

---

*Trabajo de fin de grado*

*DESARROLLO DE APLICACIONES  
MULTIPLATAFORMA*

---

# Índice

Introducción .....	6
Explicación del proyecto.....	6
Objetivos .....	6
Objetivos del proyecto.....	6
Objetivos personales.....	7
Otros objetivos .....	7
Arquitectura.....	7
MVVM .....	7
Patrón repository .....	8
Objetos observables (LiveData).....	8
Arquitectura final .....	8
Metodologías ágiles.....	8
Scrum .....	8
Gestión de Sprints.....	9
Explicación.....	9
Tablas .....	10
Fases del proyecto .....	13
Diseño .....	13
Nombre del proyecto .....	13
Logo.....	13
Personaje .....	14
Interfaces móviles – Esquema funcional.....	14
Modelo entidad relación .....	21
Explicación del modelo entidad relación .....	21
Tecnologías utilizadas .....	22
Figma – Diseño de las interfaces (iniciales).....	22
Android Studio – Creación de la aplicación .....	22
Room – Gestor de base de datos .....	23
Implementación y configuración .....	24
Servidor .....	24
Implementación de los minijuegos.....	27
Animaciones .....	28
Sonidos.....	31

Pruebas unitarias .....	33
ACCIONES PROGRAMADAS/AUTOMÁTICAS .....	33
ACCIONES DE PULSACIÓN .....	36
COMPROBACIONES .....	44
EFECTOS DE SONIDO / AUDIOS.....	49
Errores y soluciones .....	52
Ampliación y posibles.....	53
Conclusión .....	54
Bibliografía .....	55

## Índice de imágenes

Img 1- Arquitectura MVVM.....	7
Img 2 - Arquitectura final.....	8
Img 3- Esquema Scrum.....	9
Img 4 - Logo .....	14
Img 5 - Icono .....	14
Img 6 - Ratón blanco.....	14
Img 7 -Ratón gris .....	14
Img 8 -Ratón marrón .....	14
Img 9 - Pantalla de carga .....	15
Img 10 - Pantalla de inicio de sesión .....	15
Img 11 - Pantalla de registro .....	15
Img 12 - Pantalla perfil usuario 1.....	16
Img 13 - Pantalla perfil usuario 2.....	16
Img 14 -Pantalla de personalización .....	16
Img 15 - Pantalla de información .....	17
Img 16 - Pantalla del salón .....	17
Img 17 - Pantalla de la cocina .....	17
Img 18 - Pantalla del baño .....	17
Img 19 - Pantalla del dormitorio.....	17
Img 20 - Pantalla de la terraza.....	17
Img 21- Pantalla del armario.....	18
Img 22 - Pantalla de la tienda de complementos.....	18
Img 23 - Pantalla de la nevera .....	19
Img 24 - Pantalla del supermercado.....	19
Img 25 - Pantalla de la ducha .....	20
Img 26 - Pantalla del dormitorio de noche .....	20
Img 27 - Pantalla de la sala de juegos.....	21
Img 28 - Modelo entidad relación.....	21
Img 29 - Dependencias de Room .....	23
Img 30 - Esquema de Room.....	24
Img 31 - Clase MainService 1 .....	24
Img 32 - Clase MainService 2 .....	25
Img 33 - Clase MainService 3 .....	26
Img 34 - Clase MainService 4 .....	26
Img 35 - Clase MainServiceThread 1 .....	27
Img 36 - Clase MainServiceThread 2 .....	27
Img 37 - Pantalla inicio juego 3 en raya .....	28
Img 38 - Pantalla juego 3 en raya.....	28
Img 39 - Pantalla inicio juego memory .....	28
Img 40 - Pantalla juego memory 1 .....	28
Img 41 - Pantalla juego memory 2 .....	28
Img 42 - Editor lottifiles 1 .....	29
Img 43 - Editor lottifiles 2 .....	29
Img 44 - Editor lottifiles 3 .....	29
Img 45 - Descarga animación .....	29
Img 46 - Directorio RAW .....	29

Img 47 - Dependencias de lottifiles .....	30
Img 48 - XML componente lottie .....	30
Img 49 - Declaración del componente lottie .....	30
Img 50 - Referencia a la imagen animada .....	30
Img 51 - Uso de la animación .....	31
Img 52 - Búsqueda pixabay .....	31
Img 53 - Directorio RAW sonidos .....	32
Img 54 - Declaración MediaPlayer.....	32
Img 55 - Asignación del audio.....	32
Img 56 - Métodos necesarios.....	32
Img 57 - Uso del audio .....	33

## Introducción

"¡Bienvenido a Tailor-Mouse, el juego donde puedes cuidar a tu propia mascota virtual! En esta emocionante aventura, te convertirás en el mejor amigo de un adorable ratoncito, encargándote de todas sus necesidades y dándole un toque único con tu estilo personal.

Desde alimentarlo hasta darle un relajante baño, pasarás horas de diversión atendiendo a tu pequeño amigo. Además, podrás personalizarlo a tu gusto, ¡incluso vistiéndolo con la última moda en ropa y accesorios!

Con Tailor-Mouse, los más pequeños podrán disfrutar de un juego diseñado especialmente para ellos, donde aprenderán la importancia de cuidar a sus mascotas virtuales mientras se divierten. ¡Únete a la diversión y conviértete en el mejor cuidador de ratoncitos!"

## Explicación del proyecto

Para garantizar que disfrutes al máximo con tu entrañable mascota virtual, es necesario que te registres utilizando tu dirección de correo electrónico. Si ya tienes una cuenta, simplemente puedes iniciar sesión para reencontrarte con tu compañero virtual.

Durante el proceso de registro inicial, se te ofrecerá la posibilidad de elegir el color de tu ratón entre blanco, marrón y gris. Una vez hayas seleccionado el ratón que más te gusta, tendrás la libertad de ponerle el nombre que elijas.

Con tu mascota elegida y su nombre decidido, asumirás la responsabilidad de cuidarla. Será fundamental mantener un equilibrio en su nivel de cansancio, diversión y hambre. Controlar la fatiga y la diversión será fácil, pero para satisfacer el apetito de tu ratón, necesitarás ganar monedas participando en diversos minijuegos. Cada vez que juegues, el nivel de diversión de tu ratón aumentará.

Además, tendrás la oportunidad de personalizar la apariencia de tu mascota agregándole accesorios y atuendos únicos. Sin embargo, ten en cuenta que esta personalización también requerirá monedas.

## Objetivos

### Objetivos del proyecto

- › Implementar un sistema de registro e inicio de sesión seguro y fácil para los usuarios.
- › Desarrollar la personalización del ratón tanto con el color y el nombre.
- › Crear minijuegos desafiantes y entretenidos para ganar puntos.
- › Diseñar una tienda virtual con una interfaz intuitiva para la compra de atuendos y accesorios además de comida.
- › Poder incrementar las barras de necesidad del ratón realizando diferentes acciones.

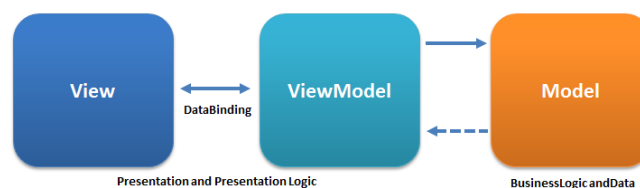
## Objetivos personales

- › Consolidar los conocimientos de programación para móviles en el entorno Android.
- › Aprender a gestionar Room como capa de abstracción para la base de datos SQLite.
- › Utilizar correctamente los conocimientos sobre programación de servicios y procesos (servidores e hilos).
- › Ampliar mis conocimientos en funciones de Android, como la implementación de animaciones y sonidos.

## Otros objetivos

Otro de los objetivos a nivel social, es desarrollar un juego enfocado en la crianza y cuidado de mascotas virtuales, inicialmente solo ratoncitos. Estas mascotas virtuales contarán con barras de estado que reflejarán su nivel de hambre, sed, sueño/energía y diversión. Esta dinámica ayudará a los usuarios a asumir responsabilidades a través del cuidado de mascotas con necesidades "vitales".

## Arquitectura



Img 1- Arquitectura MVVM

La arquitectura que he empleado para la aplicación es un modelo MVVM, empleando como capa de abstracción Room.

La arquitectura MVVM se centra en la segregación de los componentes de UI de la parte de persistencia empleando un presentador de información llamado ViewModel.

## MVVM

1. Cuando una actividad necesite un dato para mostrar por pantalla, se lo pide a Viewmodel.
2. Si viewmodel desconoce el dato que pide la activity, se lo pedirá a la base de datos.
3. La base de datos recibirá la petición del viewmodel y le devolverá una respuesta.
4. El viewmodel tomará la respuesta de la base de datos y la convertirá en un dato que espera la activity.
5. La activity modificará la interfaz con el dato recibido de viewmodel.

Para que las peticiones a base de datos no den problemas de concurrencia o bloqueo del mainthread, el viewmodel debe realizar peticiones asíncronas, por lo que una práctica recomendada es el uso del patrón "Repository".

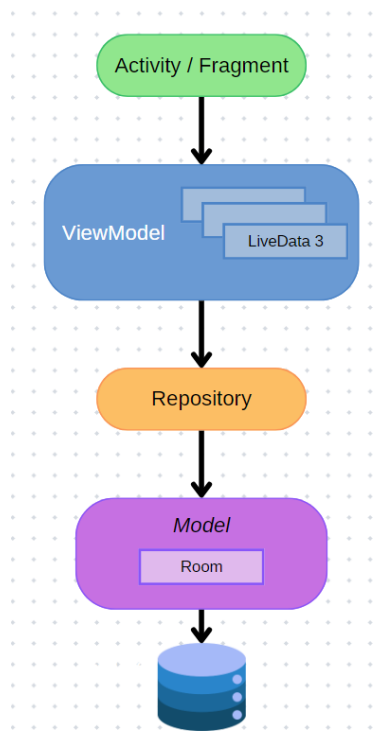
## Patrón repository

El patrón repository brinda a los componentes de la aplicación de un punto de entrada común y abstracto para la realización de peticiones a la persistencia (sin necesidad de saber el tipo de persistencia existente; base de datos local, base de datos en internet, archivo de preferencias...).

## Objetos observables (LiveData)

Son unos objetos especiales que pueden quedarse a la escucha del estado de la base de datos para poder reaccionar si algún campo cambia, y así poder reconstruir la vista al momento.

## Arquitectura final



Img 2 - Arquitectura final

## Metodologías ágiles

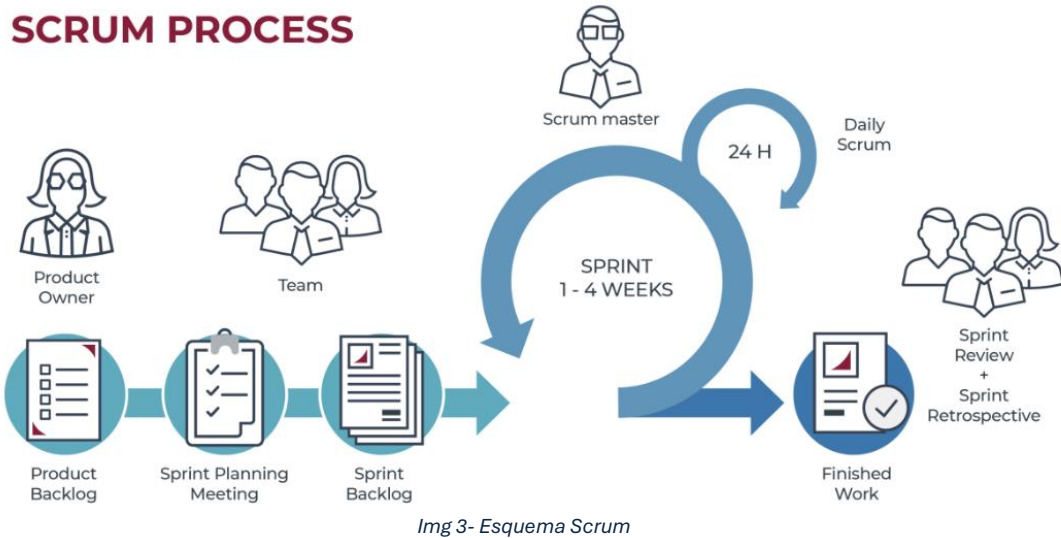
### Scrum

Scrum es un marco de trabajo ágil usado principalmente en desarrollo de software, basado en un enfoque iterativo e incremental. El trabajo se divide en "Sprints", iteraciones cortas en las que un equipo colaborativo entrega un incremento del producto. Los roles principales en Scrum son:

- › **Product Owner:** maximiza el valor del producto y gestiona el Backlog del Producto.
- › **Scrum Master:** facilita el proceso Scrum y elimina obstáculos.
- › **Equipo de Desarrollo:** desarrolla y entrega incrementos del producto.



