

Grado en Ingeniería Informática de Gestión y Sistemas de Información

**MARCADOR DE PÁDEL POR VOZ
DESARROLLO DE UNA APLICACIÓN MÓVIL PARA ANDROID**

MEMORIA

DANIEL SALINAS ABELLÁN

TUTORA: ROSA HERRERO ANTÓN

2023-2024

ABSTRACT

The objective of this project is to create an intuitive and easy-to-use application, through the use of voice, that eases monitoring of the score at all times of a paddle tennis match using voice commands. In addition, it seeks to provide users with a simple and quick way to view previously played matches, thus improving their experience and knowledge of their performance in the game.

RESUMEN

El objetivo de este proyecto es crear una aplicación intuitiva y fácil de usar, mediante el uso de la voz, que facilite el seguimiento del marcador en todo momento de un partido de pádel mediante comandos de voz. Además, se busca proporcionar a los usuarios una manera sencilla y rápida de visualizar los partidos jugados anteriormente, mejorando así su experiencia y conocimiento sobre su desempeño en el juego.

RESUM

L'objectiu d'aquest projecte és crear una aplicació intuïtiva i fàcil de fer servir, mitjançant l'ús de la veu, que faciliti el seguiment del marcador en tot moment d'un partit de pàdel mitjançant ordres de veu. A més, es busca proporcionar als usuaris una manera senzilla i ràpida de visualitzar els partits jugats anteriorment, millorant així la seva experiència i coneixement sobre el seu exercici en el joc.

ÍNDICE

ABSTRACT.....	2
RESUMEN.....	2
RESUM.....	2
1. Introducción.....	1
2. Marco Teórico.....	3
2.1 Contexto Pádel.....	3
2.2 Contexto Reconocimiento de Voz.....	4
3. Objetivos y Alcance.....	7
3.1 Objetivos del producto.....	7
3.2 Objetivos de usuario.....	9
3.3 Target.....	15
4. Antecedentes históricos.....	17
4.1 Antecedentes actuales No Relacionados con el Pádel.....	18
4.1.1 Cortana.....	18
4.1.2 Siri.....	18
4.1.3 Alexa.....	18
4.2 Antecedentes actuales Relacionados con el Pádel.....	19
4.2.1 SCOREAPP.....	19
4.2.2 Padel Watch: Padel Scorekeeper.....	20
4.2.3 Padel (Marcador por Voz).....	20
5. Metodología.....	21
5.1 Definición del proyecto.....	21
5.2 Metodología de trabajo.....	25
5.2.1 Sprint 1: Implementación Marcador y Reconocimiento Voz.....	26
5.2.2 Sprint 2: Funcionamiento del Marcador por Voz y Selección de Cantidad de Sets.....	28
5.2.3 Sprint 3: Implementación de Modo de Ventaja y Desempate de Sets.....	29
5.2.4 Sprint 4: Diseño, Testing e Implementación de la Interfaz Definitiva.....	31
5.2.5 Sprint 5: Integración con Firebase y Gestión de Usuarios.....	33
5.2.6 Sprint 6: Personalización de Idioma, Voz del Narrador y Cantidad de Información.....	35
5.2.7 Sprint 7: Testing de Interfaces e Iteración.....	36
5.2.8 Sprint 8: Mejora de Calidad y Nuevas Funcionalidades.....	37
5.3 Tareas de codificación y diseño.....	38
5.4 Requerimientos Funcionales.....	39
5.5 Requerimientos Tecnológicos.....	41
6. Desarrollo.....	43
6.1 Tecnología Usada.....	43
6.1.1 Android Studio.....	43
6.1.2 Firebase Firestore Database.....	45
6.1.3 Firebase Authentication.....	47

6.2 Reconocimiento de Voz.....	49
6.2.1 Reconocimiento de Voz Recursivo.....	52
6.3 Interfaz.....	53
6.3.1 Pantalla Inicio de Sesión.....	53
6.3.2 Pantalla Registro.....	54
6.3.3 Pantalla Menú Principal.....	56
6.3.4 Pantalla Ajustes.....	57
6.3.5 Desarrollo Ajustes.....	59
6.3.6 Pantalla Datos de Usuario.....	60
6.3.7 Pantalla Iniciar Partido.....	62
6.3.8 Pantalla Marcador.....	65
6.3.9 Restablecimiento de Contraseña.....	67
6.3.10 Personalización Componentes.....	68
6.4 Base de Datos.....	68
6.4.1 Firebase.....	68
6.4.2 Estructura.....	70
6.4.3 Conexión.....	73
6.4.3 Uso.....	74
6.5 Icono de la aplicación.....	77
7. Testing.....	79
6.5.1 Primera Iteración: Entrevistas Interfaces.....	79
6.5.2 Segunda Iteración: Entrevistas Interfaces.....	80
6.5.3 Tercera Iteración: Entrevistas Interfaces.....	81
6.5.4 Cuarta Iteración: Entrevistas Interfaces.....	82
6.5.5 Primera Iteración: Entrevistas Funcionales.....	82
8. Conclusiones.....	83
9. Posibles Ampliaciones.....	85
9.1 Funcionamiento completo mediante el uso de la voz.....	85
9.2 Posibilidad de guardar estadísticas del partido.....	86
9.3 Creación de Modo de Campeonato.....	86
10. Bibliografía.....	87

ÍNDICE DE FIGURAS

Figura 3.2.1 Caso de Uso - Cambiar cantidad de sets.....	9
Figura 3.2.2 Caso de Uso - Establecer equipo y jugador que empieza sacando.....	9
Figura 3.2.3 Caso de Uso - Iniciar el partido.....	10
Figura 3.2.4 Caso de Uso - Visualizar Marcador.....	10
Figura 3.2.5 Caso de Uso - Cambiar Marcador.....	11
Figura 3.2.6 Caso de Uso - Escuchar la evolución del partido.....	11
Figura 3.2.7 Caso de Uso - Finalizar Partido.....	12
Figura 3.2.8 Caso de Uso - Crear cuenta de usuario.....	12
Figura 3.2.9 Caso de Uso - Iniciar sesión.....	13
Figura 3.2.10 Caso de Uso - Cambiar Datos de Usuario.....	13
Figura 3.2.11 Caso de Uso - Visualizar partidos anteriores.....	14
Figura 6.2.1 Diagrama de secuencia de Reconocimiento de Voz.....	51
Figura 6.3.1.1 Esquema de pantalla de inicio de sesión.....	53
Figura 6.3.1.2 Pantalla de inicio de sesión final.....	54
Figura 6.5.2.1 Esquema de pantalla de registro.....	55
Figura 6.5.2.2 Pantalla de registro final.....	55
Figura 6.3.3.1 Esquema de pantalla de menú principal.....	56
Figura 6.3.3.2 Pantalla de menú principal final.....	57
Figura 6.3.4.1 Esquema de pantalla de ajustes.....	58
Figura 6.3.4.2 Pantalla de ajustes final.....	59
Figura 6.3.6.1 Esquema de pantalla de datos de usuario.....	61
Figura 6.3.6.2 Pantalla de datos de usuario final.....	61
Figura 6.3.7.1 Esquema de pantalla de inicio de partido.....	63
Figura 6.3.7.2 Pantalla de inicio de partido final.....	64
Figura 6.3.7.3 Pantalla añadir jugadores al partido.....	64
Figura 6.3.8.1 Esquema de pantalla marcador vertical y horizontal, respectivamente.....	65
Figura 6.3.8.2 Pantalla marcador vertical y horizontal finales, respectivamente.....	66
Figura 6.3.9.1 Diagrama de secuencia de restablecimiento de Contraseña.....	67
Figura 6.4.2.1 Estructura Base de datos Autenticación.....	70
Figura 6.4.2.2 Estructura General Base de datos.....	70
Figura 6.4.2.3 Estructura base de datos Resultados de partidos.....	71
Figura 6.4.2.4 Estructura base de datos Datos de Usuario.....	72
Figura 6.4.3.1 Diagrama de secuencia de registro de usuario.....	74
Figura 6.4.3.2 Diagrama de secuencia de inicio de sesión.....	75
Figura 6.4.3.3 Diagrama de secuencia de obtención de datos de usuario.....	75
Figura 6.4.3.4 Diagrama de secuencia de actualización de datos de usuario.....	75
Figura 6.4.3.5 Diagrama de secuencia obtención historial de partidos.....	76
Figura 6.4.3.6 Diagrama de secuencia de guardado de resultados.....	76

1. Introducción

En la era digital actual, las aplicaciones móviles han revolucionado numerosos aspectos de nuestra vida cotidiana, incluido el ámbito deportivo. Con el crecimiento del pádel como uno de los deportes de raqueta más populares, surge la necesidad de innovar en la manera en que los jugadores gestionan y disfrutan de sus partidos. En este contexto, el presente trabajo se enfoca en el desarrollo de una aplicación móvil que simula el marcador de un partido de pádel, pero con una peculiaridad que lo hace destacar entre otras herramientas existentes: la capacidad de entender las órdenes verbales de los jugadores y actualizar el marcador de forma automática en respuesta a estas instrucciones.

Esta aplicación no solo busca simplificar la tarea de llevar la puntuación durante un partido de pádel, sino que también introduce una nueva dimensión de interactividad y comodidad para los jugadores. Al permitir que la aplicación reconozca y procese las comandas verbales de los jugadores, se elimina la necesidad de intervención manual en la actualización del marcador, lo que agiliza el desarrollo del juego y minimiza posibles errores o disputas sobre la puntuación.

A lo largo de este trabajo, se explorarán los fundamentos técnicos necesarios para implementar esta funcionalidad, incluyendo el procesamiento de voz, la lógica de actualización del marcador y la integración con una plataforma móvil. Además, se analizarán las implicaciones prácticas de esta innovación en el contexto del pádel, así como las posibles mejoras y expansiones futuras que podrían incorporarse a la aplicación.

En resumen, este trabajo representa un paso adelante en la convergencia entre la tecnología móvil y el mundo del deporte, ofreciendo una solución inteligente y práctica para mejorar la experiencia de juego de los aficionados al pádel. La aplicación propuesta no solo

simplifica la gestión del marcador, sino que también promueve la interactividad y la comodidad para los jugadores, consolidando su posición como una herramienta indispensable en cualquier partido de pádel.

