



# PROYECTO FINAL FLUTTER: MEMORAMA CARD MANUAL TÉCNICO

*Jorge Benítez Lladó*

2DAM

*Programación Multimedia y  
Dispositivos Móviles*

## Índice

<b>1. Introducción.....</b>	<b>3</b>
<b>2. Análisis del problema.....</b>	<b>3</b>
2.a Problemática.....	3
2.c Análisis DAFO.....	3
2.d Monetización y beneficios.....	4
<b>3. Diseño de la solución.....</b>	<b>4</b>
3.a Tecnologías elegidas.....	4
3.b Arquitectura.....	4
3.c Diagrama de clases.....	4
3.d Diagrama E/R.....	4
3.e Consideraciones técnicas.....	4
<b>4. Documentación de la solución.....</b>	<b>5</b>
<b>5. Enlaces de interés.....</b>	<b>5</b>

# 1. Introducción

Este proyecto consiste en una aplicación de **juego de memoria con niveles de dificultad** desarrollado en **Flutter** con integración de **Firestore**. El objetivo principal es proporcionar una experiencia de entretenimiento educativa y desafiante para los usuarios, mejorando la memoria y la agilidad mental.

La importancia de este software radica en su capacidad para:

- Mejorar la retención y concentración en jugadores de todas las edades.
- Ofrecer una experiencia personalizable con diferentes niveles de dificultad.
- Permitir a los usuarios **guardar y comparar su rendimiento** mediante Firestore.

## 2. Análisis del problema

### 2.a Problemática

Existen diversas aplicaciones de juegos de memoria, pero muchas carecen de **un sistema de progresión, almacenamiento de récords personales y una interfaz atractiva**.

### 2.b Clientes potenciales

- **Niños y estudiantes:** Para mejorar la memoria y concentración de manera divertida.
- **Adultos y personas mayores:** Como herramienta de estimulación cognitiva.
- **Usuarios casuales de juegos móviles:** Que buscan desafíos rápidos y entretenidos.

### 2.c Análisis DAFO

Fortalezas	Oportunidades
Interfaz atractiva y moderna en Flutter	Creciente interés en apps educativas
Integración con Firestore para almacenamiento de datos	Posibilidad de expandir con más modos de juego
Múltiples niveles de dificultad	Monetización a través de anuncios o suscripción
Debilidades	Amenazas
Competencia con apps similares	Posibles problemas de compatibilidad en algunos dispositivos
Dependencia de Firestore para datos en la nube	Eliminación de la api

## 2.d Monetización y beneficios

Actualmente ninguna, pero en un futuro podríamos implementar:

- **Publicidad integrada:** Anuncios en pantalla después de cada partida.
- **Compras dentro de la app:** Eliminación de anuncios o desbloqueo de modos adicionales.
- **Suscripción premium:** Acceso a estadísticas avanzadas y personalización de cartas.

## 3. Diseño de la solución

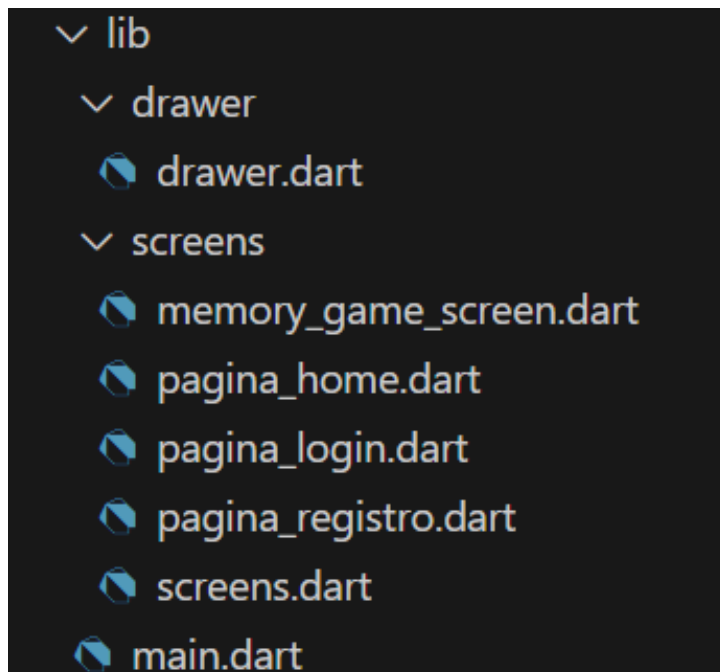
### 3.a Tecnologías elegidas

- **Flutter:** Framework para desarrollo multiplataforma.
- **Firebase:** Para autenticación y almacenamiento de datos de usuario.
- **Deck of Cards API:** Para obtener imágenes dinámicas de cartas.

