PRACTICA T2

DESARROLLO DE APLICACIONES PARA DISPOSITIVOS MOVIL

Alberto Blanco Barrios

JeSUs Tellez Serrano

**Tabla de contenido**

[**Introducción** 3](#_Toc529888893)

[**Requisitos completados** 3](#_Toc529888894)

[Requisitos mínimos 3](#_Toc529888895)

[Desafíos plata 5](#_Toc529888896)

[Desafío Oro 5](#_Toc529888897)

[Pantallas 7](#_Toc529888898)

[Menús 7](#_Toc529888899)

[Juego 8](#_Toc529888900)

[Conclusión 8](#_Toc529888901)

[Referencias 8](#_Toc529888902)

## **Introducción**

En este documento se explicará el desarrollo de una aplicación para móviles en Android Studio API 28. La aplicación es un juego tipo Quiz sobre Spiderman.

A la versión anterior se ha añadido la gestión de usuarios y conjuntos de preguntas que se almacenan de forma persistente en una base de datos SQLite. La implementación de la gestión de la base de datos se ha realizado sobre la librería ROOM que nos proporciona una capa de abstracción sobre SQLite.

En el apartado de Configuración, se han implementado la gestión del nombre de usuario, puntuación, fecha de última partida, número de partidas, dificultad, conjunto de preguntas elegido y foto que se puede adquirir a través de la cámara fotográfica del teléfono o en el caso de que no estuviese disponible, de la biblioteca del dispositivo.

En el apartado de Clasificación se encuentra una lista de los 10 mejores jugadores de la aplicación, según sus puntuaciones obtenidas.

En el apartado de preguntas se han añadido tres tipos diferentes: de video, de audio y de imagen, todas ellas con un selector radial de cuatro respuestas. Además existen dos dificultades (utilizando conjuntos de preguntas) y tres duraciones de juego (5, 10 o 15 preguntas).

## **Requisitos completados**

### Requisitos mínimos

#### R.1. Arquitectura básica

1. **Configuración:** permitirá introducir o modificar los valores necesarios para el funcionamiento del juego.

La actividad configuración consta de:

1. Gestión de usuarios. Se hablará más detalladamente sobre este punto en el R.10.
2. La modificación de la dificultad (**5**,**10**,**15** preguntas)
3. La modificación de los conjuntos de preguntas
4. **Clasificación:** mostrará una tabla con las mejores puntuaciones obtenidas en partidas anteriores.

Tabla que contiene una lista con los 10 usuarios que mejores puntuaciones han obtenido en el Quiz, en orden descendente según puntuaciones. Cada ítem de la lista contiene **nombre**, **foto**, y **puntuación** del usuario.

1. **Juego:** al finalizar se debe mostrar una pantalla de fin de partida con la puntuación obtenida.

Se trata de una pantalla con un texto de fin de juego y la puntuación que ha conseguido el usuario en esa partida. También dispone de dos botones, uno para volver al menú principal y otro para ir a la pantalla de clasificación.

#### R.2. Conjunto de preguntas

Se han creado dos conjuntos de preguntas diferentes que se almacenan en la base de datos.

1. Conjunto Fácil: Se trata de un conjunto base de 20 preguntas, que incluyen los tres tipos (video, audio e imagen) y tienen una dificultad reducida.
2. Conjunto Difícil: Se compone de las 20 preguntas del conjunto fácil más otras 5 de dificultad más elevada. Es decir, al jugar en fácil se seleccionan las preguntas de un conjunto de 20, y al jugar en difícil se seleccionan de un conjunto de 25.

Todas estas preguntas se almacenan en la base de datos de la aplicación (gestionada por ROOM) en una tabla llamada *question\_table* y gestionada por las clases *Question* y *QuestionDao*. Al ejecutarse la actividad *Intro* por primera vez se ejecuta la inserción de las preguntas de manera que se queden almacenadas de manera persistente.