2. Marco Teórico

2.1 Contexto Pádel

El pádel es un deporte que se juega en parejas y que combina elementos del tenis y del squash. El objetivo principal del juego es golpear la pelota con la pala para que pase por encima de la red y no pueda ser devuelta por el equipo contrario.

[1] En cuanto al marcador, el pádel se juega a partidos al mejor de tres sets. Cada set se gana alcanzando primero seis juegos, pero con una diferencia mínima de dos juegos sobre el equipo contrario. Si el marcador llega a 6-6, se juega un tie-break para determinar el ganador del set. En el tie-break, se juegan puntos de manera continua y el equipo que gane al menos siete puntos con una diferencia de dos sobre el otro equipo gana el set.

Durante el juego, se utiliza un sistema de puntuación similar al del tenis. El marcador comienza en 0-0 y se cuenta de la siguiente manera: 15, 30, 40 y juego. Si ambos equipos alcanzan 40 puntos, se llama "iguales" o "deuce". A partir de ahí, uno de los equipos debe ganar dos puntos consecutivos para ganar el juego, lo que se conoce como "ventaja". Si el equipo que tiene la ventaja gana el siguiente punto, ganan el juego. Si el equipo contrario gana el punto, el marcador vuelve a "deuce" y el juego continúa hasta que un equipo gane por dos puntos de diferencia.

No obstante, en 2020, se añadió una nueva forma de jugar el punto de ventaja. Esta nueva forma consiste en que, cuando ambos equipos llegan a 40 iguales, se juega un único punto más y el que gana ese punto, automáticamente gana el juego. Es decir, sin necesidad de ganar por dos puntos de diferencia en el punto de ventaja.

En cuanto al cambio de lado de la pista para sacar, los jugadores cambian de lado después de que se haya disputado un número impar de juegos en cada set. Por lo general, esto significa que se cambia de lado después de los primeros juegos (es decir, después del primer, tercer y quinto juego) y luego después de cada juego completo (es decir, después del séptimo juego y así sucesivamente).

Por último, en cuanto al cambio de lado del equipo, este se produce al final de cada set. Después de que se haya completado un set, los equipos cambian de lado de la pista para empezar el siguiente set desde una posición equitativa.

2.2 Contexto Reconocimiento de Voz

El reconocimiento de voz es una tecnología que permite a las máquinas entender, interpretar y procesar el lenguaje humano hablado, conocidos como Sistemas de reconocimiento automático del habla (ASR).

Utiliza algoritmos y modelos de aprendizaje automático para convertir las señales de audio, es decir, las palabras habladas, en texto o comandos entendibles por la computadora.

En general, el proceso de reconocimiento de voz implica varias etapas [2]:

- **Captura del Audio:** El proceso comienza con la grabación del audio que contiene el habla humana. Esto puede ser a través de un micrófono en un dispositivo, como un teléfono móvil o una computadora.
- **Preprocesamiento:** El audio capturado puede contener ruidos o interferencias. En esta etapa, se realiza un preprocesamiento para limpiar el audio, eliminando ruidos no deseados o mejorando la calidad para facilitar el reconocimiento.

- **Extracción de Características:** El habla humana tiene componentes distintivos, como fonemas, tonos, ritmos y patrones. Aquí, se extraen estas características clave del audio para comprender mejor el contenido y la intención del hablante.
- **Modelo de Reconocimiento:** Se utiliza un modelo matemático o algoritmos de aprendizaje automático para interpretar las características extraídas. Estos modelos han sido entrenados para reconocer patrones y convertirlos en texto.
- **Decodificación y Transcripción:** El modelo de reconocimiento analiza las características extraídas y las convierte en texto, transcribiendo lo que se ha dicho en lenguaje humano.
- **Salida del Texto Reconocido:** El resultado final es el texto transcrito que representa la información hablada, que puede ser utilizado por dispositivos, aplicaciones o sistemas para ejecutar comandos, realizar búsquedas, transcripciones de texto, etcétera.

3. Objetivos y Alcance

3.1 Objetivos del producto

- Proporcionar un correcto funcionamiento del marcador de pádel.
- Permitir seleccionar la cantidad de sets por partido.
- Permitir seleccionar el equipo que empieza sacando.
- Facilitar la visualización del marcador del partido.
- Reconocer los comandos por voz del usuario.
- Procesar los comandos.
- Cambiar la puntuación del marcador a través del procesamiento de los comandos.
- Emitir sonidos para comunicar a los jugadores la evolución del partido.

- **Objetivos Secundarios**
 - Permitir inicio de sesión a los usuarios introduciendo el correo electrónico y una contraseña.
 - Recuperar los datos de inicio de sesión de los usuarios de una base de datos externa.

- Permitir registro de los usuarios introduciendo el correo electrónico, una contraseña, nombre y apellidos y sexo.
- Comprobar si el correo electrónico introducido por el usuario ya está en uso.
- Guardar los datos de registro de los usuarios en una base de datos externa.
- Permitir a los usuarios cambiar sus datos personales.
- Actualizar los datos personales de los usuarios en el usuario correcto de la base de datos.
- Guardar el resultado del partido en la base de datos.
- Permitir la visualización de los partidos anteriores jugados de los usuarios.

3.2 Objetivos de usuario

- Seleccionar la cantidad de sets que se jugará.

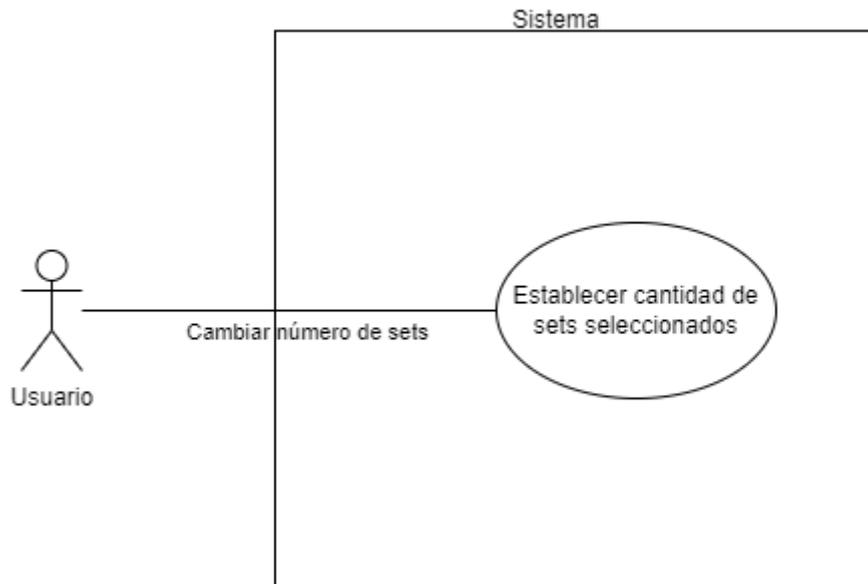


Figura 3.2.1 Caso de Uso - Cambiar cantidad de sets

- Seleccionar el equipo y jugador que empieza sacando.

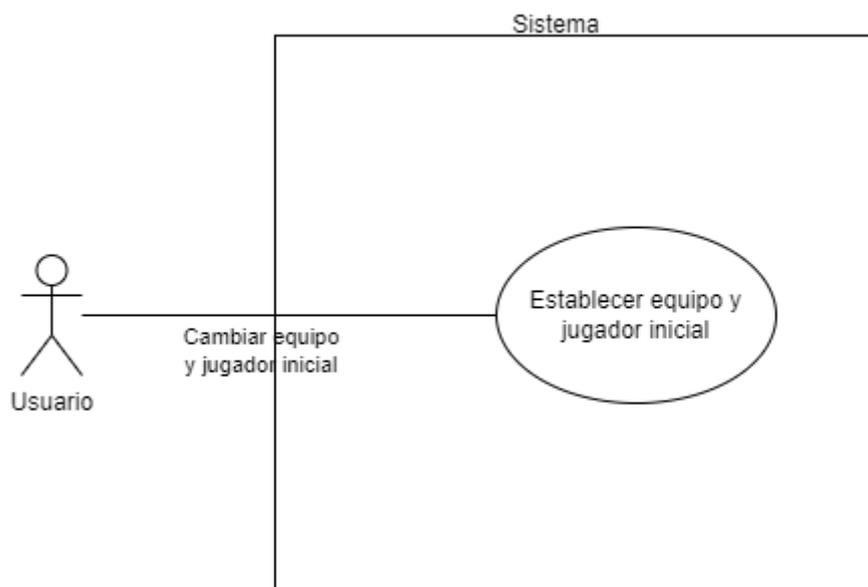


Figura 3.2.2 Caso de Uso - Establecer equipo y jugador que empieza sacando

- Iniciar el partido.

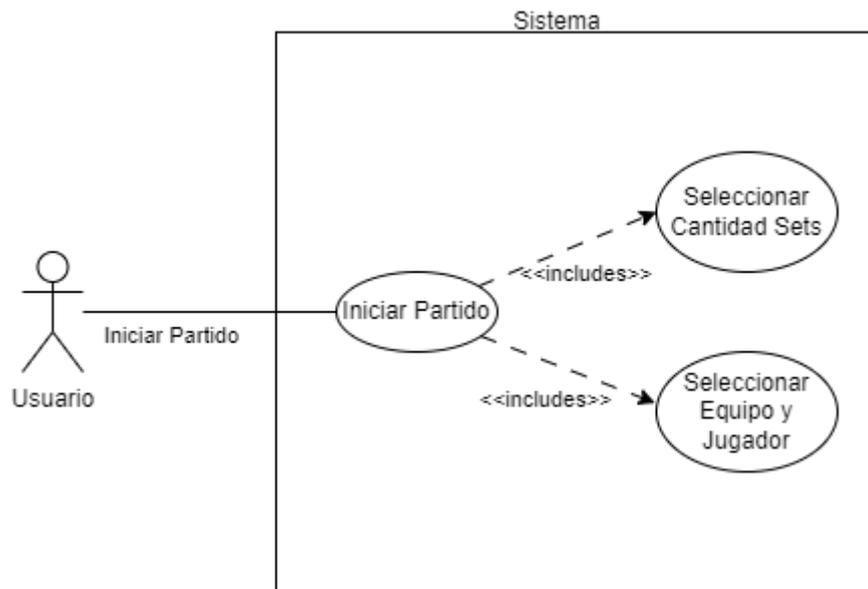


Figura 3.2.3 Caso de Uso - Iniciar el partido

- Visualizar el marcador del partido.