En mi proyecto utilizo Scrum porque facilita la organización y el desarrollo. Cada mañana planifico las tareas a realizar, soluciono los problemas a medida que surgen y desarrollo e incorporo mejoras e ideas nuevas al proyecto diariamente.



## Gestión de Sprints

### Explicación

#### 1º Sprint

El primer sprint, inicialmente de 19 días, se reduce a 16 días debido a la anticipación de tres días de inactividad. Durante este periodo, la principal tarea consiste en la organización integral del sprint. Se establecerán las bases del proyecto, incluyendo la introducción, explicación y los objetivos principales. Además, se abordarán las primeras ideas para el diseño de interfaces y la estructura de la aplicación.

#### 2º Sprint

El segundo sprint, originalmente planeado para 14 días, se reduce a 11 días tras descontar los periodos de inactividad. Durante este tiempo, me enfocaré en documentarme e informarme sobre las tecnologías necesarias para mi aplicación. También dedicaré esfuerzos a diseñar el concepto inicial de la base de datos.

#### 3º Sprint

Durante el tercer sprint, programado para 11 días, se realizarán las actividades centradas en la base de datos. Se dedicará tiempo a crear los DAOs y las entidades necesarias. Asimismo, se avanzará gradualmente en el diseño definitivo de la aplicación.

#### 4º Sprint

El cuarto sprint, originalmente planificado para 21 días, se reduce a 14 días activos al considerar los periodos de inactividad. Este sprint se concentra en corregir y mejorar las actividades ejecutadas en los sprints previos.

### 5ºSprint

El quinto sprint tiene una duración de 12 días, descontando ya los días inactivos. Principalmente en este sprint me he dedicado a crear e implementar los complementos/artículos del ratón. Además de crear los minijuegos e implementarlos.

### 6ºSprint

Este sprint tiene una duración inicial de 14 días, descontando los días inactivos, se me queda una duración de 11 días. En él me he dedicado principalmente a corregir errores del anterior sprint. Además de corregir los ids y alguna que otra pantalla de diseño.

### 7ºSprint

Este sprint ha estado dedicado a la creación de la presentación, actualización de la memoria, modificar algún diseño de la base de datos para mejorar la lógica. En resumen, corregir y mejorar la aplicación en diversos puntos. Ha tenido una duración de 12 días descontando ya los días inactivos.

### 8ºSprint

Este sprint es el último y tiene una duración de 9 días. En él principalmente he ido probando la aplicación y solucionando errores. Han salido diversos errores inesperados o mejoras para el usuario por lo que también lo he dedicado a mejorar la aplicación.

## Tablas

FEBRERO				
Sprints	Día	Tareas	Horas estimadas	Horas reales
1ºSprint	1	Creación de la planificación de las tareas del primer sprint y general. Explicar la idea de forma general	1	2
	2			
	3	Introducción	1	1
	4	Objetivos (ideas)	1	1
	5	Descripción de dichos objetivos	1	1
	6		2	1
	7		2	2
	8	Corrección de los puntos anteriores	2	2
	9			
	10	Ir sacando ideas de diseño	2	2
	11		2	3
	12		2	1
	13	Diseñar interfaces	3	2
	14		3	3
	15	Planificación del resto del mes	1	0
	16			
	17	Creación de alguna de las interfaces de Android Studio	2	2
	18	Diseñar más interfaces	2	2
	19		2	1
	20		2	3

	21		2	1
	22	Información room	2	2
	23			
	24	Cursos Android Studio	2	2
	25		2	2
	26		2	2
	27		2	2
	28		2	2
	29		2	2

MARZO				
Sprints	Día	Tareas	Horas estimadas	Horas reales
2º Sprint	1			
	2			
	3	Planificar mes	1	1
	4	Generar el modelo entidad-relación	1	2
3º Sprint	5	Hacer las entidades	2	2
	6		2	1
	7	Hacer los DAOs	2	2
	8			
	9	Hacer los DAOs	2	2
	10	Investigar el gestor de base de datos e ir probando	2	2
	11		2	3
	12		2	0
	13		1	2
	14			
	15			
	16	Actualizar la documentación	1	1
	17	Documentarme sobre los diseños y layouts	2	3
	18	Crear las interfaces en Android Studio	2	6
4º Sprint	19		2	4
	20		2	4
	21		2	0
	22			
	23	Actualizar documentación	1	2
	24	Agregar consultas a las entidades y DAOs	2	2
	25		2	3
	26		2	3
	27		2	1
	28			
	29			
	30			
	31			

ABRIL				
Sprints	Día	Tareas	Horas estimadas	Horas reales
4º Sprint	1			
	2	Planificar mes	1	0,3

	3	Acabar las tablas de la base de datos	1	2
	4	Mejorar el diseño para el usuario	2	3
	5		2	2
	6			
	7	Modificar la documentación	1	1
	8	Mejorar el rendimiento de la aplicación	1	2
5ºSprint	9	Hacer los complementos del ratón	2	1
	10		2	1
	11		2	1
	12			
	13	Creación del primer minijuego	3	3
	14		2	3
	15	Corregir errores del diseño	3	3
	16		3	3
	17	Creación del 2º minijuego	2	2
	18		3	2
	19			
	20	Implementar los minijuegos a la aplicación	1,3	2
	21	Actualizar la documentación	1	1,3
	22	Investigación e implementación de algunos efectos de sonido	2	1
6ºSprint	23	Corregir el primer minijuego	3	1
	24	Implementar correctamente los minijuegos	2	0
	25	Acabar la pantalla de perfil de usuario	2	1,3
	26			
	27	Acabar la pantalla del armario y nevera	3	2
	28	Mejorar la metodología y calidad de las pantallas	2	3
	29	Renombrar los id correctamente	1	1,3
	30	Implementar algún método	2	1,3

MAYO				
Sprints	Día	Tareas	Horas estimadas	Horas reales
6ºSprint	1			
	2	Crear los métodos del SharedPreferences	1	1
	3			
	4	Crear las funcionalidades de las primeras pantallas	2	1
	5		2	1,3
	6	Modificar el diseño para mejorar el uso que pueda tener el usuario con él	1	2
7ºSprint	7	Crear funcionalidad de las pantallas restantes	2	2
	8		2	1,3
	9		2	1,3
	10			
	11	Ir generando la presentación y actualizar documentación	1	2
	12	Implementar los minijuegos con todos sus diseños y pantallas de inicio funcionales	2	2
	13		2	1,3
	14		2	1
	15	Modificación de la base de datos	1	1
	16	Crear consultas y métodos necesarios	2	2,3
	17			
	18	Corrección de errores	2	2

8ºSprint	19	Creación del servidor para las tablas de progreso	2	2
	20		2	1,3
	21		2	1
	22	Implementación y uso del servidor con los diferentes hilos en las pantallas necesarias	2	1
	23		2	1,3
	24		2	2,3
	25	Creación de las tiendas	4	2
	26		2	2
	27	Creación del armario y de la nevera	2	2
	28	Pruebas unitarias	2	1
	29	Último repaso completo	0,3	2
	30	<b>ENTREGA DEL TRABAJO DE FIN DE GRADO</b>		
	31			

HORAS TOTALES	
Horas estimadas	Horas reales
180	175,3

## Fases del proyecto

### Diseño

#### Nombre del proyecto

El nombre de mi proyecto, "Tailor-Mouse", fue seleccionado cuidadosamente considerando el significado de las palabras. "Tailor" se refiere a un sastre y "Mouse" a un ratón. La elección se basó en el objetivo del juego: la personalización de una mascota virtual. Opté por un ratón para evitar los clichés de las mascotas típicas y permití a los usuarios personalizar no solo el color y el nombre de la mascota, sino también sus accesorios y ropa. La combinación directa de "Ratón-Sastre" en español no resultaba atractiva, por lo que decidí traducirlo al inglés, dando lugar al nombre "Tailor-Mouse".

#### Logo

El logotipo de la aplicación se corresponde con la Img4, incorpora tanto el nombre como la mascota. La tipografía del nombre presenta un estilo "infantil" e informal, acorde con la audiencia a la que se dirige la aplicación, transmitiendo una sensación de juego y diversión. Los pequeños círculos en los vértices de las letras evocan la forma de un ojo de aguja, mientras que los trazos restantes sugieren la presencia de hilo.

En el lado izquierdo, se muestra la mascota de la aplicación, un ratón, seleccionado para reflejar el nombre del programa. La imagen del ratón fue diseñada para proyectar una apariencia amigable, y se ajustó en consecuencia. En línea con la elección de la tipografía, la mano izquierda del ratón sostiene una aguja, haciendo eco al nombre de la aplicación. Además, se le añadió un abrigo para hacer referencia a la personalización.

Notablemente, el ratón está envuelto en una circunferencia con aperturas. La elección de la circunferencia pretende hacer referencia al mundo del ratón. Las 4 separaciones hacen referencia a cada una de las necesidades que tiene el ratón.

La Img5 se refiere al ícono de la aplicación. Este ícono se muestra como entrada al descargar la aplicación. Solo presenta la mascota y la circunferencia, ya que son elementos bastante característicos de la misma.



Img 4 - Logo



Img 5 - Icono

## Personaje

La elección de este personaje se originó en el deseo de evitar lo convencional al pensar en mascotas. El diseño del ratón, tal como se presenta en las imágenes, fue cuidadosamente elaborado con el objetivo de incorporar un toque adorable y amistoso, con la intención de captar la atención y el afecto de los niños.

Una vez establecido el diseño base, se tomó la decisión de ofrecer aún más opciones a los usuarios. Por esta razón, se procedió a crear variantes del ratón en tres colores diferentes, brindando así una amplia gama de elecciones para personalizar y adaptar la mascota virtual según las preferencias individuales de cada usuario. Este enfoque no solo busca proporcionar variedad estética, sino también fomentar la conexión emocional de los usuarios con su ratón virtual, permitiéndoles elegir y cuidar de una mascota única y personalizada que refleje su estilo y preferencias.

El personaje fue creado mediante la herramienta Paint, diseñado y creado en base a un dibujo ya existente, pero modificado y retocado para que sea único.



Img 6 - Ratón blanco



Img 7 - Ratón gris

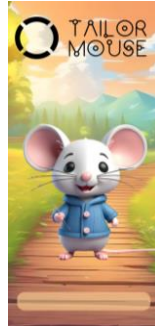


Img 8 - Ratón marrón

## Interfaces móviles – Esquema funcional

### Pantalla de carga

El diseño es muy simple, compuesto por uno de los logos de la aplicación, la mascota general de la misma y una barra de progreso. Una vez que la barra de progreso alcance el 100%, la aplicación avanzará a la siguiente pantalla.



Img 9 - Pantalla de carga

### Inicio de sesión

Después de la pantalla de carga, la aplicación nos llevará a la pantalla de inicio de sesión (Img10). En esta pantalla podemos realizar dos acciones diferentes:

- › Si tenemos una cuenta creada previamente, únicamente introducimos nuestras credenciales y nos llevará a nuestro perfil de usuario.
- › Si no tenemos una cuenta creada, pulsamos en el botón que indica la pantalla y nos dirigirá a la pantalla correspondiente para poder crearnos una cuenta.



Img 10 - Pantalla de inicio de sesión

### Registro

Esta pantalla (Img11) es la que nos permitirá crearnos una cuenta en caso de que no tengamos una previa. Accedemos a ella a través de la pantalla anterior de inicio de sesión (Img10). Debemos completar todos los campos para poder crearnos nuestra cuenta y una vez creada la cuenta nos dirigirá de nuevo a la pantalla de inicio de sesión para comprobar que las credenciales se han guardado correctamente y podremos acceder a nuestro perfil de usuario.

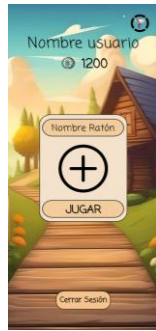


Img 11 - Pantalla de registro

### Perfil de usuario

Esta pantalla es la que nos muestra nuestro perfil, a la cual podremos llegar desde la pantalla de inicio de sesión (Img10), en ella podremos crear nuestra mascota pulsando el botón más central, en caso de que no la tengamos (Img12). También podremos ver nuestro nombre de usuario y las monedas que poseemos, además de cerrar sesión.

Si ya teníamos nuestra mascota creada, se mostrará el color elegido y su nombre correspondiente (Img13).



Img 12 - Pantalla perfil usuario 1



Img 13 - Pantalla perfil usuario 2

### Personalización

Esta pantalla es la que nos permitirá personalizar nuestra mascota por primera vez, si no tenemos nuestra mascota creada (Img14), desde la pantalla de nuestro perfil (Img12), pulsando el botón más, nos dirigirá a la siguiente pantalla donde podremos elegir el nombre y el color de nuestra mascota. Tenemos que elegir las dos cosas para que nos permita continuar

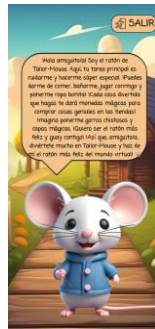


Img 14 -Pantalla de personalización

### Información

Esta pantalla sirve simplemente para mostrar el objetivo del juego. Se puede acceder a ella desde el perfil de usuario (Img13).





Img 15 - Pantalla de información

### Escenas del juego

La aplicación cuenta con 5 escenarios distintos: el salón (Img16), la cocina (Img17), el baño (Img18), el dormitorio (Img19) y la terraza (Img20), donde podremos interactuar con nuestro ratón. Cada escena ofrece una acción específica, accesible mediante el botón central en la parte inferior.

Para cambiar entre escenarios, en todas las escenas encontraremos dos botones en la parte inferior: uno a la derecha y otro a la izquierda. Al presionar uno de ellos, nos dirigirá a la siguiente escena en orden secuencial.



Img 16 - Pantalla del salón



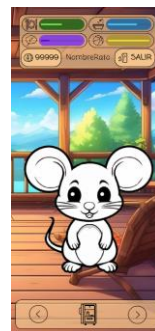
Img 17 - Pantalla de la cocina



Img 18 - Pantalla del baño



Img 19 - Pantalla del dormitorio



Img 20 - Pantalla de la terraza

### Acción-Armario

A esta pantalla (Img21) llegamos a través del botón de acción de la pantalla del salón (Img16). En esta pantalla podemos personalizar a nuestro ratón con los artículos que tengamos bajo nuestra propiedad. Podemos seleccionar un artículo del armario y dándole al botón inferior de confirmar se aplicarán los cambios al ratón y se le pondrá el artículo seleccionado. Podemos volver al salón pulsando el botón de la flecha situado en la esquina superior izquierda.

También podemos ir a la tienda pulsando el botón de la esquina superior derecha, donde podremos comprar más elementos.



Img 21- Pantalla del armario

### Acción-Tienda

Esta imagen corresponde a la pantalla de la tienda (Img22), accesible desde la pantalla del armario (Img21). En esta sección, podemos adquirir nuevos complementos.

Se pueden elegir complementos para la cabeza y el torso. Es posible seleccionar varios elementos de la misma categoría simultáneamente. El costo total de los elementos seleccionados se mostrará en un rectángulo en la parte inferior. Si el total aparece en rojo, significa que no tenemos suficiente dinero para realizar la compra; si aparece en verde, indica que la compra es posible.



Img 22 - Pantalla de la tienda de complementos

### Acción-Nevera

La siguiente imagen (Img23) corresponde a la pantalla de la nevera, accesible desde la pantalla de la cocina (Img17). En esta pantalla, podemos alimentar a nuestro ratón seleccionando cualquier elemento disponible. Para volver a la

pantalla de la cocina, simplemente hay que pulsar la flecha ubicada en la esquina superior izquierda.

Aquí podemos observar cómo el hambre del ratón aumenta a medida que le damos comida o bebida. Además, podemos acceder al supermercado pulsando el ícono del carrito en la esquina superior derecha.



Img 23 - Pantalla de la nevera

### Acción-Supermercado

La siguiente imagen (Img24) corresponde a la pantalla del supermercado, accesible desde la pantalla de la nevera (Img23). Para volver a la pantalla de la nevera, solo hay que pulsar el botón en la esquina superior izquierda.

En esta pantalla, podemos comprar elementos de dos categorías: comida y bebida. Seleccionamos los elementos que deseamos adquirir y se nos mostrará el total de la compra. Como en la pantalla de la tienda (Img22), el total aparecerá en rojo si no tenemos suficiente dinero para realizar la compra, y en verde si sí podemos comprarlo. Al completar la compra, los elementos se añadirán a la pantalla de la nevera (Img23).



Img 24 - Pantalla del supermercado

### Acción-Ducha

Esta imagen (Img25) se refiere a la pantalla de la ducha del ratón, a la cual se accede pulsando el botón de acción del baño (Img18).

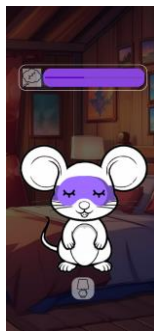
En esta pantalla, la interacción es puramente visual. Al pulsar el botón, ordenamos al ratón que se duche. Pulsando el botón de la ducha unas burbujas aparecerán por encima de la cortina además de accionar el efecto de sonido de la ducha. La barra de higiene aumentará gradualmente y, una vez que la higiene del ratón alcance el 100%, regresará automáticamente a la pantalla del baño (Img18).



Img 25 - Pantalla de la ducha

### *Acción-Luz del dormitorio*

Esta pantalla (Img26) se relaciona con la habitación del ratón con la luz apagada, y podemos acceder a esta acción pulsando el icono de la lámpara en la pantalla del dormitorio (Img19). Es muy similar a la pantalla de la ducha (Img25), con la diferencia de que, al pulsar el icono de la lámpara, podemos volver a encender la luz de la habitación. Si decidimos dejar la luz apagada, el ratón la encenderá automáticamente cuando su barra de sueño/cansancio alcance el 100%. Cuando la barra se encuentre al 70% se activará un efecto de sonido de un despertador.



Img 26 - Pantalla del dormitorio de noche

### *Acción-Sala de juegos*

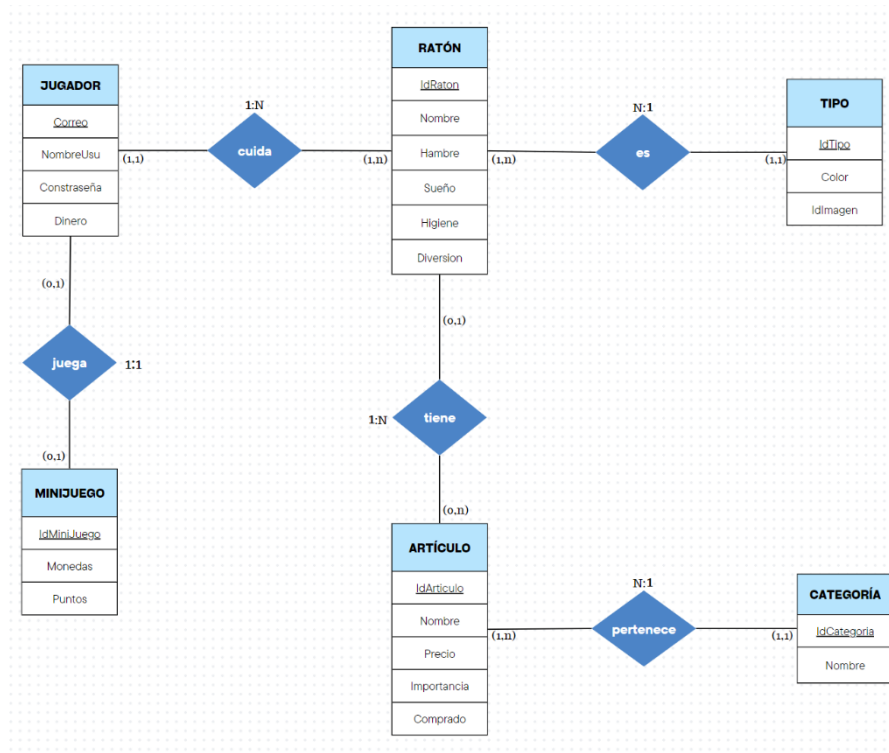
Esta pantalla (Img27) hace referencia a una sala de juegos, la cual se accede pulsando el botón con el icono de un mando de videojuegos en la pantalla de la terraza (Img20). En esta sala, encontraremos la barra de diversión de nuestro ratón, así como el saldo de nuestras monedas y los videojuegos disponibles.

Para aumentar el nivel de diversión de nuestro ratón, debemos jugar con él. Para ello, basta con pulsar sobre el juego deseado, lo que nos llevará a su correspondiente pantalla de juego. Una vez que regresemos después de haber jugado, la barra de diversión del ratón habrá aumentado. Además, el saldo de nuestras monedas variará dependiendo de si hemos ganado o perdido la partida del juego.



Img 27 - Pantalla de la sala de juegos

## Modelo entidad relación



Img 28 - Modelo entidad relación

## Explicación del modelo entidad relación

### Entidades:

**JUGADOR** (correo, nombreUsuario, contraseña, dinero, idRaton↑)

**RATON** (idRaton, idTipo↑, nombre, sueño, hambre, higiene, diversión)

**TIPO** (idTipo, color, idImagen)

**ARTICULO** (idArticulo, idCategoria↑, nombre, importancia, precio, comprado)

**CATEGORIA** (idCategoria, nombre)

**JUEGO**(idJuego, nombreUsuario↑, monedas, puntos)

### Relaciones:

**juega:** esta relación se define como la conexión entre las entidades "jugador" y "juego". Esta relación se caracteriza por el hecho de que un jugador puede participar

o no en un minijuego en un momento específico (0,1), y a su vez, un juego en particular puede ser jugado o no por un jugador en un momento dado (0,1). Por consiguiente, esta relación presenta una cardinalidad de 1:1.

**cuida:** esta relación se define como la conexión entre las entidades "jugador" y "ratón". Un jugador puede cuidar uno o varios ratones (1,n), mientras que un ratón específico solo puede ser cuidado por un jugador (1,1). Por tanto, la cardinalidad de esta relación es N:1.

**es:** esta relación se define como la conexión entre las entidades "ratón" y "tipo". Un ratón sólo puede ser de un tipo específico (1,1), pero puede haber varios ratones del mismo tipo (1,n). En consecuencia, la cardinalidad de esta relación es N:1.

**tiene:** esta relación forma una nueva tabla denominada "RATONARTICULO". Se define como la conexión entre las entidades "ratón" y "artículo". Un ratón puede poseer varios artículos o ninguno (0,n), mientras que un artículo específico solo puede pertenecer a un ratón o no estar asociado a ninguno (0,1). Por lo tanto, la cardinalidad de esta relación es 1:N.

**pertenece:** esta relación se define como la conexión entre las entidades "artículo" y "categoría". Un artículo debe pertenecer específicamente a una categoría (1,1), mientras que una categoría puede contener varios artículos (1,n). Por consiguiente, la cardinalidad de esta relación es N:1.

## Tecnologías utilizadas

### Figma – Diseño de las interfaces (iniciales)

En mi proyecto, he elegido utilizar Figma, una herramienta basada en la nube para el diseño de interfaces de usuario (UI) y experiencia de usuario (UX), que ha ganado popularidad recientemente. Opté por Figma debido a su accesibilidad desde cualquier lugar, su capacidad de prototipado integrado, su sistema de versionado automático y la posibilidad de crear componentes reutilizables.

Aunque evalué alternativas como Adobe XD y InVision Studio, me decidí por Figma debido a diferencias clave entre ellas. Mientras que Adobe XD está integrado en el ecosistema de Adobe, con una estrecha relación con productos como Photoshop e Illustrator, y InVision Studio se centra en el diseño y prototipado, Figma destaca por su combinación de diseño en tiempo real y colaboración, lo que la hace más versátil y adecuada para mis necesidades específicas.

