**Centro de Bachillerato Tecnológico Industrial Y De Servicios No 44**

**Especialidad:**

**Programación**

**Asignatura:**

**Desarrolla Aplicaciones Móviles Para Android**

**MANUAL TÉCNICO**

**Presenta:**

**Fernanda Baez Pérez**

**Docente:**

**Maricela Hernández Morales**

**Teziutlán, Pue.**

**14 de Junio del 2025**

[1. Introducción 2](#_Toc201313263)

[1.1 Propósito 2](#_Toc201313264)

[1.2 Alcance de la aplicación 2](#_Toc201313265)

[2. Descripción general del sistema 3](#_Toc201313266)

[2.1 Arquitectura de la app 3](#_Toc201313267)

[2.2 Diagrama de flujo de pantallas 3](#_Toc201313268)

[3. Requisitos 4](#_Toc201313269)

[3.1 Funcionales 4](#_Toc201313270)

[3.2 No funcionales 4](#_Toc201313271)

[4. Componentes principales 5](#_Toc201313272)

[4.1 AndroidManifest.xml 5](#_Toc201313273)

[4.2 Recursos 5](#_Toc201313274)

[5. Módulo MainActivity 7](#_Toc201313275)

[5.1. Funcionalidades y responsabilidades 7](#_Toc201313276)

[5.2. Video de fondo y música 8](#_Toc201313277)

[5.3. Navegación a PreguntaActivity 8](#_Toc201313278)

[6. Módulo PreguntaActivity 9](#_Toc201313279)

[6.1. Definición de las clases Pregunta y Respuesta 9](#_Toc201313280)

[6.2. Renderizado dinámico de preguntas y respuestas 9](#_Toc201313281)

[6.3. Gestión de selección e imágenes interactivas 10](#_Toc201313282)

[6.4. Cálculo de la casa y paso a ResultadoActivity 10](#_Toc201313283)

[7. Módulo ResultadoActivity 11](#_Toc201313284)

[7.1. Recepción de datos por Intent 11](#_Toc201313285)

[7.2. Visualización de imagen y texto según la casa 11](#_Toc201313286)

[7.3. Reinicio de la aplicación 11](#_Toc201313287)

[8. Diseño de interfaces 12](#_Toc201313288)

[8.1. activity\_main.xml 12](#_Toc201313289)

[8.2. activity\_pregunta.xml 12](#_Toc201313290)

[8.3. activity\_resultado.xml 13](#_Toc201313291)

[8.4. Estilo y temas 13](#_Toc201313292)

[9. Gestión de recursos 14](#_Toc201313293)

[9.1. Audio 14](#_Toc201313294)

[9.2. Video 15](#_Toc201313295)

[9.3. Imágenes de respuestas y escudos de casas 15](#_Toc201313296)

[9.4. Strings y colores 16](#_Toc201313297)

[10. Ciclo de vida y gestión de recursos 18](#_Toc201313298)

[10.1. onCreate(), onDestroy() y manejo de MediaPlayer 18](#_Toc201313299)

[10.2. Prevención de fugas de memoria 19](#_Toc201313300)

[11. Pruebas y aseguramiento de calidad 20](#_Toc201313301)

[11.1. Casos de prueba funcionales 20](#_Toc201313302)

[11.2. Pruebas de UI y de usabilidad 21](#_Toc201313303)

[11.3. Pruebas de rendimiento (video y audio) 22](#_Toc201313304)

[12. Apéndices 23](#_Toc201313305)

[A. Estructura de carpetas y archivos 23](#_Toc201313306)

[B . Referencias y bibliografía 24](#_Toc201313307)

[C. Contacto y soporte 24](#_Toc201313308)

# Introducción

Nombre del proyecto: Sombrero Seleccionador

Versión: 1.0

## 1.1 Propósito

Este manual técnico describe a fondo la estructura, funcionalidades, diseño, recursos y ciclo de vida de la aplicación “Hogwartsquiz”. Su objetivo es servir como forma de entretenimiento para los aficionados a la saga Harry Potter o los interesados. Esta aplicación está diseñada con Android Studio usando el lenguaje Kotlin. Integra video y música de fondo, interacción gráfica con botones de imagen, y lógica para determinar resultados.

## 1.2 Alcance de la aplicación

Una app de Android que reproduce música y video de fondo mientras permite que el usuario responda un test de personalidad. El test se basa en valores relacionados a las casas de Hogwarts, con resultados visuales y descriptivos finales.