#### R.3. Configuración o ajustes

1. Se puede fijar el número de preguntas con un botón que cambia entre (5,10,15) cada vez que lo pulsamos.
2. Se puede fijar el conjunto de preguntas a elegir con un botón que cambia entre fácil (Conjunto 1) y difícil (Conjunto 2) cada vez que lo pulsamos.
3. La creación de usuario se ha implementado dentro de la actividad *Profile*, que se accede a ella a través del botón **Gestionar usuarios** en la actividad de *Configuration.*
4. Si no se ha elegido o creado usuario al intentar comenzar la partida, la aplicación nos preguntará si estamos seguros de comenzar la partida como usuario anónimo. La gestión del usuario anónimo se ha realizado como la de cualquier otro usuario a excepción de que este se elige al comenzar la aplicación.
5. Todos los parámetros anteriores son persistentes, ya que se almacenan en la base de datos, más específicamente en la tabla de Jugadores. Cada fila corresponde a un Jugador con los campos: **Nombre de usuario, dificultad, conjunto de preguntas, fecha de última partida, número de partidas, puntuación y foto de perfil.**

#### R.4. Información en pantalla durante el juego

Durante la ejecución de la partida se muestra en todo momento en pantalla la siguiente información, en la parte superior:

1. Número de pregunta (por ejemplo 1/10): Es el número de la pregunta actual.
2. Cronómetro: Es el tiempo en minutos y segundos que ha transcurrido desde que se ha iniciado la partida. Se actualiza en tiempo real.
3. Aciertos y fallos (por ejemplo A/F: 2/3): Son los aciertos y los fallos que se han realizado hasta el momento.

#### R.5. Desarrollo del juego

Al entrar al juego se carga la actividad *Questions,* que tiene un layout dividido en tres partes: la barra superior de información en la que aparecen los aciertos y fallos, el cronómetro, el título de la pregunta, etc; el contenedor de la pregunta y la barra inferior con el botón de continuar.

Las preguntas se recuperan de la base de datos de manera aleatoria al comenzar la partida, es decir, si el usuario está jugando a 10 preguntas en el conjunto difícil, se seleccionarán 10 preguntas aleatorias de 25 que hay en total.

Cada vez que se carga una pregunta, el contenido de la misma se introduce en un layout que dependerá del tipo de pregunta (*radialbutton\_layout, video\_layout, audio\_layout*) y el contenido del mismo se introduce en el contenedor de pregunta mediante la función *InflateLayout.*

En cada pregunta se muestra el contenido multimedia (audio, video o audio) y cuatro opciones estilo radial, entre los que se puede marcar una respuesta y cambiarla tantas veces como se quiera, hasta pulsar el botón siguiente, que cambiará de pregunta.

Al terminar la partida, el juego automáticamente cargará la actividad *EndGame*, que muestra una pantalla final con la puntuación del usuario y se actualizan todos los datos de la partida en base de datos, es decir, el número de partidas jugadas por el usuario, la fecha de la última partida y la puntuación del usuario, si esta es mayor que la que ya hay almacenada.

#### R.6. Tabla de puntuaciones

1. El juego almacena de forma persistente la puntuación en la base de datos, más concretamente en la tabla de Jugadores como se explicó en el **R.3.5.**
2. La tabla de clasificación consta de una lista con los 10 usuarios que han obtenido las mejores puntuaciones **(Foto de perfil, Nombre de usuario y puntuación).** La lista se ha diseñado sobre **list\_item2.xml** y se ha implementado en una clase llamada **CustomAdapter2.java** que extiende **BaseAdapter** y utiliza los **Viewholders** para distribuir el contenido.

### Desafíos plata

#### R.7.Diseño

El diseño de la aplicación se ha realizado pensando en que sea homogéneo, con el aspecto y colores clásicos de Spiderman. Para ello se han utilizado las siguientes características:

1. Se ha utilizado una paleta de colores homogénea, en la que incluimos los colores típicos de Spiderman, como son el rojo, el azul, el blanco y el negro.
2. La actividad de **Introducción** contiene una imagen de Spiderman, y un ***ImageButton*** *de telarañas*
3. Se ha añadido una imagen de fondo de una especie de araña a la zona de los menús.
4. Se ha añadido una fuente de Spiderman a casi todos los textos de la aplicación.
5. Se han creado nuevos botones, para dotar a la aplicación de un estilo propio.
6. Se ha añadido un icono a la aplicación desde el archivo AndroidManifest.

#### R.8.Varios conjuntos de preguntas

Se han creado dos conjuntos de preguntas, como se ha explicado en el apartado R.2.

#### R.9.Cronómetro

Se ha implementado un cronómetro que se muestra durante la partida en la parte superior de la pantalla.

Para ello, en la actividad *Questions* se ha creado un Runnable y un Handler que actualiza el texto utilizando el tiempo de partida.

