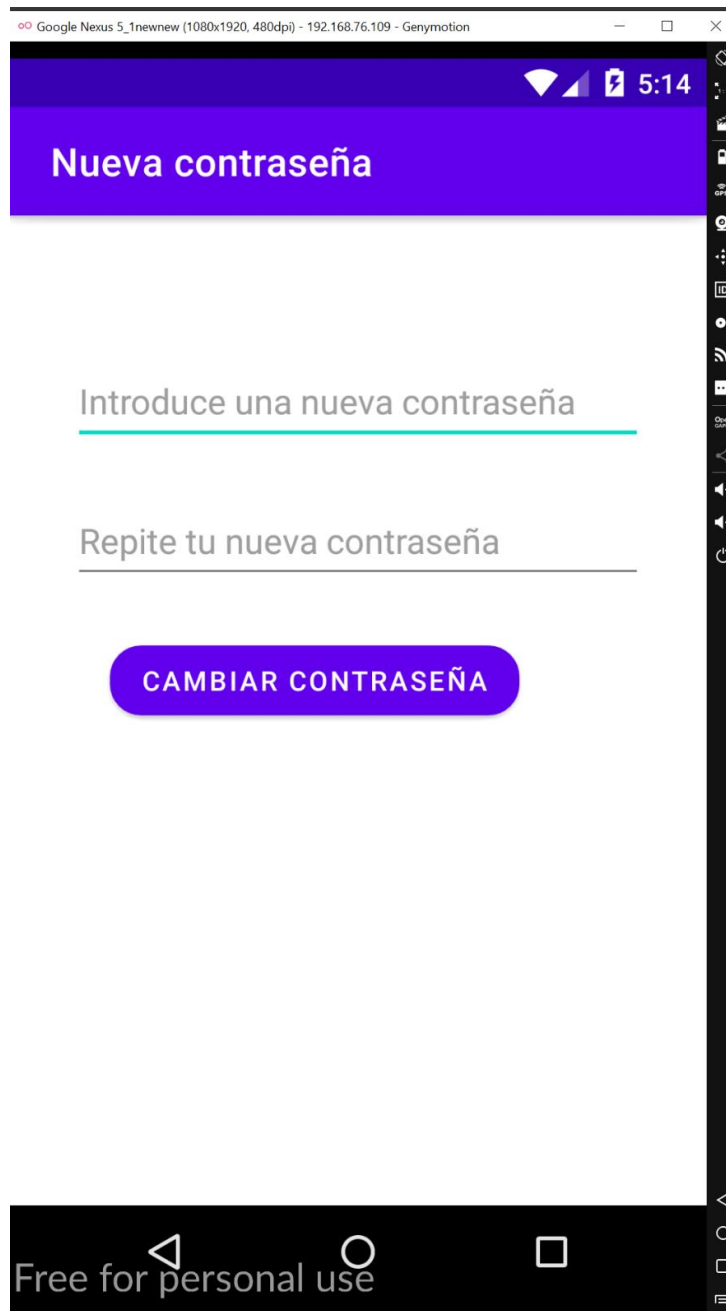


Figura 4.4 Pantalla Nueva contraseña

Aquí se presentan dos EditText donde poner nuestra nueva contraseña y repetir nuestra nueva contraseña, con las validaciones correspondientes, y un botón para cambiar la contraseña que nos mostrará un mensaje si se ha cambiado correctamente y nos dirige a la pantalla de inicio.