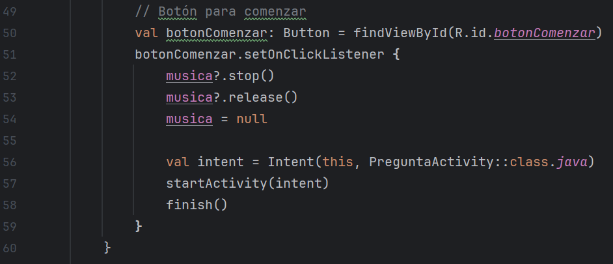
# 2. Descripción general del sistema

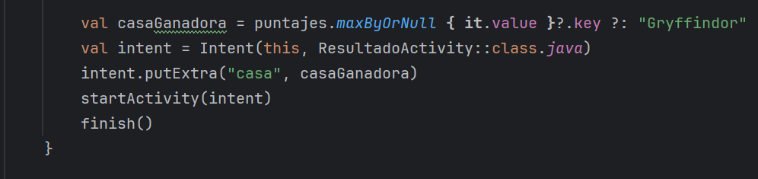
## 2.1 Arquitectura de la app

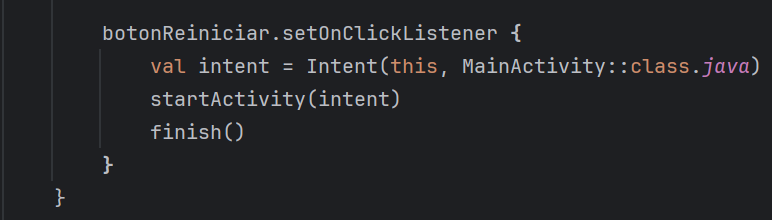
La app sigue el patrón tradicional de actividades (Activities) con UI declarativa XML y lógica en Kotlin. USA MediaPlayer para audio, VideoView para video, y flujo de datos entre pantallas con Intent.

## 2.2 Diagrama de flujo de pantallas

MainActivity --[clic "Comenzar"]--> PreguntaActivity --[clic "Terminar"]--> ResultadoActivity







Incluye el ciclo de inicio -> preguntas dinámicas -> resultado -> reinicio.

# 3. Requisitos

## 3.1 Funcionales

• Reproducir música y video al inicio

• Mostrar preguntas y respuestas gráficas

• Permitir selección única por pregunta

• Calcular y mostrar casa ganadora

• Posibilidad de reiniciar el flujo

## 3.2 No funcionales

• Compatible Android API 21+

• Uso moderado de recursos (no bloquear UI)

• Interfaz fluida y atractiva (colores, fuentes)

• Limpieza de recursos al salir de cada actividad

# 4. Componentes principales

## 4.1 AndroidManifest.xml

• Define tres actividades (MainActivity, PreguntaActivity, ResultadoActivity)

• Asigna MainActivity como LAUNCHER

• Configura permisos, back-up, íconos y tema

## 4.2 Recursos

Layouts:

• activity\_main.xml (video + botón superpuesto)

• activity\_pregunta.xml (scroll, botones, contenido dinámico)

• activity\_resultado.xml (resultado + imagen + reinicio)

Drawables:

• Respuestas (valentía, lealtad…)

• Escudos de Hogwarts (gryffindor, ravenclaw…)

Audio:

• hedwi.mp3 (música de inicio)

• hat.mp3 (música en preguntas)

Video:

• hogwarts.mp4

Strings:

• Textos estáticos: título, botón, frases por casa

Colores:

•Fondo, tintes de botones, texto

# 5. Módulo MainActivity

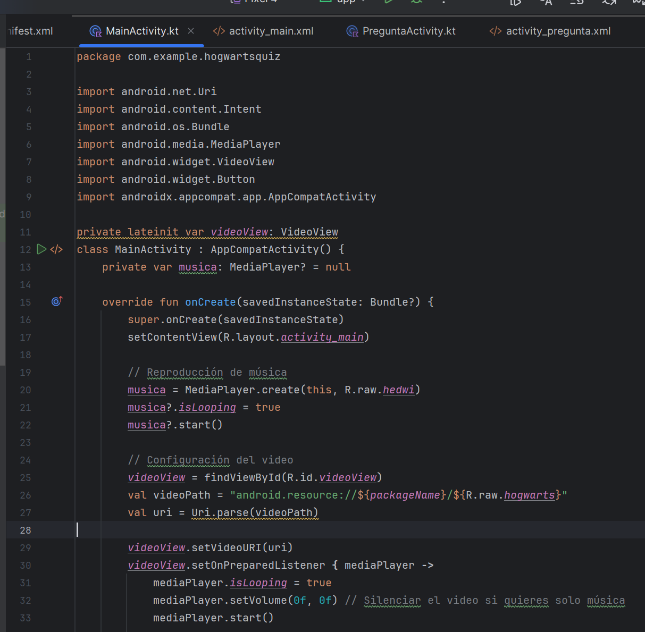
## 5.1. Funcionalidades y responsabilidades