### Desafío Oro

#### R.10. Perfiles de usuario

La gestión de los perfiles de usuario se ha realizado sobre la actividad ***Profile***. Esta consta de tres botones:

-Gestionar usuarios: En la que podemos seleccionar el usuario, modificar su fotografía de perfil y borrar el usuario.

-Crear usuario: Donde podemos crear nuestro usuario, dotándole de Nombre y fotografía.

1. **Elección de perfil:**

La elección del perfil se ha realizado en el botón Gestionar usuarios. Para ello se ha creado una lista como se especifica en el **R.6.2.** que contiene **foto de perfil**, **el nombre de usuario**, **la puntuación máxima**, **número de partidas jugadas** y **fecha de la última partida**. Para seleccionar un usuario solo hay que hacer *click* sobre su elemento en la lista, este cambiará el fondo del ítem de color rojo a blanco. Si no se ha seleccionado ninguno previamente, podremos observar que está seleccionado el usuario anónimo.

1. **Creación de usuario:**

Para la creación de usuario se ha añadido un botón crear usuario. Una vez que pulsemos sobre éste, aparecerá una caja de texto en la que se debe introducir el nombre de usuario. Tras este sencillo paso, si se encuentra disponible la cámara del dispositivo móvil se procederá a realizar una fotografía con la aplicación de captura del dispositivo. Si este no fuese el caso, se procederá a elegir una fotografía de las disponibles en la galería del dispositivo. Una vez completado este paso, el usuario estará creado en la base de datos.

Los datos del usuario se almacenan como una fila en la tabla de los jugadores. Cada fila contiene los datos pertenecientes a este, incluyendo la fotografía en formato ***Bitmap****.*

1. **Modificar perfil:**

La modificación del perfil se encuentra en el apartado de **Gestión de usuarios**, para ello solo hay que seleccionar el perfil del cual se quiere modificar la fotografía y presionar el botón de actualizar. Una vez realizado este paso, accederemos a la aplicación de la cámara o a la galería de fotos como se explicó en el apartado anterior.

1. **Eliminar perfil:**

La eliminación del perfil se realiza de la misma forma que la modificación, exceptuando que se deberá pulsar el botón eliminar en vez del botón actualizar. Este se elimina, y la selección del usuario pasará a ser la del usuario anónimo.

