

PROGRAMACIÓN MULTIMEDIA Y DISPOSITIVOS MÓVILES

MEMORIA

ACTIVIDAD EVALUABLE

INDICE

DESCRIPCIÓN DE LA APP	3
GUIA DE USUARIO	3
Vista principal.	3
Vista de la partida.....	4
Vista final o resolución de partida.	5
JUSTIFICACIÓN DEL CODIGO	5
Start Activity	5
Game Activity	5
Final Activity.....	7
DIAGRAMA DE LA APP.....	8

DESCRIPCIÓN DE LA APP

Un grupo de amigos nos ha pedido que realicemos una aplicación del popular juego pasapalabra adaptado para entretener una pareja de novios a los cuales quieren darles el regalo en una caja fuerte con un código secreto de 3 dígitos concretos.

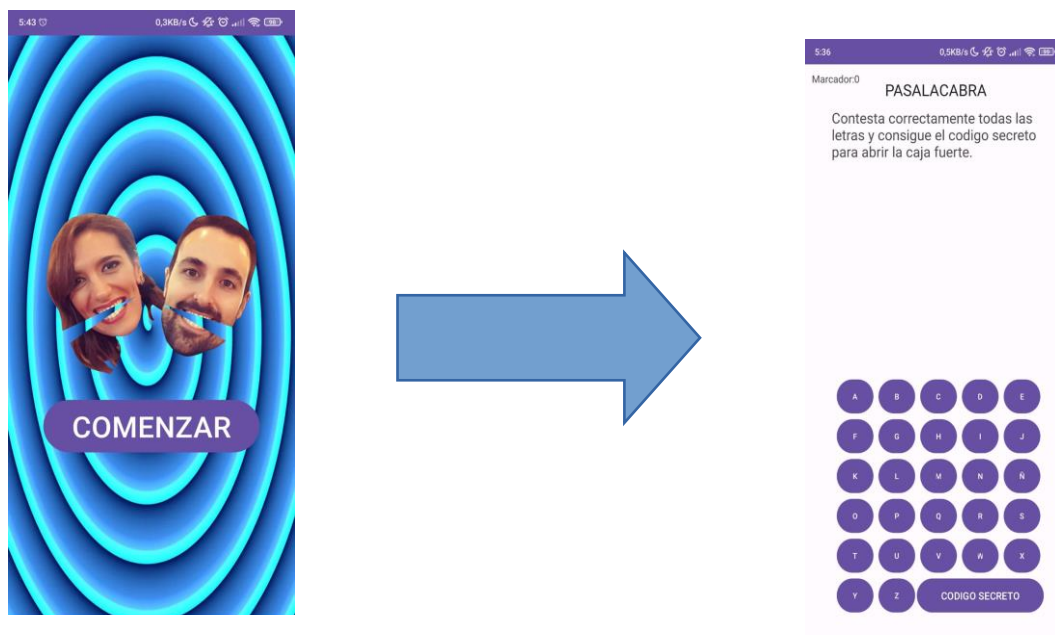
La dinámica es sencilla: los participantes deben responder preguntas asociadas con sus vidas y la que los rodea para conseguir el código secreto y poder abrir la caja fuerte.

La aplicación consta de tres ventanas totalmente configurables por el programador con los mínimos cambios en el código ya que es posible que en un futuro otros clientes quieran una aplicación parecida.

GUIA DE USUARIO

Vista principal.

Al iniciar la aplicación aparecerá la primera vista, la cual será una ventana con un fondo divertido, un botón para comenzar el reto y las caras de los novios, que en este caso se decidió, para darles más simpatía, partirles la cara por la boca dándole un toque a los famosos dibujos de South Park.



Vista de la partida.

En esta pantalla es en la que el jugador va a estar la mayor parte del juego y es donde se deben responder todas las preguntas de forma correcta.