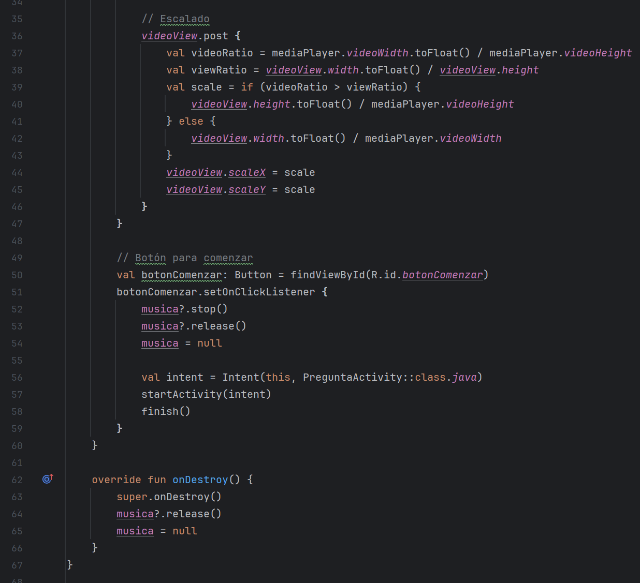
MainActivity es la primera pantalla de la aplicación. Sus responsabilidades incluyen:

• Reproducir un video de fondo.

• Superponer una capa oscura para mejorar la legibilidad del texto.

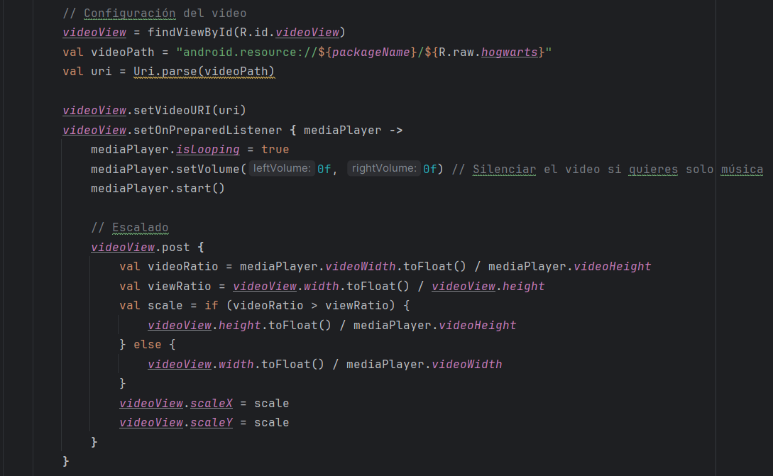
• Mostrar el título de bienvenida.

 • Proveer el botón de navegación hacia la sección de preguntas (PreguntaActivity).



Esta actividad tiene un diseño estático pero estéticamente inmersivo, alineado con la temática de Harry Potter.

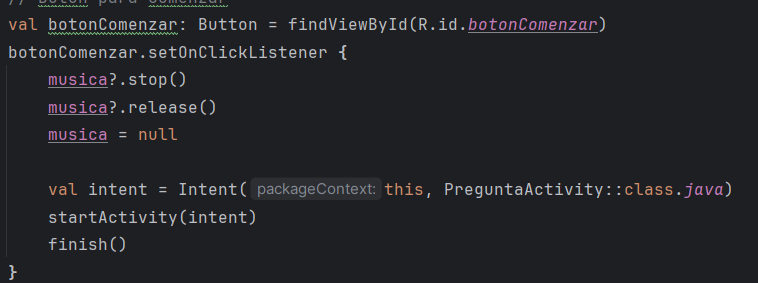
## 5.2. Video de fondo y música

El elemento VideoView carga un video local (por ejemplo, hogwarts.mp4) que se reproduce en bucle al inicio. La capa semitransparente oscurece el video para dar énfasis al contenido textual.

La música en esta pantalla puede reproducirse en un futuro usando MediaPlayer, aunque por simplicidad, se usa solo en la siguiente actividad.

