App “Sums”

RAUL GARCIA PABLOS

desarrollo de aplicaciones multiplataforma

Grupo DAM2C

Programación multimedia y dispositivos móviles

Análisis de la app final de Android

Raúl García Pablos

Contenido

[Resumen General 2](#_Toc126524066)

[Pantalla principal 2](#_Toc126524067)

[Cambio de personaje inicial 2](#_Toc126524068)

[Conexión con la base de datos 3](#_Toc126524069)

[Niveles 3](#_Toc126524070)

[Imágenes de los números 3](#_Toc126524071)

[Vidas 4](#_Toc126524072)

[Modo infinito 4](#_Toc126524073)

# Resumen General

He creado una aplicación móvil enfocada hacia niños pequeños que estén aprendiendo a sumar, restar y multiplicar. Se trata de un pequeño juego en el que deberán ir acumulando puntos con sus aciertos, en distintos niveles de dificultad ascendente. Se comenzará con tres vidas y esos serán los intentos fallidos que podrán acumular hasta que termine la partida.

El juego contará con seis niveles de dificultad que explicaré después.

En la aplicación se recogerá el alias del jugador para guardar su puntuación en una pequeña base de datos (sqlite) y poder mostrar la mejor puntuación en la pantalla principal.

# Pantalla principal

En la pantalla principal aparecerá una mascota entre 5 posibilidades de forma aleatoria cada vez que entremos en la aplicación. He escogido distintos elementos de material de clase como un rotulador, una goma o una escuadra.

Todos ellos han sido caracterizados por mí con plantillas de paint3D usando imágenes modificadas sin copyright.

En esta activity el usuario debe introducir un alias que servirá para guardar su puntuación de esa partida. Este campo es obligatorio y no se podrá avanzar sin que se rellene. También tiene un control con un límite de 16 caracteres para evitar posibles errores o mala visualización.

El botón “play” iniciará el juego.

En la parte inferior aparecerá (si la hay) la mayor puntuación alcanzada junto al alias del jugador que la haya logrado.

## Cambio de personaje inicial

Este cambio de personaje de la pantalla principal se controla con un Random() dentro de la clase “MainActivity.java” gracias al método .*setImageResource()*

## Conexión con la base de datos

Para la base de datos e optado por una opción sencilla como lo es la clase “SQLiteDatabase”. Para la creación y administración de la base de datos he creado una clase llamada “AdminSQLiteOpenHelper.java” que extiendo de la clase anterior.

Sobrescribo los métodos necesarios de la clase para su buen funcionamiento en mi aplicación y creo la tabla que voy a usar para guardar la puntuación de cada alias introducido en la app.

# Niveles

Los niveles cuentan con su propia activity en la que está incorporada la lógica de cada nivel.

Cada nivel cuenta con una clase java “Nivel*n*.java” donde *n* avanza de 1 a 5 y por último un “NivelFinal.java”.

Lo niveles tienen una dificultad ascendente cada 10 puntos en la puntuación de la siguiente manera:

1. Sumas cuyo resultado sea < 10
2. Sumas con un rango de resultado de 0 a 18
3. Restas (>=0)
4. Sumas y restas
5. Multiplicaciones
6. Modo infinito (explicado más adelante)

La lógica de cada nivel se encuentra dentro del método *numAleatorio()* que tienen todas las clases “Nivel*n*.java”.

El resto del código es similar en todas estas clases (conexión con su *layout*, variables, comunicación con la base de datos y la sobreescritura del método del botón “atrás” de Android para evitar que se pueda volver a la *activity* anterior).

La interfaz gráfica, es decir, el *xml* de los niveles, es muy similar:

Se mostrará el nombre del jugador “Player:” y la puntuación que lleva por el momento. En la pantalla aparecerán dos números a través de imágenes.

## Imágenes de los números

Cada número está vinculado a una imagen en la que aparece material escolar o uno de los personajes de la pantalla principal. Para el uno aparece un personaje, para el dos, dos, para el 5 aparecen 5 objetos iguales y así con todos los números del 1 al 9. Con el cero solo aparece un 0.

De esta manera ayudamos al usuario (esta app está destinada a niños) para que pueda visualizar y entender mejor la cantidad con la que está operando.

## Vidas

Las vidas están representadas por la imagen de un corazón, aparecen tras la palabra “Intentos:“ y se comienza la partida con 3, que irán disminuyendo con los errores que se cometan.

El juego consiste en introducir una respuesta en un campo que controlamos que sea numérico.

Tras introducirlo pulsamos el *check* verde que aparece al lado y esto ejecuta el método *comprobar()* de la clase del nivel correspondiente. Ese método comprobará si el resultado es correcto y, si es así, añadirá un punto al *score* y continúa el juego. De no ser así, se perderá un intento y en la pantalla aparecerá un corazón menos. Cuando el jugador se quede sin corazones, acaba la partida.

## Modo infinito

Si el jugador no falla, a partir de la puntuación 60 entrará en el modo infinito. Esto quiere decir que si no falla o no para la aplicación podrá aumentar su puntuación tanto como quiera. Este nivel tiene una lógica un tanto especial que puede verse en su método *numAleatorio()* de la clase “NivelFinal.java”. En resumen, hace una mezcla de los niveles anteriores y cada mínimo 5 puntos aparece una multiplicación (que presuponemos es lo más dificultoso), el resto de puntos se ganarán con una operación aleatorio (suma, resta o multiplicación).

# Conclusión final

Este proyecto me ha servido para afianzar y utilizar lo aprendido sobre Android Studio e investigar sobre soluciones a problemas que se me han presentado.

Por ejemplo, tenía varias dificultades con la rotación y adaptación de los *layout* a distintos dispositivos. Esto lo he solucionado usando *RelativeLayout*. El problema añadido a esto era el reinicio de las *activities* cuando se cambiaba de perspectiva. La solución ha sido agregar una línea de código en el *Manifest* para cada actividad:

android:configChanges="orientation|screenSize"