

## **Doble Titulación en Grado en Informática de Gestión y Sistemas de Información & Grado en Diseño y Producción de Videojuegos**

### **Explorando la interactividad: Virtualización del clásico juego de mesa ¿Quién es quién?**

#### **Memoria**

**Aida Chavero Pérez**  
**Tutora: Laura Rivera Sánchez**

**Curso 2022/2023**



## **Dedicatoria**

A mi familia, por apoyarme durante todos estos años y ayudarme a tomar las decisiones que han definido el rumbo de mi vida.

A mis amigos, por estar a mi lado en los momentos cruciales y animarme a seguir adelante.



## **Agradecimientos**

Agradezco a mi tutora la ayuda durante el proceso de desarrollo y los ánimos y positivismo cuando más lo necesitaba.

A las personas que me han acompañado estos meses, por la paciencia.



## **Abstract**

This project consists of creating a mobile application of the traditional board game "Guess Who?". It has been designed and developed to be a faithful reflection of the original game, providing users with an interactive and entertaining digital experience. The application has been developed using Flutter and Back4App, which has allowed the development of a multiplatform product capable of handling a wide range of functionalities. This work has provided valuable experience in mobile application development.

## **Resum**

Aquest projecte consisteix en la creació d'una aplicació mòbil del joc de taula tradicional "Qui és qui?". Ha estat dissenyat i desenvolupat per ser un reflex fidel del joc original, proporcionant als usuaris una experiència digital interactiva i entretinguda. L'aplicació s'ha desenvolupat utilitzant Flutter i Back4App, la qual cosa ha permès el desenvolupament d'un producte multiplataforma capaç de gestionar una àmplia gamma de funcionalitats. Aquest treball ha aportat una valuosa experiència en el desenvolupament d'aplicacions mòbils.

## **Resumen**

Este proyecto consiste en la creación de una aplicación móvil del juego de mesa tradicional "¿Quién es Quién?". Se ha diseñado y desarrollado para ser un fiel reflejo del juego original, proporcionando a los usuarios una experiencia digital interactiva y entretenida. La aplicación ha sido desarrollada utilizando Flutter y Back4App, lo que ha permitido el desarrollo de un producto multiplataforma capaz de manejar una amplia gama de funcionalidades. Este trabajo ha aportado una valiosa experiencia en el desarrollo de aplicaciones móviles.



# Índice

Índice de figuras .....	III
Glosario de términos .....	V
1. Introducción.....	1
1.1. Motivación.....	1
1.2. Objeto del proyecto .....	1
1.3. Estructura de la memoria .....	2
2. Marco teórico y análisis de referentes .....	3
2.1. Contexto.....	3
2.2. Antecedentes y referentes .....	3
2.3. Propuesta .....	8
3. Objetivos y alcance .....	9
3.1. Objetivos.....	9
3.2. Alcance .....	9
4. Metodología de trabajo.....	11
5. Desarrollo .....	13
5.1. Requerimientos.....	13
5.1.1. Requerimientos de negocio.....	13
5.1.2. Requerimientos funcionales.....	13
5.1.3. Requerimientos no funcionales.....	14
5.1.4. Requerimientos tecnológicos.....	14
5.2. Tecnologías.....	15
5.3. Necesidades de información.....	16
5.4. Implementación .....	17
5.4.1. Diseño de la solución.....	17
5.4.2. Estructura del proyecto .....	19
5.4.3. Base de datos .....	20
5.4.4. Frontend .....	21
5.4.5. Backend .....	27
6. Test .....	35
6.1. Test de usuario.....	35
6.1.1. Descripción de una muestra ideal .....	35
6.1.2. Descripción y resultados de los test de usuario .....	35
6.2. Test de aceptación .....	37
6.2.1. Descripción y resultados de los test de aceptación.....	37
7. Conclusiones .....	41
8. Posibles ampliaciones.....	43

9. Bibliografía .....45

## Índice de figuras

Figura 2.1. Juego de mesa tradicional ¿Quién es quién?. Fuente: Hasbro, 2005.....	3
Figura 2.2. Aplicación Guess who I am. Fuente: GooglePlay, 2019. ....	4
Figura 2.3. Aplicación Guess who I am 2. Fuente: GooglePlay, 2021. ....	4
Figura 2.4. Aplicación ¿QUIÉN ES QUIÉN?. Fuente:GooglePlay, 2020.....	5
Figura 2.5. Aplicación Whooo?. Fuente: GooglePlay, 2020. ....	5
Figura 2.6. Aplicación Chess Online. Fuente: GooglePlay, 2013.....	6
Figura 2.7. Aplicación Parchisi STAR Online. Fuente: GooglePlay, 2017.....	7
Figura 2.8. Aplicación Akinator. Fuente: GooglePlay, 2012.....	7
Figura 5.1. Diagrama de arquitectura. Fuente: Elaboración propia, 2023. ....	18
Figura 5.2. Diagrama de clases. Fuente: Elaboración propia, 2023.....	19
Figura 5.3. Diagrama Entidad-Relación. Fuente: Elaboración propia, 2023. ....	20
Figura 5.4. Wireframe de la aplicación. Fuente: Elaboración propia, 2023. ....	22
Figura 5.5. Prototipo inicial de la aplicación. Fuente: Elaboración propia, 2023.....	23
Figura 5.6. Prototipo final de la aplicación. Fuente: Elaboración propia, 2023.....	24
Figura 5.7. Diagrama de Flujo de Acceso y Registro. Fuente: Elaboración propia, 2023.....	25
Figura 5.8. Diagrama de Flujo de Juego. Fuente: Elaboración propia, 2023.....	26
Figura 5.9. Diagrama de Flujo de Instrucciones. Fuente: Elaboración propia, 2023.....	26
Figura 5.10. Diagrama de Flujo de Configuración. Fuente: Elaboración propia, 2023.....	26
Figura 5.11. Diagrama de Flujo de Perfil. Fuente: Elaboración propia, 2023. ....	27
Figura 5.12. Diagrama de Flujo de Ranking. Fuente: Elaboración propia, 2023.....	27



## Glosario de términos

API	Application Programming Interface. Mecanismo que permite a dos componentes de software comunicarse entre sí.
IDE	Integrated Development Environment. Aplicación de software que ayuda a los programadores a desarrollar código de software de manera eficiente.
DDD	Domain Driven Design. Técnica de diseño de software que se centra en el análisis y diseño del dominio del problema.



# 1. Introducción

## 1.1. Motivación

La era digital ha transformado la vida y el día a día de las personas trasladando las tareas cotidianas al cloud y digitalizando temas como la comunicación, el entretenimiento, la educación, etc. Centrándose en el tema del entretenimiento, se ha observado un gran aumento en la oferta de juegos en el mercado de aplicaciones móvil, incluyendo aplicaciones que simulan populares juegos de mesa tradicionales. Este aumento se intuye que es debido a la evolución y transformación en la forma de juego que se produce actualmente, dado que la población más joven prefiere jugar en un móvil o tableta que en un tablero tradicional [1].

El hecho de formar parte del segmento de población que consume más juegos para móvil permite tener una percepción directa de la situación del mercado y de las demandas de los usuarios. Tras analizar dicho mercado, se observa una diferencia notable entre las aplicaciones dedicadas a los juegos de mesa más populares y los juegos menos populares. Los primeros, siendo el ajedrez, el parchís o la oca, cuentan con abundancia de aplicaciones; mientras que los segundos apenas cuentan con un par de aplicaciones que los simulen. En relación con el juego ¿Quién es quién?, se observa cierta escasez en el número de aplicaciones que lo simulen, aún y siendo este uno de los juegos más populares en la infancia de los principales consumidores de juegos móvil. El poder contar con una visión en primera persona de las necesidades del segmento de clientes objetivo permite tomar decisiones con una seguridad mayor, decisiones como la elección del juego tradicional a digitalizar.

Una vez escogido el juego alrededor del cual se desarrolla el proyecto, se realiza un análisis de las aplicaciones que lo simulan. Este análisis permite concluir que hay una gran necesidad de innovación y actualización en las aplicaciones existentes, las cuales están desfasadas tanto gráficamente como en temas de diseño e interacción con los usuarios.

Los tres aspectos mencionados anteriormente - el creciente interés en los juegos móviles, la escasez de aplicaciones que simulen el juego seleccionado y la necesidad de innovación y actualización en las existentes - constituyen la motivación y justificación para el desarrollo de este proyecto.

## 1.2. Objeto del proyecto

El objeto del proyecto a realizar es el diseño y desarrollo de una aplicación multiplataforma que simule el juego de mesa tradicional ¿Quién es quién?. La aplicación debe ofrecer tres métodos de juego a los usuarios:

- Competir contra una inteligencia artificial.
- Competir contra un usuario aleatorio en una partida pública.
- Competir contra un usuario en una partida privada.

### 1.3. Estructura de la memoria

La memoria que documenta el proyecto a realizar y su proceso de desarrollo se encuentra dividida en ocho secciones, excluyendo la sección de introducción al proyecto. Cada una de las secciones tiene el objetivo de documentar y detallar una de las cuestiones relacionadas con el proyecto, de manera que en su conjunto se documente el proceso completo de ideación, planificación, investigación y desarrollo del proyecto.

**Marco teórico y Análisis de referentes.** La primera sección del documento presenta el contexto en el que se plantea y desarrolla el proyecto, junto con los antecedentes que le preceden y los referentes analizados para el desarrollo del producto final. Se detalla también la propuesta de proyecto que se quiere realizar.

**Objetivos y alcance.** La segunda sección del documento presenta los objetivos que se quieren lograr con el desarrollo del proyecto y el producto final a entregar, junto con el alcance al que se propone llegar.

**Metodología de trabajo.** La tercera sección del documento describe las metodologías y métodos de trabajo a seguir para llevar a cabo el diseño y desarrollo del proyecto.

**Desarrollo.** La cuarta sección del documento documenta e informa de forma exhaustiva y detallada sobre el proceso completo de desarrollo del proyecto y el producto final que se desea obtener. Se describen en profundidad las distintas acciones llevadas a cabo y los posibles problemas y sus soluciones que hayan podido surgir en el proceso. Se indican también las tecnologías usadas en cada fase del desarrollo y las necesidades de información que se requieren para llevar a cabo el proyecto. Finalizado el proceso de desarrollo, se muestran los resultados obtenidos en los test de usuario y test de aceptación realizados.

**Conclusiones.** La sexta sección del documento analiza los resultados obtenidos durante el desarrollo del proyecto y se concluye conforme a ellos.

**Posibles ampliaciones.** La séptima sección del documento plantea posibles ampliaciones a añadir al proyecto y al producto final en caso de una futura continuación del proyecto posterior a la entrega del proyecto presentado.

## 2. Marco teórico y análisis de referentes

### 2.1. Contexto

El avance tecnológico ha cambiado la vida de las personas al trasladar las actividades cotidianas al cloud y digitalizar prácticamente todos los aspectos de sus vidas. En particular, en el ámbito del entretenimiento, se ha registrado un incremento exponencial en la disponibilidad de juegos en el mercado de aplicaciones móviles, incluyendo versiones digitales de juegos de mesa tradicionales. Este crecimiento se debe a la transformación en la forma de juego de hoy en día, sobre todo por parte de la población más joven. Se ha observado que los jóvenes y niños prefieren jugar en un dispositivo móvil y de forma online en lugar de jugar en un tablero tradicional de forma presencial.

Se estima que los jóvenes de entre 12 y 24 años, siendo este el público objetivo del proyecto, dedican de 2 a 6 horas al día al uso de dispositivos electrónicos, como smartphones, tabletas y computadoras [2]. Parte de este tiempo se dedica al consumo de juegos en línea, por lo que una aplicación como el producto a desarrollar durante el proyecto podría llegar a ser un producto popular entre el sector más joven de la población.

### 2.2. Antecedentes y referentes

El mercado móvil cuenta con multitud de aplicaciones que simulen juegos de mesa tradicionales como el propuesto a desarrollar en el proyecto o juegos no tradicionales que comparten características similares con el juego propuesto.

El principal antecedente a la aplicación propuesta como producto final del proyecto es el propio juego de mesa ¿Quién es quién? [3].



Figura 2.1. Juego de mesa tradicional ¿Quién es quién?. Fuente: Hasbro, 2005.

A continuación, se muestran aplicaciones en el mercado móvil que simulan el juego ¿Quién es quién?. Estas aplicaciones se consideran antecedentes a la aplicación a desarrollar.

### Guess who I am – Board games de la compañía Offs Games [4]

En esta aplicación, las partidas están formadas por dos jugadores. Las partidas pueden jugarse contra otro usuario aleatorio de forma online o contra un usuario de forma local.

La mecánica de preguntas se hace mediante la elección de una pregunta entre un conjunto de preguntas predeterminadas, mostradas como opciones durante el turno del jugador.



Figura 2.2. Aplicación *Guess who I am*. Fuente: GooglePlay, 2019.

### Guess who I am 2 – Board games de la compañía Offs Games [5]

En este juego, las partidas están formadas por dos jugadores, ya sea por dos usuarios o por un usuario y una inteligencia artificial.

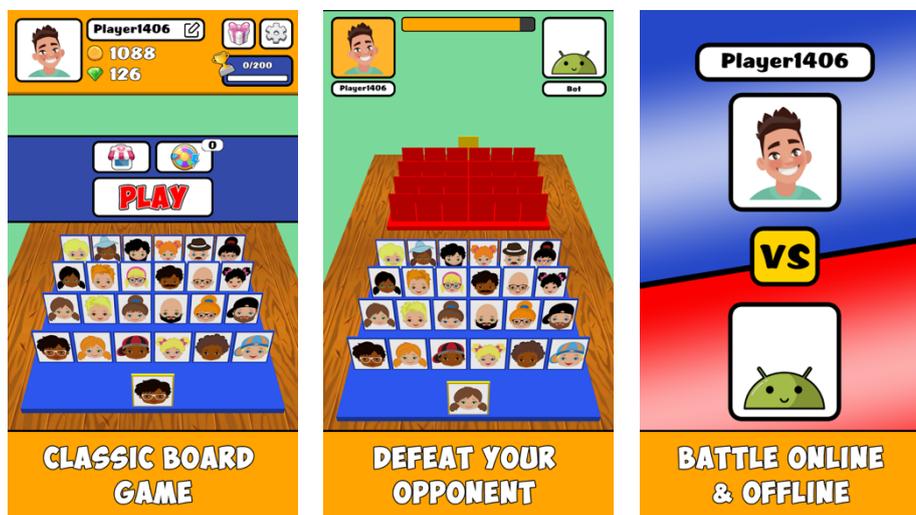


Figura 2.3. Aplicación *Guess who I am 2*. Fuente: GooglePlay, 2021.

