# El Juego Del Diccionario

# (Explicación del código)

Sergio Pérez Fernández 2º DAM

Logotipo

Descripción generada automáticamente

# Índice

1. Cómo funciona Jetpack Compose
2. Gradle
3. Temas
4. Activities
5. Composables reutilizables
6. Resources

**Cómo funciona Jetpack Compose**

Jetpack Compose es el kit de herramientas moderno de Android para compilar IU nativas. Respecto al antiguo sistema basado en el uso de .xml, Jetpack Compose es bastante más intuitiva y potente.

Adicionalmente, permite el uso de *Material Design 3*, diseñado para ser coherente con el nuevo estilo visual y la IU del sistema en Android 12 y versiones posteriores.

Es tan fácil de implementar como simplemente añadir el “*setContent*” y aplicar el tema.

Texto

Descripción generada automáticamente

Dentro del tema se hace llamada a *MainView*, una función que Jetpack Compose detecta como “visible” gracias a que va acompañada de *“@Composable”.*

*Material Design 3* cuenta con todo lo necesario para hacer vistas como botones, columnas, textos, *lazycolumns* (anteriormente *recycler views*). Además estos elementos son mucho más fáciles de implementar que sus anteriores versiones con .xml, ya que no requiere enlazar el xml con un controlador y vista propio para cada actividad.

Además, permiten el uso de modificadores y la posibilidad de poder aplicarlos múltiples veces. En la imagen inferior podemos ver cómo se aplica un separado (*padding*), se crea un color de fondo, y luego se vuelve a separar otra vez. Esto se debe a que los modificadores se aplican en orden, haciendo que las posibilidades sean casi infinitas.

Texto

Descripción generada automáticamente

Un factor a tener en cuenta usando Jetpack Compose es que se “regenera” la activity cuando se realizan cambios en esta cuando un jugador se encuentra en dicha activity, entonces para conseguir datos persistentes dentro de la activity entran en juego los remember, los cuales suponen un nuevo reto nunca antes visto, que es poder diferenciar variables nuevas y volátiles y otras persistentes que se guardan incluso si finaliza la actividad.

Texto

Descripción generada automáticamente

**Gradle**

Jetpack Compose necesita realmente pocas dependencias para funcionar, es por eso que el Gradle de este proyecto está algo vacío.

Texto

Descripción generada automáticamente

Simplemente he tenido que añadir media3 para el vídeo que se reproduce al abrir la aplicación y *Firebase*, la base de datos de Google.

**Temas**

Texto

Descripción generada automáticamente con confianza media

Jetpack Compose aplica sus temas manteniendo el código en estos 3 archivos:

* Color: Donde se especifican los colores que usará la aplicación. Estos no se aplican por defecto, hay que aplicarlos manualmente añadiendo el tema dentro del setContent, y luego aplicar cada color a cada elemento. Cabe recalcar que Material 3 integra la posibilidad de aplicar temas dinámicos, siendo posible aplicar un tema claro y uno oscuro, siendo estos aplicados automáticamente según se tenga aplicado en el dispositivo.

Texto

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto

Descripción generada automáticamente

* Theme: Aquí es donde se configura cada aspecto del tema, como si se aplican los colores dinámicos o no y los esquemas de color

Texto

Descripción generada automáticamente

* Type: Donde puedes crear tipografías para los textos y especificar si se usa negrita, el tipo de fuente, etc

Texto

Descripción generada automáticamente

Para este proyecto se ha elegido una paleta de colores principalmente amarillos que irradian energía e imaginación y anima a pensar y pasarlo bien.

**Activities**

Las activities son, resumidamente, las “pantallas” que nos encontramos, pudiendo iniciar una nueva o finalizar la actual para moverse entre pantallas. Esto se puede apreciar viendo que MainActivity te lleva a la activity JuegoActivity al pulsar sobre el botón de jugar y dentro de JuegoActivity se vuelve a MainActivity al pulsar sobre el botón de volver. Dentro de cada activity hay que volver a implementar Jetpack Compose, aunque Android Studio lo implementa automáticamente. Cabe recalcar que se pueden pasar variables de una activity a otra, no son 100% independientes una de otra.

Las activities se mantienen respecto al modelo basado en .xml y siguen utilizando un AndroidManifest.xml para funcionar correctamente, al cual he añadido la función de que la aplicación pueda conectarse a internet.

Texto

Descripción generada automáticamente

Este código implementa un total de 9 activities

Composables Reutilizables

Los Composables, al ser técnicamente funciones, podemos apartar en un fichero aparte aquellos Composables que sepamos que vayamos a reutilizar muchas veces. En ExtraComposables.kt he almacenado el código para crear un contador, un pase de diapositivas con controles, alertas *popup* y el reproductor de vídeo.

Todos estos composables son completamente reutilizables, ya que, por ejemplo, el contador pregunta cuán larga debe ser la cuenta atrás, el pase de diapositivas pide una lista para mostrar por pantalla, las alertas preguntan qué funciones deben realizar cuando se acepta o rechaza y el reproductor pide la url del vídeo. Como todas esas variables se especifican en la declaración, permite evitar la repetición de código innecesario y es, en general, una buena práctica.

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

Resources

Esta aplicación cuenta con una serie de recursos concretos y siendo diferenciados en diferentes carpetas.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Realmente los importantes se encuentran en *drawable* y *raw*. En ambos se encuentran las imágenes de la aplicación, pero adicionalmente, en *raw* se almacena el vídeo del inicio y todas las palabras y definiciones en archivos de texto diferentes para cada letra del abecedario.

Estos archivos tienen un formato algo raro, pero se debe a que una palabra puede tener varias definiciones, y con esto se pueden procesar de una manera dinámica las ~500000 líneas totales de los .txt

Texto

Descripción generada automáticamente