

# Manual Técnico **QuizTime**

## *-Índice*

<b>-Introducción.....</b>	<b>2</b>
<b>-Análisis del problema.....</b>	<b>2</b>
<b>-Diseño de la solución.....</b>	<b>4</b>
<b>-Documentación.....</b>	<b>5</b>

## *-Introducción*

Este proyecto surge con el objetivo de combinar entretenimiento y aprendizaje, ofreciendo una experiencia entretenida que desafía la mente de los usuarios mientras se divierten. QuizTime es una aplicación de trivia interactiva basada en Flutter que integra el consumo de una API pública y utiliza Firebase para gestionar la autenticación y el almacenamiento de datos a tiempo real de forma global.

## *-Análisis del problema*

Muchas personas encuentran poco efectivas las formas tradicionales de estudio, lo que puede llevar a la frustración y la creencia de que no son capaces de aprender. Sin embargo, el problema no siempre radica en la falta de capacidad, sino en no haber encontrado el método de estudio adecuado. QuizTime puede hacer al usuario darse cuenta de que su manera de estudio es un trivia sobre el tema en cuestión. También es una opción ya no solo para la gente que quiera aprender sino también para gente que quiera pasárselo bien.

El público de esta aplicación va dirigido hacia todo el mundo pero principalmente para: gente que le gustan los videojuegos, gente que le gustan la informática, gente con TDA, gente que le gusta la historia y para gente que le gustan los quiz/trivias.

## Análisis DAFO:

Plataforma interactiva y divertida: Permite aprender jugando, lo que mejora la retención del conocimiento.

## FORTALEZAS

- Uso de tecnologías modernas desarrolladas en Flutter, lo que facilita la compatibilidad multiplataforma (Android, iOS y Web).
- Base de datos en la nube gracias a Firebase Firestore, la aplicación almacena y gestiona datos en tiempo real de forma global.
- Diseño divertido y atractivo.
- Permite aprender jugando, lo que mejora la retención del conocimiento.

## DEBILIDADES

- La aplicación necesita internet para funcionar.
- Las preguntas pueden ser difíciles incluso en los modos más fáciles.
- Los usuarios no pueden introducir sus propias preguntas

## AMENAZAS

- Existen otras apps de trivia y aprendizaje como Kahoot!, Preguntados o Duolingo
- Si la api deja de funcionar la aplicación también
- Si firebase cambia sus políticas puede causar problemas

## OPORTUNIDADES

- Cada vez más personas buscan alternativas digitales para aprender de forma entretenida.
- Gracias a Flutter puedo con el mismo código expandirme en mas dispositivos (Android, IOS, web..)

## *-Diseño de la solución*

La solución es hacer juego desafiante con diferentes dificultades. Para que el juego no sea aburrido tienes solo 10 segundos para responder cada pregunta antes de perder la partida. Además gracias al sistema de puntos el juego te premia al elegir más dificultad haciéndolo todavía más competitivo.

Los puntos que se obtiene en el intento se calcula mediante esta simple ecuación  $[n \text{ respuestas correctas} * \text{índice de dificultad}]$ .

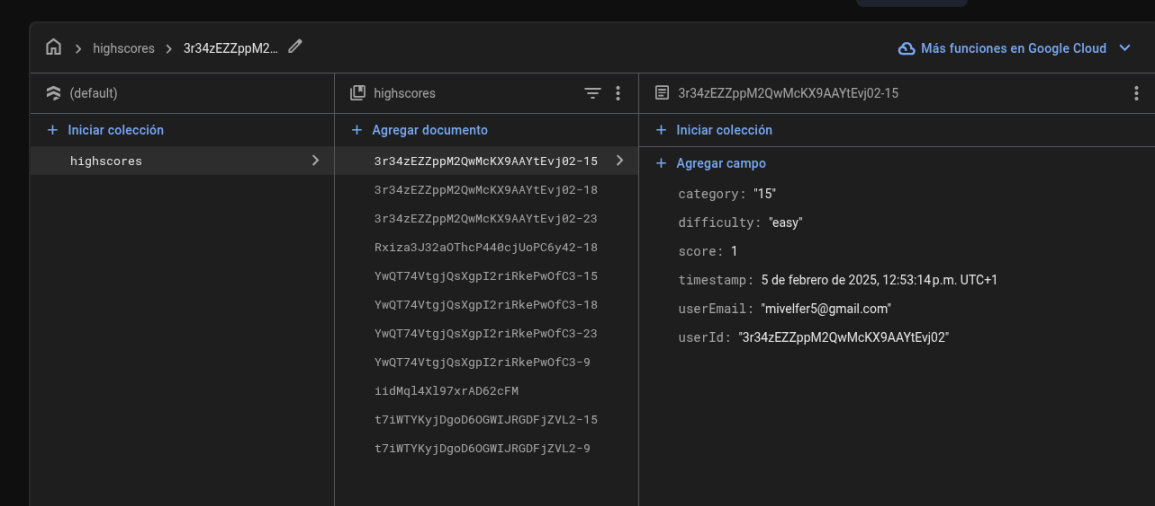
El índice de dificultad es un número oculto que se obtiene según la dificultad obtenida:

- Si has elegido Easy el índice es 1
- Si has elegido Medium el índice es 1,5
- Si has elegido Hard el índice es 2

Podrás compararte con los demás usuarios en la pantalla “HighScore”.

Las tecnologías elegidas son Flutter y Firebase. Con el frontal de Flutter he conseguido hacer un interfaz agradable y divertida además de implementar toda la lógica del programa. Con Firebase, he logrado implementar la persistencia de datos, permitiendo el registro y autenticación de usuarios, así como el almacenamiento de sus mejores puntuaciones según la categoría seleccionada.

Al ser firebase una BBDD NoSQL no tengo tablas entidad relación. Tengo colecciones y documentos que tienen esta estructura:



highscores > 3r34zEZZppM2...			Más funciones en Google Cloud		
(default)	highscores			3r34zEZZppM2QwMcKX9AAyEvj02-15	
+ Iniciar colección	+ Agregar documento			+ Iniciar colección	
highscores >	3r34zEZZppM2QwMcKX9AAyEvj02-15 >			+ Agregar campo	
	3r34zEZZppM2QwMcKX9AAyEvj02-18			category: "15"	
	3r34zEZZppM2QwMcKX9AAyEvj02-23			difficulty: "easy"	
	Rx1za3J32a0ThcP448cjUoPC6y42-18			score: 1	
	YwQT74VtgjQsXgpI2r1RkePwOfc3-15			timestamp: 5 de febrero de 2025, 12:53:14p.m. UTC+1	
	YwQT74VtgjQsXgpI2r1RkePwOfc3-18			userEmail: "mivelfer5@gmail.com"	
	YwQT74VtgjQsXgpI2r1RkePwOfc3-23			userId: "3r34zEZZppM2QwMcKX9AAyEvj02"	
	YwQT74VtgjQsXgpI2r1RkePwOfc3-9				
	i1dMq14x197xrAD62cFM				
	t71WTYKyjDgoD6OGWIJRGDFjZVL2-15				
	t71WTYKyjDgoD6OGWIJRGDFjZVL2-9				

## *-Documentación*

La aplicación es open source y puedes leer su código en este repositorio de GitHub: <https://github.com/mirip1/flutter>

He utilizado widgets que no hemos visto en clase:

LinearProgressIndicator - documentación Flutter: [HazClickAqui](#)

SafeArea - documentación Flutter: [HazClickAqui](#)

Y he utilizado una librería lucide\_icons para casi todos los iconos.