### ¿QUIÉN ES QUIÉN? de la compañía Didacta [6]

En este juego, las partidas están formadas por dos jugadores, ya sea por dos usuarios o por un usuario y una inteligencia artificial.

La mecánica de preguntas se hace mediante la elección de una pregunta entre un conjunto de preguntas predeterminadas, mostradas como opciones durante el turno del jugador.



Figura 2.4. Aplicación ¿QUIÉN ES QUIÉN?. Fuente: GooglePlay, 2020.

### Whooo? de la compañía Yso Corp [7]

En este juego, las partidas están formadas por dos jugadores, un usuario y una inteligencia artificial.

La mecánica de preguntas se hace mediante la elección de una pregunta entre un conjunto de preguntas predeterminadas, mostradas como opciones durante el turno del jugador.

El juego incluye un sistema de vidas que corresponde a cada turno jugable por el jugador. Las vidas disminuyen tras cada turno si no se adivina el personaje del contrincante. En caso de agotar todas las vidas disponibles, el usuario pierde la partida.



Figura 2.5. Aplicación Whooo?. Fuente: GooglePlay, 2020.

A continuación, se muestra un análisis de los juegos con mayor popularidad en el mercado móvil que comparten características y similitudes con la aplicación a desarrollar propuesta.

### Chess Online de la compañía Chess Friends [8]

La aplicación Chess Online simula el juego tradicional de tablero Ajedrez.

En este juego, las partidas están formadas por dos jugadores, ya sea por dos usuarios o por un usuario y una inteligencia artificial.

La aplicación ofrece varias posibilidades a la hora de elegir contrincante. Las partidas pueden jugarse contra una inteligencia artificial, en salas públicas contra otros usuarios aleatorios o en salas privadas contra otro usuario.

El tablero de juego y las figuras están modeladas en 3D y se ha animado su movimiento.

La aplicación incluye la opción de registrarse mediante correo electrónico.



Figura 2.6. Aplicación Chess Online. Fuente: GooglePlay, 2013.

### Parchisi STAR Online de la compañía Gameberry Labs [9]

La aplicación Parchisi STAR Online simula el juego tradicional de tablero Parchís.

En este juego, las partidas están formadas entre dos y cuatro jugadores, ya sea por usuarios o por un usuario y una inteligencia artificial.

La aplicación ofrece varias posibilidades a la hora de elegir contrincante. Las partidas pueden jugarse contra una inteligencia artificial, en salas públicas contra otros usuarios aleatorios o en salas privadas contra otros usuarios. En partidas de cuatro jugadores, la aplicación ofrece la posibilidad de jugar en equipos de dos jugadores.

La aplicación incluye una tabla de clasificación de jugadores según las monedas que han ganado en juegos públicos.

La aplicación incluye la opción de registrarse mediante correo electrónico.



Figura 2.7. Aplicación Parchisi STAR Online. Fuente: GooglePlay, 2017.

### Akinator de la compañía Elokence SAS [10]

La aplicación Akinator simula un juego en el que un genio puede adivinar el personaje en el que está pensando el jugador.

En este juego, las partidas están formadas dos jugadores, más concretamente por un usuario y una inteligencia artificial.

La lógica de juego de Akinator es similar a la lógica de juego de ¿Quién es quién?. El objetivo de ambos juegos es adivinar un personaje a partir de preguntas.

La mecánica principal se basa en preguntas lanzadas por una inteligencia artificial a las que el usuario puede responder “Sí” / “No” / “Probablemente sí” / “Probablemente no” / “No lo sé” con el objetivo de que el genio adivine el personaje en el que el usuario está pensando.

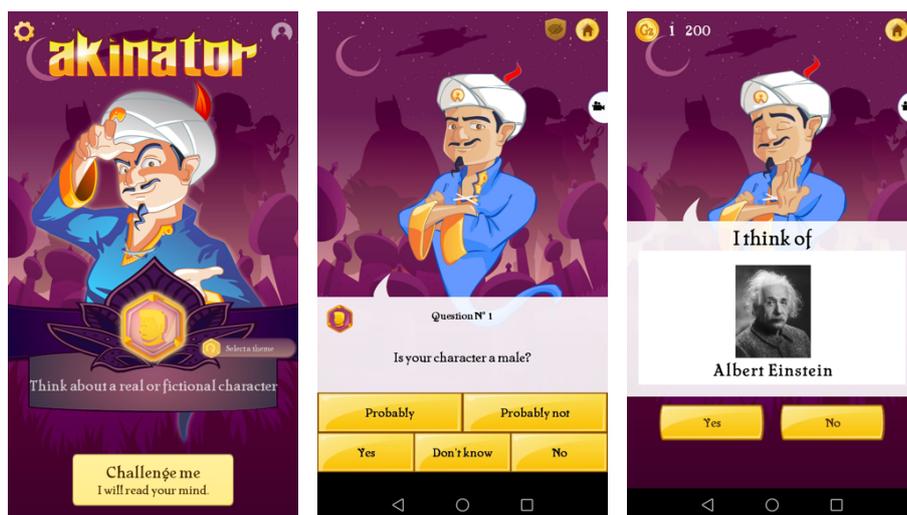


Figura 2.8. Aplicación Akinator. Fuente: GooglePlay, 2012.

## 2.3. Propuesta

El proyecto de digitalización del juego de mesa tradicional ¿Quién es quién?, con el consecuente desarrollo de una aplicación móvil como producto final, se propone a partir del análisis de mercado realizado en el que se muestra la falta de aplicaciones que repliquen el juego escogido.

Se propone como valor final a entregar una aplicación multiplataforma que permita a los usuarios de dispositivos móviles y tabletas poder disfrutar del juego ¿Quién es quién?, el cual ha formado parte de la infancia de gran parte del público al que se dirige la aplicación.

La aplicación se ha pensado de forma que incluya distintas funcionalidades para satisfacer el mayor número de necesidades de los usuarios. Las funcionalidades más relevantes son los métodos de juego que se ofrecen, permitiendo jugar contra una inteligencia artificial, jugar contra un usuario aleatorio en una partida pública o jugar contra un usuario en una partida privada. Una funcionalidad que realzar se trata de la posibilidad que se ofrece a los usuarios de poder escoger entre distintas interfaces para que la aplicación sea más accesible para ellos y se adapte a posibles discapacidades que puedan sufrir.

En resumen, este proyecto busca llenar un vacío identificado en el mercado digitalizando el tradicional juego de mesa ¿Quién es quién? en una aplicación móvil multiplataforma. No solo se replica la experiencia del juego original, sino que también se mejoran las funcionalidades, permitiendo enfrentarse a una inteligencia artificial, competir con otros jugadores y personalizar la interfaz para mejorar la accesibilidad. El objetivo es revivir este popular juego, capturando el interés tanto de los fans originales como de una nueva generación de jugadores en el entorno digital.

## 3. Objetivos y alcance

### 3.1. Objetivos

El objetivo principal del proyecto es digitalizar un juego de mesa tradicional, concretamente ¿Quién es quién? de la compañía Hasbro, Inc. La virtualización del juego se propone mediante el desarrollo de una aplicación móvil que adapte el juego clásico a las formas de juego actual. El producto final a entregar es la aplicación móvil y tiene como objetivo suplir la falta de aplicaciones que simulen el juego ¿Quién es quién? presente en el mercado móvil.

A continuación, se especifican los objetivos a cumplir por el producto final:

- La aplicación debe poder usarse en distintos sistemas operativos, por lo que debe ser multiplataforma, además de asegurar su correcto funcionamiento en todos ellos.
- La aplicación debe ser atractiva para los usuarios y captar su atención, por lo que debe contar con una parte visual fresca e innovadora.
- La aplicación debe tener una interfaz intuitiva y adaptable a las posibles discapacidades de los usuarios, por lo que se deben ofrecer varias opciones de interfaz a elegir por los usuarios.
- El contenido visual del juego, concretamente las tarjetas de personajes, debe ofrecer representación de todos los posibles usuarios, contando con personajes de distintas culturas e ideologías.
- Los usuarios deben poder registrarse usando una cuenta de Google, por lo que la aplicación deberá tener integrados los servicios de Google.
- La aplicación debe ofrecer tres modos de juego, siendo estos competir contra una inteligencia artificial, competir contra un usuario aleatorio en una partida pública y competir contra un usuario en una partida privada.
- Las partidas y sus respectivos datos deben gestionarse según el tipo de partida.
- Los datos de los usuarios y sus partidas deben gestionarse y relacionarse correctamente en una base de datos.
- Los usuarios deben poder ver una tabla de clasificación donde aparezcan todos los usuarios de la aplicación ordenados según su número de partidas públicas ganadas.
- La inteligencia artificial debe ser capaz de comprender, analizar y dar una respuesta válida a la entrada de texto de los usuarios.
- El servidor para partidas online debe ofrecer una conexión estable y una jugabilidad adecuada, sin tiempos de espera ni desconexiones que interrumpan las partidas.

### 3.2. Alcance

La aplicación a desarrollar como producto final del proyecto planea cubrir un amplio abanico de funcionalidades, nombradas en el apartado anterior, aun así, su alcance está limitado en los siguientes aspectos:

- La inteligencia artificial se limita a responder sólo preguntas relacionadas con el juego y, específicamente, con las características descriptivas de los personajes. Además, se limita a responder sólo en el idioma seleccionado en el menú de configuración, siendo el idioma por defecto el español.
- La inteligencia artificial no se entrena con las preguntas recibidas por parte de los usuarios, esto se ha decidido así para evitar el uso de lenguaje inapropiado o inofensivo y filtrar la información.
- Las respuestas de los usuarios están restringidas a “Sí” o “No” para evitar ambigüedades.
- Las partidas sólo pueden estar formadas por un máximo de dos jugadores, tanto en partidas públicas contra usuarios aleatorios como en partidas privadas.

## **4. Metodología de trabajo**

El proyecto a realizar se llevará a cabo usando una metodología de trabajo Agile [11]. Esta metodología es de las más usadas en desarrollo de software y se adecua al plan de desarrollo del proyecto propuesto.

Se ha escogido un método de trabajo Agile dado que “se trata de una metodología iterativa, lo que permite realizar entregas cíclicas en las que se entrega valor constantemente y se cuenta con un producto funcional durante todo el proceso de desarrollo. En dichas entregas se realizan todas las fases del ciclo, desde análisis de requisitos y diseño hasta verificación y entrega de valor.” [11]

Se plantea una planificación siguiendo la metodología Scrum [12] aplicada a Agile, adaptada a las necesidades del proyecto y el equipo de desarrollo. El equipo de desarrollo está formado únicamente por el desarrollador. Siguiendo la metodología escogida, “el tiempo de desarrollo se divide en sprints. Antes del inicio de cada sprint, se escogen las tareas que se realizarán durante el sprint, basándose en el esfuerzo necesario para realizar cada tarea. Al finalizar el sprint, se reflexiona sobre cómo se ha desarrollado dicho sprint y el trabajo que se ha realizado, y se evalúan las posibles mejoras que podrían implementarse en la siguiente iteración, en caso de haberlas.” [12]



## 5. Desarrollo

### 5.1. Requerimientos

El producto final a entregar debe satisfacer las expectativas de los usuarios o clientes y de los interesados, tanto internos como externos, por lo que se deben cumplir los objetivos detallados en la definición del proyecto.

Lograr los objetivos del proyecto implica que el producto debe cumplir ciertos requerimientos [14], considerando estas las condiciones o capacidades que debe tener el producto para satisfacer las especificaciones propuestas.

#### 5.1.1. Requerimientos de negocio

“Los requerimientos de negocio son aquellos que describen los objetivos y resultados esperados del desarrollo del proyecto.” [14]

Los requerimientos de negocio que debe cumplir la aplicación son los siguientes:

- Digitalizar el juego de mesa tradicional ¿Quién es quién?.
- Suplir la falta de aplicaciones que simulen el juego ¿Quién es quién? presente en el mercado móvil.

#### 5.1.2. Requerimientos funcionales

“Los requerimientos funcionales son aquellos que describen de manera explícita las capacidades y los comportamientos que debe tener una solución, junto con qué información debe manejar.” [14]

Los requerimientos funcionales que debe cumplir la aplicación son los siguientes:

- Recoger, tratar y almacenar datos internos.
- Recoger, tratar y almacenar datos externos.
- Relacionar los datos internos y externos en una base de datos relacional.
- Desplegar y mantener activo un servidor.
- Mantener un control de los usuarios.
- Mantener un control de las partidas jugadas.
- Gestionar el estado de las partidas.
- Clasificar a los usuarios según su desempeño.
- Permitir una jugabilidad mecánica e interactiva que transmita a los usuarios una experiencia de juego satisfactoria.
- Permitir competir contra una inteligencia artificial capaz de comprender y analizar el lenguaje natural.

- Permitir competir contra un usuario aleatorio en partidas públicas o contra un usuario en partidas privadas mediante el ingreso de un código de partida.
- Permitir realizar la mecánica principal del juego mediante la entrada de texto por parte de los usuarios.
- Proveer de distintas interfaces adaptadas a las necesidades de los usuarios.
- Proporcionar feedback a los usuarios de los eventos producidos en la aplicación a través de estímulos visuales o auditivos.

### 5.1.3. Requerimientos no funcionales

“Los requerimientos no funcionales son aquellos que describen las condiciones necesarias para el correcto funcionamiento de la solución, junto con los requisitos de calidad que esta debe cumplir.” [14]

Los requerimientos no funcionales que debe cumplir la aplicación son los siguientes:

- Los datos deben actualizarse para todos los usuarios en menos de 10 segundos.
- El servidor debe estar activo para todos los usuarios durante las 24 horas del día y los 7 días de la semana, exceptuando en caso de mantenimiento.
- Las partidas deben desarrollarse de forma fluida y sin interrupciones.
- La inteligencia artificial debe comprender la entrada de datos de los usuarios y responder de forma coherente y correcta.
- La interfaz de la aplicación debe ser clara y sencilla de usar para usuarios no experimentados.
- La interfaz de la aplicación debe ofrecer adaptabilidad a aquellos usuarios que lo necesiten.
- La aplicación debe ser multiplataforma y asegurar su correcto funcionamiento en los distintos dispositivos en los que se ofrece.