## 5.3. Navegación a PreguntaActivity

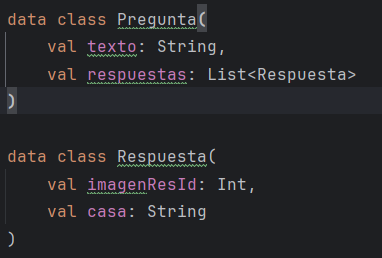
El botón “Comenzar” inicia PreguntaActivity mediante un Intent. Esta navegación permite al usuario empezar el cuestionario. La navegación es directa y no requiere pasar datos adicionales.



# 6. Módulo PreguntaActivity

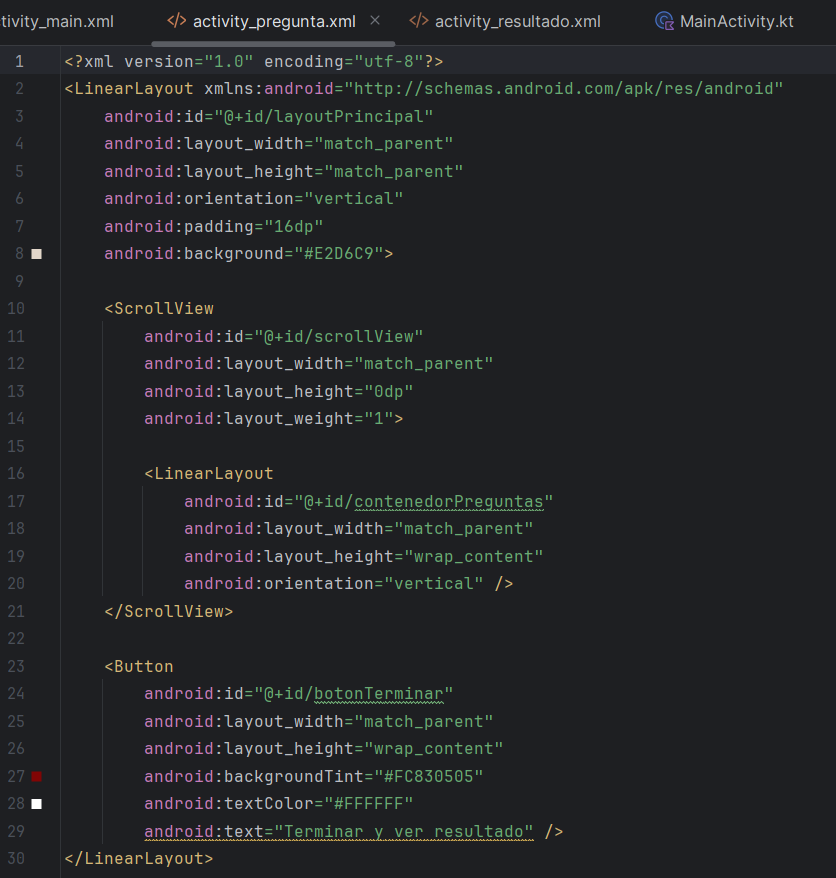
## 6.1. Definición de las clases Pregunta y Respuesta

Se definen dos data classes:



Estas clases modelan la estructura de cada pregunta y sus opciones, asociadas a casas de Hogwarts. Esto permite un sistema escalable para nuevas preguntas.

## 6.2. Renderizado dinámico de preguntas y respuestas

Las preguntas se renderizan en un LinearLayout dentro de un ScrollView. Cada pregunta contiene un TextView y un GridLayout con ImageButtons que representan las posibles respuestas. Todo esto se genera dinámicamente con código en tiempo de ejecución.

## 6.3. Gestión de selección e imágenes interactivas

Cada ImageButton tiene un listener que aplica un filtro dorado cuando se selecciona, usando setColorFilter. Las selecciones se almacenan en un arreglo IntArray que guarda la opción elegida por índice.



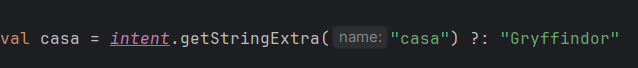
## 6.4. Cálculo de la casa y paso a ResultadoActivity

Al pulsar el botón “Terminar”, se recorren las selecciones y se suman los puntos por casa. La casa con mayor puntaje es enviada mediante un Intent a la ResultadoActivity.