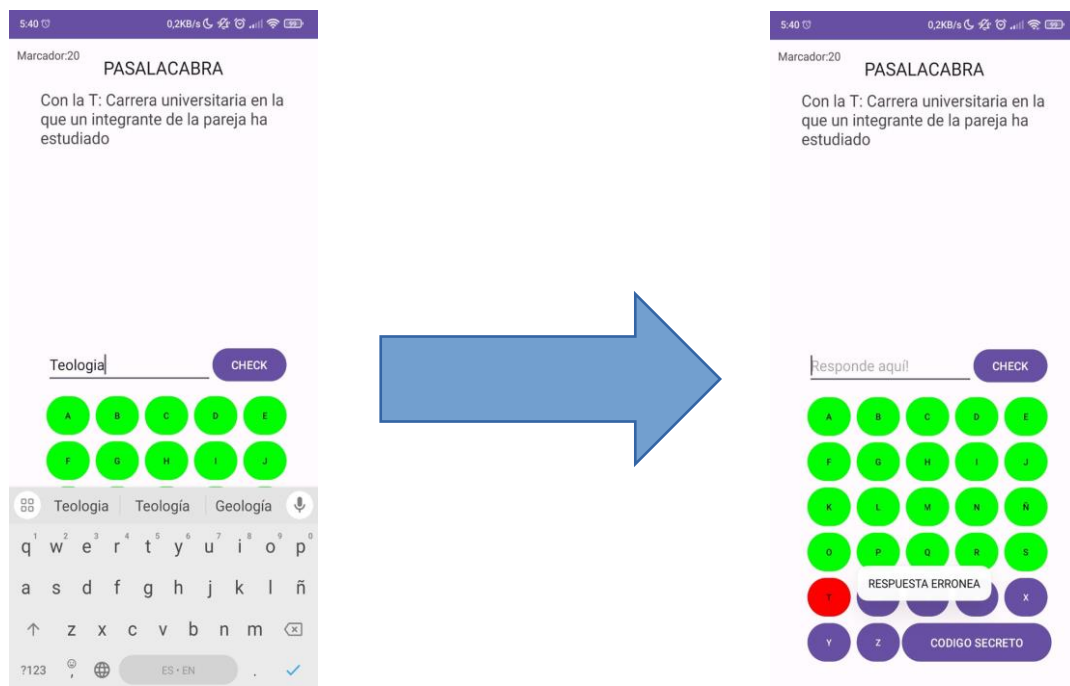
Lo primero que nos arriba a la izquierda es el marcador y una línea más abajo, centrado, nos encontramos el título de “PASALACABRA”, lo cual a los usuarios ya debe de ponerlos en contexto, un texto dándole las instrucciones del juego para conseguir el código secreto para abrir la caja fuerte, una serie de botones de la “A” a la “Z”, que se utilizan para seleccionar que pregunta quieres responder y el botón de “CODIGO SECRETO”, el cual para darle más dinamismo se ha decidido dejar como clicable, pero solo mostrará el código secreto si todas las letras están correctamente respondidas, por lo que el marcador tiene que llegar a 27 respuestas correctas.

Una vez pulsamos cualquiera de las letras, el texto introductorio cambiará por la pregunta concreta y ser hará visible una imagen de ayuda sobre la pregunta (si lo hubiera), una zona para escribir la respuesta y un botón de “CHECK” para ver si la respuesta introducida es correcta.

Al pulsar sobre “Responde Aquí!” se desplegará el teclado del terminal para que se pueda responder.

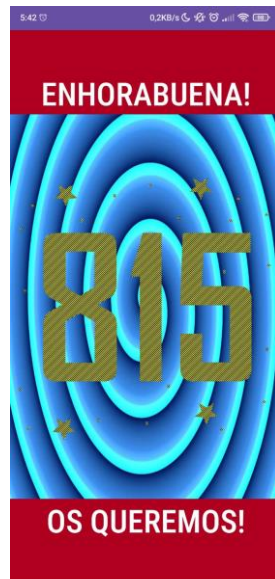
Si la respuesta es incorrecta, saltará un sonido de desaprobación, un aviso de que es incorrecta y además la tecla de la letra se pondrá en rojo hasta que se ponga la respuesta correcta.

Si fuera la respuesta correcta sonará el sonido de aprobación, el botón se pondrá en verde y se sumará un punto al marcador.



Vista final o resolución de partida.

En esta pantalla solamente se muestra el mismo fondo que en la vista principal, un par de mensajes de enhorabuena, y una imagen con el código secreto de la caja fuerte para que los novios pueda rescatar el dinero que les han regalado sus amigos.

**JUSTIFICACIÓN DEL CODIGO**

La aplicación consta de 3 activities con sus correspondientes layouts y en las tres he implantado un SCREEN_ORIENTATION_PORTRAIT para que se bloqueen en posición vertical.

En mi caso he usado el manifest para decidir con este bloque cuál de las 3 activities será la el main:

```
<intent-filter>
    <action android:name="android.intent.action.MAIN" />

    <category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
```

Ademas solamente he utilizado el método de ciclo de vida onCreate ya que no he visto necesario usar ningún otro puesto que no dispongo de una base de dato que necesite cerrar conexión ni nada parecido.