### 5.1.4. Requerimientos tecnológicos

“Los requerimientos tecnológicos son aquellos que describen los aspectos técnicos que deben cumplirse para el correcto uso y funcionamiento de la solución.” [14]

Los requerimientos tecnológicos de los que requiere la aplicación en caso de querer ser usada por un usuario son los siguientes:

- El usuario debe disponer de un dispositivo móvil o tableta que soporte la aplicación, siendo Android 5.0 y iOS 14 las versiones mínimas de los sistemas operativos soportados. Además, el dispositivo debe disponer del almacenamiento necesario para la instalación de la aplicación.
- El dispositivo debe estar conectado a Internet para realizar la descarga de la aplicación. Se debe estar conectado también para poder acceder a la opción de jugar contra otros usuarios.

Los requerimientos tecnológicos de los que requiere el proyecto en caso de querer ser replicado o desarrollado son los siguientes:

- El desarrollador debe disponer de un dispositivo de trabajo con sistema operativo Windows 10 o macOS 10.15.7.
- El dispositivo de trabajo debe estar conectado a internet, tener el almacenamiento necesario para almacenar el proyecto y tener instalados los softwares necesarios para la codificación del proyecto y la creación del material visual de la aplicación.

## 5.2. Tecnologías

El proyecto se compone de un amplio número de tareas y funcionalidades, las cuales abarcan distintas áreas del diseño y el desarrollo de software. Partiendo de los requerimientos expuestos en la sección anterior, se detallan cada una de las tecnologías y softwares necesarios para el desarrollo de las distintas funciones de la aplicación, junto con los lenguajes de programación a usar:

- **Frontend.** El frontend de la aplicación se ha de desarrollar usando el IDE Visual Studio, complementado con el plug-in de Flutter [15]. El lenguaje de programación usado para el desarrollo es Dart [16].  
“Flutter es un kit de desarrollo de software de UI de código abierto creado por Google. Permite crear diseños atractivos para aplicaciones de forma rápida y sencilla. Se utiliza para desarrollar aplicaciones multiplataforma para Android y iOS.” [15]
- **Backend.** El backend de la aplicación se ha de desarrollar usando el IDE Visual Studio. El lenguaje de programación usado para el desarrollo es Dart. Inicialmente, se planteó el uso de Java para el desarrollo del backend, pero finalmente se ha desarrollado en Dart. La decisión de utilizar Dart para el backend se debe a su compatibilidad con Flutter y a su capacidad para facilitar el mantenimiento y la coherencia del código.
- **Inteligencia Artificial.** La inteligencia artificial no se ha implementado en esta versión del proyecto debido a desviaciones en la planificación, aunque se considera para futuras ampliaciones.  
La inteligencia artificial debe entrenarse mediante Azure [16]. Los recursos usados para el entrenamiento deben ser los provistos por Azure.  
“El servicio Análisis de texto de Azure permite analizar fácilmente los sentimientos, extraer frases clave y reconocer las entidades con nombre, solamente con unas pocas líneas de código.” [16]
- **Servidor.** El servidor y la API se ha de alojar en Back4App [17].  
“Back4App es una plataforma de desarrollo de aplicaciones para web y móvil que ofrece una solución backend escalable y fácil de usar para desarrolladores, lo que agiliza el proceso de desarrollo de la aplicación.” [17]
- **Base de datos.** La base de datos se ha de alojar en Back4App.  
Back4App utiliza PostgreSQL [18] para las bases de datos, permitiendo la implementación de un modelo de datos relacional.

- **Test.** El testeo de la aplicación y el correcto funcionamiento del servidor se ha de realizar con un grupo de posibles usuarios reales para obtener una retroalimentación directa y personalizada.
- **Prototipado.** El prototipado de la aplicación se ha de diseñar usando Figma [19]. Figma es una aplicación web colaborativa para el diseño de interfaces. El motivo de elección de Figma se debe a la experiencia del desarrollador con esta herramienta.
- **Diseño.** La interfaz de la aplicación y sus elementos, así como sus distintos estilos, se ha de diseñar usando Figma.
- **Ilustración.** Las ilustraciones de las tarjetas de personajes y el logo de la aplicación se han externalizado para esta versión del proyecto debido a desviaciones en la planificación. Sin embargo, se plantea la creación de ilustraciones propias para futuras ampliaciones, a fin de conseguir una mayor personalización y originalidad.
- **Modelado.** El diseño y modelado del tablero de juego y sus animaciones se ha de diseñar, modelar y animar usando Autodesk 3DS Max [20]. Autodesk 3DS Max es un programa profesional de gráficos 3D para la creación de modelos, animaciones 3D, juegos e imágenes.
- **Sonido.** Los sonidos incluidos en el juego deben externalizarse.

### 5.3. Necesidades de información

El proyecto a desarrollar se basa en el juego tradicional ¿Quién es quién?, un juego popularmente conocido con el cual el desarrollador está familiarizado en términos de lógica de juego y reglas. En lo que respecta al juego, no se requiere más información de la que ya se posee.

En relación con las tecnologías y lenguajes de programación usados en el desarrollo del proyecto, es necesario adquirir mayor conocimiento en torno a los siguientes aspectos:

- **Flutter.** “Flutter es un marco de desarrollo desarrollado por Google que utiliza Dart para crear aplicaciones multiplataforma con un solo código fuente. Este se compila a código nativo, lo que mejora el rendimiento en comparación con aplicaciones basadas en web-views. Además, Flutter no utiliza componentes nativos, sino que cuenta con sus propios elementos llamados "widgets", lo que garantiza que la apariencia de la aplicación sea consistente en cualquier dispositivo, independientemente de su sistema operativo o versión.” [15] Esto hace que su desarrollo sea cómodo y eficiente, lo que aumenta la viabilidad del proyecto.  
El desarrollador del proyecto carece de experiencia en el uso de Dart como lenguaje de programación para desarrollo de software. A pesar de esta falta de experiencia previa con el lenguaje, su aprendizaje es factible gracias a su similitud con otros lenguajes como JavaScript, Java y C++. Aprender Dart con un conocimiento previo en alguno de estos lenguajes solo toma unas horas.
- **Back4App.** “Back4App es una plataforma Backend as a Service que proporciona una solución fácil de usar y escalable para los desarrolladores. Utiliza PostgreSQL para la base de datos, permitiendo la implementación de un modelo relacional.” [17]

El desarrollador del proyecto carece de experiencia previa directa con Back4App, aún así, la familiaridad con los principios de las bases de datos relacionales y los servicios en la nube facilita su comprensión y uso.

- **Azure.** Azure es una plataforma que ofrece más de 200 productos y servicios en la nube, brindando todas las herramientas necesarias para el desarrollo del proyecto. Se propone su uso para la implementación de la inteligencia artificial en futuras versiones del proyecto.

El desarrollador del proyecto carece de experiencia previa con Azure, aún así, la familiaridad con las tecnologías en la nube y el aprendizaje rápido inherente a la formación en ingeniería informática permitirá su manejo.

Para el aprendizaje de cada una de estas tecnologías, el acceso a documentación oficial, tutoriales, foros de la comunidad, y otros recursos educativos es invaluable para adquirir la competencia necesaria para su uso efectivo en el desarrollo del proyecto.

## 5.4. Implementación

El diseño y desarrollo de la aplicación es un proceso que involucra la interacción y colaboración de distintos componentes para formar una solución completa. Estos componentes, aunque desarrollados individualmente con tecnologías específicas para cada uno, se comunican y colaboran entre sí para garantizar la funcionalidad completa del producto. La estructura de la implementación se divide principalmente en dos componentes clave: el frontend y el backend.

1. **Frontend:** El frontend de la aplicación, la interfaz con la que interactúan los usuarios, se ha construido utilizando Flutter y Dart. Durante el desarrollo del frontend, se ha utilizado una variedad de plugins para mejorar la funcionalidad y la experiencia del usuario.
2. **Backend:** El backend de la aplicación ha sido desarrollado usando Back4App. Esta plataforma utiliza Parse Server para gestionar la lógica de negocio de la aplicación y asegurar la integridad y seguridad de los datos del usuario. La gestión de los datos se realiza mediante PostgreSQL. El desarrollo del backend se ha centrado en garantizar la escalabilidad, la seguridad y el rendimiento óptimo de la aplicación.

A través de la colaboración efectiva entre estos componentes de frontend y backend, la aplicación es capaz de ofrecer a los usuarios una experiencia de juego enriquecedora y atractiva, manteniendo al mismo tiempo la seguridad y la privacidad de sus datos.

### 5.4.1. Diseño de la solución

Previamente al inicio del desarrollo, se realiza el diseño de la solución, en el que se determina como deben relacionarse los distintos servicios que la forman. El diagrama de arquitectura de la aplicación está diseñado para proporcionar una visión completa y coherente de la forma

en que los componentes del sistema interactúan entre sí y con los usuarios, así como la relación entre los servicios que forman parte del el frontend y el backend.

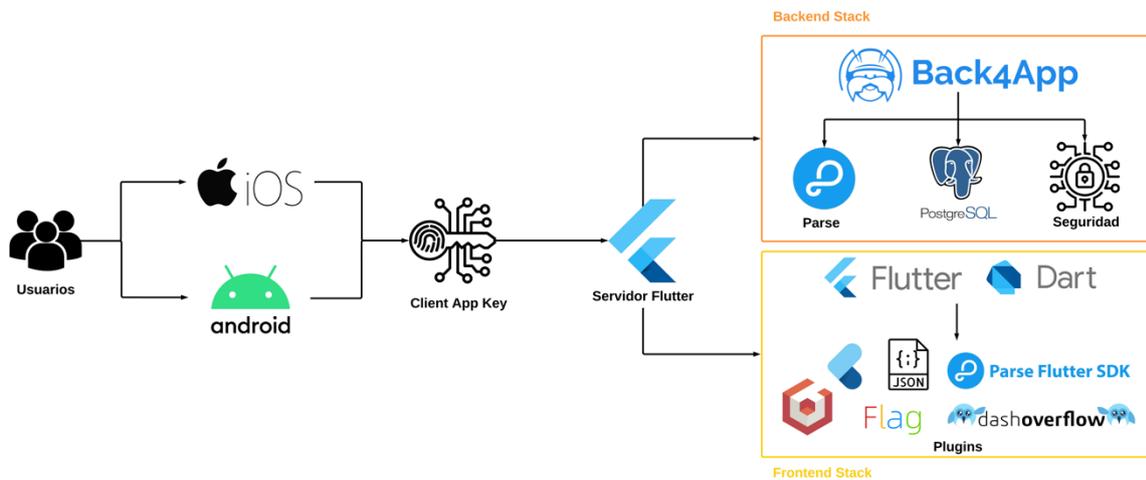


Figura 5.1. Diagrama de arquitectura. Fuente: Elaboración propia, 2023.

Los usuarios acceden a la aplicación desde sus dispositivos móviles, que pueden ser tanto dispositivos iOS como Android. La client app key proporciona a los usuarios la autorización necesaria para utilizar los servicios de frontend y backend de la aplicación.

A partir de aquí, la conexión se establece con el servidor de Flutter. Esta conexión sirve como un puente entre los usuarios y el frontend y el backend de la aplicación.

El frontend de la aplicación se ha construido utilizando Flutter y Dart. Flutter es un framework de desarrollo de aplicaciones móviles. Dart es el lenguaje de programación utilizado para escribir el código de Flutter. En este proyecto, se utilizan varios plugins en el frontend para mejorar la funcionalidad y la usabilidad de la aplicación. Estos incluyen:

- Async [21]: Para la manipulación de operaciones asíncronas en Dart.
- Parse\_server\_sdk\_flutter [22]: Para facilitar la comunicación entre la aplicación y el servidor de Back4App.
- Package\_info\_plus [23]: Para acceder a información del paquete de la aplicación.
- Json\_annotation [24]: Para la conversión de objetos JSON en modelos de datos Dart.
- Model\_viewer\_plus [25]: Para visualizar y manipular modelos 3D.
- Perfect\_volume\_control [26]: Para controlar el volumen del dispositivo.
- Flag [27]: Para mostrar banderas de diferentes países en la aplicación.

En cuanto al backend de la aplicación, se ha desarrollado utilizando Back4App, una plataforma basada en la nube que proporciona características de backend como autenticación de usuarios, almacenamiento en la nube y notificaciones push. Back4App utiliza Parse Server, una tecnología de código abierto que permite a los desarrolladores crear su propio backend de aplicaciones móviles. Para la gestión de los datos, se utiliza PostgreSQL, un sistema de gestión de bases de datos relacional de código abierto. El backend se encarga de la lógica de negocio de la aplicación y de la seguridad de los datos del usuario.

### 5.4.2. Estructura del proyecto

En el diseño del proyecto se emplea un enfoque basado en el Diseño Guiado por el Dominio (DDD), pero con modificaciones según las necesidades específicas del proyecto. “DDD es una técnica de desarrollo de software que busca colocar la lógica y la complejidad del negocio en el centro de la aplicación, en lugar de ser dominada por la tecnología o la infraestructura subyacente. En su forma convencional, DDD divide la estructura del código en tres capas principales: Infraestructura, Aplicación y Dominio. El núcleo o la capa de Dominio contiene las reglas de negocio y los objetos del dominio del problema. La capa de Aplicación proporciona una interfaz para interactuar con el dominio, mientras que la capa de Infraestructura se encarga de las operaciones técnicas como la persistencia de datos, la red y la interfaz de usuario.” [29]

El proyecto sigue en general esta estructura, pero se han hecho modificaciones para adaptarse mejor a las características de una aplicación móvil creada con Flutter y a las necesidades específicas del proyecto. En lugar de tener una separación estricta entre las capas de Infraestructura, Aplicación y Dominio, el proyecto las entrelaza para hacer frente a los aspectos prácticos del desarrollo de la aplicación y a la arquitectura dirigida por eventos de Flutter. En lugar de tener una capa de Infraestructura separada, los aspectos de la interfaz de usuario (views) y la persistencia de datos (DAO) están dispuestos en sus propios directorios en el mismo nivel que el dominio. Esto se debe en parte a la naturaleza de Flutter, que integra la interfaz de usuario y la lógica de negocio más estrechamente que en otros marcos de trabajo. También se ha introducido la carpeta `utils`, que no forma parte del patrón DDD estándar, pero es útil para organizar funciones y clases auxiliares que se utilizan en toda la aplicación, como las funciones de serialización de JSON y la configuración del tema.