#### 4.5 Pantalla de menú principal:

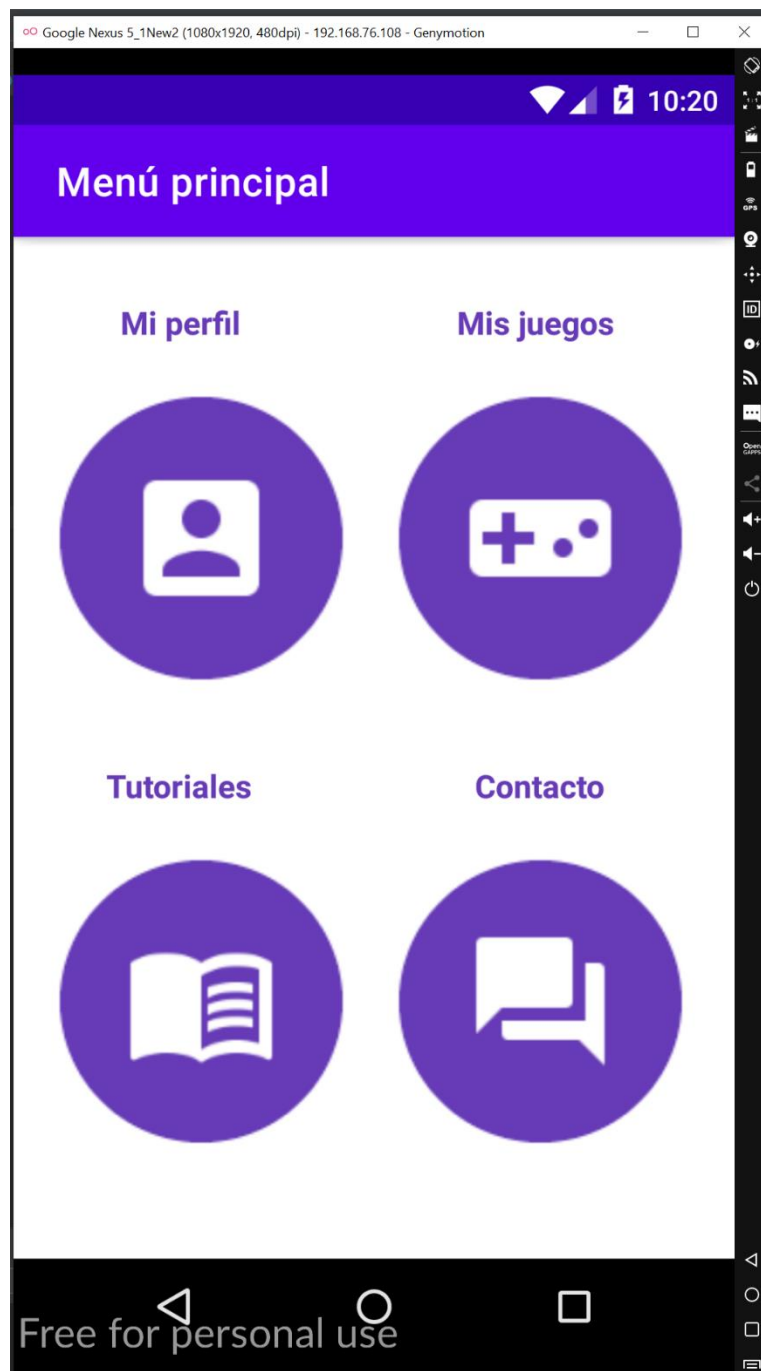


Figura 4.5 Pantalla Menú principal

Tras poner nuestros datos para hacer login en la aplicación, esta nos lleva a la pantalla de menú principal, donde se presentan cuatro botones que nos llevan a las pantallas de "Mi perfil", "Mis juegos", "Tutoriales" y "Contacto" respectivamente.