#### R.11. Almacenamiento persistente.

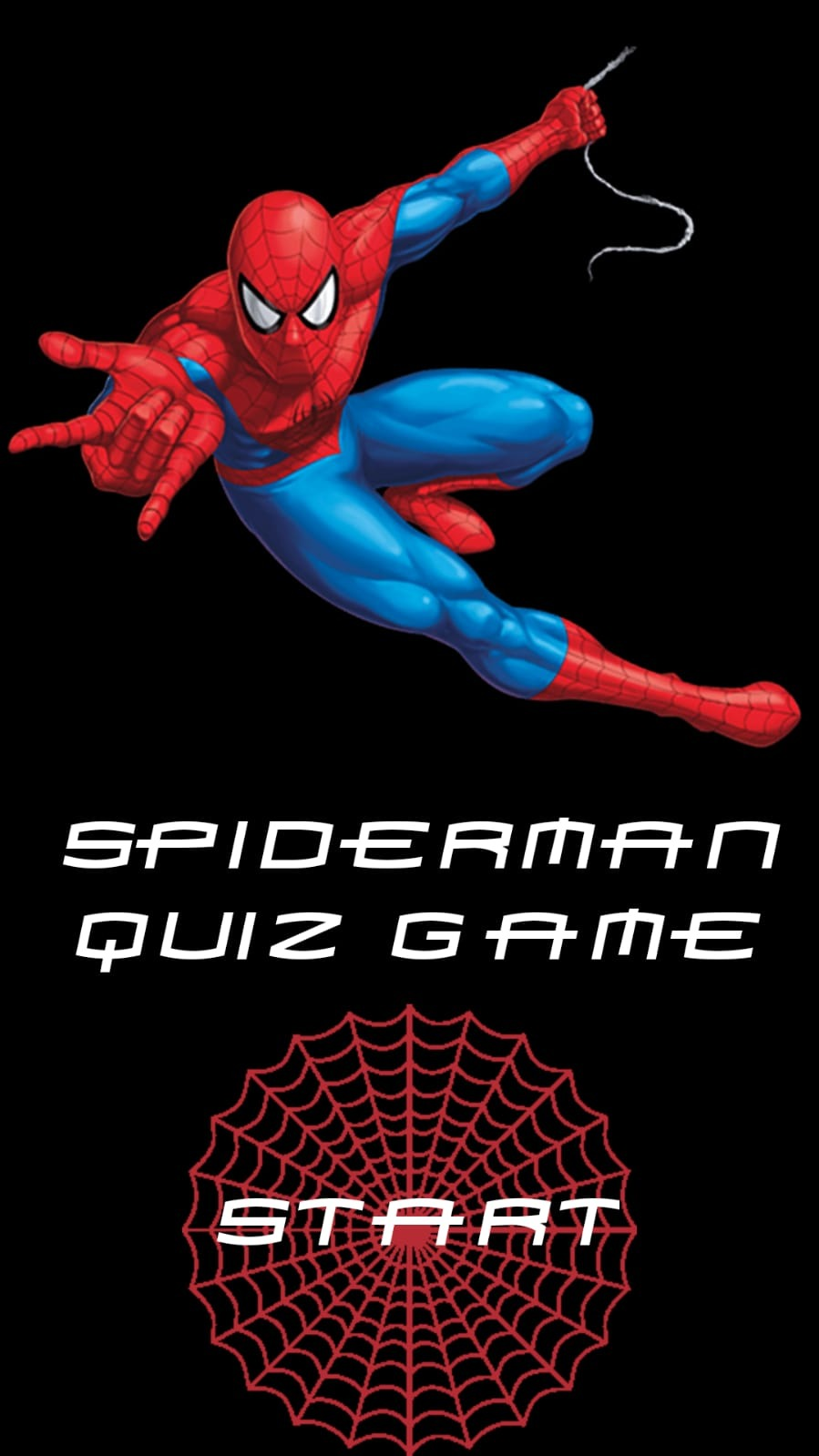
Se ha implementado una base de datos con la biblioteca ROOM que contiene dos tablas:

1. **Jugadores**: Esta tabla contiene el nombre del jugador, la puntuación máxima, el número de partidas, la fecha de la última partida, la fotografía de perfil en formato ***Bitmap***, la dificultad seleccionada (5, 10 ,15 preguntas) y el conjunto de preguntas seleccionado.
2. **Preguntas**: Esta tabla contiene toda la información de cada pregunta: el texto, el tipo de pregunta (video, audio o imagen), el grupo (fácil o difícil), las preguntas, que son un string separado por “;” al que se hace explode para convertirlo en un array, el id de la respuesta correcta, y la ruta del video/audio/imagen relacionado a la pregunta.