En resumen, aunque se ha inspirado en DDD, la estructura del proyecto se ha adaptado para enfrentarse de manera efectiva a las necesidades de un proyecto Flutter, proporcionando una organización lógica y manejable del código mientras se mantiene la filosofía central de DDD: centrarse en la lógica de negocio y el dominio del problema.

La estructura del proyecto está representada por el siguiente diagrama de clases:

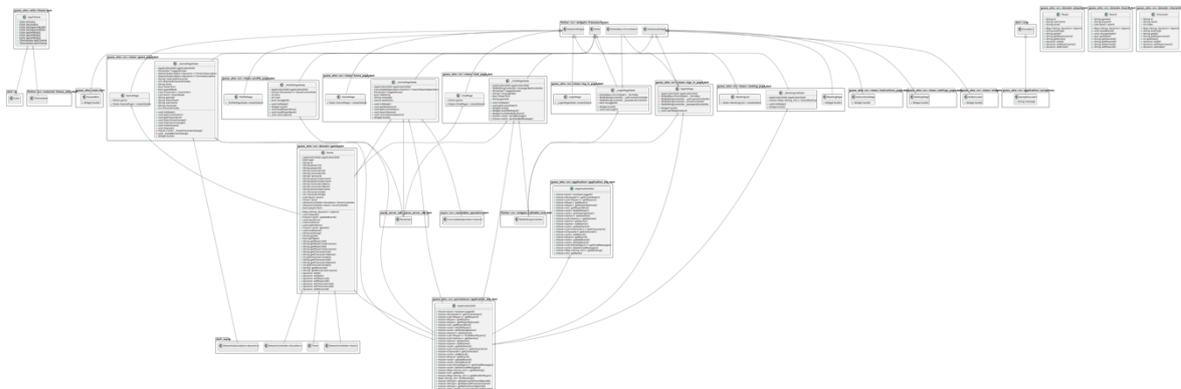


Figura 5.2. Diagrama de clases. Fuente: Elaboración propia, 2023.

### 5.4.3. Base de datos

En el desarrollo se utiliza una base de datos relacional para gestionar los datos generados por la aplicación. La elección de una base de datos relacional es estratégica para el desarrollo de la aplicación debido a las características inherentes a este modelo. Las bases de datos relacionales permiten una gestión eficaz de los datos interrelacionados. Además de permitir un acceso y reagrupación versátil de los datos sin necesidad de reorganizar las estructuras de la base de datos física. El sistema de gestión de la base de datos utilizado es PostgreSQL, ya que proporciona una gran flexibilidad y control sobre la manipulación de los datos.

Esta combinación de una base de datos relacional alojada en Back4app y gestionada a través de PostgreSQL, permite un manejo eficiente de los datos, a la vez que ofrece escalabilidad y flexibilidad para adaptarse a las cambiantes necesidades de la aplicación. La base de datos se estructura en las siguientes tablas:

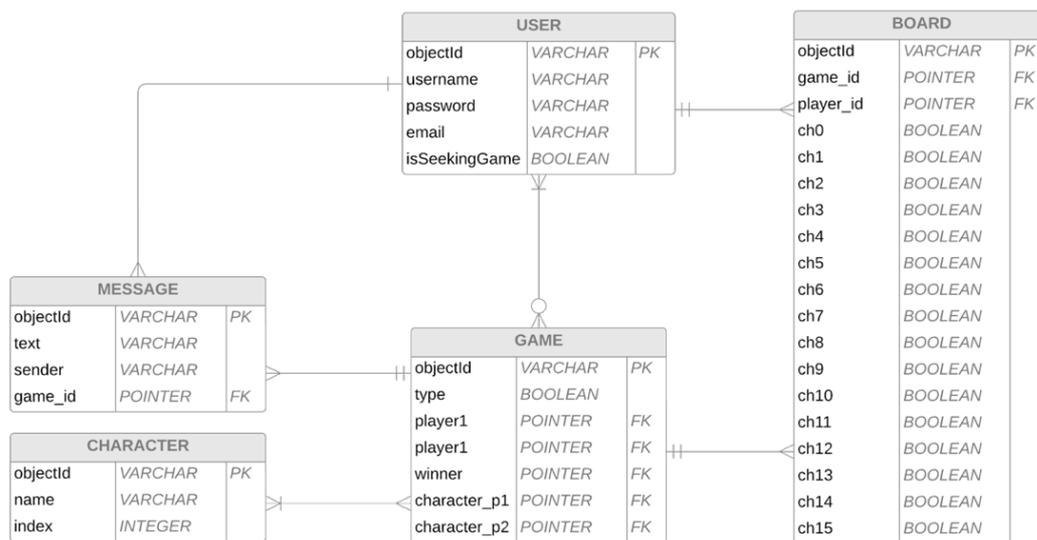


Figura 5.3. Diagrama Entidad-Relación. Fuente: Elaboración propia, 2023.

La tabla User almacena los datos relacionados con los usuarios de la aplicación. Los atributos de esta tabla son los siguientes:

- **ObjectId.** Atributo único para cada usuario que permite su identificación.
- **Username.** Atributo que almacena el nombre de usuario.
- **Password.** Atributo que almacena la contraseña del usuario.
- **Email.** Atributo que almacena el email de registro del usuario.
- **IsSeekingGame.** Atributo que indica el estado de búsqueda de partida del usuario.

La tabla Character almacena los datos relacionados con los personajes del juego. Los atributos de esta tabla son los siguientes:

- **ObjectId.** Atributo único para cada personaje que permite su identificación.
- **Name.** Atributo que almacena el nombre del personaje.
- **Index.** Atributo que asigna un índice único a cada personaje.

La tabla Game almacena los datos relacionados con las partidas. Los atributos de esta tabla son los siguientes:

- **ObjectId.** Atributo único para cada juego que permite su identificación.
- **Type.** Atributo que indica el tipo de juego.
- **Player1.** Puntero que relaciona el usuario correspondiente al jugador 1 con el juego.
- **Player2.** Puntero que relaciona el usuario correspondiente al jugador 2 con el juego.
- **Winner.** Puntero que relaciona el usuario correspondiente ganador del juego.
- **Character\_p1.** Puntero que relaciona el personaje correspondiente al jugador 1 con el juego.
- **Character\_p2.** Puntero que relaciona el personaje correspondiente al jugador 2 con el juego.

La tabla Board se usa para controlar el estado de los tableros de la partida tras cada turno jugado. Almacena los datos relacionados con el estado de cada casilla del tablero según las decisiones tomadas por cada jugador. Los atributos de esta tabla son los siguientes:

- **ObjectId.** Atributo único para cada tablero que permite su identificación.
- **Game\_id.** Puntero que relaciona la partida a la que corresponde el tablero.
- **Player\_id.** Puntero que relaciona el usuario al que corresponde el tablero.
- **Ch0, ..., Ch15.** Atributos que almacenan el estado de la casilla de cada personaje.

La tabla Message almacena los mensajes enviados por los usuarios durante el transcurso de una partida. Los atributos de esta tabla son los siguientes:

- **ObjectId.** Atributo único para cada mensaje que permite su identificación.
- **Text.** Atributo que almacena el texto del mensaje.
- **Sender.** Atributo que almacena el nombre de usuario del emisor del mensaje.
- **Game\_id.** Puntero que relaciona la partida a la que corresponde el tablero.

#### 5.4.4. Frontend

Las tecnologías usadas en este apartado son: Figma, Flutter, Dart, 3DS Max.

El frontend de la aplicación no es solo la cara de la solución que ven e interactúan los usuarios, sino también una parte integral del juego que lleva a cabo varias tareas críticas. Principalmente, el frontend presenta la información a los usuarios mediante una serie de elementos visuales, proporcionando una interacción directa entre el usuario y la aplicación. Pero, más allá de esa responsabilidad, el frontend también juega un papel vital en la comunicación con el backend. Es el encargado de solicitar a la lógica del backend los datos necesarios para que la aplicación funcione correctamente, y a continuación, presenta esos datos a los usuarios de una forma intuitiva y fácil de entender.

El diseño de la interfaz de usuario es un proceso que ha evolucionado a lo largo del proceso de desarrollo y ha estado muy influenciado por la retroalimentación de los usuarios.

Inicialmente, se crea el wireframe inicial de la aplicación en Figma. Este wireframe es un bosquejo simplificado que muestra la disposición de los elementos de la interfaz de usuario. Es la base sobre la que se construye la primera versión del prototipo de la aplicación.



Figura 5.4. Wireframe de la aplicación. Fuente: Elaboración propia, 2023.

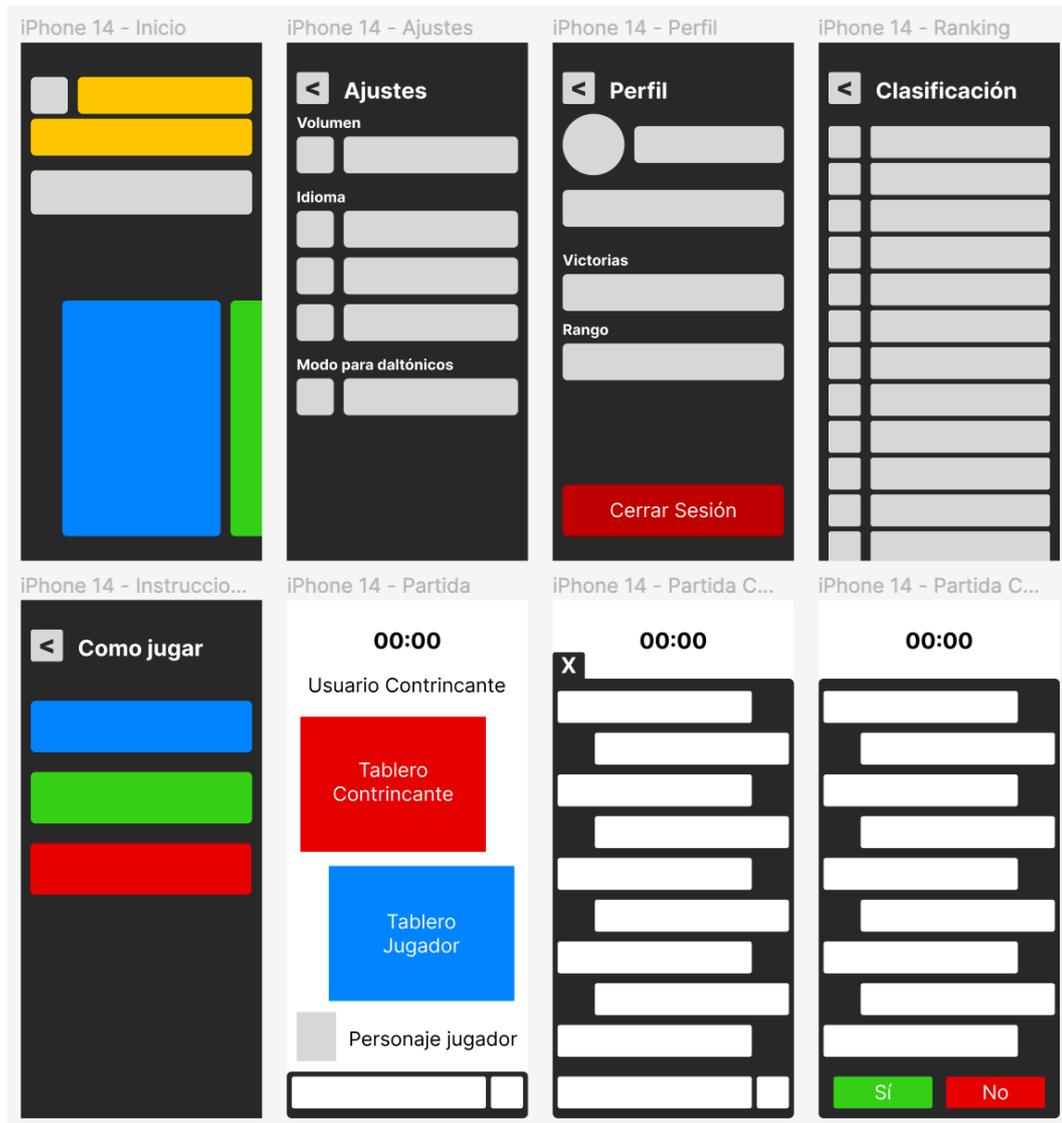


Figura 5.5. Prototipo inicial de la aplicación. Fuente: Elaboración propia, 2023.

A partir de este prototipo inicial, se inicia el desarrollo del frontend utilizando Flutter y Dart. Flutter permite la construcción de una aplicación móvil multiplataforma a partir de un único código base, lo que permite ahorrar tiempo y recursos. Dart, por otro lado, es un lenguaje de programación de alto rendimiento que permite un desarrollo rápido y eficiente.

Una vez desarrollada la primera versión del frontend, se realizan una serie de pruebas de usuario para evaluar la usabilidad de la interfaz de usuario y la experiencia general del usuario. Los resultados obtenidos en estas pruebas muestran un conjunto de ajustes y mejoras que se aplican al diseño original.

Teniendo en cuenta los comentarios y las respuestas de los usuarios, se modifica y adapta el diseño de la interfaz de usuario para mejorar la experiencia del usuario. Estos ajustes van desde la reubicación de los botones y la mejora de la navegación entre las páginas, hasta la optimización de las interacciones del usuario con los elementos de la interfaz. Estas

modificaciones no solo hacen que la aplicación sea más intuitiva y fácil de usar, sino que también se alinean mejor con las expectativas y preferencias de los usuarios.

Este ciclo iterativo de desarrollo, pruebas y refinamiento es fundamental para lograr un frontend que sea no solo funcional, sino también atractivo y fácil de usar. A través de este proceso, se ha conseguido una interfaz de usuario que realmente responde a las necesidades y preferencias de los usuarios, proporcionando una experiencia de juego enriquecedora y atractiva en la aplicación.