# 7. Módulo ResultadoActivity

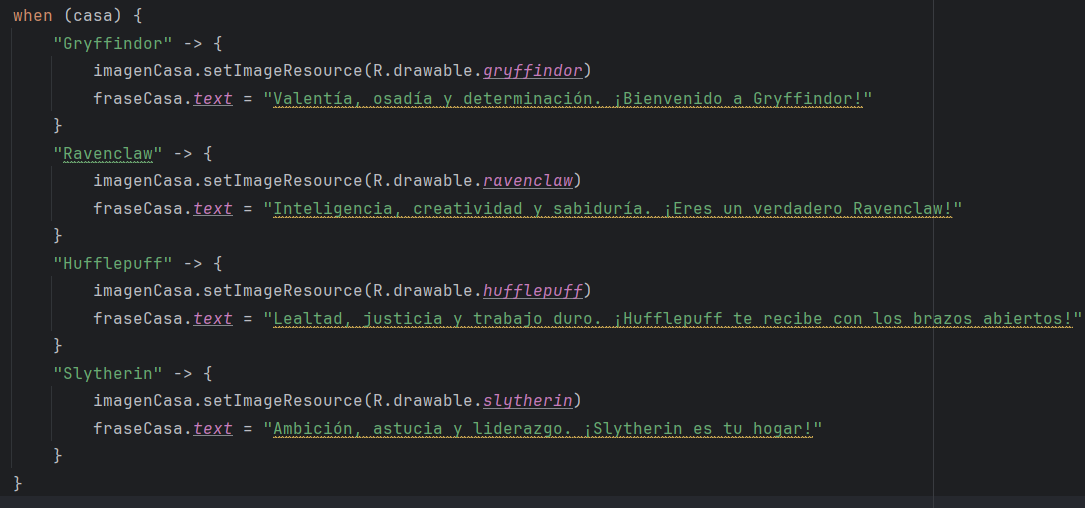
## 7.1. Recepción de datos por Intent

La casa ganadora se recupera con:

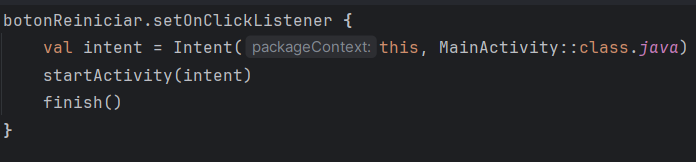


Esto asegura que haya un valor predeterminado en caso de fallo.

## 7.2. Visualización de imagen y texto según la casa

Se actualiza dinámicamente un TextView, un ImageView y otro TextView con la descripción/frase de la casa. Cada casa tiene una imagen y frase motivacional personalizada.

## 7.3. Reinicio de la aplicación

El botón “Volver al inicio” lanza MainActivity y termina la actual, reiniciando el ciclo de uso de la app.

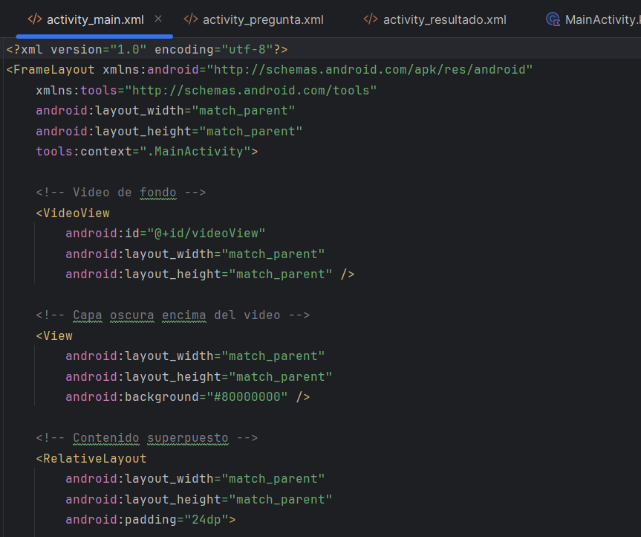
# 8. Diseño de interfaces

## 8.1. activity\_main.xml

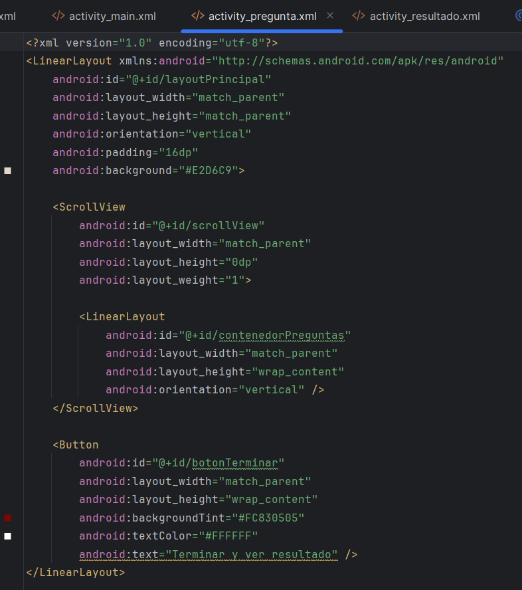
Usa FrameLayout con:

• VideoView a pantalla completa.