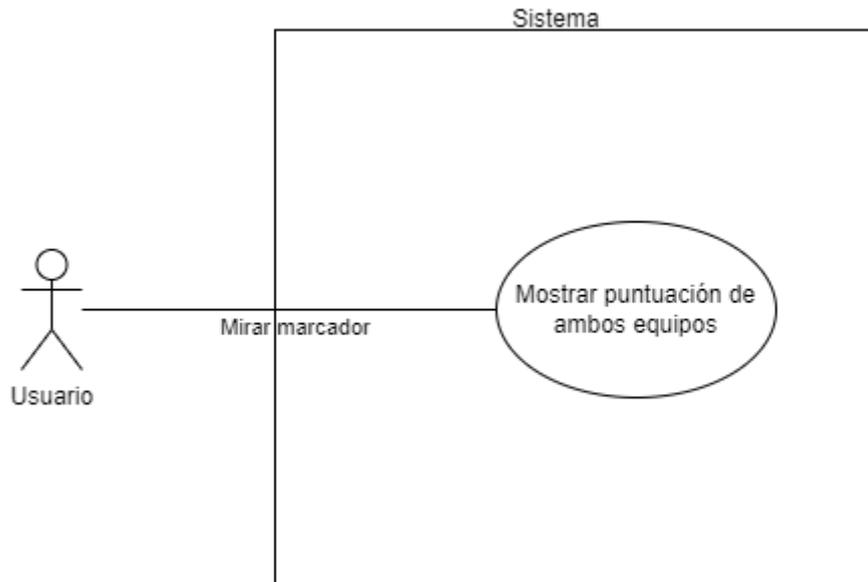


Figura 3.2.4 Caso de Uso - Visualizar Marcador

- Poder cambiar el marcador del partido mediante comandos de voz.

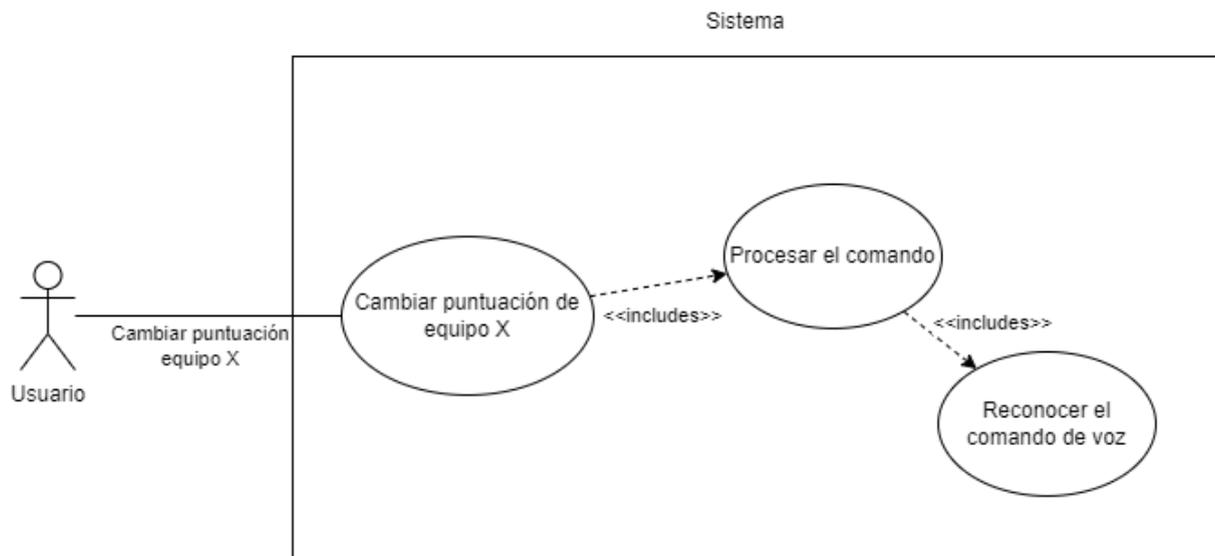


Figura 3.2.5 Caso de Uso - Cambiar Marcador

- Poder escuchar la evolución del partido a medida que se juega.

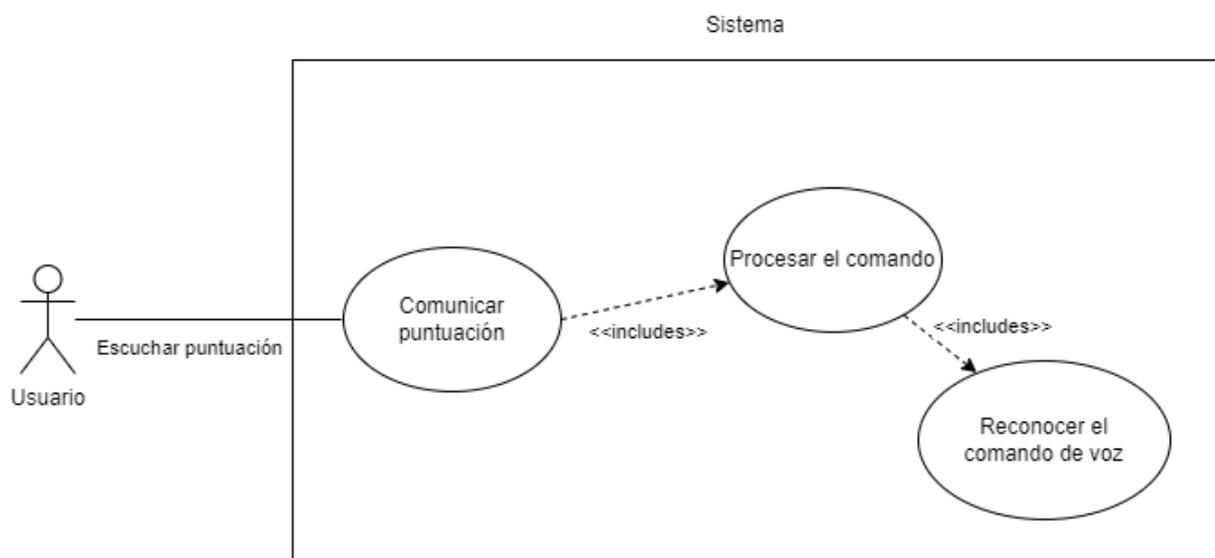


Figura 3.2.6 Caso de Uso - Escuchar la evolución del partido

- Poder finalizar el partido.

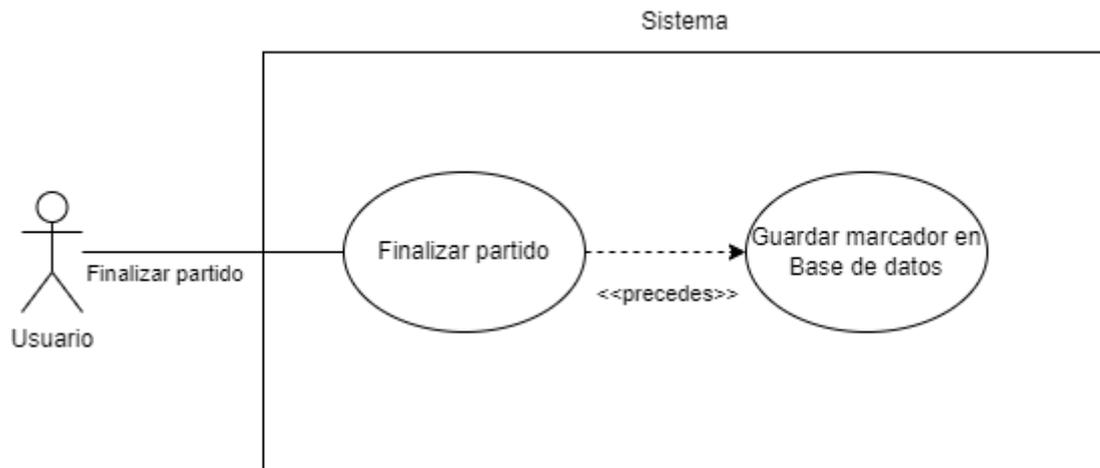


Figura 3.2.7 Caso de Uso - Finalizar Partido

- **Objetivos Secundarios**

- Poder crear una cuenta de usuario.

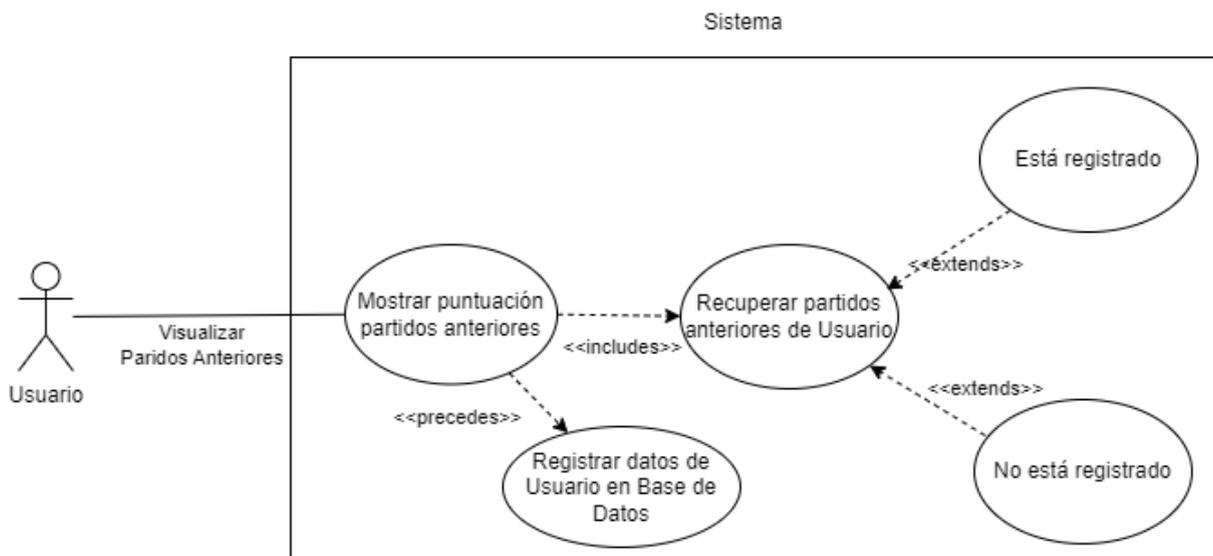


Figura 3.2.8 Caso de Uso - Crear cuenta de usuario

- Poder iniciar sesión.

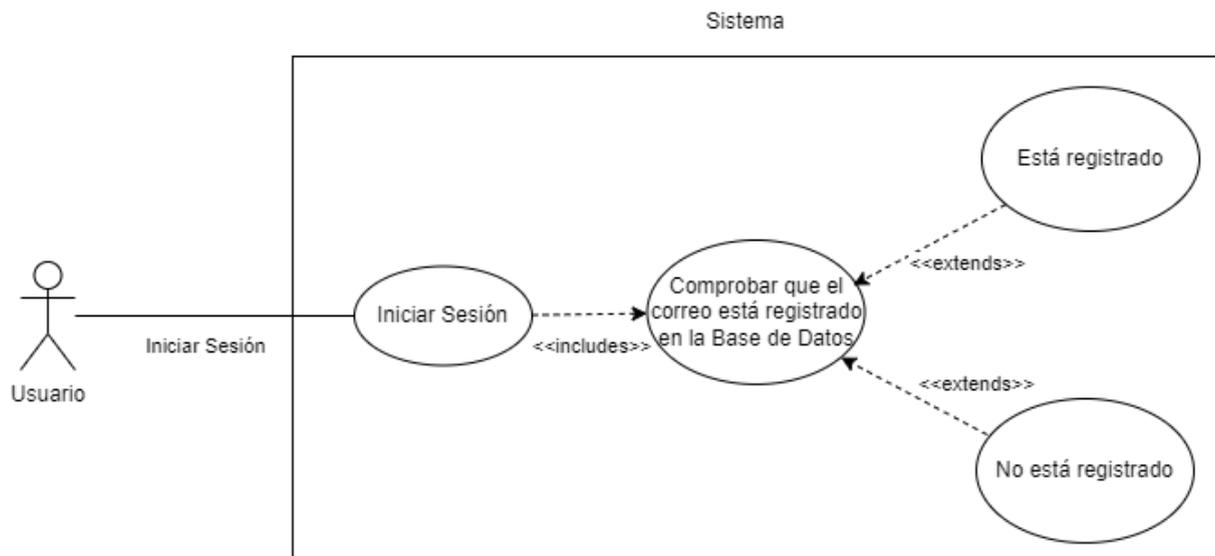


Figura 3.2.9 Caso de Uso - Iniciar sesión

- Poder cambiar los datos de usuario introducidos anteriormente.

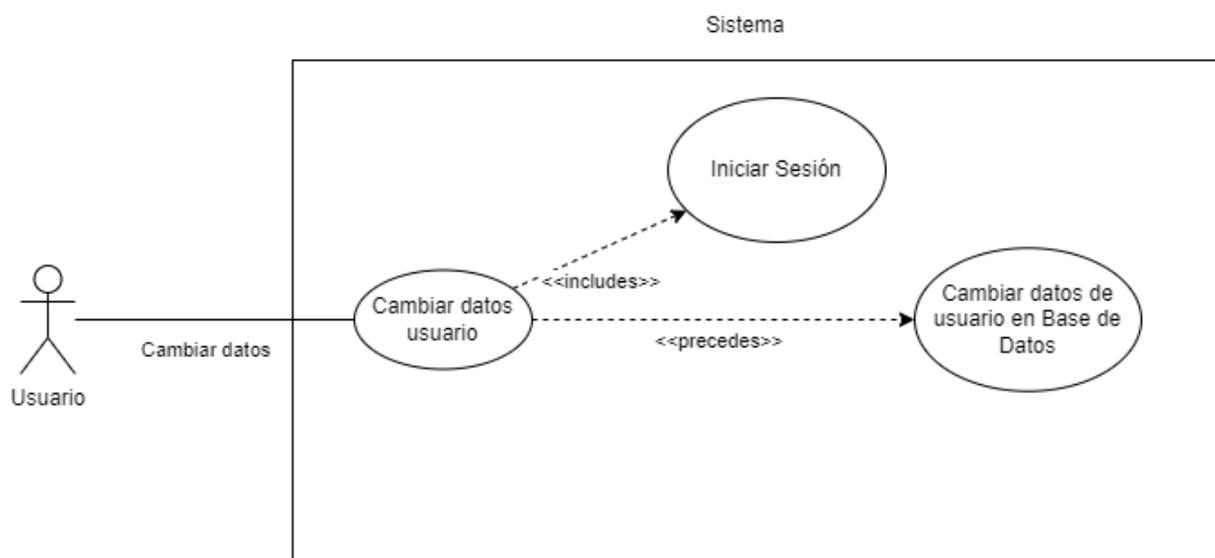


Figura 3.2.10 Caso de Uso - Cambiar Datos de Usuario

- Poder visualizar partidos anteriores.

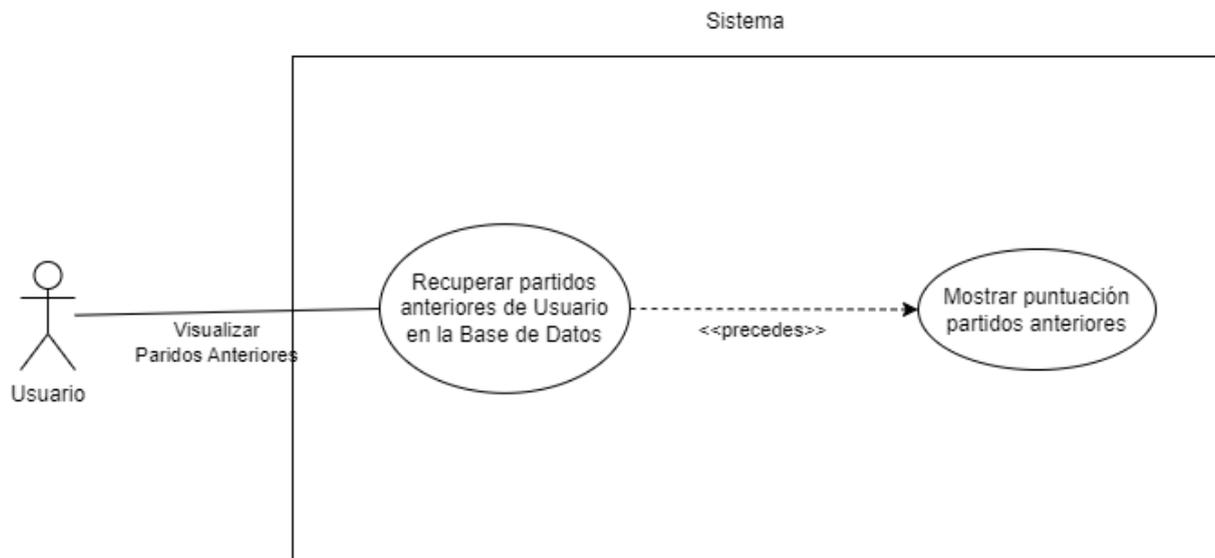


Figura 3.2.11 Caso de Uso - Visualizar partidos anteriores

3.3 Target

La aplicación está destinada a las personas que juegan a pádel de manera casual. Este grupo incluye desde principiantes hasta personas que juegan esporádicamente por diversión o como actividad social. Se dirige a quienes buscan mejorar su experiencia en el pádel sin estar enfocados en un rendimiento profesional.

La encuesta nacional llevada a cabo por Media Sports Marketing [3] entre personas de 18 a 55 años revela una evolución significativa en el pádel. Este deporte ha experimentado un crecimiento constante, con proyecciones que indican un aumento del doble en el número de practicantes en los próximos cinco años, pasando de 2 a 4 millones.

De los 26 millones en el rango de edad estudiado, el 8% juega regularmente, un 7,3% ocasionalmente, y un 22,5% está dispuesto a probarlo. Cerca de 2 millones de individuos participan frecuente u ocasionalmente, y 1,8 millones lo hacen esporádicamente, sumando casi 4 millones de jugadores en total.

Este informe resalta un cambio social importante. En los últimos 6 años, el pádel ha pasado de ser considerado un deporte elitista, mayormente practicado por grupos socioeconómicos selectos y adultos maduros, a ser más accesible para las clases medias y jóvenes. En 2004, el 31% de los jugadores pertenecía a la alta sociedad y el 23% tenía entre 45 y 55 años. En contraste, en 2010, el 62% pertenece a la clase media y media-alta, mientras que el 66% tiene entre 20 y 45 años.

Un aspecto relevante es el crecimiento en la participación femenina, que ahora representa más del 35% de los practicantes, especialmente en edades comprendidas entre los 20 y 35 años. Este cambio en la demografía del pádel es un punto clave para la comprensión del público objetivo de este deporte en evolución.

Se busca simplicidad y utilidad, una herramienta que mejore la experiencia de juego sin requerir un conocimiento técnico avanzado. Por tanto, la aplicación se enfoca en facilitar el seguimiento del marcador de manera práctica y sencilla, manteniendo el disfrute del juego como prioridad principal.

4. Antecedentes históricos

El reconocimiento de voz ha experimentado una evolución a lo largo de décadas, gracias a numerosos precursores clave en su desarrollo [4]:

- **Bell Labs:** En los años 50, crearon "Audrey", un dispositivo para reconocer dígitos en entornos controlados, estableciendo las bases para la industria telefónica.
- **IBM:** En los 60, desarrollaron el primer sistema de reconocimiento basado en patrones y en los 80 lograron reconocer 20.000 palabras.
- **Dragon Systems:** En los 90, lanzaron Dragon NaturallySpeaking, el primer reconocimiento de voz para PC de uso comercial masivo.
- **Google:** Alrededor de 2010, con Google Voice Search, evolucionando a Google Assistant con IA y aprendizaje automático para interactuar y responder a usuarios.
- **Amazon:** También alrededor de 2010, presentó Alexa, utilizando reconocimiento de voz para controlar dispositivos y obtener información.

Esta tecnología ha evolucionado enormemente y se emplea en diversas aplicaciones actualmente, con expectativas de más avances en el futuro.

4.1 Antecedentes actuales No Relacionados con el Pádel

En la era digital actual, hay diversas compañías que utilizan el reconocimiento de la voz humana para crear los llamados “Asistentes Personales”. Algunos de los más famosos son:

4.1.1 Cortana

[5] Es la asistente personal de Microsoft que fue lanzada en abril de 2014 en Windows y que sirve para gestionar el calendario, establecer recordatorios y alarmas, abrir aplicaciones, etc...

4.1.2 Siri

[6] Es la asistente personal de Apple que fue lanzada en octubre de 2011 en los dispositivos Apple y que sirve para realizar tareas cotidianas con la voz.

4.1.3 Alexa

[7] Es la asistente personal de Amazon que fue lanzada en noviembre de 2014 y que sirve para crear una forma más intuitiva de interactuar con la tecnología que usan las personas a diario.

[8] - [10] Estas “Asistentes Personales”, se basan en sistemas de reconocimiento automático del habla (ASR). Estos sistemas incluyen capacidades mejoradas para filtrar el ruido de fondo, reconocer el habla con acento y distinguir entre palabras y frases que deben ser consideradas y las que deben ser ignoradas.

El machine learning se emplea de manera efectiva para potenciar la comprensión del lenguaje y personalizar las experiencias de los usuarios con el tiempo. Esta tecnología

permite que los sistemas aprendan de las interacciones individuales, adaptándose a las preferencias de cada usuario.

A medida que se interactúa con estas plataformas, el machine learning las capacita para comprender mejor el habla, el contexto y los patrones de comportamiento, para dar así una experiencia más personalizada y precisa a lo largo del tiempo.

4.2 Antecedentes actuales Relacionados con el Pádel

En la actualidad, aún no se ha desarrollado una aplicación de marcador de pádel que funcione exclusivamente mediante comandos de voz.

Pero sí existen dos aplicaciones que tienen un funcionamiento muy similar al de la aplicación que se desarrollará pero estas funcionan para Apple Watch y Smartphones.

4.2.1 SCOREAPP

Es una aplicación que está pensada para ligas y torneos amateurs.

Está vinculada con el sistema SCOREPAD [11], el cual es una forma de gestionar todas las puntuaciones y cronometrajes deportivos.

El modo de funcionamiento de esta aplicación es simple [12].

Los jugadores llevarán una pulsera con la instalación SCOREPAD instalada en ella, mientras que el sistema SCOREPAD es manejado por un árbitro o persona ajena al partido.

Cuando un equipo consigue un punto, esta persona tendrá que sumarlo al marcador.

A su vez, el marcador de todas las pulseras y dispositivos smartphone con la aplicación SCOREAPP, se verán actualizados, mostrando el marcador actual.

El problema de esta aplicación es que, a pesar de ser muy cómoda para los jugadores, se depende de una persona externa que lleve el marcador.

4.2.2 Padel Watch: Padel Scorekeeper

Es una aplicación que se debe instalar tanto en el Apple Watch como en un Iphone. Una vez instalados en ambos, se debe vincular uno con otro.

El funcionamiento de esta aplicación es simple [13].

El jugador debe primero seleccionar la cantidad de sets que se jugarán en el partido y, de forma manual, ir actualizando el marcador del partido, pulsando sobre el equipo que ha conseguido el punto.

Una vez acabado el partido, le dará a “Guardar” y el resultado del partido se guardará.

En la propia aplicación se podrán ver los resultados de partidos anteriores y estadísticas de estos.

4.2.3 Padel (Marcador por Voz)

Esta aplicación, desarrollada en Android 4.0 [14], ofrecía la posibilidad de seleccionar la cantidad de sets y el equipo que inicia el juego, además de permitir el control del marcador mediante comandos de voz. Sin embargo, debido a que fue diseñada para una versión anterior de Android, ya no es compatible con las versiones más recientes admitidas por la Play Store (a partir de Android 13.0) [15]. Actualmente, no está disponible para instalación en dispositivos Android.

5. Metodología

5.1 Definición del proyecto

Estudio Previo

Esta fase consiste en la recopilación y análisis de información relevante antes de comenzar una tarea o desarrollo. En este proceso se investigan antecedentes, datos, recursos y cualquier detalle necesario para comprender el contexto y los desafíos a enfrentar.

Estudio del mercado

Antes de iniciar el desarrollo de la aplicación, se lleva a cabo un exhaustivo análisis del mercado. Esto incluye un estudio detallado de las aplicaciones existentes relacionadas con el pádel. Se identifican y analizan sus fortalezas y debilidades para comprender mejor el panorama actual y encontrar oportunidades de mejora y diferenciación.

Listado de tareas

Una vez completado el análisis del mercado, se procede a la creación de un listado detallado de tareas para el desarrollo de la aplicación.

Requerimientos de la aplicación

Se establecen los requisitos específicos de la aplicación en base a las conclusiones obtenidas del análisis del mercado y del listado de tareas. Esto abarca desde la interfaz de usuario hasta la arquitectura técnica necesaria para garantizar un rendimiento óptimo y una experiencia de usuario satisfactoria.

Estudio de viabilidad

Para asegurar la viabilidad del proyecto, se realiza un estudio detallado que evalúa aspectos técnicos, financieros y operativos. Esto implica la estimación de costos, recursos necesarios, plazos de desarrollo y la evaluación de posibles obstáculos que puedan surgir durante el proceso.

Identificar objetivos de la aplicación

En la fase de identificación de objetivos de una aplicación, se define claramente el propósito y las metas que se buscan alcanzar con su desarrollo. Esto implica comprender las necesidades del usuario, establecer qué problemas resolverá la app y determinar qué beneficios ofrecerá a quienes la utilicen. Se analizan y priorizan funcionalidades, se definen las características principales y se delimitan las expectativas y el alcance del proyecto.

Implementación marcador

Antes de iniciar con el reconocimiento de voz, se hará la implementación del marcador. Durante esta fase, se utilizarán dos botones para subir la puntuación de los equipos y dos botones para reducir la puntuación. No obstante, solo se hará el funcionamiento de un único set durante esta fase.

Implementación reconocimiento de voz

Una vez acabada la implementación del marcador, se iniciará con el reconocimiento de voz. De esta manera, cuando el usuario hable, la aplicación mostrará en un cuadro de texto lo que ha dicho, para saber si lo dicho por el usuario es correcto o no.

Conjuntar marcador y reconocimiento de voz

Una vez testeado y comprobado que el reconocimiento de voz funciona correctamente, se empezará a conjuntar con el funcionamiento del marcador.

Implementación de selección de sets

Durante esta fase, se busca ajustar la duración del partido según la cantidad de sets que el usuario elija previamente, asegurando que la aplicación se adapte a las preferencias del jugador antes de iniciar el juego.

Implementación comunicación del estado del marcador.

Durante esta etapa del desarrollo, se incorporará la funcionalidad para que la aplicación informe a los usuarios sobre el estado actual del marcador. Esto permitirá una interacción dinámica, brindando actualizaciones en tiempo real sobre el progreso del partido o juego en curso.

Diseño de la interfaz de la aplicación

En esta etapa, el enfoque principal se dirige hacia la creación de una interfaz intuitiva y fácil de usar para los usuarios, asegurando una experiencia clara y accesible en la aplicación.

Implementación del diseño de la Interfaz

En esta fase, el enfoque principal está en llevar a cabo la implementación de la interfaz de usuario diseñada previamente. Se busca plasmar el diseño visual y funcionalidades previstas para asegurar una experiencia intuitiva y amigable para el usuario.

Implementación conexión con Base de Datos

Durante esta fase, se emplea Android Studio para establecer la conexión con la base de datos de Firebase. Esta integración permitirá el intercambio de datos necesario para el funcionamiento de la aplicación, garantizando así un flujo eficiente y seguro de la información entre la app y la base de datos.

Implementación registro de cuenta

Una vez establecida la conexión, se procederá a la codificación que permitirá a los usuarios crear sus cuentas dentro de la aplicación. Esta información se almacenará de manera segura en la base de datos de Firebase, asegurando la integridad y privacidad de los datos de cada usuario.

Implementación inicio de sesión

En esta etapa se codifica el proceso de inicio de sesión para los usuarios. Se verificará en la base de datos de Firebase si la cuenta de correo proporcionada existe y si la contraseña ingresada coincide con los registros almacenados. Este proceso asegura el acceso autorizado y la autenticación segura de los usuarios en la plataforma.

Cambio de datos de usuario

Durante esta fase de desarrollo, se implementa la funcionalidad que permite a los usuarios modificar sus datos de inicio de sesión. Esto incluye la capacidad de actualizar la dirección de correo electrónico o cambiar la contraseña, brindando a los usuarios un control activo sobre su información de acceso a la plataforma.

Comprobación resultado partidos anteriores

En esta fase, se trabajará en la codificación para que los usuarios puedan acceder y visualizar los partidos anteriores que hayan jugado. Esto implica recuperar estos registros de la base de datos y presentarlos de manera organizada para su fácil acceso y consulta.

5.2 Metodología de trabajo

Metodología SCRUM

En el proceso de desarrollo de la aplicación, se ha decidido implementar la metodología ágil conocida como Scrum. Esta metodología se centra en dividir el proyecto en partes más pequeñas y manejables, denominadas "sprints", durante los cuales se enfoca en tareas específicas para lograr avances concretos en períodos de tiempo definidos.

Scrum promueve la flexibilidad y la colaboración, permitiendo adaptaciones rápidas a medida que se avanza en el proyecto. A través de reuniones periódicas y una comunicación cercana, se busca mantener un ritmo constante de progreso y ajustar el enfoque según sea necesario para alcanzar los objetivos establecidos.