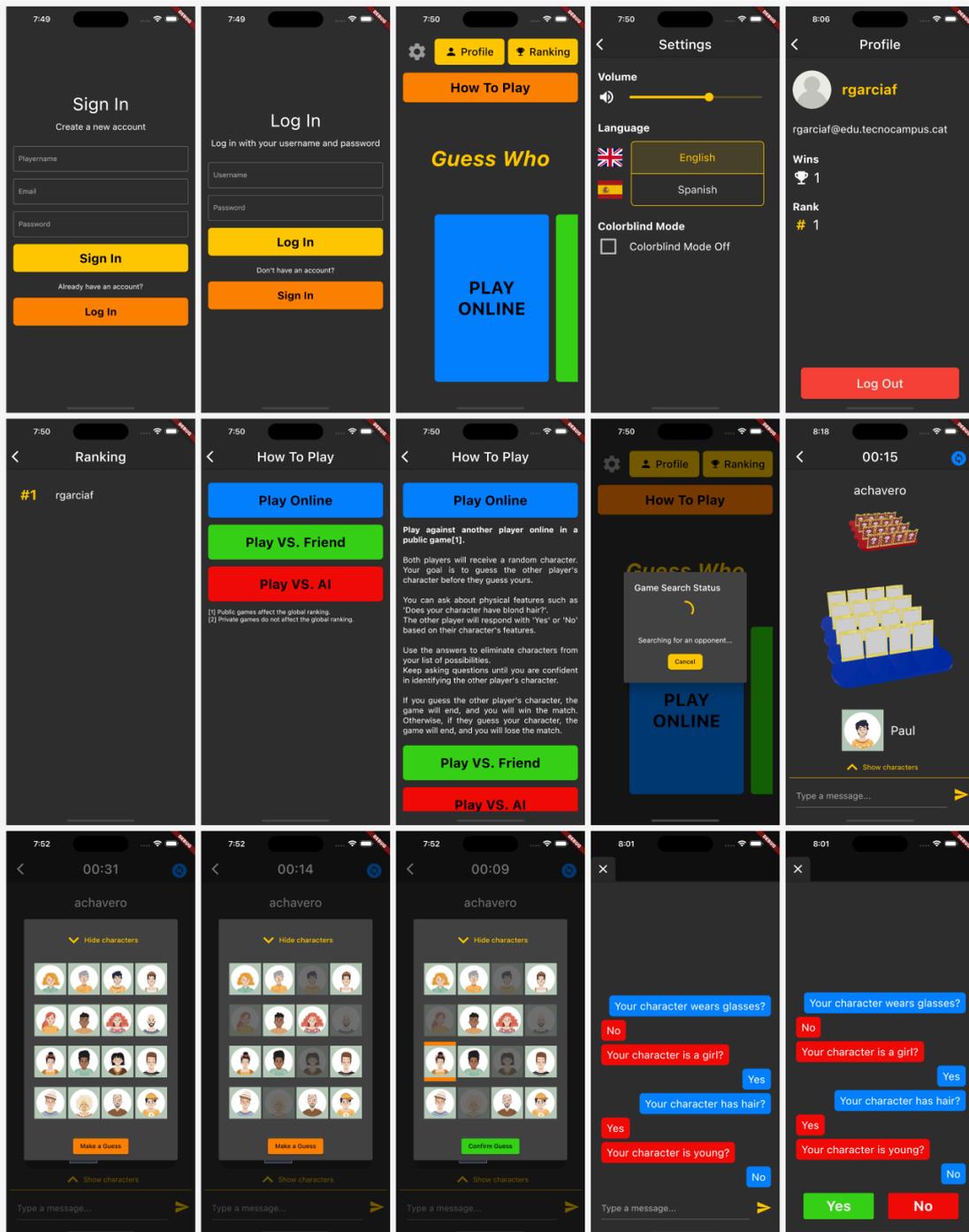


Figura 5.6. Prototipo final de la aplicación. Fuente: Elaboración propia, 2023.

Los flujos de navegación de la interfaz de usuario son esenciales para entender cómo se mueven los usuarios a través de la aplicación. Este movimiento se puede describir en términos de 'viajes' que un usuario puede hacer a través de varias páginas o vistas de la aplicación para lograr un objetivo en particular.

### Flujo de Acceso y Registro

Cuando los usuarios inician la aplicación son recibidos por la pantalla de inicio de sesión (**LogInPage**). En esta pantalla, tienen la opción de registrarse o acceder con un usuario y contraseña. En caso de tener una cuenta, deben proporcionar su usuario y contraseña para acceder a la aplicación. En caso de no tener una cuenta, son redirigidos a la pantalla de registro (**SignInPage**). El flujo de registro lleva a los usuarios a través de un proceso en el cual deben proporcionar su nombre de usuario, dirección de correo electrónico y una contraseña para crear una nueva cuenta.

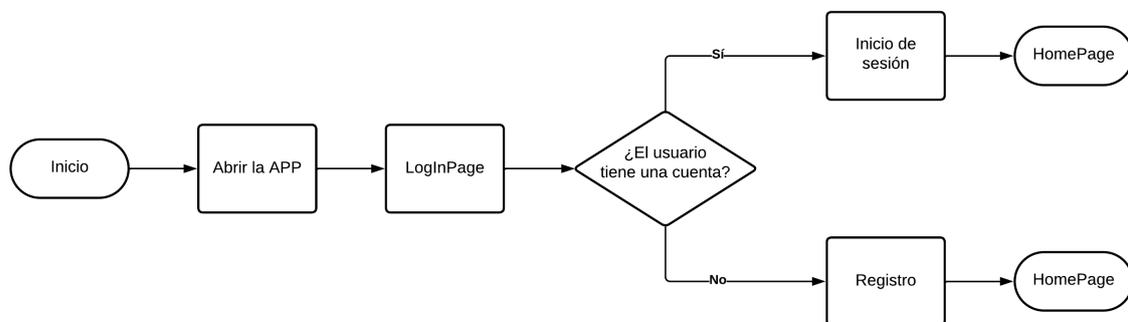


Figura 5.7. Diagrama de Flujo de Acceso y Registro. Fuente: Elaboración propia, 2023.

Una vez que los usuarios han accedido a la aplicación, llegan a la página de inicio (**HomePage**). Desde aquí, tienen varias opciones disponibles. Pueden iniciar un nuevo juego, acceder a su perfil, ver el ranking de jugadores, cambiar la configuración o ver las instrucciones de cómo jugar a cada modo de juego.

### Flujo de Juego

Si un usuario decide iniciar un nuevo juego desde la página de inicio (**HomePage**), se inicia un proceso de búsqueda de oponentes. En caso de encontrar oponente, el usuario es redirigido a la página de juego (**GamePage**). Aquí, se muestra un tablero de personajes y se asigna un personaje aleatorio a cada usuario. Los usuarios pueden hacer preguntas para adivinar el personaje del oponente mediante la pantalla de chat (**ChatPage**), que es accesible desde la pantalla de juego.

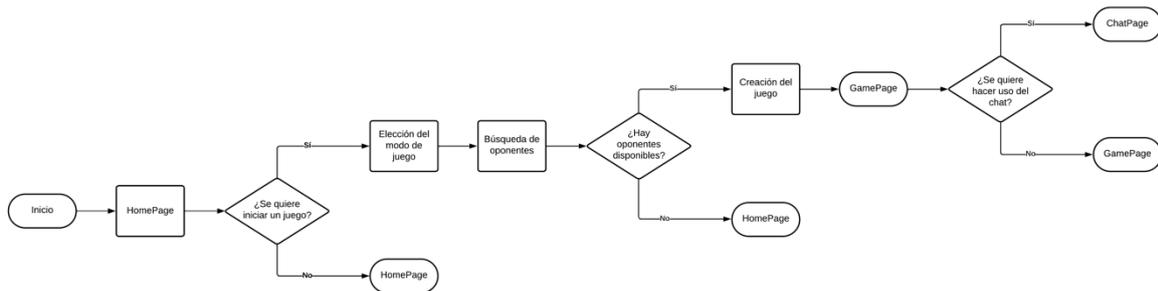


Figura 5.8. Diagrama de Flujo de Juego. Fuente: Elaboración propia, 2023.

### Flujo de Instrucciones

Si un usuario necesita ayuda para entender cómo se juega, puede acceder a la página de instrucciones (**InstructionsPage**) desde la página de inicio (**HomePage**), donde se muestra una guía sobre cómo jugar a cada modo de juego.

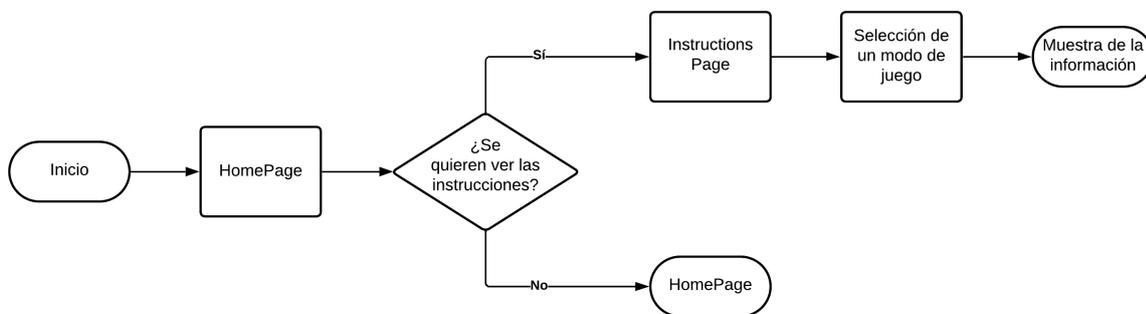


Figura 5.9. Diagrama de Flujo de Instrucciones. Fuente: Elaboración propia, 2023.

### Flujo de Configuración

Desde la página de inicio (**HomePage**), los usuarios pueden acceder a la página de configuración (**SettingsPage**). Aquí, pueden cambiar varias configuraciones de la aplicación, como el volumen del sonido, el idioma, etc.

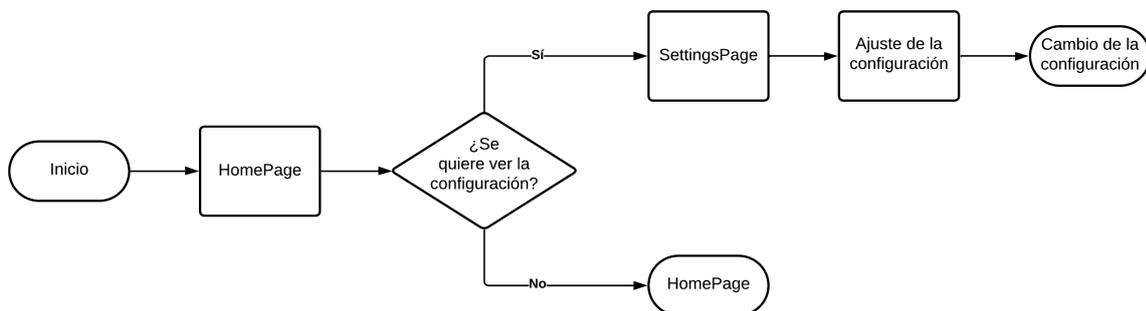


Figura 5.10. Diagrama de Flujo de Configuración. Fuente: Elaboración propia, 2023.

### Flujo de Perfil

Desde la página de inicio (**HomePage**), los usuarios pueden acceder a su perfil (**ProfilePage**), donde pueden ver su información, sus estadísticas y su rango.

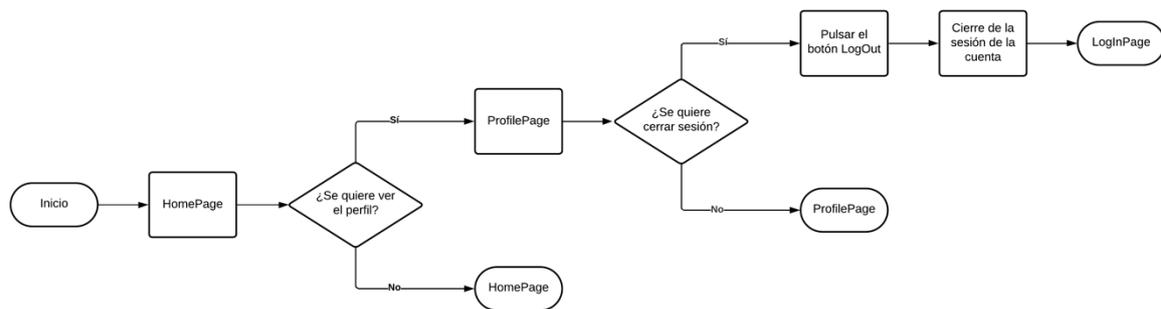


Figura 5.11. Diagrama de Flujo de Perfil. Fuente: Elaboración propia, 2023.

## Flujo de Ranking

Desde la página de inicio (**HomePage**), los usuarios pueden acceder a la página de ranking (**RankingPage**), donde pueden ver el estado del ranking de usuarios.

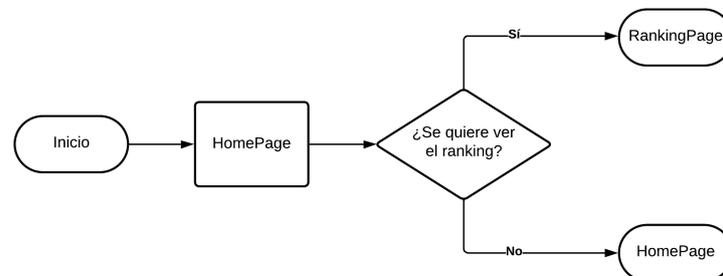


Figura 5.12. Diagrama de Flujo de Ranking. Fuente: Elaboración propia, 2023.

Estos flujos de navegación no solo son esenciales para entender cómo los usuarios interactúan con la aplicación, sino que también proporcionan una forma de evaluar y mejorar la experiencia del usuario. Mediante la realización de pruebas de usuario y la recopilación de comentarios, se hacen ajustes en estos flujos de navegación para hacerlos más intuitivos y eficientes, mejorando así la experiencia general del usuario.

### 5.4.5. Backend

Las tecnologías usadas en este apartado son: Dart, Back4app, Parse, PostgreSQL.