• View semitransparente.

 • RelativeLayout con título y botón.

## 8.2. activity\_pregunta.xml

Diseñado en un LinearLayout vertical con:

• ScrollView que contiene un LinearLayout para preguntas.

• Botón inferior para terminar el cuestionario.

## 8.3. activity\_resultado.xml

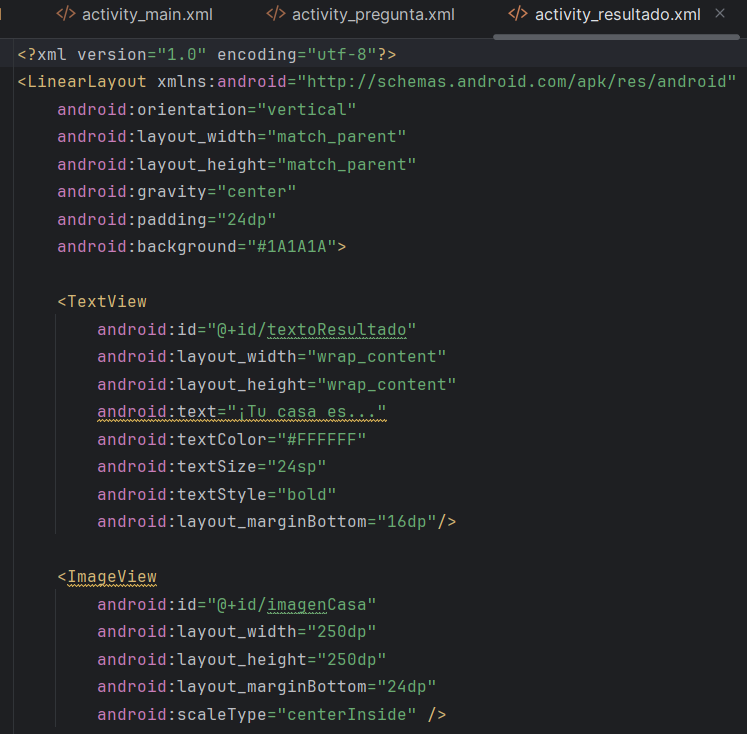
Estructura centrada verticalmente:

• Título (TextView)

• Imagen representativa de casa

• Frase final

• Botón de reinicio



## 8.4. Estilo y temas

La aplicación utiliza colores sobrios como negro, dorado y marrón, acordes con el estilo mágico de Hogwarts. Los textos están bien contrastados y la experiencia visual es inmersiva.

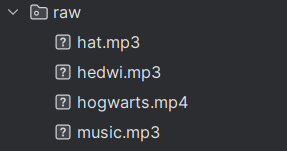
# 9. Gestión de recursos

La aplicación hace un uso intensivo de recursos multimedia y visuales que están organizados adecuadamente en las carpetas res/ de Android Studio. Estos recursos están clasificados en distintas categorías que contribuyen a la experiencia inmersiva de la app temática de Harry Potter.

## 9.1. Audio

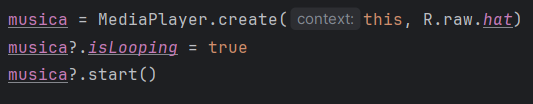
• hat.mp3: Este archivo de audio contiene la voz del Sombrero Seleccionador. Se reproduce al inicio del cuestionario para ambientar al usuario y reforzar la temática.

• hedwi.mp3: Puede usarse como música de fondo en otras actividades, ideal para reforzar la inmersión y la nostalgia de los fans.



Ambos archivos están almacenados en la carpeta res/raw/, y se reproducen con MediaPlayer.

Ejemplo:

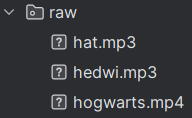


## 9.2. Video

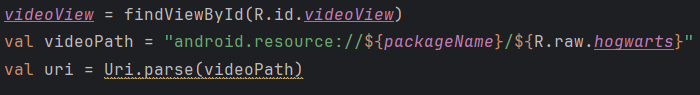
• Se encuentra en la carpeta res/raw/ como hogwarts.mp4.

• Es reproducido en la MainActivity mediante un VideoView a pantalla completa.

• Este video muestra una vista icónica del castillo de Hogwarts, contribuyendo a la ambientación mágica.