Android Studio

El software escogido para desarrollar la aplicación es Android Studio. Es un entorno de desarrollo integrado (IDE) que ofrece herramientas esenciales para la creación, depuración y pruebas de aplicaciones. Su amplio conjunto de características, como el emulador de dispositivos, la gestión de versiones y la compatibilidad con diferentes lenguajes de programación, facilita la creación de apps móviles de manera eficiente y con soporte para múltiples versiones de Android.

Firestore

La base de datos escogida para usar durante el desarrollo del proyecto es Firestore. Esta base de datos ofrece una infraestructura sólida y fácil de usar para el almacenamiento y gestión de datos en aplicaciones móviles. Su integración con Android Studio facilita enormemente el desarrollo, permitiendo a los desarrolladores enfocarse en la experiencia del usuario en lugar de preocuparse por la configuración de servidores y la gestión de la base de datos.

Esta plataforma provee servicios esenciales como la autenticación de usuarios, almacenamiento en la nube y bases de datos en tiempo real entre otros. Su capacidad para escalar según las necesidades del proyecto, garantiza un crecimiento flexible y eficiente de la aplicación. Además, Firebase ofrece una interfaz amigable y una amplia documentación que simplifica el proceso de desarrollo y reduce considerablemente el tiempo requerido para implementar funcionalidades clave en la aplicación móvil.

5.2.1 Sprint 1: Implementación Marcador y Reconocimiento Voz

Duración del Sprint: cuatro semanas

Objetivo: Desarrollar una interfaz de prueba para un marcador de pádel de un único set que permite subir y bajar la cantidad de puntos obtenidos por cada jugador, y añadir la funcionalidad de reconocimiento de voz.

Tareas:

1. Configuración Inicial del Proyecto:

- Se configura un nuevo proyecto en Android Studio.

2. Desarrollo del Marcador de Pádel (Interfaz de Prueba):

- Se crea una interfaz de usuario básica para el marcador de pádel.
 - Se diseña una pantalla simple con dos secciones para cada jugador.
 - Se incluyen botones para incrementar y decrementar los puntos de cada jugador.
 - Se muestra el puntaje actual de cada jugador con un diseño simple.
- Se implementa la lógica para actualizar los puntos.
 - Se programan los botones de incremento y decremento de puntos.
 - Se asegura de que los puntos no sean negativos.

3. Integración de Reconocimiento de Voz:

- Se añaden funcionalidades de reconocimiento de voz utilizando la API de reconocimiento de voz de Android.

- Se configuran los permisos necesarios para el uso del micrófono y la API de reconocimiento de voz.
- Se implementan comandos de voz para incrementar y decrementar puntos de cada jugador.
 - Ejemplos de comandos: "Punto Equipo A", "Quitar Equipo B".
- Se proporciona feedback visual y auditivo al usuario después de cada comando de voz reconocido.

4. Pruebas y Validación:

- Se realizan pruebas funcionales para asegurar que los botones de incremento y decremento funcionan correctamente.
- Se prueba la funcionalidad de reconocimiento de voz.
- Se asegura de que la aplicación maneje correctamente los errores de reconocimiento de voz.

5.2.2 Sprint 2: Funcionamiento del Marcador por Voz y Selección de Cantidad de Sets

Duración del Sprint: dos semanas

Objetivo: Implementar el control del marcador de pádel mediante comandos de voz y añadir la funcionalidad de seleccionar la cantidad de sets.

Tareas:

1. Integración Completa del Reconocimiento de Voz:

- Se configura y optimiza la API de reconocimiento de voz para controlar el marcador.
- Se definen y programan los comandos de voz para las siguientes acciones:
 - Incrementar y decrementar puntos para cada jugador (por ejemplo, "Aumentar punto jugador 1", "Disminuir punto jugador 2").
 - Establecer y restablecer el marcador (por ejemplo, "Restablecer marcador").
- Se proporciona feedback visual o auditivo al usuario después de cada comando de voz reconocido.
- Se realizan pruebas para asegurar la precisión y efectividad del reconocimiento de voz en diferentes condiciones (ruido de fondo, distintas pronunciaciones, etc.).

2. Desarrollo de la Funcionalidad para Seleccionar la Cantidad de Sets:

- Se actualiza la interfaz de usuario (de prueba) para incluir una opción que permita seleccionar la cantidad de sets.
 - Se añade un menú desplegable o una serie de botones para seleccionar entre 1, 3 o 5 sets.
 - Se asegura que la interfaz sea intuitiva y fácil de usar.
- Se implementa la lógica para manejar múltiples sets.

- Se ajusta la lógica del marcador para rastrear y mostrar los resultados de cada set.
- Se asegura de que la funcionalidad de incrementar y decrementar puntos se aplique al set actual.
- Se maneja la transición entre sets y determina el ganador final basado en la cantidad de sets seleccionada.

3. Pruebas y Validación:

- Se realizan pruebas funcionales para asegurar que los comandos de voz para controlar el marcador funcionan correctamente.
- Se asegura que la transición entre sets funciona sin problemas y que el marcador se actualiza correctamente.

5.2.3 Sprint 3: Implementación de Modo de Ventaja y Desempate de Sets

Duración del sprint: una semana y media

Objetivo: Permitir al usuario establecer el modo de ventaja y la forma de desempatar los sets, e implementar estas funcionalidades.

Tareas:

1. Configuración del Modo de Ventaja:

- Se actualiza la interfaz de usuario para incluir una opción que permita seleccionar el modo de ventaja.
 - Se añade un interruptor o menú desplegable para elegir entre "Ventaja" y "Punto de Oro".
 - Se asegura que la interfaz sea intuitiva y fácil de usar.
- Se implementa la lógica para manejar ambos modos:

- Para el modo "Ventaja": el juego sigue la regla tradicional donde los jugadores deben ganar por dos puntos consecutivos después de llegar a 40-40.
- Para el modo "Punto de Oro": el siguiente punto después de 40-40 decide el ganador del juego.

2. Configuración del Desempate de Sets:

- Se actualiza la interfaz de usuario para incluir una opción que permita seleccionar la forma de desempatar los sets.
 - Se añade un menú desplegable para elegir entre diferentes métodos de desempate.
- Se implementa la lógica para manejar los diferentes métodos de desempate:
 - Se ajusta la lógica del marcador para iniciar y rastrear el tie break cuando los jugadores llegan al final de un set empatados.
 - Se asegura de que la puntuación y las reglas específicas del tie break se aplican correctamente.

3. Integración y Actualización de la Interfaz:

- Se integran las nuevas opciones en la pantalla de configuración del partido.
- Se actualizan las pantallas de marcador y estadísticas para reflejar los modos de ventaja y desempate seleccionados.
- Se proporciona feedback visual o auditivo cuando se aplican las reglas de ventaja y desempate durante el partido.

4. Pruebas y Validación:

- Se realizan pruebas funcionales para asegurar que las opciones de ventaja y desempate se configuran y aplican correctamente.
- Se prueban los distintos escenarios de juego para verificar que las reglas de ventaja y desempate se ejecutan según lo esperado.
- Se asegura que la interfaz de usuario refleja correctamente el estado del partido y las reglas aplicadas.

5.2.4 Sprint 4: Diseño, Testing e Implementación de la Interfaz Definitiva

Duración del sprint: tres semanas

Objetivo: Diseñar la interfaz definitiva para la aplicación, realizar pruebas de usabilidad con usuarios y proceder a la implementación de la interfaz final.

Tareas:

1. Diseño de la Interfaz Definitiva:

○ Creación de Prototipos:

- Se diseñan todas las interfaces.
- Se aseguran que los diseños sean intuitivos, atractivos y fáciles de navegar.

2. Pruebas de Usabilidad:

○ Selección de Usuarios:

- Se selecciona un grupo diverso de usuarios para realizar pruebas de usabilidad.

○ Organización de Sesiones de Testing:

- Se organizan sesiones de testing en las que los usuarios interactúan con los prototipos de la interfaz.
- Se utilizan métodos de testing como entrevistas, encuestas y observación directa.

○ Recopilación y Análisis de Feedback:

- Se recopila el feedback de los usuarios sobre la usabilidad, funcionalidad y diseño estético de la interfaz.
- Se analizan los datos recopilados para identificar áreas de mejora y posibles problemas de usabilidad.

3. Implementación de la Interfaz Definitiva:

- **Desarrollo Frontend:**
 - Se implementan los diseños finales en Android Studio, asegurando que el código sea limpio, eficiente y bien documentado.
 - Se aplican estilos y componentes visuales según el diseño definitivo.
- **Integración con Funcionalidades Existentes:**
 - Se integran las funcionalidades existentes (marcador, reconocimiento de voz, selección de sets, modos de ventaja y desempate) con la nueva interfaz.
 - Se asegura que todas las funciones operen sin problemas dentro del nuevo diseño.
- **Pruebas y Validación:**
 - Se realizan pruebas exhaustivas para garantizar que la nueva interfaz funcione correctamente en diversos dispositivos y tamaños de pantalla.
 - Se aseguran que no haya errores ni problemas de rendimiento con la nueva interfaz.

4. Optimización y Mejoras:

- **Refinamiento Basado en Feedback:**
 - Se hacen ajustes y mejoras adicionales basadas en el feedback recibido durante las pruebas de usabilidad.
- **Optimización de UX/UI:**
 - Se optimizan elementos de la interfaz para mejorar la experiencia del usuario, como tiempos de carga, animaciones y transiciones.

5.2.5 Sprint 5: Integración con Firebase y Gestión de Usuarios

Duración del sprint: una semana

Objetivo: Establecer la conexión con Firebase Firestore y Firebase Authentication, e implementar los métodos de registro, inicio de sesión, y la funcionalidad para que los usuarios puedan cambiar sus datos.

Tareas:

1. Configuración Inicial de Firebase:

○ Configuración de Firebase en el Proyecto Android:

- Se crea un proyecto en Firebase y se configura la base de datos Firestore y Authentication.
- Se añade el archivo de configuración **google-services.json** al proyecto Android.
- Se añaden las dependencias necesarias para Firebase Firestore y Firebase Authentication en el archivo **build.gradle**.

2. Implementación del Registro de Usuarios:

○ Lógica de Registro:

- Se implementa la lógica para crear una nueva cuenta de usuario utilizando Firebase Authentication.
- Se guarda la información adicional del usuario (por ejemplo, nombre) en Firestore.