### Android Studio – Creación de la aplicación

Android Studio, la IDE oficial para crear aplicaciones Android, es mi elección debido a su respaldo por Google, su constante actualización y su sólido respaldo comunitario y documentación oficial. Está específicamente diseñada para trabajar con el ecosistema Android, simplificando el proceso de desarrollo, prueba y despliegue de aplicaciones. Además, ofrece un intuitivo editor de diseño, herramientas avanzadas de depuración y perfilado, emuladores de dispositivos Android y soporte para dispositivos físicos. Su extensa comunidad de desarrolladores proporciona una rica fuente de recursos, tutoriales y soluciones para desafíos comunes.

## Room – Gestor de base de datos

He optado por utilizar Room de Android Studio como la capa de abstracción de mi aplicación, ya que ofrece ventajas como la verificación del tiempo de compilación de las consultas SQL, anotaciones convenientes que reducen el código estándar repetitivo y propenso a errores, y rutas de migración de bases de datos optimizadas.

Room, es una biblioteca de persistencias que forma parte de Android Jetpack. Es una capa de abstracción que se ubica sobre una base de datos SQLite.

Para implementar Room en mi aplicación, he tenido que agregar algunas dependencias al archivo "build.gradle" correspondiente. Estas dependencias se pueden visualizar en la siguiente imagen.

```
dependencies {
    val room_version = "2.6.1"

    implementation("androidx.room:room-runtime:$room_version")
    annotationProcessor("androidx.room:room-compiler:$room_version")

    // To use Kotlin annotation processing tool (kapt)
    kapt("androidx.room:room-compiler:$room_version")
    // To use Kotlin Symbol Processing (KSP)
    ksp("androidx.room:room-compiler:$room_version")

    // optional - Kotlin Extensions and Coroutines support for Room
    implementation("androidx.room:room-ktx:$room_version")

    // optional - RxJava2 support for Room
    implementation("androidx.room:room-rxjava2:$room_version")

    // optional - RxJava3 support for Room
    implementation("androidx.room:room-rxjava3:$room_version")

    // optional - Guava support for Room, including Optional and ListenableFuture
    implementation("androidx.room:room-guava:$room_version")

    // optional - Test helpers
    testImplementation("androidx.room:room-testing:$room_version")

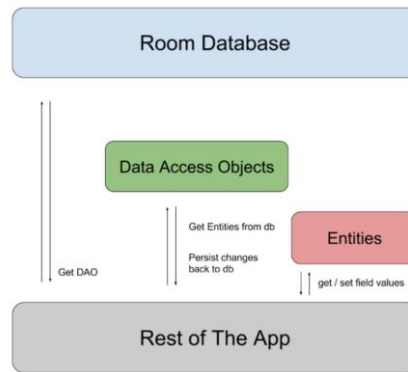
    // optional - Paging 3 Integration
    implementation("androidx.room:room-paging:$room_version")
}
```

*Img 29 - Dependencias de Room*

Room consta de tres componentes principales:

- › La clase de la base de datos que actúa como punto de acceso principal a los datos persistentes.
- › Las entidades de datos que representan las tablas de la base de datos.
- › Los DAOs que proporcionan métodos para interactuar con la base de datos.
- › La clase de la base de datos proporciona instancias de los DAOs asociados, que la aplicación puede usar para consultar, actualizar, insertar y eliminar datos. Las entidades de datos se utilizan para actualizar filas en las tablas o insertar nuevas filas.





Img 30 - Esquema de Room

## Implementación y configuración

### Servidor

He tenido que desarrollar un pequeño servidor para resolver un error que encontré al manejar hilos para las barras de progreso de las necesidades del ratón mientras jugaba. Este servidor, representado por la clase `MainService` en Android, supervisa la ejecución del hilo definido en `MainServiceThread`. Proporciona métodos para iniciar y detener el servicio, así como para verificar su estado. Al iniciarse, crea una instancia de `MainServiceThread` y lo activa. Tanto el inicio como la detención del servicio pueden ser controlados interna y externamente mediante métodos estáticos. Además, `MainService` ofrece métodos estáticos para verificar si el servicio y su hilo asociado están en ejecución.

Además, he creado la clase `MainServiceThread` para gestionar un hilo de ejecución en segundo plano. Esta clase se encarga de tareas periódicas, como la actualización del estado de un objeto `Raton`. Específicamente, disminuye los valores de hambre, higiene, diversión y sueño del objeto `Raton` a intervalos regulares. El hilo se ejecuta en un bucle continuo hasta que se detiene explícitamente al cambiar la variable `loophthread` a `false`. `MainServiceThread` también se comunica con un repositorio para actualizar los datos del `Raton`.

### Código de la clase `MainService`

En la siguiente imagen (Img31), se muestran las etiquetas que utilicé para enviar acciones al servicio. También se visualiza la instancia del hilo principal del servicio y, por último, un booleano para verificar si el servicio está en ejecución o no.

```

public class MainService extends Service {
    3 usages
    private static final String SERVICE_TAG = "action";
    2 usages
    private static final String SERVICE_START = "start";
    2 usages
    private static final String SERVICE_STOP = "stop";

    2 usages
    private MainServiceThread thread = null;

    4 usages
    private boolean serviceStatus = false;
  
```

Img 31 - Clase `MainService` 1



En la imagen adjunta (Img32), se pueden apreciar varios métodos. El método onCreate es invocado cuando el servicio se crea por primera vez, inicializando así el hilo. Asimismo, se encuentra el método onDestroy, el cual se ejecuta cuando el servicio está a punto de ser destruido, deteniendo su funcionamiento. Además, se destaca el método onStartCommand, que se activa cada vez que se inicia el servicio mediante startService. Aquí se determina la acción que debe llevar a cabo el servicio según la acción enviada a través del intent.

```
@Override
public void onCreate() {
    super.onCreate();
    thread = new MainServiceThread(getApplication());
}

@Override
public void onDestroy() {
    stopService();
    super.onDestroy();
}

4 usages
@Override
public int onStartCommand(Intent intent, int flags, int startId) {
    if(intent != null) {
        String action = intent.getStringExtra(SERVICE_TAG);

        if(action != null) {
            if(action.equals(SERVICE_START)) startService();
            if(action.equals(SERVICE_STOP)) stopService();
        }
    }

    return START_STICKY;
}
```

Img 32 - Clase MainService 2

La siguiente imagen(Img33), nos muestra diferentes métodos. El método startService, el cual inicia el servicio si no está en ejecución, lo que implica iniciar el hilo y establecer la variable loopthread en true. Otro método que podemos observar es stopService, el cual detiene el hilo si está en ejecución. Realiza lo contrario al método anterior, detiene el hilo, establece la variable loopthread en false y detiene el servicio.

```

@Override
public IBinder onBind(Intent intent) { return null; }

1 usage
private void startService() {
    //En caso de tener el servicio ya corriendo no es necesario volver a lanzarlo
    if (serviceStatus) { return; }
    serviceStatus = true;

    //Lanzamos el hilo de ejecución principal
    thread.start();

    MainServiceThread.loopthread = true;
}

2 usages
private void stopService() {
    if (!serviceStatus) { return; }
    serviceStatus = false;

    //Modificamos la variable del thread
    MainServiceThread.loopthread = false;

    //Detenemos el thread
    stopSelf();
}

```

Img 33 - Clase MainService 3

Y, por último, en la imagen (Img34), podemos ver los métodos estáticos para manipular el servicio desde fuera. `isMyServiceRunning` verifica si el servicio está en ejecución. `startService` inicia el servicio desde una actividad externa. `stopService` detiene el servicio desde una actividad externa. `isMyServiceThreadRunning` verifica si el hilo del servicio está en ejecución.

```

public static boolean isMyServiceRunning(Context context) {
    ActivityManager manager = (ActivityManager) context.getSystemService(Context.ACTIVITY_SERVICE);
    Class<?> serviceClass = MainServiceThread.class;

    for (ActivityManager.RunningServiceInfo service : manager.getRunningServices(Integer.MAX_VALUE)) {
        if (serviceClass.getName().equals(service.service.getClassName())) {
            return true;
        }
    }
    return false;
}

1 usage
public static void startService(Context context) {
    if(!isMyServiceRunning(context)) {
        //Iniciar el servicio
        Intent intentService = new Intent(context, MainService.class);
        intentService.putExtra(SERVICE_TAG, SERVICE_START);
        context.startService(intentService);
    }
}

1 usage
public static void stopService(Context context) {
    if(isMyServiceRunning(context)) {
        //Detenemos el servicio
        Intent intentService = new Intent(context, MainService.class);
        intentService.putExtra(SERVICE_TAG, SERVICE_STOP);
        context.stopService(intentService);
    }
}

no usages
public static boolean isMyServiceThreadRunning() { return MainServiceThread.loopthread; }

```

Img 34 - Clase MainService 4

### Código de la clase MainServiceThread

En la siguiente imagen (Img35) podemos ver, las variables de la clase y el constructor de esta.

```
public class MainServiceThread extends Thread{
    4 usages
    public static boolean loopthread = true;
    3 usages
    private Repository repository;

    1 usage
    public MainServiceThread(@NonNull Application application) {
        repository = new Repository(application);
    }
}
```

Img 35 - Clase MainServiceThread 1

En esta imagen (Img36) encontramos el método run, el cual está diseñado para ejecutar en un hilo separado y periódicamente actualiza las necesidades del ratón mientras el hilo esté en ejecución. Esto se hace a través de un bucle continuo que se detiene cuando la variable loopthread se establece en false.

```
@Override
public void run() {
    super.run();

    Log.v( tag: "tailorMouse", msg: "thread lanzado");

    do {
        Raton ratonActivo = repository.activeRaton();

        if(ratonActivo.getHambre() > 0) {
            ratonActivo.setHambre(ratonActivo.getHambre() -1);
        }
        if(ratonActivo.getHigiene() > 0) {
            ratonActivo.setHigiene(ratonActivo.getHigiene() -1);
        }
        if(ratonActivo.getDiversion() > 0) {
            ratonActivo.setDiversion(ratonActivo.getDiversion() -1);
        }
        if(ratonActivo.getSuenio() > 0) {
            ratonActivo.setSuenio(ratonActivo.getSuenio() -1);
        }

        repository.updateRaton(ratonActivo);

        //Paramos el servicio el tiempo que queramos
        try {
            Thread.sleep( millis: 500);
        } catch (InterruptedException e) {
            throw new RuntimeException(e);
        }
    } while (loopthread);
}
```

Img 36 - Clase MainServiceThread 2

## Implementación de los minijuegos

### Tres en raya

Este minijuego es un desafío estratégico para el jugador. Se desarrolla en una cuadrícula de 3x3 donde el jugador se enfrenta a la máquina. Ambos colocan sus fichas por turnos: el jugador usa un ratón y la máquina una imagen de gato.

Si el jugador gana, la línea de victoria se resalta en verde, indicando su triunfo y las monedas ganadas. En caso de perder, la línea de la máquina se destaca en rojo, señalando la derrota y las monedas perdidas. Si hay un empate, no se resalta ninguna línea y se notifica el empate.

El minijuego está diseñado para iniciar automáticamente una nueva partida cinco segundos después de que se complete una. Para salir del juego en cualquier momento, basta con pulsar el botón de salida en la esquina superior izquierda.



Img 37 - Pantalla inicio juego 3 en raya



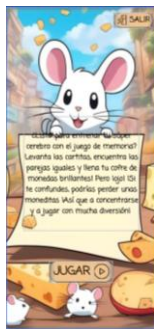
Img 38 - Pantalla juego 3 en raya

## Memory

Este minijuego desafía la memoria del jugador en una cuadrícula de 4x4, con un total de 16 cartas que forman 8 parejas. En cada turno, el jugador puede levantar dos cartas distintas y descubrir si coinciden. Hay dos resultados posibles:

- › Si las cartas coinciden, se mantendrán descubiertas y el jugador ganará 5 monedas.
- › Si las cartas son diferentes, se mostrarán brevemente antes de volver a su estado inicial, y al jugador se le restarán 2 monedas. Luego, podrá intentar levantar dos cartas nuevamente.

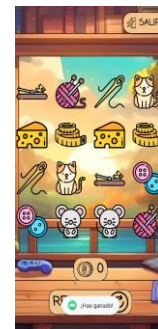
Este proceso continuará hasta que el jugador encuentre todas las parejas. Además, el jugador tiene la opción de reiniciar la partida en cualquier momento.



Img 39 - Pantalla inicio juego memory



Img 40 - Pantalla juego memory 1



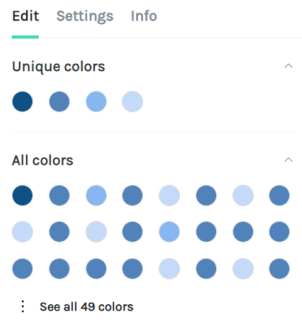
Img 41 - Pantalla juego memory 2

## Animaciones

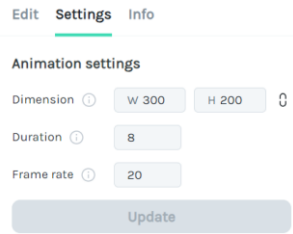
Para las animaciones, he optado por una plataforma web que ofrece una variedad de animaciones gratuitas. Específicamente, he empleado Lottifiles, que permite descargar animaciones preexistentes, crear nuevas desde cero o modificar las existentes.

En mi caso, opté por una animación preexistente y la modifiqué ligeramente. Esto implicó ingresar al editor de texto de la plataforma y ajustar las propiedades

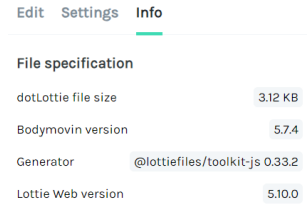
deseadas, como colores, dimensiones, duración y fotogramas por segundo. Se puede ver un ejemplo en la imagen (Img42, Img43, Img44).



Img 42 - Editor lottifiles 1

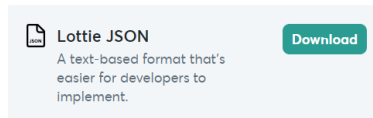


Img 43 - Editor lottifiles 2



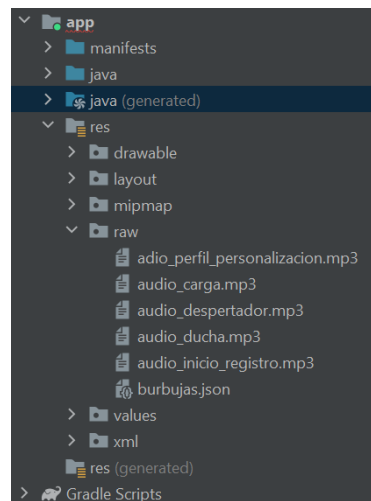
Img 44 - Editor lottifiles 3

A la hora de realizar la descarga para poder implementarlo en Android Studio es necesario descargarlo en formato JSON como se ve en la siguiente imagen (Img45).



Img 45 - Descarga animación

Cuando importamos la animación en Android Studio para poder utilizarla, necesitamos crear un nuevo directorio dentro de la carpeta “res”, al que llamaremos “raw”. Luego, dentro de este directorio, colocaremos el archivo descargado en formato JSON, como se muestra en la siguiente imagen (Img46).



Img 46 - Directorio RAW

Para hacer funcional la animación, es necesario importar las dependencias necesarias dentro del archivo de configuración de la clase “*build.gradle.kts*”. En este caso, la dependencia es “*implementation(“com.airbnb.android:lottie:6.4.0”)*”. En la última sección de la dependencia se especifica la versión que se desea

implementar, por lo que es posible modificarla según sea necesario. Esto se ilustra en la siguiente imagen (Img47).

```
implementation("androidx.appcompat:appcompat:1.6.1")
implementation("com.google.android.material:material:1.10.0")
implementation("androidx.constraintlayout:constraintlayout:2.1.4")
testImplementation("junit:junit:4.13.2")
implementation("com.airbnb.android:lottie:6.4.0")
androidTestImplementation("androidx.test.ext:junit:1.1.5")
androidTestImplementation("androidx.test.espresso:espresso-core:3.5.1")
```

Img 47 - Dependencias de lottifiles

Ahora, para usarlo, necesitaremos dirigirnos al XML del diseño de la actividad donde deseemos emplearlo. En mi caso, lo he utilizado en la pantalla de la ducha (Img25). Allí, debemos definirlo como un componente de Lottie y configurarlo según nuestras preferencias, como se muestra en la imagen (Img48). Luego, en la programación de dicha animación, en su clase respectiva, declaramos el componente como un LottieAnimationView, tal como se observa en la imagen (Img49). Lo referenciamos como cualquier otro componente, como se muestra en la imagen (Img50). Finalmente, hacemos referencia al momento en que deseamos que la animación comience. En mi caso, deseo que se inicie cuando el usuario presione un botón específico. Utilizamos el método ".start", que viene por defecto en las animaciones de esta clase, y así tendremos la animación en funcionamiento. Esto último se puede apreciar en la imagen (Img51).

```
<!--Animación burbujas-->
<com.airbnb.lottie.LottieAnimationView
    android:id="@+id/img_burbujas"
    android:layout_width="300dp"
    android:layout_height="350dp"
    android:layout_marginBottom="400dp"
    app:layout_constraintBottom_toBottomOf="@+id/img_raton"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/linearLayout"
    app:layout_constraintVertical_bias="0.261" />
```

Img 48 - XML componente lottie

```
public class Ducha extends AppCompatActivity {

    1 usage
    Router router;
    4 usages
    SharedPreferences sharedPreferences;
    2 usages
    private LottieAnimationView img_burbujas;
    3 usages
```

Img 49 - Declaración del componente lottie

```
img_burbujas = findViewById(R.id.img_burbujas);
```

Img 50 - Referencia a la imagen animada

```

btn_alcachofaDucha.setOnClickListener(view -> {
    btn_alcachofaDucha.setVisibility(View.INVISIBLE);
    ducha = burbujas(img_burbujas, R.raw.burbujas, ducha);
    aguaDucha.start();

    if (!progressBarStarted) {
        startProgressBarUpdates(higieneDucha, prgHigiene, updateDelay: 50);
        progressBarStarted = true;
        guardarProgreso(100);
    }
});

```

Img 51 - Uso de la animación

## Sonidos

Para los sonidos, tanto la música como los efectos de sonido me he decantado por la web de Pixabay, es un banco de imágenes, ilustraciones, vectores y vídeos de alta calidad, libres de derechos de autor.

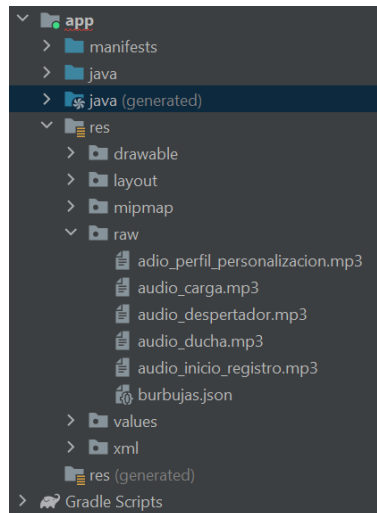
En mi primer lugar debemos buscar el efecto de sonido o la música que queramos, por ejemplo, en mi caso era necesario el sonido de un despertador, por ello busco despertador en el buscador y me salen diferentes sonidos, como podemos ver en la imagen (Img52).

Después de seleccionar el que más se ajuste a nuestras necesidades debemos descargarlo, simplemente pulsando en el botón de la derecha de cada sonido que dice “Descargar” se nos descarga “en el formato correcto para poder usarlo.



Img 52 - Búsqueda pixabay

Una vez descargado, como ya tenemos el directorio “raw” que necesitábamos para las animaciones, simplemente agregamos los audios en formato “mp3” en ese directorio. Como podemos ver en la siguiente imagen (Img53).



Img 53 - Directorio RAW sonidos

Para poder hacer uso de los efectos de sonido, primero lo declaramos de tipo MediaPlayer en la clase correspondiente como podemos ver en la imagen (Img54). Después lo asignamos al audio que queramos como se muestra en la imagen (Img55). Seguido de esto, implementamos métodos como el onPause (Img56), onResume y onDestroy que nos serán muy útiles a la hora de usar los audios. Después ya, indicamos con el método .start() cuando queramos que se reproduzca el sonido, como podemos ver en la siguiente imagen (Img57).

```
private MediaPlayer despertador;
```

Img 54 - Declaración MediaPlayer

```
despertador = MediaPlayer.create(context: this, R.raw.audio_despertador);
```

Img 55 - Asignación del audio

```
@Override
protected void onPause() {
    super.onPause();
    if (despertador.isPlaying()) {
        despertador.pause();
        despertador.release();
    }
    guardarProgreso(suenio.getProgress());
}
```

Img 56 - Métodos necesarios



```
private void startProgressBarUpdates(final ProgressBar progressBar, final int progress, final int updateDelay) {
    // Crear un nuevo hilo para actualizar la barra de progreso
    new Thread(new Runnable() {
        @Override
        public void run() {
            Handler handler = new Handler(getMainLooper());

            int currentProgress = progress;

            while (currentProgress < progressBar.getMax()) {
                currentProgress++;
                final int finalProgress = currentProgress;
                handler.post(new Runnable() {
                    @Override
                    public void run() {
                        progressBar.setProgress(finalProgress);
                    }
                });
                try {
                    Thread.sleep(updateDelay);
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }

                if (finalProgress == progressBar.getMax() * 0.7) {
                    despertador.start();
                }
            }
            guardarProgreso(progressBar.getMax());
            finish();
        }
    }).start();
}
```

Img 57 - Uso del audio

## Pruebas unitarias

ACCIONES PROGRAMADAS/AUTOMÁTICAS				
Descripción	Resultado esperado	Resultado obtenido	Tiempo obtenido por un usuario experto	Tiempo obtenido por un usuario inexperto
Iniciar la aplicación	Pulsar el icono de la aplicación, esperar 1 segundo (aprox.) a que aparezca la pantalla de inicio de carga	Resultado esperado	3 seg	3 seg
<b>Pantalla inicio carga</b>				
Completar la barra de progreso de la pantalla de inicio	Esperar a que la barra de carga llegue al 100% y esperar unos 2-3 segundos a que nos lleve al inicio de sesión	Resultado esperado	5 seg	5 seg
Ir al inicio de sesión una vez completada la pantalla de carga	Una vez completada la pantalla de carga, si no hay ninguna cuenta creada o ninguna cuenta con el inicio de sesión activo, nos deberá llevar a la pantalla de inicio de sesión	Resultado esperado	< 1 seg	< 1 seg
Ir al perfil del usuario una vez completada la	Una vez completada la pantalla de carga, si hay alguna cuenta creada y esta está activa, nos deberá llevar a la	Resultado esperado	< 1 seg	< 1 seg

pantalla de carga	pantalla de inicio de sesión			
<b>Pantallas: perfil de usuario, salón, cocina, baño, dormitorio, terraza</b>				
Actualización nombre de usuario	Debe actualizarse el campo correcto con el nombre del usuario activo	Resultado esperado	-	-
Actualización nombre del ratón	Debe actualizarse el campo correcto con el nombre del ratón activo	Resultado esperado	-	-
Actualización monedas del usuario	Debe actualizarse el campo correcto con las monedas del usuario activo	Resultado esperado	-	-
Actualización niveles de necesidades	Deben actualizarse correctamente las barras de progreso del ratón activo	Resultado esperado	-	-
Actualización artículos	Si el usuario decide ponerle algún artículo al ratón se deben actualizar en todas las diferentes pantallas	Resultado esperado	-	-
<b>Pantallas: tienda, supermercado, sala de juegos</b>				
Actualización monedas del usuario	Debe actualizarse el campo correcto con las monedas del usuario activo	Resultado esperado	-	-
<b>Pantalla armario</b>				
Actualizar armario	Debe estar en constante actualización por si se compran nuevos artículos que aparezcan	Resultado esperado	-	-
<b>Pantalla tienda</b>				
Actualizar tienda	Debe estar en constante actualización por si se compran nuevos artículos que desaparezcan	Resultado esperado	-	-
Actualizar el precio total de la compra	Se debe actualizar el precio total de la compra según los elementos seleccionados en la pantalla	Resultado esperado	-	-