### 3.b Arquitectura

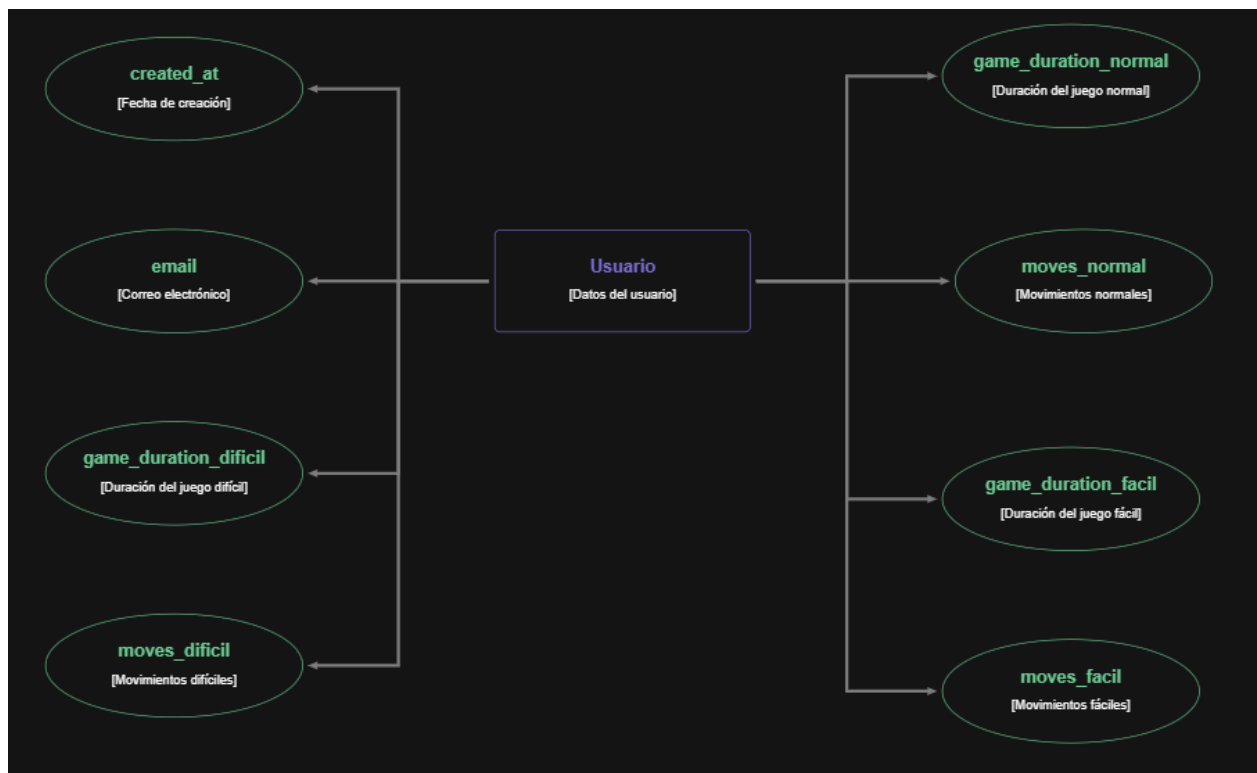
- **Frontend:** utilizando **widgets personalizados** para la interfaz
- **Backend:** **Firebase Authentication** para el inicio de sesión y **Firestore** para almacenar los registros de juego.
- **Lógica del juego:** Modelo-Vista-Controlador (**MVC**).

### 3.c Diagrama de clases



1. **main.dart** inicia la aplicación.
2. **screens.dart** facilita la importación de pantallas.
3. **drawer.dart** es un menú lateral que proporciona al usuario una forma rápida de navegar entre estadísticas del usuario o la opción de cerrar sesión.
4. **memory\_game\_screen.dart** es donde ocurre la mayor parte de la interacción del usuario con el juego. Esta pantalla usa el estado para gestionar la lógica del juego, el temporizador, y las cartas.
5. **pagina\_home.dart** es la pantalla principal donde seleccionas la dificultad de la partida.
6. **pagina\_login.dart** y **pagina\_registro.dart** son pantallas que gestionan el proceso de autenticación: una para el inicio de sesión y otra para el registro de nuevos usuarios.

### 3.d Diagrama E/R



### 3.e Consideraciones técnicas

- **Optimización:** Manejo eficiente del estado y actualización de la UI para mantener un rendimiento fluido.
- **Seguridad:** Restricciones en Firestore para evitar manipulación de datos.
- **Usabilidad:** Diseño adaptado tanto para teléfonos como para tablets.

## 4. Documentación de la solución

El código fuente está disponible en el siguiente repositorio:

<https://github.com/jorgebenitezz05/Proyecto-Final-Flutter.git>

## 5. Enlaces de interés

Firestore: <https://firebase.google.com/>

Deck of Cards API: <https://deckofcardsapi.com/>