○ Validación y Manejo de Errores:

- Se añaden validaciones para los campos de entrada (correo electrónico válido, contraseña segura, etc.).
- Se manejan errores comunes como correo electrónico ya registrado, errores de red, etc.

3. Implementación del Inicio de Sesión:

○ Diseño de la Pantalla de Inicio de Sesión:

- Se diseña la pantalla de inicio de sesión donde los usuarios pueden introducir su correo electrónico y contraseña.
 - **Lógica de Inicio de Sesión:**
 - Se implementa la lógica para autenticar usuarios existentes utilizando Firebase Authentication.
 - **Manejo de Sesiones:**
 - Se maneja el inicio y cierre de sesión, asegurando que los usuarios autenticados permanezcan conectados entre sesiones de la aplicación.
- 4. Implementación de la Funcionalidad para Cambiar Datos del Usuario:**
- **Diseño de la Pantalla de Perfil:**
 - Se diseña la pantalla de perfil donde los usuarios pueden ver y editar su información personal (nombre, correo electrónico, etc.).
 - **Lógica para Actualizar Datos del Usuario:**
 - Se implementa la lógica para actualizar la información del usuario en Firestore.
 - Se implementa la lógica para actualizar la información de autenticación en Firebase Authentication, como el correo electrónico y la contraseña.
 - **Validación y Manejo de Errores:**
 - Se añaden validaciones para los campos de entrada y se maneja la actualización de datos de manera segura.
 - Se manejan errores comunes como fallos de autenticación, errores de red, etc.

5.2.6 Sprint 6: Personalización de Idioma, Voz del Narrador y Cantidad de Información

Duración del sprint: tres semanas.

Objetivo: Permitir a los usuarios cambiar el idioma de la aplicación, la voz del narrador y la cantidad de información que el narrador proporciona.

Tareas:

1. Configuración Inicial y Preparación:

○ Internacionalización:

- Preparar archivos de recursos de traducción (strings.xml) para los idiomas soportados.

2. Implementación de Cambio de Idioma:

○ Lógica de Cambio de Idioma:

- Implementar la lógica para cambiar el idioma de la aplicación dinámicamente basándose en la selección del usuario.
- Asegurar que todos los textos de la aplicación se actualicen al nuevo idioma seleccionado.

○ Validación:

- Realizar pruebas para asegurar que el cambio de idioma se aplique correctamente en toda la aplicación.

3. Implementación de Cambio de Voz del Narrador:

○ Lógica de Selección de Voz:

- Implementar la lógica para cambiar la voz del narrador utilizando el motor TTS.
- Proveer opciones de diferentes voces (masculina, femenina).

○ Validación:

- Realizar pruebas para asegurar que la voz del narrador se cambie correctamente según la selección del usuario.

4. Implementación de la Cantidad de Información Proporcionada por el Narrador:

- **Lógica de Cantidad de Información:**
 - Implementar la lógica para ajustar la cantidad de información que el narrador proporciona basándose en la selección del usuario.
- **Validación:**
 - Realizar pruebas para asegurar que la cantidad de información proporcionada por el narrador se ajuste correctamente según la selección del usuario.

5.2.7 Sprint 7: Testing de Interfaces e Iteración

Duración del sprint: una semana

Objetivo: Realizar pruebas exhaustivas de las interfaces desarrolladas, corregir errores descubiertos durante el testing y realizar nuevos tests para garantizar la calidad y usabilidad de las interfaces.

Tareas:

1. Pruebas de Interfaces Existente:

- Realizar pruebas de usabilidad en todas las interfaces desarrolladas hasta el momento.
- Identificar y registrar cualquier problema de diseño, navegación o experiencia del usuario.

2. Corrección de Errores y Problemas:

- Priorizar los problemas identificados durante las pruebas de usabilidad.
- Realizar correcciones en las interfaces según los problemas encontrados.
- Asegurar que las correcciones sean consistentes con los requisitos del proyecto y la experiencia de usuario deseada.

3. Nuevo Testing de Interfaces Corregidas:

- Realizar pruebas adicionales en las interfaces corregidas para verificar que los problemas han sido resueltos satisfactoriamente.
- Solicitar feedback de usuarios beta si es posible para validar las correcciones realizadas.

4. Iteración y Mejoras Adicionales:

- Basándose en los resultados del nuevo testing, realizar iteraciones adicionales en las interfaces si es necesario.
- Realizar mejoras adicionales en el diseño, la navegación o la usabilidad según sea necesario.
- Asegurar que las interfaces estén listas para su implementación final en el siguiente sprint.

5.2.8 Sprint 8: Mejora de Calidad y Nuevas Funcionalidades

Duración del sprint: una semana

Objetivo: Realizar el testing del funcionamiento de la aplicación, arreglar los problemas encontrados durante el testing y desarrollar dos funcionalidades clave: el botón de comandos y desarrollar un nuevo comando para poder para el reconocimiento de voz.

Tareas:

1. Testing del Funcionamiento de la Aplicación:

- Ejecutar pruebas funcionales en toda la aplicación para asegurar su correcto funcionamiento.

2. Arreglo de Problemas Encontrados:

- Analizar y priorizar los problemas vistos durante la fase de testing.
- Verificar que los problemas se han solucionado adecuadamente mediante pruebas adicionales.

3. Desarrollo de la Funcionalidad del Botón de Comandos:

- Diseñar e implementar el botón de comandos en la interfaz de usuario.
- Integrar el botón con las funcionalidades correspondientes del sistema.
- Asegurar el correcto funcionamiento del botón.

4. Desarrollo de la Capacidad de Reconocimiento de Voz:

- Desarrollar la lógica necesaria para que la aplicación procese y responda al nuevo comando de voz.
- Asegurar el correcto funcionamiento del comando.

5.3 Tareas de codificación y diseño

Se codifica la configuración y el funcionamiento del marcador, el funcionamiento de la puntuación durante el partido y al final de este. Posteriormente, se procede con la codificación del reconocimiento de voz, seguido por la implementación de la traducción de voz a instrucciones.

A continuación, se centra en el reconocimiento de voz junto a su posterior traducción a instrucciones.

Finalmente, se diseña la interfaz de la aplicación, haciendo que sea muy fácil e intuitiva de usar. Una vez finalizada, se realiza la codificación necesaria para ponerla en la aplicación y poder utilizarla en ella.

Este proceso se realiza de forma estructurada para asegurar un avance productivo en cada etapa del desarrollo.

5.4 Requerimientos Funcionales

Los usuarios deben poder:

- Configurar la cantidad de sets antes del partido.
- Seleccionar el equipo que empieza sacando
- Iniciar el seguimiento del partido.
- Poder controlar el marcador por voz.
- Finalizar el partido.
- Ver el resultado del partido.
- Guardar resultados.
- Ver resultado de partidos anteriores
- Crear una cuenta.
- Iniciar sesión en la aplicación.

El sistema debería poder:

- Codificar el correcto funcionamiento del marcador, basándose en un marcador real de pádel.
- Permitir el desarrollo correcto del marcador en función de los sets establecidos por el jugador.
- Modificar la puntuación de los jugadores una vez detectado el comando de voz.
- Corregir la puntuación de los jugadores una vez detectado el comando de voz.
- Comentar la puntuación de los jugadores cuando se haya anotado un punto.
- Finalizar el partido automáticamente cuando uno de los jugadores gana.
- Emitir un sonido para que los jugadores comprendan que se ha acabado el partido.
- Mostrar el resultado final a los jugadores.
- Guardar el resultado en la base de datos si el jugador así quiere.
- Mostrar el resultado de los partidos anteriores, recuperando los datos de la base de datos.
- Permitir a los usuarios crear una cuenta de usuario.
- Comprobar si los datos introducidos, a la hora de crear una cuenta, no son duplicados.

- Guardar los datos introducidos.
- Permitir a los usuarios iniciar sesión en la aplicación.
- Comprobar si los datos introducidos son correctos.
- Permitir modificar los datos de usuario comprobando si los nuevos datos introducidos están duplicados.

5.5 Requerimientos Tecnológicos

- Usar Firebase Firestore Database como base de datos.
- Usar Firebase Authentication para registrar los datos de usuario.
- Desarrollar la aplicación con Android Studio 2022.3.1.21
 - Utilizar API 27: Android 8.1.
 - Usar Java para el desarrollo del marcador.
 - Usar XML para el desarrollo de la interfaz.
 - Usar NoSQL para el uso de la base de datos.

6. Desarrollo

6.1 Tecnología Usada

6.1.1 Android Studio

Se ha escogido Android Studio para el desarrollo del proyecto por varios motivos.

En primer lugar, [16] es un IDE (Entorno Integrado de Desarrollo) especializado, diseñado específicamente para el desarrollo de aplicaciones Android. Proporciona un conjunto completo de herramientas y características que simplifican el proceso de desarrollo, incluyendo un editor de código, depurador, emulador de dispositivos y gestión de versiones.

Además, Android Studio ofrece una interfaz intuitiva y fácil de usar que facilita la creación y edición de código, así como la navegación por el proyecto. También cuenta con plantillas y asistentes que agilizan el proceso de desarrollo.

Android Studio está integrado con servicios y bibliotecas de Google, como Firebase, Google Maps y Google Play Services, lo que facilita la incorporación de funcionalidades avanzadas a las aplicaciones, como análisis de datos, autenticación de usuarios y mapas interactivos. Además, cuenta con una amplia comunidad de desarrolladores y una sólida documentación que facilita la resolución de problemas y el aprendizaje continuo.

Por último, Android Studio se actualiza regularmente para incluir nuevas características y mejoras, manteniendo el entorno de desarrollo actualizado y optimizado. En resumen, Android Studio ofrece un entorno completo y especializado para el desarrollo de aplicaciones Android, con características avanzadas, facilidad de uso y soporte de la comunidad, lo que lo convierte en la opción ideal para los desarrolladores de aplicaciones móviles.

Alternativas a Android Studio

Flutter: [17] Es un kit de herramientas de código abierto para crear aplicaciones nativas de alta calidad para Android e iOS desde una sola base de código y utiliza un lenguaje de programación llamado Dart.