<b>Pantalla nevera</b>				
Actualizar nevera	Debe estar en constante actualización por si se compran nuevos artículos que aparezcan	Resultado esperado	-	-
Actualización nivel del hambre	Deben actualizarse correctamente la barra de progreso del hambre ratón activo	Resultado esperado	-	-
<b>Pantalla supermercado</b>				
Actualizar el precio total de la compra	Se debe actualizar el precio total de la compra según los elementos seleccionados en la pantalla	Resultado esperado	-	-
<b>Pantalla ducha</b>				
Ir directamente a la pantalla del baño	Al finalizar la barra de progreso, se establezca su valor máximo, nos debe dirigir automáticamente a la pantalla del baño	Resultado esperado	< 1 seg	< 1 seg
Actualización nivel de la higiene	Deben actualizarse correctamente la barra de progreso de la higiene ratón activo	Resultado esperado	-	-
<b>Pantalla dormitorio noche</b>				
Articulos ratón	Independientemente de si el ratón lleva artículos colocados antes, en esta pantalla aparecerá con el pijama predefinido	Resultado esperado	-	-
Ir directamente a la pantalla del dormitorio	Al finalizar la barra de progreso, se establezca su valor máximo, nos debe dirigir automáticamente a la pantalla del dormitorio	Resultado esperado	< 1 seg	< 1 seg
Actualización nivel de la energía	Deben actualizarse correctamente la barra de progreso de la energía del ratón activo	Resultado esperado	-	-
<b>Pantalla sala de juegos</b>				

Actualización nivel de la diversión	Deben actualizarse correctamente la barra de progreso de la diversión del ratón activo	Resultado esperado	-	-
<b>Pantalla juego 3 en raya</b>				
Colocar la ficha del rival una vez puesto nuestra ficha	Una vez hemos colocado una ficha del usuario, debe colocar una ficha la IA en algún espacio libre. Así sucesivamente hasta que no queden huecos libres	Resultado esperado	< 1 seg	< 1 seg
Reiniciar la partida si hemos finalizado	Una vez estén todos los espacios completados, y nos hayan indicado si hemos perdido o ganado, después de unos segundos, debe reiniciarse la partida automáticamente	Resultado esperado	5 seg	5 seg
Actualizar monedas	Una vez se salga del juego, se deben actualizar las monedas del usuario	Resultado esperado	-	-
<b>Pantalla juego memoria</b>				
Voltear todas las tarjetas medio segundo después de empezar la partida	Si el usuario inicia la partida a este juego, las cartas deben verse durante medio segundo y después deben ponerse boca abajo	Resultado esperado	< 1 seg	< 1 seg
Actualizar monedas	Una vez se acabe una partida se deben actualizar las monedas del usuario	Resultado esperado	-	-

ACCIONES DE PULSACIÓN				
Descripción	Resultado esperado	Resultado obtenido	Tiempo obtenido por un usuario experto	Tiempo obtenido por un usuario inexperto
<b>Pantalla inicio de sesión</b>				
Pulsar sobre el texto editable	Si pulsamos sobre este campo nos deberá permitir escribir en él	Resultado esperado	< 1 seg	< 1 seg

del nombre de usuario				
Pulsar sobre el texto editable de la contraseña	Si pulsamos sobre este campo nos deberá permitir escribir en él	Resultado esperado	< 1 seg	< 1 seg
Pulsar sobre el botón entrar	Pulsando sobre este botón, si nuestras credenciales son correctas nos debe dirigir a la pantalla de nuestro perfil	Resultado esperado	< 1 seg	< 1 seg
Pulsar el elemento que cambia de la pantalla de inicio de sesión a la de registro	Pulsando este elemento nos deberá dirigir a la pantalla de registro	Resultado esperado	< 1 seg	< 1 seg
<b>Pantalla registro</b>				
Pulsar sobre el texto editable del nombre de usuario	Si pulsamos sobre este campo nos deberá permitir escribir en él	Resultado esperado	< 1 seg	< 1 seg
Pulsar sobre el texto editable del correo electrónico	Si pulsamos sobre este campo nos deberá permitir escribir en él	Resultado esperado	< 1 seg	< 1 seg
Pulsar sobre el texto editable de la contraseña	Si pulsamos sobre este campo nos deberá permitir escribir en él	Resultado esperado	< 1 seg	< 1 seg
Pulsar sobre el texto editable de la confirmación de la contraseña	Si pulsamos sobre este campo nos deberá permitir escribir en él	Resultado esperado	< 1 seg	< 1 seg
Pulsar sobre el botón confirmar	Si hace las comprobaciones necesarias y si son correctas, nos deberá guardar los datos en la base de datos y dirigirnos a la pantalla de inicio de sesión	Resultado esperado	< 1 seg	< 1 seg
Pulsar el elemento que cambia de la pantalla de registro a la de inicio de sesión	Pulsando este elemento nos deberá dirigir a la pantalla de inicio de sesión	Resultado esperado	< 1 seg	< 1 seg

Pantalla perfil de usuario				
Pulsar sobre el botón de información	Pulsando sobre este botón nos debe dirigir a la pantalla de información	Resultado esperado	< 1 seg	2 seg
Pulsar sobre la imagen del más	Pulsando sobre este botón nos debe dirigir a la pantalla de personalización	Resultado esperado	< 1 seg	1 – 2 seg
Pulsar sobre el botón de jugar	Este botón solo debe funcionar cuando el ratón ya este creado, si todo esta correcto y le damos nos debe dirigir a la pantalla del salón con nuestro ratón	Resultado esperado	< 1 seg	< 1 seg
Pulsar sobre el botón de cerrar sesión	Pulsando este botón la base de datos se actualiza poniéndonos como inactivos y nos dirige a la pantalla de inicio de sesión	Resultado esperado	< 1 seg	< 1 seg
Pantalla información				
Pulsar sobre el botón salir	Pulsando sobre este botón nos debe dirigir a la pantalla de nuestro perfil	Resultado esperado	< 1 seg	< 1 seg
Pantalla personalización				
Pulsar sobre el texto editable del nombre del ratón	Si pulsamos sobre este campo nos deberá permitir escribir en él	Resultado esperado	< 1 seg	< 1 seg
Pulsar sobre cualquiera de los botones de color	Pulsando sobre uno de ellos nos debe de cambiar la imagen del ratón por defecto y pintarlo del color concreto seleccionado	Resultado esperado	< 1 seg	< 1 seg
Pulsar sobre el botón continuar	Únicamente nos dirigirá a la pantalla de perfil de usuario cuando hayamos asignado un nombre y un color al ratón	Resultado esperado	< 1 seg	< 1 seg
Pantalla salón				

Pulsar sobre el botón salir	Pulsando sobre este botón nos debe dirigir a nuestro perfil	Resultado esperado	< 1 seg	2 seg
Pulsar sobre el botón de acción	Pulsando sobre este botón nos debe dirigir a la pantalla del armario	Resultado esperado	< 1 seg	1 – 2 seg
Pulsar sobre el botón de ir a la derecha	Pulsando sobre este botón nos debe dirigir a la pantalla de la cocina	Resultado esperado	< 1 seg	1 – 2 seg
Pulsar sobre el botón de ir a la izquierda	Pulsando sobre este botón nos debe dirigir a la pantalla de la terraza	Resultado esperado	< 1 seg	1 – 2 seg
<b>Pantalla armario</b>				
Pulsar sobre el botón de la tienda	Pulsando sobre este botón nos debe dirigir a la pantalla de la tienda	Resultado esperado	< 1 seg	2 seg
Pulsar sobre el botón de ir atrás	Pulsando sobre este botón nos debe dirigir a la pantalla del salón	Resultado esperado	< 1 seg	< 1 seg
Pulsar sobre el botón de confirmar	Pulsando sobre este botón nos debe dirigir a la pantalla del salón con el artículo seleccionado	Resultado esperado	< 1 seg	5 seg
Pulsar sobre un artículo	Pulsando sobre un artículo, este debe de colorear su borde en un tono azulado	Resultado esperado	< 1 seg	2 seg
<b>Pantalla tienda – tienda de ropa</b>				
Pulsar sobre el botón de ir atrás	Pulsando sobre este botón nos debe dirigir a la pantalla del armario	Resultado esperado	< 1 seg	2 seg
Pulsar sobre el radioButton de la categoría torso	Deben cambiar los artículos, y que aparezcan los de dicha categoría	Resultado esperado	< 1 seg	1 seg
Pulsar sobre el radioButton de la categoría cabeza	Deben cambiar los artículos, y que aparezcan los de dicha categoría	Resultado esperado	< 1 seg	1 seg
Pulsar sobre uno o varios artículos	Pulsando sobre uno o varios artículos, se deben colorear su borde en un tono azulado, además de actualizar el precio final	Resultado esperado	< 1 seg	3 seg
Pulsar sobre el botón de confirmar compra	Pulsando sobre este botón, con las validaciones correctas nos debe de dirigir a la	Resultado esperado	< 1 seg	3 seg

	pantalla del armario actualizando esta última con los artículos adquiridos. Además de actualizar las monedas del usuario			
Pulsar sobre el botón de cancelar compra	Pulsando sobre este botón, nos deselecciona todos los artículos que teníamos seleccionados	Resultado esperado	< 1 seg	3 seg
<b>Pantalla cocina</b>				
Pulsar sobre el botón salir	Pulsando sobre este botón nos debe dirigir a nuestro perfil	Resultado esperado	< 1 seg	2 seg
Pulsar sobre el botón de acción	Pulsando sobre este botón nos debe dirigir a la pantalla de la nevera	Resultado esperado	< 1 seg	1 – 2 seg
Pulsar sobre el botón de ir a la derecha	Pulsando sobre este botón nos debe dirigir a la pantalla del baño	Resultado esperado	< 1 seg	1 – 2 seg
Pulsar sobre el botón de ir a la izquierda	Pulsando sobre este botón nos debe dirigir a la pantalla del salón	Resultado esperado	< 1 seg	1 – 2 seg
<b>Pantalla nevera</b>				
Pulsar sobre el botón del super	Pulsando sobre este botón nos debe dirigir a la pantalla del super	Resultado esperado	< 1 seg	2 seg
Pulsar sobre el botón de ir atrás	Pulsando sobre este botón nos debe dirigir a la pantalla de la cocina	Resultado esperado	< 1 seg	2 seg
Pulsar sobre un artículo	Pulsando sobre un artículo, este debe de actualizar la barra de progreso además de desaparecer de la nevera.	Resultado esperado	< 1 seg	2 seg
<b>Pantalla tienda – supermercado</b>				
Pulsar sobre el botón de ir atrás	Pulsando sobre este botón nos debe dirigir a la pantalla de la nevera	Resultado esperado	< 1 seg	2 seg
Pulsar sobre el radioButton de la categoría bebida	Deben cambiar los artículos, y que aparezcan los de dicha categoría	Resultado esperado	< 1 seg	1 seg
Pulsar sobre el radioButton de	Deben cambiar los artículos, y que	Resultado esperado	< 1 seg	1 seg



la categoría comida	aparezcan los de dicha categoría			
Pulsar sobre uno o varios artículos	Pulsando sobre uno o varios artículos, se deben colorear su borde en un tono azulado, además de actualizar el precio final	Resultado esperado	< 1 seg	2 seg
Pulsar sobre el botón de confirmar compra	Pulsando sobre este botón, con las validaciones correctas nos debe de dirigir a la pantalla de la nevera actualizando esta última con los artículos adquiridos. Además de actualizar las monedas del usuario	Resultado esperado	< 1 seg	2 seg
Pulsar sobre el botón de cancelar compra	Pulsando sobre este botón, nos deselecciona todos los artículos que teníamos seleccionados	Resultado esperado	< 1 seg	2 seg
<b>Pantalla baño</b>				
Pulsar sobre el botón salir	Pulsando sobre este botón nos debe dirigir a nuestro perfil	Resultado esperado	< 1 seg	2 seg
Pulsar sobre el botón de acción	Pulsando sobre este botón nos debe dirigir a la pantalla de la ducha	Resultado esperado	< 1 seg	1 – 2 seg
Pulsar sobre el botón de ir a la derecha	Pulsando sobre este botón nos debe dirigir a la pantalla del dormitorio	Resultado esperado	< 1 seg	1 – 2 seg
Pulsar sobre el botón de ir a la izquierda	Pulsando sobre este botón nos debe dirigir a la pantalla de la cocina	Resultado esperado	< 1 seg	1 – 2 seg
<b>Pantalla ducha</b>				
Pulsar sobre el botón de la ducha	Pulsando sobre este botón, deberá ocultarse, y debe de iniciar a elevarse la barra de progreso, además de iniciar las respectivas animaciones	Resultado esperado	< 1 seg	5 seg
<b>Pantalla dormitorio</b>				