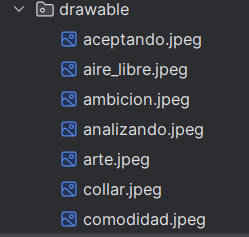
Configuración en MainActivity.kt:



## 9.3. Imágenes de respuestas y escudos de casas

Imágenes de respuestas:

• Cada opción de respuesta visual está representada por una imagen, como: quidditch.png, libros.png, serpiente.png, etc.

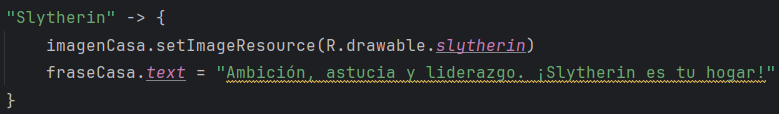
• Estas imágenes están en res/drawable/ y son mostradas dinámicamente en botones (ImageButton) dentro de la actividad de preguntas.

Escudos de casas:

• Imágenes como gryffindor.png, slytherin.png, hufflepuff.png, ravenclaw.png.



• Se muestran en ResultadoActivity para representar gráficamente la casa seleccionada.

• Están en alta resolución y adaptadas a centerInside para que escalen correctamente.

9.4. Strings y colores

Strings (res/values/strings.xml):

• Se utilizan para centralizar textos como el título, frases de cada casa y etiquetas de botones.

• Facilita la internacionalización y el mantenimiento del contenido textual.

Ejemplo:

<string name="titulo\_app">Sombrero Seleccionador</string>

<string name="boton\_comenzar">Comenzar</string>

Colores (res/values/colors.xml):

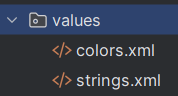
• La app define una paleta propia: marrón oscuro (#3C231B), dorado (#FC830505), gris (#1A1A1A), blanco puro y colores complementarios.

• Estos colores refuerzan el estilo visual de la saga.

Ejemplo:

<color name="fondo\_negro">#1A1A1A</color>

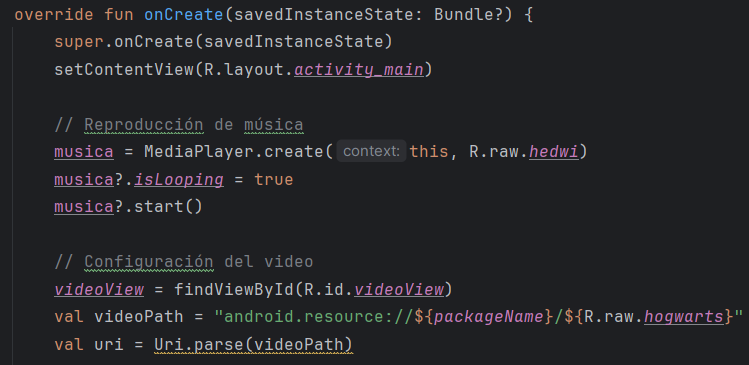
<color name="dorado">#FC830505</color>

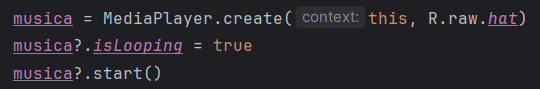


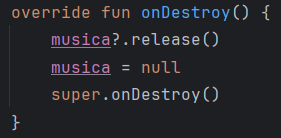
# 10. Ciclo de vida y gestión de recursos

El manejo adecuado del ciclo de vida de la aplicación es crucial para evitar fugas de memoria, errores de reproducción y brindar una experiencia estable al usuario. En esta aplicación, el ciclo de vida se gestiona principalmente en relación con el uso de MediaPlayer, VideoView y la navegación entre actividades.

## 10.1. onCreate(), onDestroy() y manejo de MediaPlayer

En cada Activity, el método onCreate() se usa para inicializar componentes visuales, configurar la lógica y cargar recursos multimedia:

PreguntaActivity – Inicio de audio



Esto evita que el recurso se mantenga en memoria después de salir de la pantalla, lo cual podría causar fugas de memoria o reproducción no deseada.

## 10.2. Prevención de fugas de memoria

Se siguen buenas prácticas en cada módulo para evitar leaks:

• Liberación explícita de MediaPlayer (release() en onDestroy()).

• Evitar referencias largas a contextos, como guardar Activity o View en variables estáticas.

• Uso de finish() al navegar a otra pantalla, para liberar el Activity anterior cuando ya no se necesita:

startActivity(intent)

finish()

• ScrollViews y LinearLayouts se utilizan sin imágenes de fondo pesadas para no saturar la memoria en dispositivos de baja gama.

