



# PASALACABRA

Programación Multimedia y Dispositivos Móviles  
Evaluable Segunda Evaluación 23/24

Jose Manuel Moreno Bolivar  
[j.moreno.bolivar@gmail.com](mailto:j.moreno.bolivar@gmail.com)

## Contenido

Guía del usuario .....	2
Inicio de sesión .....	2
Pantalla principal .....	2
Inicio del Juego .....	2
Dinámica del juego .....	2
Puntuación y logros .....	2
Mensaje final .....	2
Progreso .....	2
Cambiar de boda o borrar partida .....	2
Justificación del código.....	3
Clase SQLiteOpenHelper Implementación: .....	3
Estructura de la Base de Datos: .....	3
Métodos de Creación y Actualización: .....	3
Seguridad: .....	3
Funciones Específicas: .....	3
Clase StartActivity Uso de SharedPreferences: .....	3
Navegación entre ventanas: .....	3
Desvinculación y Reinicio de la Aplicación: .....	3
Clase GameActivity Recuperación de Datos de StartActivity: .....	4
Reproducción de Sonidos: .....	4
Actualización de la Base de Datos: .....	4
Función de Pregunta: .....	4
Botón 'Ver Mensaje': .....	4
Clase FinalActivity.....	4
Recuperación de Datos de GameActivity: .....	4
Inicialización del Reproductor de Medios: .....	4
Mostrar el Mensaje Final desde la Base de Datos: .....	4
Aspectos para destacar .....	5
Esquemas o Diagramas .....	5
Esquema E-R: .....	5
Diagrama de flujo: .....	6

## Guía del usuario

Este manual te guiará a través de todas las funciones de la aplicación para que puedas disfrutar de este juego de palabras. Sigue los pasos para una experiencia única.

### Inicio de sesión

Abre la aplicación y escribe el código que se te ha proporcionado y dale al botón enlazar. (Modo demo: 1)

### Pantalla principal

En la pantalla principal aparecerán el nombre de los novios y un botón para comenzar la partida.

### Inicio del Juego

Para comenzar el juego únicamente debes de darle a comenzar o a continuar, si es que ya la comenzaste anteriormente.

### Dinámica del juego

Pulsa sobre cada letra y aparecerá la pregunta correspondiente, pulsa sobre la caja de texto y se desplegará el teclado, responde la pregunta y pulsa el botón 'Check'. Si la respuesta es correcta sonará un sonido asertivo y el botón de la letra cambiará a color verde, si no es así sonará un sonido desaprobador y el botón cambiara a color rojo.

### Puntuación y logros

Arriba a la izquierda está el marcador que irá refrescando el numero de respuestas correctas que llevas. Desbloquea el mensaje cuando todas las letras sean correctas, o lo que es lo mismo, el marcador sea de 27 puntos.



### Mensaje final

Para acceder al mensaje final, primero deberás tener el marcador en 27 puntos y después pulsar sobre el botón 'Ver mensaje'. Aparecerá otra ventana dándote la enhorabuena y mostrando el mensaje personalizado.

### Progreso

El progreso tiene autoguardado por lo que en cualquier momento podrás salir de la aplicación y volver a entrar, estando el juego en el mismo estado que lo dejaste.

### Cambiar de boda o borrar partida

Para cambiar de boda o borrar partida únicamente tendrás que volver a la pantalla principal y pulsar sobre el botón 'Desvincular'. Se borrará el estado de tu partida actual y aparecerá en pantalla la caja de texto para volver a poner el código de boda deseado.

## Justificación del código

### Clase SQLiteOpenHelper

#### Implementación:

He utilizado esta clase para realizar la creación, actualización y gestión de la base de datos SQLite en la aplicación Android

#### Estructura de la Base de Datos:

Se definen dos tablas, 'cliente' y 'pregunta', con sus respectivas columnas para almacenar información sobre clientes, mensajes y preguntas del juego.

#### Métodos de Creación y Actualización:

En 'onCreate', se crean las tablas y se pobla cada una de ellas con los datos iniciales utilizando las funciones 'poblarCliente' y 'poblarPregunta'

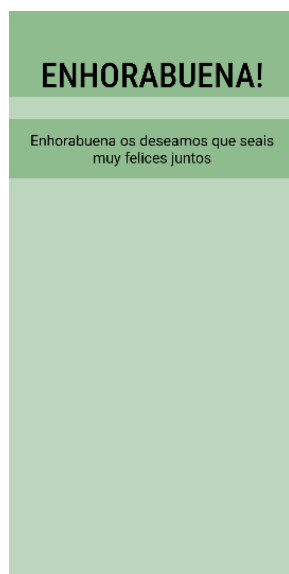
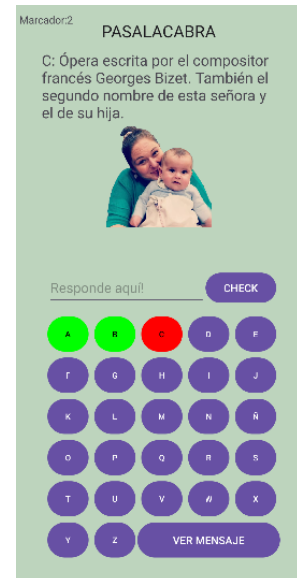
'onUpgrade' maneja las actualizaciones de la base de datos eliminando las tablas existentes y llamando a 'onCreate'.