Pulsar sobre el botón salir	Pulsando sobre este botón nos debe dirigir a nuestro perfil	Resultado esperado	< 1 seg	2 seg
Pulsar sobre el botón de acción	Pulsando sobre este botón nos debe dirigir a la pantalla del dormitorio de noche	Resultado esperado	< 1 seg	1 – 2 seg
Pulsar sobre el botón de ir a la derecha	Pulsando sobre este botón nos debe dirigir a la pantalla de la terraza	Resultado esperado	< 1 seg	1 – 2 seg
Pulsar sobre el botón de ir a la izquierda	Pulsando sobre este botón nos debe dirigir a la pantalla del baño	Resultado esperado	< 1 seg	1 – 2 seg
<b>Pantalla dormitorio noche</b>				
Pulsación en el botón de la lámpara	Si antes de llegar al 100% pulsamos de nuevo sobre el botón, nos deberá dirigir a la pantalla del dormitorio.	Resultado esperado	< 1 seg	2 seg
<b>Pantalla terraza</b>				
Pulsar sobre el botón salir	Pulsando sobre este botón nos debe dirigir a nuestro perfil	Resultado esperado	< 1 seg	2 seg
Pulsar sobre el botón de acción	Pulsando sobre este botón nos debe dirigir a la pantalla de la sala de juegos	Resultado esperado	< 1 seg	1 – 2 seg
Pulsar sobre el botón de ir a la derecha	Pulsando sobre este botón nos debe dirigir a la pantalla del salón	Resultado esperado	< 1 seg	1 – 2 seg
Pulsar sobre el botón de ir a la izquierda	Pulsando sobre este botón nos debe dirigir a la pantalla del dormitorio	Resultado esperado	< 1 seg	1 – 2 seg
<b>Pantalla sala de juegos</b>				
Pulsar sobre el botón salir	Pulsando sobre este botón nos debe dirigir a la pantalla de la terraza	Resultado esperado	< 1 seg	3 seg
Pulsar sobre el icono del juego del 3 en raya	Pulsando sobre este botón nos debe dirigir a la pantalla de inicio del juego del tres en raya	Resultado esperado	< 1 seg	3 seg
Pulsar sobre el icono del juego de memoria	Pulsando sobre este botón nos debe dirigir a la pantalla de inicio del juego de memoria	Resultado esperado	< 1 seg	3 seg

<b>Pantalla inicio juego 3 en raya</b>				
Pulsar sobre el botón salir	Pulsando sobre este botón nos debe dirigir a la pantalla de la sala de juegos	Resultado esperado	< 1 seg	3 seg
Pulsar sobre el botón jugar	Pulsando sobre este botón nos debe dirigir a la pantalla de juego del tres en raya	Resultado esperado	< 1 seg	1 seg
<b>Pantalla inicio juego memoria</b>				
Pulsar sobre el botón salir	Pulsando sobre este botón nos debe dirigir a la pantalla de la sala de juegos	Resultado esperado	< 1 seg	3 seg
Pulsar sobre el botón jugar	Pulsando sobre este botón nos debe dirigir a la pantalla de juego de memoria	Resultado esperado	< 1 seg	1 seg
<b>Pantalla juego 3 en raya</b>				
Pulsar sobre el botón salir	Pulsando sobre este botón nos debe dirigir a la pantalla de inicio del juego del tres en raya	Resultado esperado	< 1 seg	3 seg
Pulsar sobre cualquier espacio vacío	Pulsando sobre un espacio vacío debe colocarnos nuestra ficha (ratón) en el espacio pulsado	Resultado esperado	< 1 seg	< 1 seg
<b>Pantalla juego memoria</b>				
Pulsar sobre el botón salir	Pulsando sobre este botón nos debe dirigir a la pantalla de inicio del juego de memoria	Resultado esperado	< 1 seg	3 seg
Pulsar sobre cualquier carta	Pulsando sobre una carta debe "girar" la carta para ver que elemento hemos levantado	Resultado esperado	< 1 seg	< 1 seg
Pulsar sobre el botón reiniciar	Pulsando sobre este botón debe reiniciarnos la partida, debe voltearnos las cartas y reiniciar nuestro marcador de monedas	Resultado esperado	< 1 seg	2 seg

COMPROBACIONES				
Descripción	Resultado esperado	Resultado obtenido	Tiempo obtenido por un usuario experto	Tiempo obtenido por un usuario inexperto
<b>Pantalla inicio de sesión</b>				
Intentar acceder con los campos completados correctamente	Debe dirigirte a la pantalla de perfil de usuario con todos tus datos	Resultado esperado	-	-
Intentar acceder con los campos erróneos	Debe mostrar un mensaje de error donde indique que los campos son incorrectos	Resultado esperado	-	-
Intentar acceder con los campos vacíos	Debe mostrar un mensaje de error donde indique que debes completar los campos	Resultado esperado	-	-
Intentar acceder con el campo del nombre vacío	Debe mostrar un mensaje de error donde indique que debes completar el campo del nombre	Resultado esperado	-	-
Intentar acceder con el campo de la contraseña vacío	Debe mostrar un mensaje de error donde indique que debes completar el campo de la contraseña	Resultado esperado	-	-
<b>Pantalla registro</b>				
Intentar acceder con los campos completados correctamente	Debe dirigirte a la pantalla de inicio de sesión para que puedas acceder a tu perfil	Resultado esperado	-	-
Intentar registrarse con algún campo vacío	Debe mostrar un mensaje de error donde indique que debes completar los campos	Resultado esperado	-	-
Intentar registrarse con el campo del correo con un formato no válido	Debe mostrar un mensaje de error donde indique que debes introducir un correo válido	Resultado esperado	-	-
Intentar registrarse con el campo del correo con un	Debe mostrar un mensaje de error donde indique que ya existe un usuario con ese correo registrado	Resultado esperado	-	-

correo ya registrado				
Intentar registrarse con los campos de la contraseña y confirmación de la contraseña diferentes	Debe mostrar un mensaje de error donde indique que la contraseña no coincide con la confirmación de la contraseña	Resultado esperado	-	-
<b>Pantalla perfil de usuario</b>				
Intentar acceder al juego con un ratón ya creado	Debe dirigirte a la pantalla de del salón, con tus datos y ratón	Resultado esperado	-	-
Intentar crear un ratón si no tenemos creado uno	Nos debe permitir pulsar sobre el botón más y dirigirnos a la pantalla de personalización	Resultado esperado	-	-
Intentar acceder al juego sin ningún ratón	No debe permitir pulsar el botón de jugar, o simplemente no debe realizar ninguna acción si el usuario decide pulsarlo	Resultado esperado	-	-
Intentar pulsar sobre el icono del ratón ya creado	No nos debe permitir volver a la pantalla de personalización	Resultado esperado	-	-
<b>Pantalla personalización</b>				
Intentar continuar con el campo del nombre completado y el color del ratón seleccionado	Debe dirigirte a la pantalla de inicio de sesión para que puedas acceder a tu perfil	Resultado esperado	-	-
Intentar continuar con el campo del nombre vacío	Debe mostrar un mensaje de error donde indique que debes completar el campo del nombre del ratón	Resultado esperado	-	-
Intentar continuar sin el color del ratón seleccionado	Debe mostrar un mensaje de error donde indique que debes seleccionar un color para el ratón	Resultado esperado	-	-

Pantalla armario				
Intentar pulsar el botón confirmar con algún artículo seleccionado	Debe dirigírnos a la pantalla del salón con el artículo aplicado a nuestro ratón	Resultado esperado	-	-
Intentar pulsar el botón confirmar sin ningún artículo seleccionado	No nos debe dirigir a la pantalla del salón, nos debe mostrar un mensaje que indique que debemos de seleccionar un artículo	Resultado esperado	-	-
Pantalla tienda – tienda de ropa / complementos				
Intentar confirmar compra con dinero suficiente o igual	Nos debe dejar realizar la compra, nos debe salir un mensaje indicándonos el número de artículos comprados. Además de redirigirnos a la pantalla del armario	Resultado esperado	-	-
Intentar confirmar compra sin dinero	No nos debe dejar realizar la compra, nos debe salir un mensaje indicándonos que no tenemos dinero suficiente	Resultado esperado	-	-
Intentar confirmar compra sin artículos seleccionados	No nos debe dejar realizar la compra, nos debe salir un mensaje indicándonos que debemos seleccionar al menos un artículo	Resultado esperado	-	-
Pantalla nevera				
Intentar pulsar sobre un artículo	Nos debe de actualizar la barra de progreso y eliminarlo de la nevera	Resultado esperado	-	-
Barra de hambre actualizada	Al salir de la pantalla la barra de progreso de hambre debe de estar actualizada	Resultado esperado	-	-
Pantalla tienda – supermercado				
Intentar confirmar compra con dinero suficiente o igual	Nos debe dejar realizar la compra, nos debe salir un mensaje indicándonos el número de artículos comprados.	Resultado esperado	-	-

	Además de redirigirnos a la pantalla de la nevera			
Intentar confirmar compra sin dinero	No nos debe dejar realizar la compra, nos debe salir un mensaje indicándonos que no tenemos dinero suficiente	Resultado esperado	-	-
Intentar confirmar compra sin artículos seleccionados	No nos debe dejar realizar la compra, nos debe salir un mensaje indicándonos que debemos seleccionar al menos un artículo	Resultado esperado	-	-
<b>Pantalla ducha</b>				
Salir de la pantalla cuando la barra de progreso haya alcanzado el 100%	Nos debe redirigir a la pantalla del baño	Resultado esperado	-	-
Barra de higiene actualizada	Al salir de la pantalla la barra de progreso de higiene debe de estar actualizada	Resultado esperado	-	-
<b>Pantalla dormitorio noche</b>				
Salir de la pantalla cuando la barra de progreso haya alcanzado el 100%	Nos debe redirigir a la pantalla del dormitorio	Resultado esperado	-	-
Salir de la pantalla cuando el usuario pulse de nuevo el botón de la lámpara	Nos debe redirigir a la pantalla del dormitorio	Resultado esperado	-	-
Barra de sueño actualizada	Al salir de la pantalla la barra de progreso de sueño debe de estar actualizada	Resultado esperado	-	-
<b>Pantalla sala de juegos</b>				
Actualizar barra de progreso	Se debe de ir actualizando la barra de progreso según vaya pasando el tiempo y al	Resultado esperado	-	-

	salir de la pantalla la barra de progreso de diversión debe de estar actualizada			
<b>Pantalla juego 3 en raya</b>				
Ganar partida	Si tenemos alguna combinación de nuestras fichas ganadora, debe salir el mensaje correcto y debe sumarnos una cantidad de monedas	Resultado esperado	-	-
Perder partida	Si el rival tiene alguna combinación ganadora, debe salir el mensaje correcto y debe restarnos una cantidad de monedas	Resultado esperado	-	-
Empate	Si completamos todos los espacios vacíos y ni nosotros ni el rival tenemos alguna combinación ganadora, debe mostrarnos el mensaje correcto y ni nos debe sumar ni restar las monedas	Resultado esperado	-	-
Pulsar sobre un hueco completado	No debe realizar ninguna acción	Resultado esperado	-	-
<b>Pantalla juego memoria</b>				
Pareja correcta	Si levantamos dos cartas con la misma imagen, se deben quedar las dos dadas la vuelta y nos debe sumar una cantidad de monedas	Resultado esperado	-	-
Pareja incorrecta	Si levantamos dos cartas con diferente imagen, se deben ocultar las imágenes y nos debe restar una cantidad de monedas	Resultado esperado	-	-
Pulsar sobre una carta ya levantada	No debe realizar ninguna acción	Resultado esperado	-	-