## 4.6 Pantalla Mi perfil:

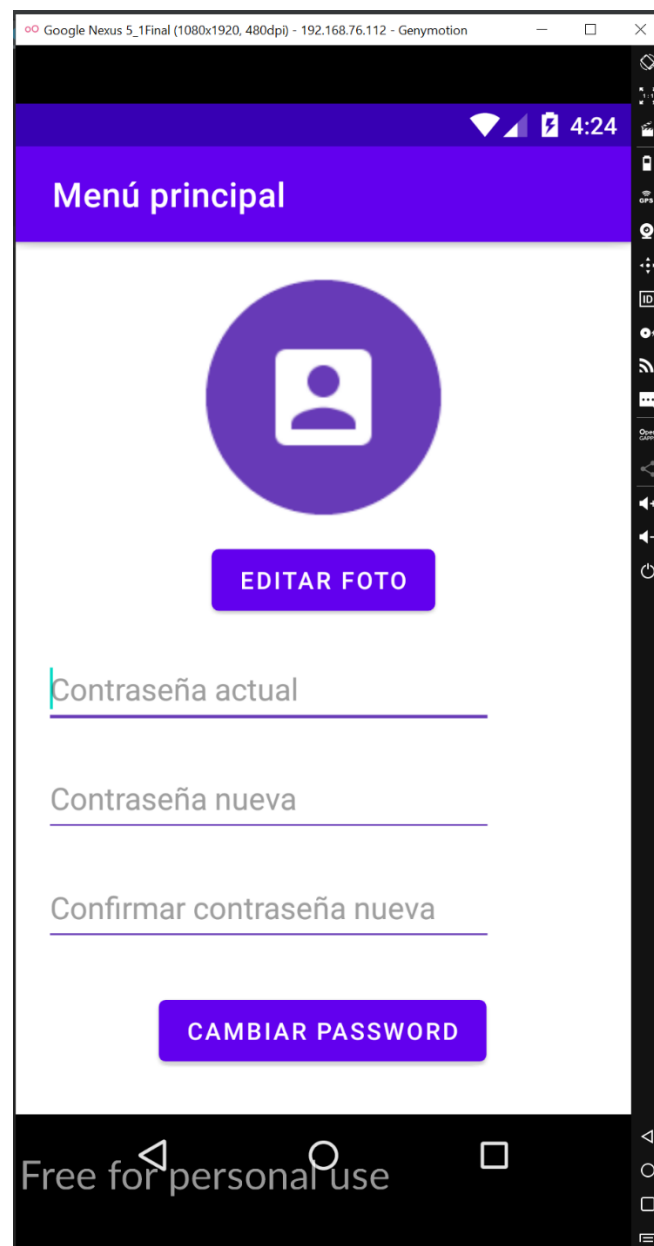


Figura 4.6 Pantalla Mi perfil

La pantalla de Menú principal presenta una imagen de perfil predefinida y un botón para editar la foto por defecto, donde podrá poner una imagen de la memoria del dispositivo móvil o hacer una foto y ponerla como foto de perfil.

También incluye tres EditText para hacer un cambio de contraseña pulsando el botón de cambiar password.