El backend de la aplicación es la parte de la solución que se ejecuta en un servidor remoto, siendo en este caso Back4App, y es el motor que impulsa todas las operaciones. Se trata de una infraestructura digital que controla la lógica central de la aplicación, permitiendo la funcionalidad del juego. Su función principal es el procesamiento de las solicitudes de los usuarios, la gestión y almacenamiento de los datos y la integración con el resto de los sistemas y servicios que forman la solución. El backend juega un papel crucial en la interacción con el frontend, facilitando los datos necesarios para la presentación y la interacción del usuario.

El proceso de desarrollo del backend se inicia con una detallada fase de definición de requerimientos. Se identifican y documentan todas las funcionalidades que debe

proporcionar el backend para soportar adecuadamente el juego. Este análisis incluye la gestión de usuarios, la lógica del juego y la persistencia de los datos, entre otros aspectos.

Inicialmente, se optó por Java y Spring para la implementación del backend. Esta elección se basó en la robustez y versatilidad de Java y la amplia comunidad de soporte de Spring, además de la experiencia previa del desarrollador con estas tecnologías. Sin embargo, en el momento de hacer la conexión del backend con el frontend, se hizo evidente que una solución basada en Dart sería más coherente y eficiente, teniendo en cuenta que el frontend se desarrollaba Flutter. Esta decisión resultó en un cambio significativo hacia Dart y la plataforma backend Back4App, una alternativa que facilita la gestión de bases de datos basadas en Parse y PostgreSQL.

Contando con el frontend desarrollado, se inicia la implementación del backend en Dart. El proceso de desarrollo del backend sigue el flujo de ejecución de un jugador entrando por primera vez a la aplicación. Las funcionalidades se implementan siguiendo el orden que seguiría un usuario al abrir la aplicación, por lo que se inicia en el registro de los usuarios, seguido de las funcionalidades secundarias de la aplicación, como la configuración del sistema o la clasificación de los usuarios, y se termina en la funcionalidad principal, es decir, la creación y gestión de un juego.

### **Registro de usuarios**

El proceso de registro de usuarios en la aplicación está formado por un conjunto de pasos que se llevan a cabo en una serie de etapas secuenciales. Tanto el frontend como el backend juegan un papel importante, interactuando de manera continua para llevar a cabo este proceso de manera eficiente y segura.

El punto de partida es la interfaz de usuario del frontend. En la pantalla de registro (**SignInPage**) se solicita al usuario información como el nombre de usuario, correo electrónico y contraseña. En el contexto de Flutter, el procedimiento de recolección de los datos se realiza mediante un **StatefulWidget** que recoge estos datos a través de widgets **TextFormField**. Los datos recopilados se guardan en variables de estado local para ser procesados posteriormente.

Antes de enviar los datos al backend, se realiza una validación de los mismos para garantizar su calidad y relevancia. Esto incluye la comprobación de que todos los campos están completos y que el formato del correo electrónico es válido. Si la validación no es exitosa, se notifica al usuario y se le solicita que corrija la información proporcionada. Este paso es crucial para mantener la integridad de los datos y evitar errores posteriores en el backend. Una vez que los datos han sido validados, el frontend realiza una solicitud al backend para registrar al usuario, utilizando el método de registro del usuario (**userRegistration**).

La solicitud para registrar al usuario es recibida por el backend, que en este caso es Back4App. Back4App verifica si el nombre de usuario y el correo electrónico proporcionados ya existen en la base de datos. En caso de que no existan, se crea un nuevo

registro en la tabla de usuarios de la base de datos, almacenando los datos proporcionados por el usuario. Una vez que se ha realizado el registro, Back4App devuelve una respuesta al frontend. Esta respuesta indica si la operación ha sido exitosa o no. Si hubo algún problema, como un nombre de usuario ya existente, se devuelve el error correspondiente.

La respuesta del backend es gestionada por el frontend, que toma las acciones adecuadas en función del resultado. Si el registro ha sido exitoso, el usuario es redirigido a la página principal de la aplicación. Si ha habido un error, se muestra un mensaje de error al usuario.

El sistema de registro de usuarios es un ejemplo de la colaboración entre el frontend y el backend en la aplicación. Cada uno tiene su papel específico, pero solo cuando trabajan juntos de forma coordinada pueden llevar a cabo operaciones tan fundamentales como el registro de usuarios. A través de este proceso, se asegura la calidad y la integridad de los datos del usuario, se gestiona la seguridad y se proporciona una experiencia fluida y satisfactoria para el usuario.

### Inicio de sesión

El sistema de inicio de sesión en la aplicación es un proceso similar al registro de usuarios. También requiere una colaboración estrecha entre el frontend y el backend de la aplicación.

Similar al registro de usuarios, el punto de partida es la interfaz de usuario del frontend. Cuando un usuario desea iniciar sesión, debe proporcionar su nombre de usuario y contraseña en los campos de la pantalla de inicio de sesión (**LogInPage**). Esto se hace mediante un **StatefulWidget** que recoge los datos a través de widgets **TextFormField**. Los datos recogidos se guardan en variables de estado local para ser procesados posteriormente.

Antes de enviar los datos al backend para el inicio de sesión, la aplicación realiza algunas validaciones en el frontend. Esta validación incluye la verificación de que todos los campos requeridos estén completos y que el nombre de usuario exista. Si la validación falla, se notifica al usuario y se pide que corrija los datos. Una vez validados los datos, la aplicación realiza una solicitud al backend para iniciar la sesión del usuario. Esta solicitud se hace utilizando el método de inicio de sesión de usuarios (**userLogIn**).

El backend de la aplicación, en este caso Back4App, recibe la solicitud de inicio de sesión. Verifica si el nombre de usuario y la contraseña proporcionados coinciden con los almacenados en la base de datos. Si los datos coinciden, Back4App reconoce al usuario y permite el acceso. La respuesta del backend es gestionada por el frontend. Si el inicio de sesión ha sido exitoso, el usuario es redirigido a la página principal de la aplicación. En caso de un error, como por ejemplo una contraseña incorrecta, se muestra un mensaje de error al usuario para informarle del problema.

El proceso de inicio de sesión es posible gracias a la colaboración entre el frontend y el backend. El frontend recoge y valida los datos del usuario, luego hace una solicitud al backend. El backend verifica los datos y concede acceso si son correctos, devolviendo una

respuesta que el frontend maneja adecuadamente. Esto es lo que hace posible que los usuarios inicien sesión en la aplicación.

## Perfil de usuario

La página de perfil de usuario es un componente que muestra al usuario la información relacionada con él y permite interactuar con ciertas funciones de la aplicación, como cerrar la sesión.

El código Flutter de la página de perfil se centra en el **StatefulWidget** de la página de perfil de usuario (**ProfilePage**). Cuando se inicia este widget, se cargan ciertas funciones y variables a través del método de inicialización (**initState**). Se obtiene información sobre el usuario actual y se cargan los datos sobre sus victorias y su posición en el ranking, para luego construir una interfaz de usuario que muestre la información de forma clara.

La interfaz de usuario se construye utilizando un **FutureBuilder** que espera a que se carguen los datos del usuario actual. Mientras espera, muestra un widget **CircularProgressIndicator** para indicar que se están cargando los datos. Si hay un error al cargar los datos, se muestra un mensaje de error. Cuando los datos del usuario están disponibles, se muestra la información del usuario, incluyendo su nombre de usuario, correo electrónico, número de victorias y posición en el ranking. En la parte inferior de la página se encuentra el botón de cierre de sesión, que permite al usuario cerrar la sesión actual.

Además de la construcción de la interfaz de usuario, la clase **\_ProfilePageState** también contiene funciones para cargar y manejar los datos del usuario. Los métodos encargados de mostrar en el frontend la información recibida del backend son los métodos de carga de las victorias del usuario (**loadPlayerWins**) y carga de la posición en el ranking del usuario (**loadPlayerRank**). Ambos métodos utilizan funciones de la clase **ApplicationDAO** para obtener estos datos. Los datos del usuario se obtienen y manejan a través de la clase **ApplicationDAO**. Esta clase contiene varios métodos que interactúan con Back4App para obtener y manejar los datos de la aplicación.

El método de cierre de sesión (**userLogout**) permite al usuario cerrar la sesión actual. Este método llama a la función para cerrar sesión del usuario actual y, si tiene éxito, navega de vuelta a la página de inicio de sesión. En caso de error, se muestra al usuario un mensaje informando del error.

## Configuración

La pantalla de ajustes es un componente que permite a los usuarios personalizar la experiencia de la aplicación según sus preferencias.

El código Flutter de la página de ajustes se basa en el **StatelessWidget** de la página de ajustes (**SettingsPage**). La interfaz de usuario se compone de una serie de campos que permiten la personalización de la aplicación, como ajustes para el volumen, selección de idioma y modo para daltónicos. Cada campo consta de un texto que describe el ajuste, seguido de un widget

interactivo que permite a los usuarios cambiar el ajuste. En este caso, los widgets interactivos son `Volume`, `LanguageButtons` y `ColorblindCheckbox`.

`Volume` es un **StatefulWidget** que muestra un **Slider** para ajustar el volumen de la aplicación. El volumen actual se obtiene al principio y se actualiza cada vez que el usuario mueve el **Slider**. Para manejar el volumen, se utiliza la biblioteca `PerfectVolumeControl`.

`LanguageButtons` es un **StatefulWidget** que muestra un conjunto de **ToggleButtons** para seleccionar el idioma de la aplicación. En este caso, los botones son para seleccionar inglés o español. Cuando un usuario selecciona un idioma, los otros botones se desactivan.

`ColorblindCheckbox` es un **StatefulWidget** que muestra un **Checkbox** para activar o desactivar el modo para daltónicos. Cuando el usuario marca o desmarca la casilla, el estado del **Checkbox** se actualiza y el texto que describe el ajuste se actualiza para reflejar el estado.

El manejo de los datos para cada ajuste se realiza en los widgets interactivos respectivos. Cuando un usuario interactúa con un widget, se llama a la función de establecer estado (**setState**) para actualizar el widget y reflejar el cambio en la interfaz de usuario. En este caso, no hay interacción con la base de datos ni con el backend. Todos los cambios se hacen localmente en la aplicación y sólo afectan a la instancia actual de la aplicación.

## Ranking

La pantalla de ranking muestra una lista de clasificación de los jugadores ordenada según el número de victorias que cada uno ha logrado. Para que un jugador aparezca en el ranking, debe tener al menos una victoria, esto evita que el ranking se llene de usuarios que nunca han usado la aplicación para jugar.

La clase del ranking (**RankingPage**) es un **StatelessWidget** que representa la pantalla de ranking. El widget `RankingList` es un **StatefulWidget** que se encarga de obtener y mostrar el ranking de los jugadores. Se utiliza **FutureBuilder** para manejar el proceso de obtención de datos de ranking de forma asíncrona. Mientras se cargan los datos, se muestra un **CircularProgressIndicator**. Si se produce un error durante la carga de los datos, se muestra un mensaje de error. Si los datos se cargan correctamente, se crea una lista ordenada con los nombres de los jugadores y se utiliza **ListView.builder** para reflejar dicha lista en la interfaz de usuario, así se muestra la posición en el ranking y el nombre de usuario de cada jugador.

La clase `_RankingListState` se encarga de manejar los datos de ranking. Al principio, obtiene los datos de ranking llamando al método de obtención del ranking (**getRanking**) del **ApplicationDAO** y guarda el `Future` resultante en una variable. El método de obtención del ranking se encarga de obtener y procesar los datos de la clasificación de los usuarios. Primero, llama al método de obtención de las victorias de cada jugador (**getWinsPerPlayer**) para obtener un mapa de los IDs de los jugadores y el número de victorias que cada uno ha obtenido. Luego, ordena el mapa según el número de victorias y reemplaza los IDs de los jugadores por sus nombres de usuario. Si no se pueden obtener los datos de ranking por alguna razón, se imprime un mensaje de error y se lanza una excepción.

El **ApplicationDAO** se comunica con la base de datos para obtener y procesar los datos de ranking. Utiliza la clase **ParseObject** para hacer consultas a la base de datos. En este caso, hace una consulta a la colección 'Game' para obtener una lista de todos los juegos públicos que tienen ganador. Luego, procesa los datos para calcular el número de victorias de cada jugador y devuelve un mapa de los nombres de usuario de los jugadores y el número de victorias que cada uno ha obtenido.

La página de ranking muestra la clasificación de los jugadores ordenada según el número de victorias de cada uno. La página obtiene los datos de ranking de forma asíncrona y maneja los posibles errores durante la obtención de los datos.

### Lógica de juego

La lógica de juego incluye aspectos clave como la búsqueda de oponente, la asignación aleatoria de los personajes a cada jugador, la generación del tablero de juego, la gestión de las rondas y la determinación del ganador.

Cuando un usuario desea jugar, desde la pantalla de inicio, debe pulsar el botón de uno de los modos de juego. Este botón está asociado a un controlador de eventos que llama al método de establecer el estado de búsqueda de juego (**setSeekingGame**) del **ApplicationDAO**. Este método se encarga de actualizar el estado del usuario en la base de datos. Para ser más específicos, modifica la columna `isSeekingGame` del usuario a `true`, lo que indica que está buscando un oponente para jugar.

Una vez que el usuario ha indicado que desea jugar, se ejecuta el método de búsqueda de juego (**seekGame**). Este método utiliza la función de búsqueda de oponentes (**findOtherPlayers**) para buscar a otros jugadores que también estén buscando un oponente. Es decir, buscan usuarios cuyo atributo `isSeekingGame` también sea `true`. Mientras se realiza esta búsqueda, se muestra una ventana emergente con un **CircularProgressIndicator** para indicar que el proceso está en marcha. La función de búsqueda de oponentes (**findOtherPlayers**) sigue buscando hasta que encuentra un oponente, o hasta que se recibe una señal de cancelación a través del **Stream** opcional `cancellationToken`. Si después de un minuto no se ha encontrado ningún oponente, el método de búsqueda de juego (**seekGame**) se encarga de actualizar el mensaje en la ventana emergente para informar al usuario que no hay oponentes disponibles en ese momento y se detiene la búsqueda de oponentes. Si se encuentra un oponente, el método de búsqueda de juego (**seekGame**) se encarga de crear un nuevo juego con los dos usuarios como jugadores. Para asignar los personajes a cada jugador, se ejecuta el método de obtención de personajes (**getCharacters**) que recoge de forma aleatoria dos personajes del conjunto total de personajes disponibles. A continuación, se crea el juego con el método de creación de juego (**addGame**) que registra el juego en la base de datos, y se generan los tableros para cada jugador con el método de creación de tablero (**addBoard**). Estos tableros reflejan el estado de los personajes, según si están descartados o en juego, de cada jugador.