Start Activity

StartActivity corresponde al menú. En el cual dentro de onCreate he declarado un botón Start el cual tiene un setOnClickListener para cuando se le da click realiza un intent a la clase GameActivity dejándolo en la pila del startActivity.

Game Activity

GameActivity es la más compleja ya que contiene la lógica del juego. Lo primero que he hecho es instanciar las variables que voy a utilizar en el programa. Entre estas variables se encuentran las dos más importantes del programa, arrayPreguntas y arraySoluciones, las cuales nos servirá para modificar las preguntas y soluciones para otros posibles clientes.

El onCreate además de abrir la vista la aprovecho para instanciar el array de los botones que más tarde se inicializarán con un iterador en un bucle for que enlaza el botón al del layout, lo deja a la espera de ser clicado por el usuario y esconde el teclado si estuviera desplegado, para cuando pase de una pregunta a otra, no quede delante de los botones.

Una vez clickado llamará a la función pregunta() con los argumentos del iterador "i" y el botón.

También instancio y dejo a la espera de ser clicado el botón Código Secreto, el cual si el marcador es diferente a 27 correctas mostrará un toast diciendo que aún no desbloqueo las 27 preguntas. Sin embargo si el jugador tiene las 27 preguntas correctas mediante un intent abrirá FinalActivity.

- La función **pregunta()** se encarga de enlazar y setear el texto de la pregunta por la que existe en la posición del iterador dentro del array Preguntas, mirará en la carpeta res/drawable, si existe una imagen para esa pregunta o más bien para la solución de esa pregunta, ya que verifica si existe o no una imagen nombrada como la solución que contiene el array Solucion en la posición del iterable, además pondrá en visible el editText y el btnCheck para que el usuario pueda introducir y verificar su respuesta.

Cuando el usuario pulse sobre el btnCheck lo primero que hace es esconder el teclado mediante la función View.hidekeyboard() y convertir a mayúsculas y quitar los acentos y espacios de la respuesta del jugador y compararlo con la solución almacenada en el array Solucion (también sin acentos) gracias a los métodos de clase uppercase(), replace(" ", "") y a función removeAccents que normaliza y devuelve sin acentos el string introducido.

En caso de que la respuesta sea correcta se realizan las siguientes acciones:

- * Crea un MediaPlayer con el sonido correct.mp3 que hay en la carpeta /res/raw y se reproduce.
- * Cambia el fondo del botón correspondiente a verde con un setBackgroundColor.
- * Cambia el color de la letra del botón a negro gracias a un setTextColor.
- * Almacena la respuesta en mayúsculas en el array Respuesta.
- * Muestra la solución en el editText.
- * Suma 1 al marcador.
- * Bloquea el editText y el btnCheck seteando el método de clase isEnabled a false.
- * Espera a que termine de sonar el audio y cierra el mediaplayer.

En caso de que la respuesta sea incorrecta:

- * Crea un MediaPlayer con el sonido negative.mp3 que hay en la carpeta /res/raw y se reproduce.
- * Setea el editText a vacío.
- * Muestra el toast como respuesta incorrecta.
- * Setea el background del botón a rojo.
- * Cambia el color de la letra a negra.
- * Espera que el sonido termine de reproducirse el sonido para cerrar el mediaplayer.

- La función **View.hidekeyboard()** se encarga de cerrar el teclado.

Primero instanciando desde el contexto al método de entrada y convirtiendolo a tipo InputMethodManager ya que el método de clase getSystemService devuelve un tipo Any.

Segundo y ultimo a través del método de entrada se utiliza un método de clase hideSoftInputFromWindows() pasándole los argumentos windowToken, y numero de flags, en este caso 0.

- La función **removeAccents()** tiene el propósito de revisar y devolver un string introducido por argumento.

Lo primero que hace es normalizar el string mediante un objeto de una clase, que separa en caracteres individuales, por lo que un carácter con acento separará el carácter base y el diacrítico.

Después se crea un pattern y por último se aplica y se devuelve el string sin diacríticos.

Final Activity

FinalActivity corresponde al final de juego, lo único que haces es, en onCreate(), instanciar e inicializar un mediaplayer mediante un mediaPlayer.create(), que es reproducido con .start().

Además con el método de clase setOnCompletionListener del mediaplayer se espera a que termine de ser reproducido y cierra el mediaplayer con un .release().

El resto son textBlocks y una imagen en el XML que mostrará el mensaje de enhorabuena y el código secreto.

DIAGRAMA DE LA APP