## 4.7 Pantalla Mis juegos:

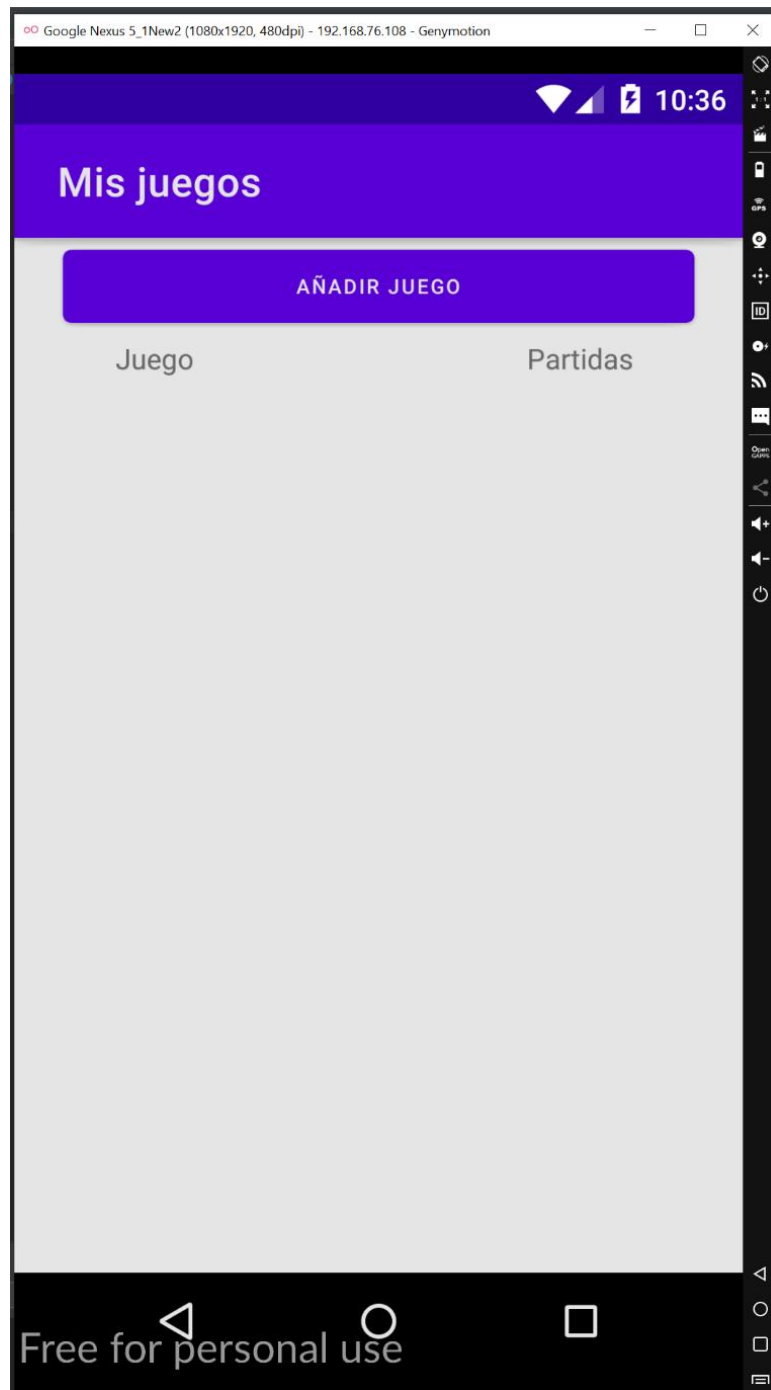


Figura 4.7 Pantalla Mis juegos

Dicha pantalla presenta un botón que nos lleva a la pantalla Nuevo juego para añadir los juegos que tengamos o que vayamos comprando.

## 4.8 Pantalla Nuevo juego:

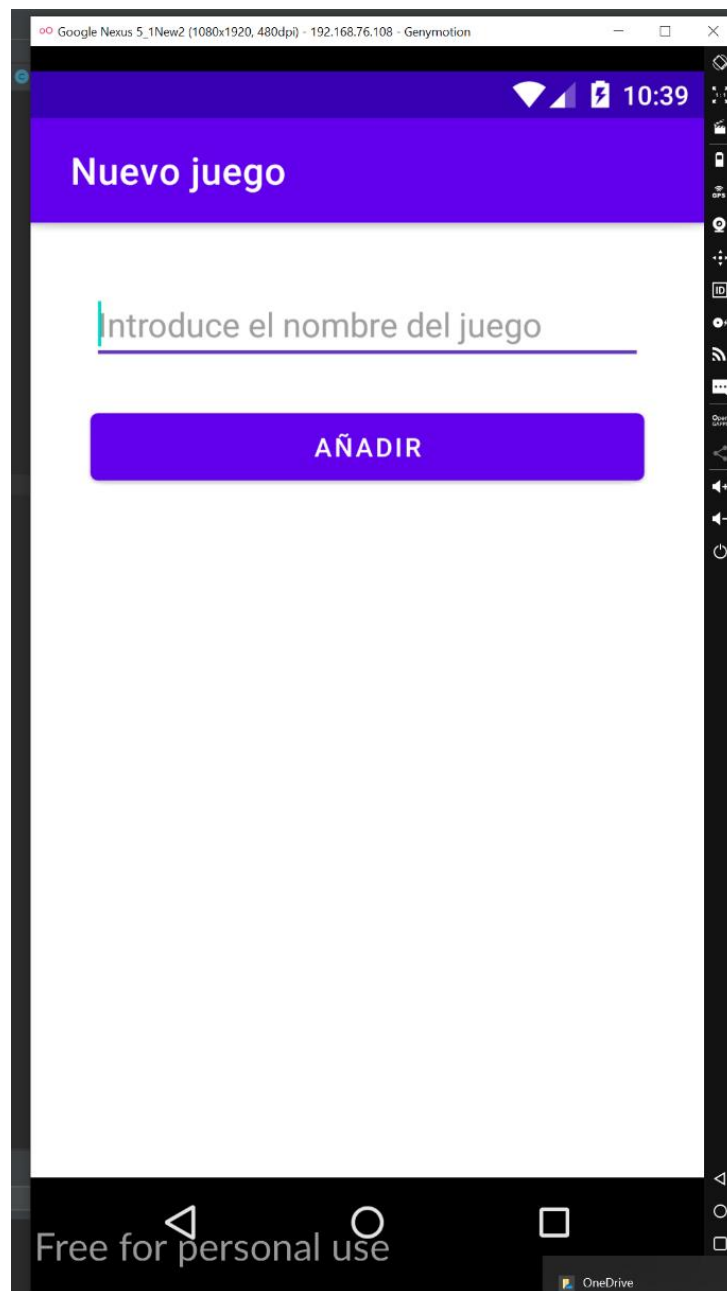


Figura 4.8 Pantalla Nuevo juego

En esta pantalla tenemos que ir introduciendo el nombre de los juegos que queramos añadir y pulsar el botón añadir, cuando hayamos añadido los juegos que queramos podemos dar al botón de volver y se presentarán en la pantalla de Mis juegos de la siguiente manera:

#### 4.9 Pantalla Mis juegos (con datos):

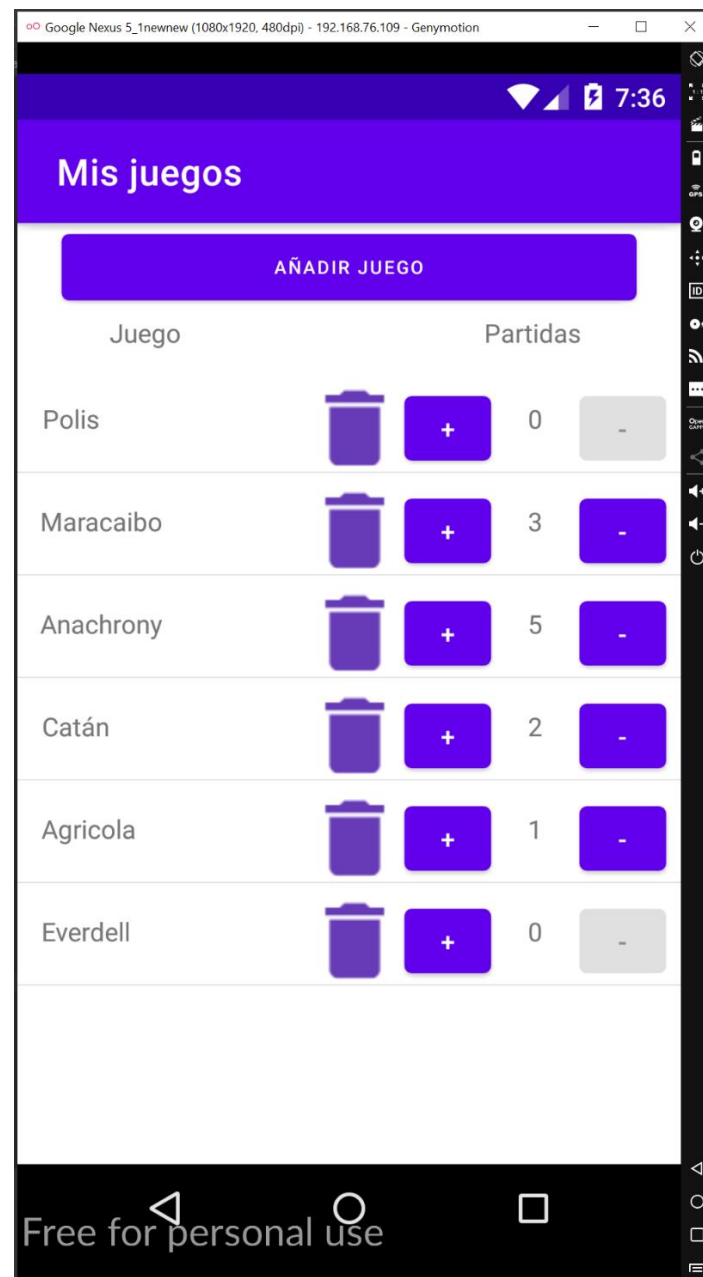


Figura 4.9 Pantalla Mis juegos (con datos)

Al añadir juegos se presenta el nombre del juego, un botón de cubo de basura para borrar el juego si lo has vendido y no quieres tenerlo en la aplicación o por qué has añadido algo por error, y unos botones de + y - para añadir las partidas que has jugado a cada juego e ir añadiendo partidas según las vallas jugando. Si el juego en cuestión tiene cero partidas jugadas, el botón de restar partida de deshabilita para que no te permita poner un número de partidas negativas.