El juego se desarrolla por turnos de un minuto de duración, donde cada jugador tiene la oportunidad de hacer preguntas al otro sobre su personaje. Estas preguntas se realizan a través de un chat, al jugador que pregunta se le muestra un input de texto, mientras que al jugador al que se le pregunta sólo se le muestran dos botones, ya que debe responder únicamente con un Sí o un No. Durante el turno, los jugadores tienen la opción de abrir la ventana de personajes y, basándose en las respuestas que se han recibido, pueden descartar o mantener ciertos personajes. Esto se hace actualizando el tablero de cada jugador mediante el método de actualización del tablero (**updateBoard**). Los personajes descartados se marcan reduciendo al 25% la opacidad de su imagen para que el jugador reciba feedback visual en la interfaz de usuario de qué personajes ha descartado. En cualquier momento durante el turno, los jugadores pueden intentar adivinar el personaje del oponente. Para hacer esto, pulsan el botón de elección de personaje en la ventana de personajes y pueden seleccionar un personaje de los que aún no se han descartado, que se resalta en naranja para indicar que se está seleccionando, y luego confirman su elección pulsando el botón de confirmar selección. Se maneja la elección del usuario y se determina si la adivinación es correcta o no, es decir, si el personaje seleccionado corresponde con el personaje del otro jugador.

Si un jugador adivina correctamente el personaje de su oponente, se registra como ganador en el `winnerId` del juego a través del método de establecer ganador (**setWinner**) del **ApplicationDAO**. Una vez registrado el ganador, se muestra un mensaje informando del fin de la partida a ambos jugadores y luego son redirigidos a la pantalla de inicio de la aplicación.

La comunicación entre los jugadores se hace mediante el chat. A lo largo del juego, los jugadores pueden interactuar enviándose preguntas y respuestas. El método de obtención de los mensajes del chat (**getChatMessages**) del **ApplicationDAO** se encarga de recuperar los mensajes de chat entre los dos jugadores, proporcionando así una comunicación en tiempo real durante el juego. Una vez finalizado el juego, se eliminan los mensajes relativos a la partida, para liberar la base de datos de datos temporales que no se usarán en el futuro.

Es importante mencionar que estas acciones se desarrollan en distintas páginas de la aplicación, por ejemplo, la página de juego (**gamePage**) y la página de chat (**chatPage**). En la página de juego se muestran los personajes y se puede hacer la adivinación, mientras que en la página de chat se llevan a cabo las preguntas y respuestas



## 6. Test

### 6.1. Test de usuario

#### 6.1.1. Descripción de una muestra ideal

La selección de una muestra de usuarios ideal para probar la aplicación desarrollada debe incorporar una gama diversa de usuarios potenciales que reflejen la amplia demografía que el juego puede atraer. Se describen las características que debería tener esta muestra ideal:

**Tamaño de la muestra:** Dada la naturaleza de la aplicación, una muestra ideal para pruebas puede incluir entre 10 y 15 usuarios. Este número proporciona una representación sólida de la diversidad de la base de usuarios sin sobrecargar el proceso de prueba.

**Diversidad de la muestra:** La aplicación tiene el potencial de atraer a una amplia gama de jugadores, por lo tanto, la muestra debe reflejar esta diversidad. Esto incluye una mezcla equilibrada de géneros, diferentes grupos de edad (desde niños hasta adultos), y una variedad de orígenes culturales.

**Niveles de habilidad tecnológica:** La muestra debe incluir personas con diferentes niveles de habilidad tecnológica. Dado que el juego es accesible para todas las edades, es importante considerar a los usuarios con habilidades tecnológicas limitadas, así como a los usuarios más avanzados y experimentados en la tecnología.

**Intereses y comportamiento del usuario:** Dado que el juego *¿Quién es Quién?* puede atraer tanto a jugadores habituales como a los que juegan de forma ocasional, es beneficioso incluir ambos tipos de jugadores en la muestra. Esto ayuda a entender mejor cómo diferentes tipos de jugadores interactúan con la aplicación y qué características pueden ser más atractivas para cada grupo.

**Necesidades especiales:** Teniendo en cuenta la inclusión y la accesibilidad, la muestra también debe incluir a usuarios con diversas necesidades especiales, como aquellos con discapacidades visuales o auditivas, para asegurar que la aplicación es accesible para todos.

En resumen, la muestra de usuarios ideal para la aplicación es un grupo diverso y representativo de usuarios potenciales del juego. Esta diversidad en la muestra proporciona una comprensión más rica y completa de cómo diferentes usuarios interactúan con la aplicación y permite identificar posibles mejoras para hacerla aún más atractiva y accesible.

#### 6.1.2. Descripción y resultados de los test de usuario

##### Flujo de acceso y registro

Se pone a prueba el registro e inicio de sesión. Se pide a los usuarios que abran la aplicación, seleccionen el botón de registro si son nuevos usuarios o el botón de inicio de sesión si ya

tienen una cuenta en la aplicación, y que introduzcan la información necesaria. El objetivo es que los usuarios puedan registrarse o iniciar sesión sin problemas, y ser redirigidos a la página principal de la aplicación.

La niña y el hombre adulto logran registrarse e iniciar sesión sin problemas, mientras que la mujer joven experimenta un pequeño retraso en el inicio de sesión. Se concluye que el flujo de registro e inicio de sesión se considera intuitivo y fácil de seguir, aunque se puede optimizar el tiempo de respuesta de la aplicación.

### **Flujo de juego**

Se quiere comprobar la interacción de los usuarios con el juego, para su ver su reacción ante el flujo de juego y asegurar la intuitividad de la interfaz de usuario. Se pide a los usuarios que seleccionen el botón para iniciar un juego, esperen a que se les asigne un oponente, hagan preguntas, eliminen personajes y adivinen el personaje del oponente. El objetivo es que los usuarios puedan seguir el flujo del juego sin problemas y comprendan las reglas y mecanismos del juego.

Todos los usuarios son capaces de seguir el flujo del juego, aunque el hombre adulto encuentra confusos algunos mecanismos del juego. Se concluye que los mecanismos del juego son en su mayoría intuitivos, pero podrían requerir una mejor explicación para los usuarios menos experimentados con este tipo de tecnologías.

### **Flujo de instrucciones**

Se quiere asegurar la claridad de las instrucciones y su correcta comprensión por parte de los usuarios. Se pide a los usuarios que seleccionen el botón de instrucciones y lean la información proporcionada. Las instrucciones deben ser claras y comprensibles.

Todos los usuarios entienden las instrucciones y cómo se juega a los distintos modos de juego ofrecidos en la aplicación. Se concluye que las instrucciones son claras y detalladas.

### **Flujo de configuración**

Se quiere comprobar la comprensión de las opciones de ajustes por parte de los usuarios. Se pide a los usuarios que seleccionen el botón de ajustes y prueben a cambiar las diferentes opciones. El objetivo es que los usuarios puedan acceder y cambiar fácilmente las configuraciones de la aplicación.

Todos los usuarios pueden acceder y cambiar las configuraciones sin problemas. Se concluye que el flujo de configuración es fácil de seguir y cambiar las configuraciones es intuitivo.

### **Flujo de perfil**

Se quiere comprobar la comprensión del perfil de usuario y la información que se muestra. Se pide a los usuarios que seleccionen el botón de perfil y visualicen la información

proporcionada. El usuario debe poder acceder a su perfil y entender la información relacionada con él.

Todos los usuarios pueden acceder a sus perfiles y entender la información que se les muestra. Se concluye que el flujo de perfil es intuitivo y la información es fácil de entender.

### **Flujo de ranking**

Se quiere comprobar la comprensión del ranking. Se pide a los usuarios que seleccionen el botón de ranking y visualicen la información que se muestra. El usuario debe poder acceder al ranking y entender la información proporcionada.

Todos los usuarios pueden acceder al ranking y entender la información proporcionada. Se concluye que el flujo de ranking es intuitivo y la información es fácil de entender.

En resumen, los usuarios son capaces de navegar por la aplicación sin problemas, con excepción de algunos aspectos del flujo de juego que podrían ser más claros para los nuevos usuarios. Los test de usuario permiten recolectar información valiosa sobre áreas que podrían mejorarse para asegurar una mejor experiencia de usuario en general.

## **6.2. Test de aceptación**

### **6.2.1. Descripción y resultados de los test de aceptación**

#### **Digitalizar el juego de mesa tradicional ¿Quién es Quién?**

Se quiere comprobar la digitalización precisa del juego de mesa tradicional ¿Quién es Quién? en la aplicación. Se realizan partidas en distintos dispositivos por usuarios de diversas edades y habilidades para asegurar que la aplicación recrea de manera exacta el juego original. Esto incluye verificar la presencia y correcto funcionamiento de las funciones del juego físico.

El objetivo es que la aplicación reproduzca con fidelidad el juego original, permitiendo a los usuarios jugar de forma fluida y sin dificultades. Los resultados indican que todos los usuarios pueden jugar al juego digital de manera similar a cómo lo harían con el juego físico, aunque se han reportado cierres inesperados de la aplicación al interactuar de manera rápida y pulsando varios botones a la vez.

En conclusión, la digitalización del juego de mesa ¿Quién es Quién? se ha logrado con éxito. La aplicación consigue recrear las dinámicas del juego físico, permitiendo a los usuarios realizar preguntas y adivinar el personaje del oponente de manera intuitiva y fluida. Sin embargo, se han identificado algunos problemas de estabilidad que provocan cierres inesperados de la aplicación. Esto sugiere la necesidad de trabajar en mejorar la robustez y capacidad de respuesta de la aplicación para evitar dichos cierres y otros posibles errores.

### **Recoger, tratar y almacenar datos internos**

Se quiere comprobar si la aplicación puede recoger, tratar y almacenar datos internos correctamente. Esto se verifica a través de pruebas en las que se juegan partidas y se verifica luego en la base de datos que todos los datos generados durante la partida se han almacenado correctamente.

El objetivo es que todos los datos internos generados durante las partidas se recogen, se procesan y se almacenan de manera adecuada. Los resultados muestran que todos los datos se recogen y almacenan correctamente.

En conclusión, la aplicación demuestra ser capaz de manejar de manera efectiva los datos internos generados durante las partidas. Todos estos datos se recogen, se procesan y se almacenan correctamente en la base de datos, lo que permite un seguimiento y análisis efectivo de las partidas.

### **Recoger, tratar y almacenar datos externos**

Se quiere comprobar si la aplicación puede recoger, tratar y almacenar datos externos correctamente. Esto se verifica a través de pruebas en las que se monitoriza la actividad y estadísticas de los usuarios en la aplicación y luego se verifica en la base de datos si se ha almacenado correctamente.

El objetivo es que todos los datos externos relacionados con los usuarios se recogen, se procesan y se almacenan de manera adecuada. Los resultados muestran que todos los datos de los usuarios se recogen y almacenan correctamente, y que estos datos se pueden recuperar y utilizar de manera efectiva en la aplicación.

En conclusión, la aplicación demuestra ser capaz de manejar de manera efectiva los datos externos generados por los usuarios. Todos estos datos se recogen, se procesan y se almacenan correctamente en la base de datos, lo que permite un seguimiento y análisis efectivo del comportamiento de los usuarios.

### **Relacionar los datos internos y externos en una base de datos relacional**

Se quiere comprobar la correcta estructuración de la base de datos y que las relaciones entre los diferentes conjuntos de datos están establecidas y funcionan como se espera. Se verifica a través de una revisión detallada de la estructura de la base de datos y realizando consultas para confirmar que las relaciones funcionan correctamente.

El objetivo es asegurar que los datos internos y externos se relacionan correctamente en la base de datos relacional. Los resultados indican que todas las relaciones entre los conjuntos de datos están establecidas y funcionan correctamente.

En conclusión, la estructura de la base de datos relacional y las relaciones entre los diferentes conjuntos de datos funcionan correctamente. Esto asegura que los datos internos y externos se pueden relacionar de manera efectiva, lo que es esencial para un correcto uso de los datos.

### **Desplegar y mantener activo un servidor**

Se quiere comprobar si el servidor puede manejar un número elevado de usuarios simultáneos sin fallos y se mantiene activo de manera constante, con excepción de los periodos de mantenimiento programados. Esto se verifica a través de pruebas de carga.

El objetivo es asegurar la estabilidad y disponibilidad del servidor ante un alto número de usuarios. Los resultados de las pruebas de carga indican que el servidor puede manejar un alto número de usuarios simultáneos. Sin embargo, durante las pruebas de carga, el tiempo de respuesta de la aplicación es más largo de lo habitual.

En conclusión, el servidor demuestra ser capaz de manejar un número elevado de usuarios simultáneos sin fallos, exceptuando que el tiempo de respuesta de la aplicación aumenta, lo que puede afectar a la experiencia del usuario. Esto indica que se necesita trabajar en la optimización del rendimiento del servidor para mejorar el tiempo de respuesta durante los períodos de alta carga.

### **Mantener un control de los usuarios**

Se quiere comprobar que cada usuario tiene un perfil único y que su actividad y estadísticas se registran y se asocian correctamente a su perfil. Se verifica a través de la creación de varios perfiles y la realización de acciones con ellos, comprobando luego si estas acciones se han registrado correctamente.

El objetivo es asegurar un control efectivo de los usuarios y su actividad. Los resultados indican que todos los usuarios tienen un perfil único y que todas las acciones realizadas se registran y se asocian correctamente a sus perfiles.

En conclusión, la aplicación consigue mantener un control efectivo de los usuarios y su información. Cada usuario tiene un perfil único y su actividad y estadísticas se registran y se asocian correctamente a su perfil.

### **Mantener un control de las partidas jugadas**

Se quiere comprobar que todas las partidas jugadas se registran correctamente, incluyendo detalles como los jugadores, los personajes y el ganador. Se verifica a través de la revisión de la base de datos después de las partidas para confirmar que todos los detalles están presentes y son correctos.