Buenas prácticas adicionales

• Reproducción de video y audio fuera del hilo principal cuando sea necesario (aunque MediaPlayer lo gestiona internamente, es buena idea vigilar la latencia en dispositivos con menos recursos).

• Evitar múltiples instancias de MediaPlayer abiertas simultáneamente.

• Minimizar el trabajo dentro de onCreate() y distribuir tareas pesadas (como carga de imágenes grandes) si es posible usando librerías como Glide (no usada aquí, pero recomendable si se amplía el proyecto).

# 11. Pruebas y aseguramiento de calidad

La calidad de una aplicación no solo depende de su funcionalidad, sino también de su estabilidad, rendimiento, usabilidad y experiencia de usuario. A continuación se describen los tipos de pruebas realizadas para validar el correcto funcionamiento de la app Sombrero Seleccionador.

## 11.1. Casos de prueba funcionales

Se realizaron pruebas manuales para validar el cumplimiento de los requerimientos funcionales:

Caso de prueba Descripción Resultado esperado Estado

CP-01 Abrir la app Se muestra el video de fondo y botón de inicio ✅

CP-02 Reproducir video de fondo El video se reproduce sin trabas ni errores ✅

CP-03 Botón “Comenzar” Navega a la pantalla de preguntas ✅

CP-04 Seleccionar respuestas Las imágenes seleccionadas se resaltan correctamente ✅

CP-05 Preguntas con múltiples opciones Se permiten distintas cantidades de respuestas por pregunta ✅

CP-06 Cálculo de casa Al presionar “Terminar”, se muestra la casa con su imagen y frase ✅

CP-07 Botón “Volver al inicio” Regresa a la pantalla principal reiniciando la app ✅

## 11.2. Pruebas de UI y de usabilidad

Se realizaron pruebas de experiencia visual para asegurar accesibilidad y facilidad de uso:

• Contraste de colores: El texto blanco sobre fondo oscuro y los botones rojos/marrones cumplen con estándares de visibilidad.

• Touch targets: Las imágenes de respuesta tienen suficiente tamaño para una selección cómoda en dispositivos móviles.

• Scroll funcional: Se garantiza que todas las preguntas sean accesibles mediante ScrollView.

• Compatibilidad: Probada en dispositivos con Android 8.0+, pantallas HD y Full HD, con orientación vertical.

Además, se hizo una revisión informal de UX (experiencia de usuario) observando el flujo completo de navegación, para asegurar que:

• El usuario entiende qué hacer desde que abre la app.

• La navegación es clara e intuitiva.

• Las imágenes ayudan a decidir y no causan confusión.

## 11.3. Pruebas de rendimiento (video y audio)

Se validó el comportamiento de los medios audiovisuales:

Prueba Descripción Resultado

Reproducción de video Se inicia automáticamente sin retraso ✅

Loop de video No se detectaron cortes bruscos entre reinicios ✅

Reproducción de música MediaPlayer en PreguntaActivity funciona correctamente ✅

Detención de música Al salir de la actividad, la música se detiene sin errores ✅

También se monitoreó el uso de CPU y memoria en un dispositivo gama media:

• Uso estable de memoria (menos de 200MB durante la sesión).

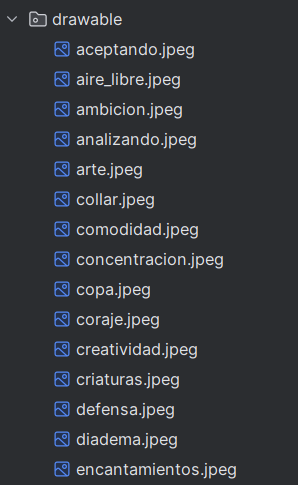
• CPU en torno al 15–20% durante reproducción multimedia.

• Sin cierres forzados ni ANRs.

# 12. Apéndices

Los apéndices ofrecen documentación complementaria que facilita la comprensión, mantenimiento y evolución de la aplicación Sombrero Seleccionador. Incluyen estructura de archivos, referencias y contactos de soporte.

## A. Estructura de carpetas y archivos



## B . Referencias y bibliografía

• Google Developers. Building Your First App

• Android Developers. Media Playback

• Kotlin Language Documentation. Data Classes

• Material Design Guidelines. Buttons and Controls

• Official Harry Potter Wiki. Hogwarts Houses

• Creative Commons. Licensing Overview

## C. Contacto y soporte

• Correo electrónico: [familyhp3421@gmail.com](mailto:familyhp3421@gmail.com)

• Número telefónico: +52 232 197 3448