#### 4.10 Pantalla Tutoriales:

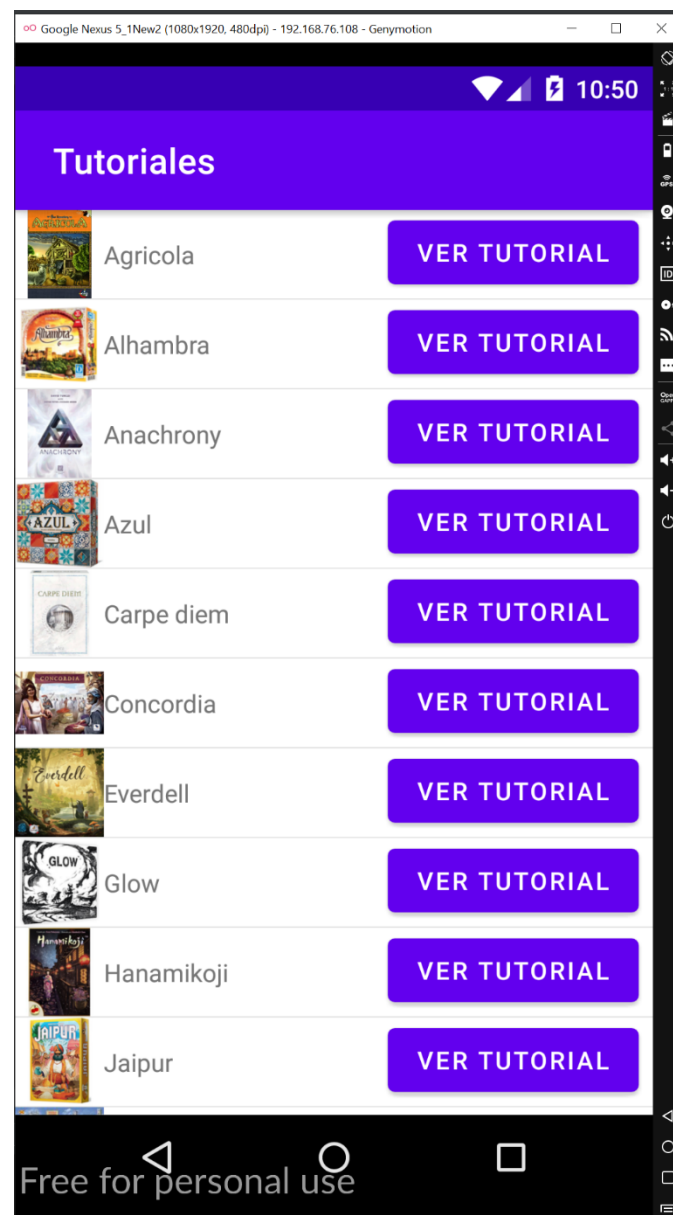


Figura 4.10 Pantalla Tutoriales

Esta pantalla presenta una selección de tutoriales recomendados, donde se ve una imagen y nombre del juego en cuestión y un botón para ver el tutorial correspondiente.



#### 4.11 Pantalla Contacto:



Figura 4.11 Pantalla Contacto

Esta pantalla presenta una breve presentación, una dirección de correo electrónico para contactar, y un botón que lleva al perfil de LinkedIn del desarrollador.

## 5. Evaluación:

Al realizar esta aplicación se ha priorizado el poder entregar un producto acabado y con una funcionalidad que cumpla con su cometido sin errores, por encima de incluir más funcionalidades que no dé tiempo a probar o implementar correctamente y puedan presentar fallos que afecten a la experiencia de usuario.

En el inicio de la planificación del desarrollo, mi idea era incluir una sección que fuera un foro donde los usuarios pudieran postear para proponer quedadas para jugar a juegos de mesa. Pero según iba avanzando el proyecto fui teniendo más dificultades de las que tenía previstas en el desarrollo de diferentes funcionalidades, consumiéndome más tiempo del que tenía previsto, así que descarté la idea por que iba a ser algo demasiado complejo para el tiempo disponible para hacer la aplicación.

Una de las partes que más tiempo me ha llevado es presentar los juegos y tutoriales en EditText personalizados, donde se muestren los elementos de la forma que yo quería, así como hacer que los nuevos juegos e imagen de perfil se guarde en la base de datos, con el id correspondiente al usuario que ha hecho login. Fue necesario ir pasando la información del id del usuario que ha hecho login entre todas las actividades para que la información mantuviera su persistencia y se guardara correctamente.