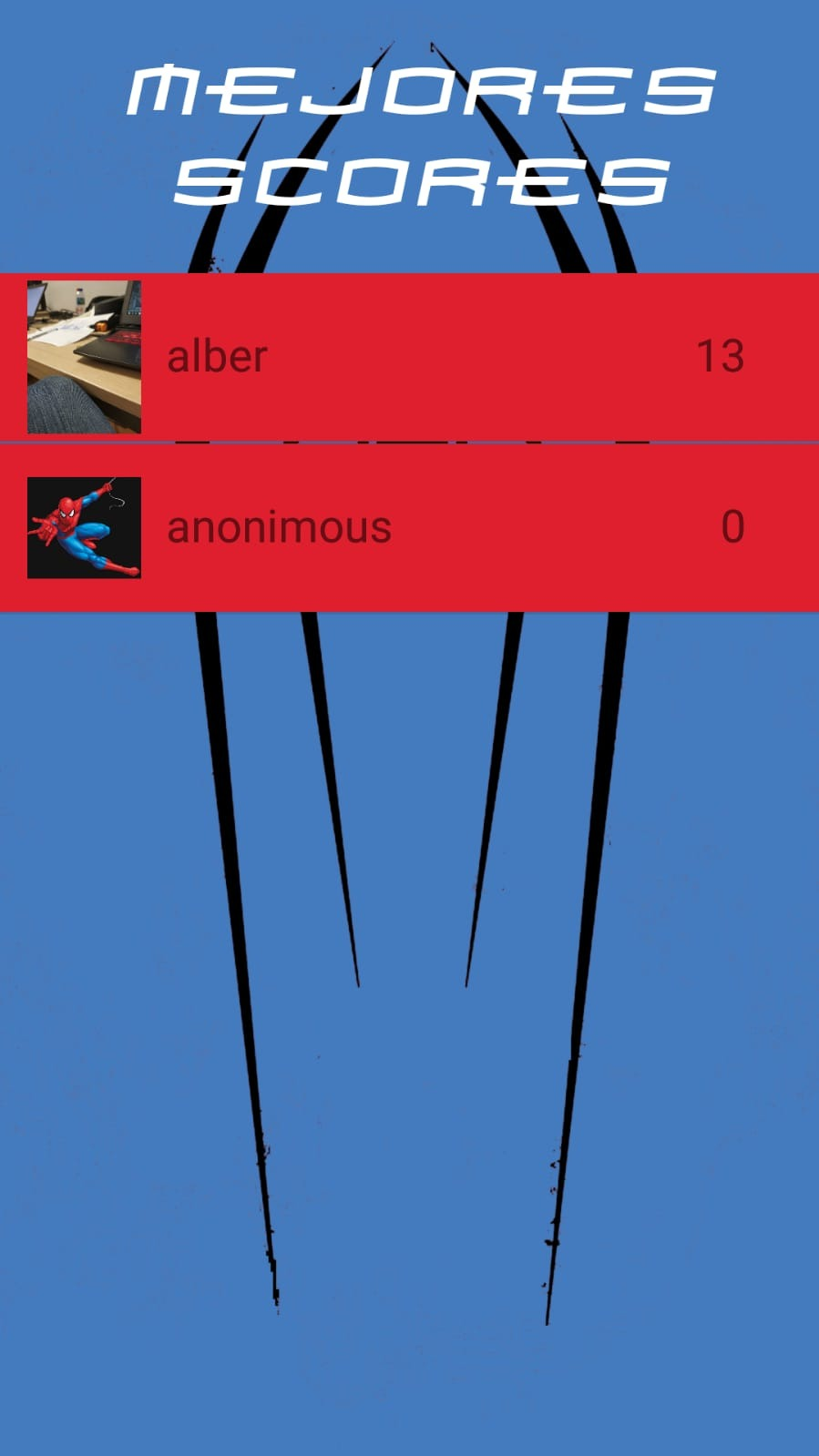
## Pantallas

### Menús

Pantalla 1: Inicio Pantalla 2: Menú inicial Pantalla 3: Configuración

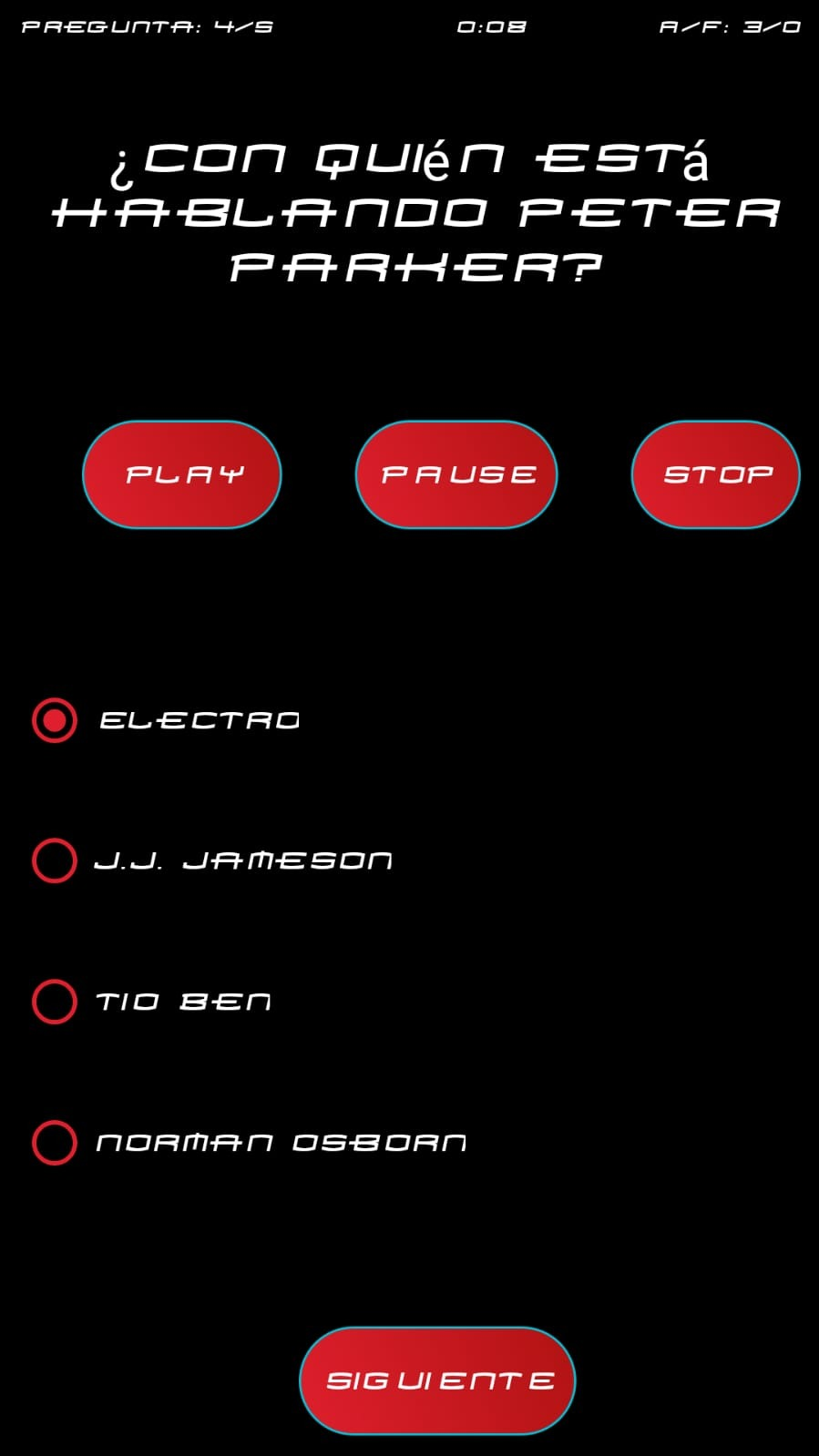
  

Pantalla 4: Fin del juego Pantalla 5: Puntuaciones

### Juego

Pantalla 6: Pregunta imagen Pantalla 7: Pregunta vídeo Pantalla 8: Pregunta audio

## Conclusión

Realizando la primera práctica en Android Studio tomamos contacto con el mismo y aprendimos lo más básico de la programación para dispositivos móviles Android.

En esta segunda práctica hemos tenido que realizar varias tareas más complejas y elaboradas que se acercan mucho más al desarrollo de una aplicación completa y funcional, encontrándonos con retos y problemas comunes de un desarrollo y aprendiendo a solucionarlos.

Ejemplos de esto serían la implementación de la base de datos local para la aplicación, algo muy útil para cualquier desarrollo, y que hemos aprendido gracias a este desarrollo, o la maquetación de una aplicación con un diseño uniforme y atractivo, con la que hemos aprendido los usos de los diferentes *layouts* y propiedades de los elementos de estos.

Quizás la parte más ardua de la práctica ha sido la cantidad de preguntas que ha habido que introducir en la aplicación, ya que hemos empleado bastante tiempo en recuperar videos e imágenes y crear el código de las propias preguntas. Quizás con grupos de preguntas más reducidos (dos grupos de 5 por ejemplo), nos habría llevado algo más de tiempo y habríamos aprendido lo mismo.

En cuanto a la toma decisiones durante el desarrollo, cabe destacar la decisión de utilizar SQLite y ROOM más en concreto, por su fácil usabilidad y su potencia a la hora de almacenar información, además de manejarse mediante lenguaje SQL, con el que ya estamos familiarizados.

## Referencias

Colours Spiderman: <https://www.schemecolor.com/spiderman.php>

Foto Spiderman: <http://pngimg.com/imgs/heroes/spider_man/>

Foto Red: <https://kathleenhalme.com/explore/spiders-clipart-spider-diagram/>

Fuente Spiderman: <https://www.1001fonts.com/homoarakhn-font.html>

Foto Background: <https://wallpaper.wiki/free-download-spiderman-backgrounds-for-iphone.html/wallpaper-wiki-art-logo-spiderman-background-for-iphone-pic-wpd005046/>

Todas las imágenes de las preguntas y los audios y videos han sido descargados desde Youtube y Google Imágenes.