El objetivo es asegurar un control efectivo de las partidas jugadas. Los resultados indican que todas las partidas jugadas se registran correctamente y que todos los detalles son correctos.

En conclusión, la aplicación consigue mantener un control efectivo de las partidas y su información. Todas las partidas jugadas se registran correctamente, incluyendo todos los detalles relevantes.

**Permitir la comunicación por chat entre los usuarios**

Se quiere comprobar que los usuarios pueden hacer preguntas y adivinar los personajes a través de la entrada de texto. Se verifica a través de la realización de partidas y la verificación de que las entradas de texto de los usuarios se procesan correctamente.

El objetivo es asegurar una comunicación fluida y efectiva entre los usuarios a través del chat. Los resultados indican que todos los usuarios pueden comunicarse efectivamente a través del chat, pudiendo hacer preguntas y responder a estas sin problemas.

En conclusión, la aplicación permite una comunicación efectiva por chat entre los usuarios.

En resumen, la aplicación demuestra un alto nivel de funcionalidad y rendimiento general. No obstante, se identifican áreas que requieren mejoras, en particular la estabilidad de la aplicación y el rendimiento del servidor durante los períodos de alta carga. Trabajar en estos aspectos mejorará la experiencia del usuario y asegurará el éxito a largo plazo de la aplicación.

## 7. Conclusiones

El proyecto de desarrollo de la aplicación ¿Quién es Quién? ha supuesto un gran reto y una oportunidad de aprendizaje significativa, especialmente considerando que se inició el proyecto con una experiencia limitada en el desarrollo de aplicaciones móviles y teniendo el desarrollador un perfil más orientado hacia el desarrollo frontend que el desarrollo backend.

El proyecto ha enfrentado dificultades técnicas durante el proceso hacia su finalización. Uno de los problemas significativos que surgieron fue la incompatibilidad entre la versión del SDK de Flutter y la versión de Parse que utiliza Back4App. Además, se enfrentaron problemas con las versiones de algunos plugins utilizados. Estos obstáculos técnicos requirieron una investigación exhaustiva para comprobar la compatibilidad entre versiones y adaptar el proyecto a las versiones compatibles. Este proceso, aunque tedioso, fue un componente esencial en la resolución de estos problemas y la eventual finalización exitosa del proyecto.

En la etapa de prueba y depuración, se encontró otro obstáculo en el desarrollo. Al desarrollar el proyecto en un dispositivo con sistema operativo macOS, se encontraron problemas puntuales con Xcode al usar el simulador para probar y depurar la aplicación. Siendo la primera vez que se trabajaba con estas herramientas para depurar aplicaciones móviles, este reto presentó una curva de aprendizaje significativa. Por suerte, tras familiarizarse con estas herramientas y entender su funcionamiento, estos problemas pudieron ser resueltos.

Pese a los desafíos técnicos y desviaciones a lo largo del desarrollo, el proyecto ha culminado con éxito en la creación de la aplicación móvil propuesta como objetivo. Esto incluye no solo el logro de construir una interfaz de usuario funcional y atractiva, sino también el desarrollo de un backend robusto y confiable. El proyecto abarcó la creación de prácticamente cada componente y funcionalidad de la aplicación, desde la recopilación y gestión de datos hasta el manejo de la interactividad del usuario y la implementación de las reglas del juego.

Además de las habilidades técnicas adquiridas durante este proyecto, el proceso de desarrollo también ha fomentado una profunda apreciación por la importancia de la planificación, la gestión de proyectos y la capacidad para adaptarse y resolver problemas en situaciones desafiantes. Estos son aspectos fundamentales del desarrollo de software que se extienden más allá del alcance técnico del proyecto y son habilidades que seguramente serán valiosas en proyectos futuros.

En resumen, el desarrollo de la aplicación ¿Quién es Quién? ha sido un proceso lleno de desafíos, pero también de aprendizaje y crecimiento. Los obstáculos encontrados en el camino sirvieron como oportunidades para aprender y mejorar, resultando en una aplicación móvil exitosa que no solo cumple con los requerimientos iniciales, sino que también proporciona una valiosa experiencia de desarrollo de software.



## 8. Posibles ampliaciones

### Implementación de la Inteligencia Artificial

Dado que la implementación de la IA no se ha realizado en la versión actual del proyecto, es una de las principales ampliaciones para el futuro. La IA podría usarse para crear un oponente virtual para el jugador, lo que permitiría partidas individuales en cualquier momento. Esto podría implementarse utilizando bibliotecas de aprendizaje automático, como TensorFlow o PyTorch.

### Despliegue en Azure

Aunque la aplicación no se ha desplegado en Azure hasta ahora, esta es una posible ampliación para el futuro. Azure es una plataforma de computación en la nube que puede proporcionar una serie de beneficios, incluyendo escalabilidad, seguridad mejorada y una mayor resistencia. Para implementar esta funcionalidad, se necesitaría familiarizarse con el conjunto de servicios de Azure y aprender cómo desplegar y gestionar aplicaciones en esta plataforma.

### Implementación de autenticación de usuarios

La autenticación actual se basa en un nombre de usuario, correo electrónico y contraseña. En el futuro se podría implementar la autenticación a través de cuentas de Google u otros servicios. Esto proporcionaría una capa adicional de seguridad y también mejoraría la experiencia del usuario al simplificar el proceso de inicio de sesión. Para implementar esta funcionalidad, se podría utilizar Firebase Authentication o el API de Google Sign-In.

### Implementación de nuevas funcionalidades

Se han observado distintas maneras de ampliar las funcionalidades de la aplicación. Algunas ideas podrían incluir la adición de logros o desafíos para los jugadores. Para implementar estas funciones, se necesitaría desarrollar la lógica del juego correspondiente en Dart, así como actualizar la interfaz de usuario en Flutter para reflejar estas nuevas funciones.

### Mejoras de la Interfaz de Usuario

La interfaz de usuario podría ser refinada y mejorada en base a la retroalimentación de los usuarios. Esto podría incluir la mejora de la navegabilidad, la implementación de nuevas características de la interfaz de usuario o la mejora del diseño visual. Para implementar estas mejoras, se podrían utilizar las capacidades de diseño y personalización de Flutter.

Además de las mejoras generales de la interfaz de usuario, se propone el desarrollo de interfaces adaptadas a las posibles discapacidades de los usuarios. Esto podría incluir la creación de versiones de las pantallas que sean más fáciles de ver y de navegar para los usuarios con daltonismo, miopía, dislexia, etc. Para implementar estas mejoras, se

necesitaría investigación en diseño inclusivo y accesibilidad, así como la utilización de las capacidades de diseño de Flutter para implementar estos cambios en la interfaz de usuario.

### **Control mejorado de los nodos del tablero**

En la versión actual del proyecto, la actualización y el control de los nodos del tablero durante el juego se ha descartado dada la complejidad de su implementación. En particular, la actualización del tablero con los personajes que se ha decidido subir o bajar podría proporcionar una experiencia de usuario más fluida. Se podría explorar el uso de patrones de diseño más avanzados o el uso de bibliotecas adicionales para facilitar esta tarea.

### **Creación de ilustraciones de personajes más inclusivas**

En la versión actual del juego, los personajes podrían no representar toda la diversidad de personas, etnias y culturas que existen en nuestra sociedad. Una posible mejora sería diseñar y añadir nuevas ilustraciones que representen a una gama más amplia de personas, promoviendo la inclusión y la diversidad.

### **Añadir sonidos a la aplicación**

Para mejorar la experiencia de juego, se podrían añadir efectos de sonido que proporcionen feedback sonoro a los jugadores. Esto podría incluir sonidos para indicar el inicio y el final de una partida, así como sonidos que se activan al realizar ciertas acciones durante el juego. Para ello, se podrían utilizar bibliotecas de sonido de Flutter.

## 9. Bibliografía

- [1] Kowert, R., & Quandt, T. (Eds.). (2015). The video game debate. Routledge. <https://doi.org/10.4324/9781315736495>
- [2] An Analysis of Ludo Board Game Play on Smartphones | Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems. (s.f.). ACM Conferences. <https://dl.acm.org/doi/abs/10.1145/3411763.3451728>
- [3] (2005). Hasbro Spielzeug und Spiele, Kinderspielzeug, Action-Figuren und Brettspiele – Hasbro. <https://shop.hasbro.com/es-es/product/guess-who-classic-game/7DEC61D9-5056-9047-F55F-FC686C17D23F>
- [4] Games, O. (2019). Guess who am I board games - apps on google play. Android Apps on Google Play. <https://play.google.com/store/apps/details?id=com.Offs.WhoIAm&pli=1>
- [5] Games, O. (2021). Guess who I am 2 - board games - apps on google play. Android Apps on Google Play. <https://play.google.com/store/apps/details?id=com.Offs.Guess.Who.Board.Game.Online>
- [6] Didacta. (2020). ¿QUIÉN ES QUIÉN? - apps on google play. Android Apps on Google Play. <https://play.google.com/store/apps/details?id=com.didacta.quienesquien>
- [7] Corp, Y. (2020). Whooo? - apps on google play. Android Apps on Google Play. <https://play.google.com/store/apps/details?id=com.YsoCorp.WhoAmI>
- [8] Friends, C. (2013). Chess online - apps on google play. Android Apps on Google Play. <https://play.google.com/store/apps/details?id=chessfriends.online.chess>
- [9] Labs, G. (2017). Parchisi STAR online - apps on google play. Android Apps on Google Play. [https://play.google.com/store/apps/details?id=com.superking.parchisi.star&hl=en\\_US&gl=US&pli=1](https://play.google.com/store/apps/details?id=com.superking.parchisi.star&hl=en_US&gl=US&pli=1)
- [10] Sas, E. (2012). Akinator - apps on google play. Android Apps on Google Play. <https://play.google.com/store/apps/details?id=com.digidust.elokence.akinator.freemium>
- [11] Agile Manifesto: Qué son las metodologías ágiles • Asana. (s.f.). Asana. <https://asana.com/es/resources/agile-methodology>
- [12] What is Scrum? (s.f.). Scrum.org. <https://www.scrum.org/resources/what-is-scrum>
- [13] Mariana de los angeles Courbenas Marcano. (s.f.). Los requerimientos de un proyecto. Academia.edu - Share research. [https://www.academia.edu/8532965/Los\\_requerimientos\\_de\\_un\\_proyecto](https://www.academia.edu/8532965/Los_requerimientos_de_un_proyecto)
- [14] Flutter documentation. (s.f.). Flutter documentation | Flutter. <https://docs.flutter.dev>

- [15] Dart documentation. (s.f.) Dart documentation | Dart. <https://dart.dev/guides>
- [16] Azure documentation. (s.f.). Microsoft Learn: Build skills that open doors in your career. <https://learn.microsoft.com/en-us/azure/?product=popular>
- [17] Inc, Back4App: Back4App - Low-code backend to build modern apps. (s.f.). Back4app™. <https://www.back4app.com/>
- Inc, Back4App: Documentation and learning content. (s.f.). Back4app™. <https://www.back4app.com/docs>
- [18] PostgreSQL 15.3 Documentation. (2023). PostgreSQL. <https://www.postgresql.org/docs/current/>
- [19] Figma. (s.f.). <https://help.figma.com/hc/en-us>
- [20] Software 3ds Max | Producto oficial 3ds Max 2024. (2024). Autodesk. <https://www.autodesk.es/products/3ds-max/overview>
- [21] Async | Dart Package. (2013). Dart packages. <https://pub.dev/packages/async>
- [22] Parse\_server\_sdk\_flutter | Flutter Package. Dart packages. (2021). [https://pub.dev/packages/parse\\_server\\_sdk\\_flutter](https://pub.dev/packages/parse_server_sdk_flutter)
- [23] Package\_info\_plus | Flutter Package. (2020). Dart packages. [https://pub.dev/packages/package\\_info\\_plus](https://pub.dev/packages/package_info_plus)
- [24] Json\_annotation | Dart Package. (2018). Dart packages. [https://pub.dev/packages/json\\_annotation](https://pub.dev/packages/json_annotation)
- [25] Provider | Flutter Package. (2019). Dart packages. <https://pub.dev/packages/provider>.
- [26] Model\_viewer\_plus | Flutter Package. (2021). Dart packages. [https://pub.dev/packages/model\\_viewer\\_plus](https://pub.dev/packages/model_viewer_plus)
- [27] Perfect\_volume\_control | Flutter Package. (2021). Dart packages. [https://pub.dev/packages/perfect\\_volume\\_control](https://pub.dev/packages/perfect_volume_control)
- [28] Flag | Flutter Package. (2020). Dart packages. <https://pub.dev/packages/flag>
- [29] Millet, S., & Tune, N. (2015). Patterns, Principles, and Practices of Domain-Driven Design. <https://books.google.es/books?hl=en&lr=&id=iefGBwAAQBAJ&oi=fnd&pg=PR35&dq=domain+driven+design&ots=DkccGMlvMu&sig=R6550I9E1CT5hCVBiVCTHjOtDCE#v=onepage&q=domain%20driven%20design&f=false>.
- González Sánchez, J. L., Padilla Zea, N., & Gutiérrez, F. L. (2009). From usability to playability: Introduction to player-centred video game development process. En Human centered design (pp. 65–74). Springer Berlin Heidelberg. [https://doi.org/10.1007/978-3-642-02806-9\\_9](https://doi.org/10.1007/978-3-642-02806-9_9)

Gross, T., Gulliksen, J., Kotzé, P., Oestreicher, L., Palanque, P., Prates, R. O., & Winckler, M. (Eds.). (2009). Human-Computer interaction – INTERACT 2009. Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-642-03655-2>

"I Love All the Bits" | Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems. (2016). ACM Conferences.  
<https://dl.acm.org/doi/abs/10.1145/2858036.2858433>

Set of people avatars round icons with faces Free Vector. (2022). Freepik.  
[https://www.freepik.com/free-vector/set-people-avatars-round-icons-with-faces\\_32886006.htm#page=2&query=characters&position=3&from\\_view=search&track=sph#position=3&page=2&query=characters](https://www.freepik.com/free-vector/set-people-avatars-round-icons-with-faces_32886006.htm#page=2&query=characters&position=3&from_view=search&track=sph#position=3&page=2&query=characters)