#### Seguridad:

Se utilizan consultas parametrizadas para prevenir posibles vulnerabilidades de seguridad del tipo SQL Inyección.

#### Funciones Específicas:

Se utilizan setters y getters para obtener información detallada sobre clientes y preguntas asociadas al juego y para actualizar los registros de la base de datos.



### Clase StartActivity

#### Uso de SharedPreferences:

Empleo las SharedPreferences para almacenar la información persistente, como la conexión del usuario y el código de boda asociado. Esto permite que la aplicación recuerde el estado de la conexión entre sesiones de la aplicación.

#### Navegación entre ventanas:

Se utiliza el 'Intent' para iniciar la actividad de 'GameActivity' y se pasa el código de la boda como parámetro.

#### Desvinculación y Reinicio de la Aplicación:

La opción de desvincular (btnDesvincular) restablece las preferencias compartidas, elimina la base de datos, y reinicia la aplicación para mantener la coherencia visual.

## Clase GameActivity

### Recuperación de Datos de StartActivity:

Se utiliza 'Intent' para recuperar los datos de la actividad anterior, en este caso, el código de boda. Esto es esencial para cargar las preguntas y respuestas asociadas a esa boda.

### Reproducción de Sonidos:

Se utilizan dos instancias a la librería multimedia 'MediaPlayer' para reproducir sonidos de respuesta correcta e incorrecta.

### Actualización de la Base de Datos:

La Base de Datos se actualiza según las respuestas del usuario, marcando las letras como respondidas correcta o incorrectamente.

### Función de Pregunta:

La función 'pregunta' es la encargada de mostrar la pregunta correspondiente a la letra seleccionada a través de la función de la clase SQLiteOpenHelper, manejar la lógica de respuesta, reproducir el sonido y actualizar la interfaz según la respuesta del usuario.

### Botón 'Ver Mensaje':

Se implementa en 'btnCodigoSecreto' que, cuando se presiona, verifica si todas las respuestas son correctas antes de pasar a la 'FinalActivity'. Si no se han respondido bien todas las preguntas correctamente se muestra un mensaje con un toast.

## Clase FinalActivity

### Recuperación de Datos de GameActivity:

Se utiliza 'Intent' para recuperar los datos de la actividad anterior, en este caso, el código de boda. Esto es esencial para cargar el mensaje deseado por el cliente.

### Inicialización del Reproductor de Medios:

Se inicializa el objeto MediaPlayer para reproducir un archivo de sonido (R.raw.win) al iniciar la actividad.

### Mostrar el Mensaje Final desde la Base de Datos:

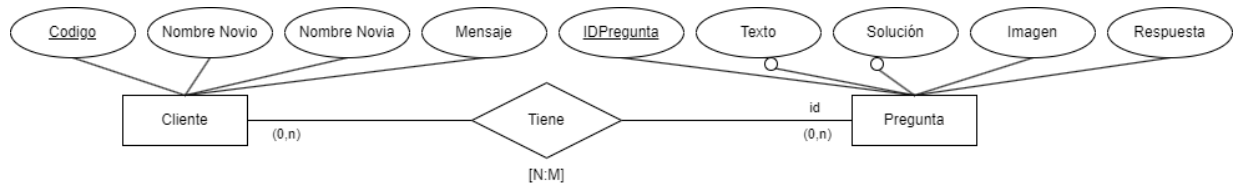
Se utiliza el objeto dbHelper (instancia de SQLiteOpenHelper) para obtener el mensaje final asociado al código de boda y se muestra en el TextView.

## Aspectos para destacar

La aplicación hace uso de la base de datos SQLite mediante la clase SQLiteOpenHelper para gestionar información como preguntas, respuestas y progreso del juego. Se emplean preferencias compartidas (SharedPreferences) para almacenar datos persistentes, como el estado de conexión y el código de boda. La integración multimedia se realiza a través de la clase MediaPlayer, reproduciendo sonidos específicos en diversas situaciones del juego, como respuestas correctas e incorrectas, y un sonido de victoria al completar todas las preguntas. Estos componentes contribuyen a la funcionalidad fluida y a la experiencia del usuario en la aplicación.

## Esquemas o Diagramas

Esquema E-R:



## Diagrama de flujo:

