



14 DE NOVIEMBRE DE 2018


# DESARROLLO DE APLICACIONES PARA DISPOSITIVOS MÓVILES

PRÁCTICA 2: JUEGO CON PERSISTENCIA Y MULTIMEDIA

EVA MARÍA PÉREZ FERNÁNDEZ & SERGIO RUS GONZÁLEZ

UNIVERSIDAD REY JUAN CARLOS

GRADO EN DISEÑO Y DESARROLLO DE VIDEOJUEGOS



## Contenido

Introducción y contexto .....	2
Descripción general.....	2
Requisitos contemplados .....	3
Arquitectura básica .....	3
Actividades .....	3
Implementación de los desafíos.....	4
Técnicas de persistencia.....	5
Conclusiones .....	6
Bibliografía y referencias.....	6

## Introducción y contexto

En esta memoria se detallan tanto el desarrollo como la implementación de la segunda práctica del bloque obligatorio de la asignatura de DADM, de acuerdo con lo requerido según las directrices de esta.

La práctica consiste en la creación de un juego quiz en el que parte de las preguntas incluyan contenido multimedia, guardado de forma local en el móvil, y almacenando de manera persistente diversos datos, como las mejores puntuaciones, la configuración, o los bloques de preguntas.

## Descripción general

Se ha optado por la realización de un quiz musical, considerando que esta era la opción más atractiva para nosotros, cumpliendo además con el requerimiento de contenido multimedia en las preguntas, ya que todas las preguntas están compuestas de audio que se puede reproducir.

La aplicación posee diversas actividades; menú, configuración, actividad principal (juego), fin del juego y mejores puntuaciones todas ellas con una estética unificada bajo los mismos criterios de diseño. Tanto los fondos de pantalla como la fuente del texto y la forma y color de los botones pretenden dar a la aplicación un aire futurista, con elementos de neón y letras estilizadas, pero poco angulosas (Ilustración 1).

Respecto al contenido de las preguntas, se han incluido dos temáticas, rock e indie. Para ambas categorías se ha optado por la inclusión de parte del repertorio de diversos grupos españoles. Durante el juego se presentan al jugador fragmentos de diversas canciones de estos grupos, teniendo este que seleccionar a qué grupo pertenece cada fragmento.



Ilustración 1: Menú principal

## Requisitos contemplados

Se han completado todos los requisitos mínimos y todos los desafíos de plata: se ha modificado el diseño, añadido un cronómetro, y empleado un conjunto de preguntas diferentes para cada temática.

### Arquitectura básica

La base del juego es similar a la de la primera práctica, adaptando la misma al uso de contenido multimedia y a la persistencia de datos, como se desarrolla más adelante.

Se han distribuido las diferentes pantallas creando diferentes actividades presentes en el proyecto: `activity_menu`, `activity_config`, `activity_score`, `activity_main` y `activity_end_game`.

Además, se han creado las clases necesarias que permiten utilizar Room para gestionar todo lo referente a memoria persistente, y otras dos clases para poder acceder de forma sencilla al contenido de las preguntas desde el juego (`Question.java` y `QuestionManager.java`).

Los dos bloques de preguntas empleados constan de veinte diferentes, todas de ellas incluyen contenido multimedia, una serie de opciones y una opción correcta. La aplicación se ha diseñado de forma que se pueden añadir de forma sencilla todas las temáticas que se deseen, por medio de la clase `QuestionManager`, que actúa como intermediaria entre el juego y la base de datos.

### Actividades

La aplicación cuenta con un menú principal desde el cual se puede acceder a un juego, a una pantalla de configuración y a otra de clasificación (Ilustración 1). El juego finaliza con una pantalla de fin de partida que nos devuelve al menú.

En la pantalla de configuración se puede fijar el número de preguntas, eligiendo entre cinco, diez o quince, así como la selección de nombre de usuario y la temática de la música con la que jugar. Si el jugador comienza una partida sin haber elegido previamente una configuración, la aplicación le avisará, dándole la opción de volver a la pantalla de configuración o jugar como anónimo. Estos parámetros también se almacenan de forma persistente en otra base de datos dedicada a la información de los usuarios.

Durante la partida, se muestran en pantalla tanto el número de pregunta actual respecto el total como el número de aciertos y fallos. También es visible el cronómetro, que contabiliza el tiempo desde que se inicia el juego (Ilustración 2).



Ilustración 2: Juego

Por último, la tabla de puntuaciones se almacena de forma persistente y se muestra en la pantalla de puntuaciones, accesible desde el menú (Ilustración 3).



Ilustración 3: Menú de puntuaciones



Ilustración 4: Pantalla de fin de juego

## Implementación de los desafíos

Se han completado todos los desafíos de plata, después de llevar a cabo la implementación básica.

Respecto al diseño, se han llevado a cabo diferentes cambios para dar a la aplicación una apariencia distintiva y homogénea. Se ha prescindido de la barra de título y modificado el icono de la aplicación con una versión simplificada (Ilustración 1) del logo del juego (Ilustración 5). Para todas las pantallas del juego se ha aplicado una estética futurista y elementos de neón, botones con transparencia y una breve animación al pulsarlos. Los fondos de pantalla son diferentes dependiendo de la temática elegida por el jugador, rock o indie. Se ha utilizado una gama de colores azules y violetas en combinación con blancos (Ilustraciones 1, 2, 3 y 4).



Respecto al uso de varios conjuntos de preguntas, se han utilizado dos bloques, uno para cada temática, música rock y música indie, como se ha comentado anteriormente. Mediante la clase QuestionManager, accedemos a las preguntas seleccionadas en función de la categoría desde la actividad principal.

El cronómetro también ha sido implementado en la actividad principal. Como es requerido, se muestra durante la ejecución del juego y también en la pantalla de fin de juego (Ilustraciones 2 y 4), y también se utiliza como variable para calcular las puntuaciones, siendo la puntuación final un porcentaje entre 0 y 100.

Ilustración 5: Logo del juego

## Técnicas de persistencia

Para la persistencia, se ha decidido emplear Room, debido a que hace sencilla la tarea de crear y utilizar bases de datos en aplicaciones Android. Se han creado dos bases de datos diferentes, una de ellas para almacenar las preguntas, y otra para los datos de usuario. Para cada una de las bases de datos, se han creado tres clases: User y Question, que contienen todos los atributos almacenados; UserDao y QuestionDao, que incluyen todas las *queries* pertinentes; y AppDatabase y QuestionDatabase, mediante las que crean las instancias de las bases de datos.

La clase QuestionManager, cuando es instanciada por primera vez, guarda todas las preguntas procedentes de un método de su clase en la base de datos, por lo que solo se insertan una vez. Tras haber sido insertadas, se accede a ellas mediante búsquedas utilizando otros métodos de esa misma clase. Las preguntas almacenan en la base de datos la temática a la que pertenecen, por lo que es sencillo tanto acceder a un bloque de preguntas por temática, como añadir más preguntas de la temática que deseemos, o añadir temáticas nuevas.

La base de datos de usuario es accedida desde la actividad de configuración, insertando en ella los jugadores nuevos que se crean. También se hacen inserciones cada vez que se crea un usuario anónimo. La puntuación final, calculada en función de los aciertos, del número de preguntas del nivel y del tiempo que se ha tardado por pregunta, se guarda en la base de datos, asociándola al usuario correspondiente, para poder obtener el ranking de mejores puntuaciones en la pantalla de clasificación. La fórmula para calcular las puntuaciones es la siguiente: **(aciertos\*total de preguntas del nivel) \* (tiempo/total de preguntas)**. Este último valor está normalizado entre 0 y 1, siendo 1 la proporción perfecta (tardar de media 2 segundos por pregunta) y 0 el peor de los casos (tardar de media 10 segundos por pregunta).

## Conclusiones

Con la elaboración de esta práctica, se ha comprobado de primera mano que realizar operaciones de bases de datos con Room es realmente sencillo; la utilización de Room simplifica en gran medida la gestión de bases de datos dentro de una aplicación.

También hemos podido comprobar que Android Studio proporciona una gran cantidad de facilidades para personalizar nuestras aplicaciones e introducir en ellas todo el contenido multimedia que deseemos.

## Bibliografía y referencias

Se ha utilizado como bibliografía para realizar todo el proyecto la librería de Android: <https://developer.android.com/?hl=es-419>

### Sobre la autoría del contenido:

*El diseño de la aplicación, incluyendo el icono de la misma, son de elaboración propia.*

*El contenido tanto de audio como de imágenes no son de elaboración propia, se han obtenido de Youtube y Google Imágenes, respectivamente.*

*Las imágenes son de licencia pública. Las canciones utilizadas no lo son. Los derechos de las canciones pertenecen a sus respectivos autores y se han utilizado únicamente para realizar esta práctica, sin fines comerciales.*