La navegación entre las distintas pantallas ha sido otra de las partes que más tiempo e investigación me ha requerido, ya que en un inicio pensaba que era suficiente con hacer distintos layouts para las diferentes vistas, pero luego vi que necesitaba distintas actividades para que la navegación entre pantallas fuese intuitiva y poder usar el botón "atrás" del móvil para navegar entre pantallas.

Sobre cómo me he visto a mí mismo durante el desarrollo. En ocasiones bastante perdido, ya que ha sido un gran salto hacer los ejemplos y actividades que me han mandado durante el curso y pasar a hacer un proyecto real, con la dificultad de buscar información en internet que no sé muy bien cómo aplicar a mi proyecto. Del mismo modo en ocasiones das con una solución que no es la más correcta para tus necesidades, todo este proceso consume muchísimo tiempo al hacerlo sin nadie que te guíe aunque sea mínimamente.

Por otro lado, si resulta muy satisfactorio cuando consigues que por fin funcione lo que estás tratando de desarrollar

## 6. Conclusión:

Esta aplicación representa un pequeño paso de lo que se pretende conseguir en un futuro con ella.

Se ha conseguido crear una aplicación con una funcionalidad contenida, pero suficiente para empezar.

En un futuro el primer paso para hacer crecer la aplicación será por un lado ampliar sustancialmente la sección de tutoriales, para abarcar el mayor número posible de juegos posible, y tratar de que cada juego importante tenga su tutorial disponible.

Por otro lado, se planea hacer uso de una base de datos con hosting en la nube, para que los datos de todos los usuarios de la aplicación estén en una sola base de datos y de esta forma añadir la funcionalidad de un foro, pensada para que los aficionados a juegos de mesa puedan postear en dicho foro para proponer quedadas para jugar a juegos de mesa y conocer a personas con la misma afición con ganas de conocer gente nueva con la que compartir este hobby, dentro de una aplicación.

Otra vía de ampliación de la aplicación sería que al añadir una nueva partida te diese acceso a un formulario donde puedas escribir los datos de la partida, como fecha, con quien la has jugado, resultados de la partida, etc., y así poder consultar dichas partidas al pulsar el nombre del juego en cuestión.

Realizar este proyecto ha sido muy positivo en todos los sentidos. A pesar de ser una aplicación con una funcionalidad contenida, hecha por un principiante, todo el proceso de investigación que ha requerido hacer cada detalle de la aplicación, es muy valioso como preparación de lo que nos espera en un futuro con esta profesión, en la que siempre tienes que estar investigando como hacer las tarea que se piden y saber buscar e interpretar la información de ayuda que encuentras en internet, tanto de fuentes oficiales como por ejemplo la Api de Java o página de Andriod Studio, como de fuentes no oficiales como páginas como stackoverflow, que proporciona una grandísima ayuda a la hora de buscar soluciones a errores que nos de nuestro código.

Los conocimientos que he mejorado tienen que ver sobre todo con el uso de la base de datos, combinada con nuestra aplicación, programar los eventos de los botones adecuados a lo que necesitamos y traspasar datos como el id del usuario de una

activity a otra, para conseguir la persistencia de los datos y que los datos guardados se guarden en la base de datos con el id correcto.

Ha sido una experiencia entretenida e inspiradora y te deja con ganas de haber tenido mas tiempo disponible para poder haber añadido más funcionalidades, que, por supuesto iré añadiendo en el futuro para lanzar la aplicación en la Google Store, pensando en tener en la aplicación un pequeño espacio para publicitar tiendas de juegos de mesa o canales de YouTube dedicados a esta afición.

**Lista de referencias:**

<https://www.obsbusiness.school/blog>

<https://developer.android.com/studio/intro?hl=es-419>

[https://es.wikipedia.org/wiki/Java\\_\(lenguaje\\_de\\_programaci%C3%B3n\)](https://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n))

<https://picodotdev.github.io/blog-bitix/2021/09/novedades-y-nuevas-caracteristicas-de-java-17/>

<https://atareao.es/software/ofimatica/sqlitestudio-o-como-trabajar-con-sqlite/>

<https://es.wikipedia.org/wiki/SQLite>

<https://academiaandroid.com/emuladores-android/#:~:text=final%20del%20art%C3%ADculo.-,GenyMotion,operativos%20Windows%2C%20Mac%20o%20Linux.>