EFECTOS DE SONIDO / AUDIOS				
Descripción	Resultado esperado	Resultado obtenido	Tiempo obtenido por un usuario experto	Tiempo obtenido por un usuario inexperto
<b>Pantalla inicio carga</b>				
Música inicio	Debe iniciar la música del inicio al iniciar la aplicación y debe parar 1-2 segundos después de que la barra de carga llegue al 100%	Resultado esperado	-	-
<b>Pantalla inicio sesión</b>				
Música inicio de sesión	Debe iniciar la música de inicio de sesión en cuanto se inicie la pantalla de inicio de sesión, y debe detenerse al cambiar de pantalla	Resultado esperado	-	-
<b>Pantalla registro</b>				
Música registro	Debe iniciar la música de registro en cuanto se inicie la pantalla de registro, y debe detenerse al cambiar de pantalla	Resultado esperado	-	-
<b>Pantalla perfil de usuario</b>				
Música perfil usuario	Debe iniciar la música del perfil de usuario en cuanto se inicie la pantalla del perfil de usuario, y debe detenerse al cambiar de pantalla	Resultado esperado	-	-
<b>Pantalla personalización</b>				
Música personalización	Debe iniciar la música de personalización en cuanto se inicie la pantalla de personalización, y debe detenerse al cambiar de pantalla	Resultado esperado	-	-

<b>Pantalla información</b>				
Música información	Debe iniciar la música de información en cuanto se inicie la pantalla de información, y debe detenerse al cambiar de pantalla	Resultado esperado	-	-
<b>Pantallas: salón, cocina, baño, dormitorio, terraza</b>				
Efecto de sonido: cambiar habitación	Debe de iniciar el efecto de sonido cuando el usuario pulse el botón de ir a la derecha o ir a la izquierda	Resultado esperado	-	-
<b>Pantalla dormitorio</b>				
Efecto de sonido: apagar luz	Debe iniciarse el efecto de sonido cuando el usuario pulse el botón de acción de esta pantalla	Resultado esperado	-	-
<b>Pantalla tienda</b>				
Efecto de sonido: comprar / gastar dinero	Debe iniciar el efecto de sonido cuando el usuario efectúe alguna compra y gaste monedas y detenerse cuando se acabe el tiempo de efecto	Resultado esperado	-	-
<b>Pantalla nevera</b>				
Efecto de sonido: acción masticar	Debe iniciar el efecto de sonido cuando el usuario pulse algún artículo de la categoría de comida y gaste monedas y detenerse cuando se acabe el tiempo de efecto	Resultado esperado	-	-
Efecto de sonido: acción beber	Debe iniciar el efecto de sonido cuando el usuario pulse algún artículo de la categoría de bebida y gaste monedas y detenerse cuando se acabe el tiempo de efecto	Resultado esperado	-	-

Pantalla supermercado				
Efecto de sonido: comprar / gastar dinero	Debe iniciar el efecto de sonido cuando el usuario efectúe alguna compra y gaste monedas y detenerse cuando se acabe el tiempo de efecto	Resultado esperado	-	-
Pantalla ducha				
Efecto de sonido: ducharse	Debe iniciar el efecto de sonido cuando el usuario pulse el botón de la ducha y debe durar hasta que salga de la pantalla	Resultado esperado	-	-
Pantalla dormitorio noche				
Efecto de sonido: despertador	Debe iniciar el efecto de sonido cuando la barra de progreso del sueño este al 70% y se debe detener cuando salga de la pantalla	Resultado esperado	-	-
Efecto de sonido: encender luz	Debe iniciarse el efecto de sonido cuando el usuario pulse el botón de la lámpara de esta pantalla	Resultado esperado	-	-
Pantalla juego 3 en raya				
Efecto de sonido: comenzar partida	Debe iniciar el efecto de sonido cuando el usuario entre en la pantalla	Resultado esperado	-	-
Efecto de sonido: ganar partida	Debe iniciar el efecto de sonido cuando el usuario complete una jugada ganadora	Resultado esperado	-	-
Efecto sonido: perder partida	Debe iniciar el efecto de sonido cuando la IA complete una jugada ganadora	Resultado esperado	-	-
Pantalla juego memoria				
Efecto de sonido: comenzar partida	Debe iniciar el efecto de sonido cuando el usuario entre en la	Resultado esperado	-	-

	pantalla y cuando pulse el botón de reiniciar			
Efecto de sonido: combinación correcta	Debe iniciar el efecto de sonido cuando el usuario complete una jugada ganadora	Resultado esperado	-	-
Efecto de sonido: combinación errónea	Debe iniciar el efecto de sonido cuando el usuario complete una jugada fallida	Resultado esperado	-	-
Efecto de sonido: terminar partida	Debe iniciar el efecto de sonido cuando el usuario haya acertado todas las combinaciones	Resultado esperado	-	-

## Errores y soluciones

### *Error de diseño*

Me encontré con un problema de diseño que me causaba desajustes en las imágenes y, especialmente, en los botones. Resolverlo resultó más complejo respecto al tiempo de lo esperado, a pesar de ser un detalle trivial. Originalmente, intenté ajustar las imágenes de fondo a la resolución de la pantalla en Android Studio en lugar de cambiar su resolución desde el principio. Esto provocó que los botones, independientemente de si estaban escalados o no, no se mostraban como los había configurado. La solución que encontré fue reescalar todas las imágenes de fondo y luego volver a implementarlas. Esto corrigió el error que estaba experimentando.

### *Error con las barras de progreso*

En los últimos días, me vi en la necesidad de desarrollar un servidor de tamaño reducido. El motivo principal radicaba en los desafíos surgidos al guardar y modificar las barras de progreso, donde ciertas acciones resultaban en superposiciones no deseadas. Para abordar este inconveniente, ideé una solución consistente en la creación de un servidor acompañado de un hilo general, lo cual permitiría su implementación en diversas pantallas de manera eficiente.

### *Error con el audio*

Cuando intentaba ejecutar la máquina virtual, noté que el audio no reproducía el sonido adecuadamente. Para abordar este problema, opté por realizar pruebas directamente desde mi dispositivo móvil. Varias semanas después, Android recibió una actualización. Tras esta actualización, el audio comenzó a funcionar correctamente tanto en la máquina virtual como en el dispositivo físico.

### *Error con los gridView*

Cuando buscaba una manera de mostrar artículos específicos en las pantallas que usaban gridView, me encontré con un obstáculo: gridView carecía del método necesario para esto. La solución que hallé fue sustituir los gridView por elementos

RecycledView. Aunque presentaban algunas diferencias, este último me proporcionaba una mayor facilidad de uso con diferentes métodos, lo que resultó en una solución más eficiente para mi problema.

#### *Error a la hora de pasar los artículos de las categorías de cabeza y torso*

Cuando me encontré con el desafío de pasar el artículo seleccionado a la pantalla del salón y otros escenarios, me di cuenta de que ninguno de los métodos disponibles era adecuado, ya que eran similares a los utilizados para las categorías de comida y bebida. Para resolver este problema, ideé la solución de almacenar el artículo completo dentro de las sharedPreferences. Una vez completado este paso, tuve que implementar estas opciones en todas las pantallas para asegurarme de que la imagen del artículo se mostrara correctamente.

## Ampliación y posibles

#### *Ampliación del catálogo de complementos*

Para aumentar la cantidad de complementos disponibles, se propone la generación de más artículos en las categorías de cabeza y torso. Esto permitirá ofrecer al usuario una variedad más amplia de opciones para personalizar su experiencia.

#### *Ampliación del catálogo de alimentos y bebidas*

Similar a la anterior ampliación, se sugiere incrementar la variedad de artículos en las categorías de comida y bebida. Esto garantizará una mayor diversidad de opciones de compra para satisfacer las preferencias del usuario.

#### *Diversificación de juegos*

Continuando con la línea de ampliaciones anteriores, se plantea la creación de nuevos juegos para proporcionar al usuario una selección más amplia y así obtener diferentes formas de obtener monedas.

#### *Posibilidad de comprar más de una unidad de un artículo dentro de las categorías comida y bebida*

Se propone la opción de adquirir más de una unidad de un artículo dentro de las categorías de comida y bebida. Esto permitirá a los usuarios comprar cantidades múltiples de sus productos favoritos de manera conveniente.

#### *Ampliación de opciones de color para el ratón*

Se sugiere aumentar las posibilidades de cambio de color para el ratón. Esto brindará a los usuarios una gama más amplia de opciones para personalizar su experiencia de usuario.

#### *Selección entre diferentes mascotas*

Se plantea la posibilidad de elegir entre una variedad de mascotas para acompañar al usuario. Esto añadirá diversidad y personalización a la experiencia del usuario.

#### *Múltiples mascotas en un correo electrónico*

Se propone la opción de tener más de una mascota en un correo electrónico. Esto permitirá a los usuarios interactuar con múltiples mascotas dentro de su correo electrónico, agregando diversión y entretenimiento a la comunicación digital.

### *Implementación de la opción de selección doble*

Por último, se propone la inclusión de una función que permita al usuario seleccionar simultáneamente dos artículos en el armario: uno de la categoría de la cabeza y otro de la categoría del torso. Al confirmar la selección, ambos artículos serán aplicados al avatar del usuario.

## Conclusión

Tras finalizar mi proyecto de grado, puedo decir que ha sido una experiencia sumamente enriquecedora, tanto a nivel individual como en términos tecnológicos y de aprendizaje. El desarrollo de esta aplicación ha sido un desafío que me ha impulsado a mejorar en varios aspectos de mis conocimientos académicos.

En primer lugar, este proyecto me ha enseñado la importancia de la organización y la planificación anticipada. Aprender a gestionar las distintas tareas con previsión y resolver problemas imprevistos de manera oportuna han sido habilidades clave que he adquirido durante este proceso. Además, he comprendido la importancia de completar cada tarea de manera exhaustiva antes de pasar a la siguiente, ya que la calidad del trabajo final es fundamental.

En el ámbito tecnológico, este proyecto ha sido una verdadera aventura de aprendizaje. Explorar campos como el manejo de servidores e hilos, que inicialmente no pensaba utilizar, me ha proporcionado un conocimiento invaluable sobre su creación y funcionamiento. También he profundizado mis habilidades en el uso de Android Studio, una herramienta que me ha fascinado y con la que he disfrutado trabajar.

Por otro lado, el tema elegido para el proyecto, así como la propia aplicación, han sido fuente de satisfacción y disfrute para mí. No solo he encontrado el proceso de desarrollo interesante, sino que también he disfrutado viendo cómo la aplicación evolucionaba y mejoraba con el tiempo.

En resumen, este proyecto no solo ha representado un hito importante en mi formación académica, sino también una experiencia enriquecedora que ha fortalecido mis habilidades técnicas, mi capacidad de organización y mi pasión por la tecnología. Estoy emocionado por continuar explorando nuevos proyectos y desafíos en el futuro.

## Bibliografía

### **Android: Fundamentos para crear tus primeras apps de calidad**

[https://www.udemy.com/share/101x2o3@vqF2EPEhXQFdFUwnSs0eQHCqT1\\_pehCB-8y4YAT3q3yUMTPblM3GtS7BJyod67o3kg==/](https://www.udemy.com/share/101x2o3@vqF2EPEhXQFdFUwnSs0eQHCqT1_pehCB-8y4YAT3q3yUMTPblM3GtS7BJyod67o3kg==/)

### **Minicurso de introducción a Material Design para Android**

<https://www.udemy.com/share/1033ms3@-G9DUflm93m7X-G0Ulq-y6wT62HlfTDJfV-W7xijl4j8zpqpcTMog48sMtlryjzEKw==/>

### **Room**

<https://developer.android.com/training/data-storage/room?hl=es-419>

### **Iconos**

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

### **Animaciones**

<https://lottiefiles.com/es/>

### **Sonido**

<https://pixabay.com/es/>

### **Elementos de la paleta de Android Studio**

[https://prezi.com/p\\_hvrfpaed2m/elementos-de-la-paleta-de-android-studio/](https://prezi.com/p_hvrfpaed2m/elementos-de-la-paleta-de-android-studio/)