[18] La conexión con Firebase en Flutter se realiza exclusivamente a través de comandos, lo que puede resultar menos intuitivo y más costoso en comparación con la configuración mediante interfaz en Android Studio. Esta diferencia puede representar un desafío para aquellos desarrolladores que están acostumbrados a trabajar con interfaces gráficas de usuario más amigables y directas en Android Studio. Por otro lado, en el contexto de Flutter, el uso de Dart como lenguaje de programación agrega otro nivel de complejidad. El cambio a Dart implica una curva de aprendizaje adicional y requeriría mucho tiempo y esfuerzo adicionales para familiarizarse con el nuevo lenguaje y sus características.

Ionic: [19] Ionic es una plataforma de desarrollo de aplicaciones híbridas que ofrece una variedad de herramientas y servicios para crear aplicaciones móviles multiplataforma utilizando tecnologías web como HTML, CSS y JavaScript. Si bien Ionic es una opción popular para el desarrollo rápido de aplicaciones multiplataforma, tiene costos asociados que pueden afectar significativamente el presupuesto de desarrollo. Con un precio de 4199\$ al mes, el uso de Ionic podría resultar costoso para algunas empresas y desarrolladores, especialmente para proyectos con presupuestos limitados.

Por otro lado, Java y Kotlin son lenguajes de programación utilizados en el desarrollo nativo de Android, que ofrecen una alternativa más rentable para el desarrollo de aplicaciones móviles. Aunque el costo de usar Java o Kotlin no está directamente asociado con el lenguaje en sí, sino más bien con el desarrollo y mantenimiento de la aplicación, el uso de estos lenguajes en combinación con Android Studio, una herramienta de desarrollo gratuita, puede ser una opción más económica en comparación con Ionic.

En cuanto a la comparación de rendimiento, Java y Kotlin pueden ofrecer un rendimiento superior en términos de velocidad y eficiencia en comparación con HTML, CSS y JavaScript. Esto se debe a que Java y Kotlin son lenguajes compilados que se ejecutan directamente en la máquina virtual de Java en el caso de Java, o en el entorno de tiempo de ejecución de Kotlin en el caso de Kotlin. Por otro lado, HTML, CSS y JavaScript son lenguajes interpretados que dependen del navegador o del motor JavaScript para su ejecución, lo que puede resultar en una mayor sobrecarga de interpretación y procesamiento.

6.1.2 Firebase Firestore Database

Firebase Firestore Database es la opción escogida como base de datos para el proyecto por varias razones.

[20] En primer lugar, ofrece una solución de base de datos en tiempo real que permite sincronizar datos entre los dispositivos de los usuarios en tiempo real, lo que garantiza una experiencia fluida y consistente.

Además, Firestore Database es parte de la plataforma Firebase de Google, lo que significa que se integra perfectamente con otras herramientas y servicios de Firebase, como la autenticación de usuarios, el almacenamiento de archivos, la mensajería en la nube y las funciones en la nube. Esto simplifica el proceso de desarrollo al proporcionar un conjunto completo de herramientas para crear y escalar aplicaciones móviles de manera eficiente.

Otra ventaja de Firestore Database es su escalabilidad y rendimiento. Puede manejar grandes volúmenes de datos y proporcionar consultas rápidas y eficientes, lo que garantiza un rendimiento óptimo incluso en aplicaciones con millones de usuarios activos.

Además, Firestore ofrece una estructura de datos flexible basada en documentos y colecciones, lo que permite a los desarrolladores modelar y organizar los datos de manera que se adapten mejor a las necesidades específicas de su aplicación. Esto facilita la creación de aplicaciones altamente personalizadas y adaptables a medida que evolucionan las necesidades del negocio.

En resumen, Firebase Firestore Database es una opción sólida como base de datos para aplicaciones desarrolladas en Android Studio debido a su capacidad de sincronización en tiempo real, integración con otras herramientas de Firebase, escalabilidad y rendimiento, y su estructura de datos flexible. Estas características hacen que Firestore sea una opción popular entre los desarrolladores para crear aplicaciones móviles modernas y altamente funcionales.

Alternativas a Firebase Firestore Database

Realm: [21] Realm es una popular base de datos de objetos para aplicaciones móviles que ofrece un almacenamiento de datos rápido y eficiente. A diferencia de las bases de datos SQL tradicionales, Realm está diseñado específicamente para trabajar con objetos en lugar de tablas.

Sin embargo, en comparación con la integración de Firebase en Android Studio, trabajar con Realm puede ser más complicado ya que la configuración y el manejo de la base de datos se realizan principalmente a través de código en lugar de una interfaz gráfica intuitiva. Esto puede requerir un mayor conocimiento técnico y tiempo de desarrollo para implementar y mantener la base de datos en una aplicación móvil. Por otro lado, Firebase proporciona una serie de herramientas y servicios que se pueden configurar y administrar fácilmente a través de una interfaz gráfica, lo que hace que la integración y el mantenimiento sean más rápidos y accesibles para los desarrolladores.

SQL (Structured Query Language): [22] Es un lenguaje de programación estándar utilizado para gestionar y manipular bases de datos relacionales.

Aunque SQL es una herramienta poderosa para trabajar con datos, su configuración y administración pueden ser más complejas en comparación con Firebase. Configurar y administrar una base de datos SQL tradicional generalmente implica instalar y configurar un servidor SQL, crear y mantener tablas y relaciones, gestionar el rendimiento y la seguridad, entre otros aspectos.

En contraste, Firebase ofrece una alternativa más rápida y sencilla para el almacenamiento de datos en la nube. Al proporcionar una base de datos en la nube completamente administrada, Firebase elimina la necesidad de instalar y configurar un servidor SQL. Esto simplifica significativamente la configuración y administración de la base de datos, lo que puede ahorrar tiempo y recursos de desarrollo.

6.1.3 Firebase Authentication

[23] Firebase Authentication es una elección sólida para el registro de usuarios en aplicaciones desarrolladas en Android Studio debido a sus múltiples ventajas y características avanzadas. En primer lugar, Firebase Authentication proporciona un método seguro y confiable para autenticar a los usuarios, lo que garantiza la protección de los datos sensibles y la privacidad de los usuarios. Utiliza tecnologías de cifrado avanzadas para proteger las credenciales de inicio de sesión y garantizar la seguridad de la información del usuario.

Además, Firebase Authentication ofrece una amplia variedad de métodos de autenticación, incluyendo correo electrónico y contraseña, números de teléfono, Google, Facebook, Twitter y muchas más. Esto permite a los usuarios registrarse y acceder a la aplicación utilizando el

método de autenticación que les resulte más conveniente y familiar, lo que mejora la experiencia del usuario y reduce las barreras de entrada.

Otra ventaja de Firebase Authentication es su integración perfecta con otras herramientas y servicios de Firebase, como Firebase Firestore Database y Firebase Cloud Functions. Esto facilita la implementación de funcionalidades adicionales, como la personalización del contenido basada en el perfil del usuario, el seguimiento de eventos de inicio de sesión y la gestión de usuarios desde el panel de Firebase.

Alternativa a Firebase Authentication

Auth0: [24] Auth0 es un servicio de gestión de identidades y acceso que proporciona soluciones de autenticación seguras y personalizables para aplicaciones web y móviles. Con Auth0, los desarrolladores pueden implementar fácilmente funcionalidades de inicio de sesión seguro, registro de usuarios y gestión de identidades, sin la necesidad de construir y mantener una infraestructura de autenticación desde cero.

No obstante, Firebase Authentication está integrado de forma nativa con Firebase y otros servicios de Google, lo que facilita la integración con el ecosistema de Google Cloud Platform. Esto simplifica el desarrollo y la gestión de la autenticación, especialmente si ya estás utilizando otros servicios de Firebase o Google en tu aplicación. Además, Firebase Authentication ofrece una configuración rápida y sencilla a través de la consola de Firebase, lo que facilita la implementación y la gestión de la autenticación de usuarios.

Por otro lado, Auth0 puede requerir una configuración más detallada y compleja, ya que ofrece una amplia gama de opciones de personalización y configuración. Si bien esto proporciona una mayor flexibilidad y control sobre la experiencia de autenticación de los

usuarios, puede implicar un mayor tiempo y esfuerzo para implementarlo en comparación con Firebase Authentication.

6.2 Reconocimiento de Voz

Para poder reconocer la voz de los usuarios, se ha creado una clase llamada `VoiceRecognition`.

Cuando el usuario inicia la aplicación por primera vez, la clase `VoiceRecognition` verifica si se han otorgado los permisos necesarios para acceder al micrófono del dispositivo. Si los permisos no están concedidos, la clase solicitará al usuario que los autorice.

Una vez que se otorgan los permisos necesarios, `VoiceRecognition` inicializa el servicio de reconocimiento de voz provisto por el sistema operativo Android, configurando parámetros como el modelo de lenguaje libre y el idioma predeterminado para el reconocimiento de voz.

Para ello:

1. Se crea una nueva instancia de `Intent`, especificando `RecognizerIntent.ACTION_RECOGNIZE_SPEECH` como acción. Esto indica al sistema Android que debe iniciar el servicio de reconocimiento de voz.
2. Se configuran los parámetros del reconocimiento de voz mediante llamadas a `putExtra()`. En este caso, se establece el modelo de lenguaje (`EXTRA_LANGUAGE_MODEL`) como `LANGUAGE_MODEL_FREE_FORM`, lo que permite al usuario hablar libremente sin restricciones gramaticales. También se especifica el idioma (`EXTRA_LANGUAGE`) como el idioma predeterminado del dispositivo.

3. Finalmente, se llama al método `startListening()` del objeto `SpeechRecognizer`, pasando el `Intent` configurado como parámetro. Esto inicia el reconocimiento de voz y comienza a escuchar las entradas del usuario.

Esta clase reconoce el texto hablado por el usuario utilizando el servicio de reconocimiento de voz proporcionado por el sistema operativo Android. Este servicio se inicializa y configura dentro del método `initializeSpeechRecognizer()`.

Cuando el usuario habla y el servicio de reconocimiento de voz detecta que ha finalizado el habla, se invoca el método `onResults()` del objeto `RecognitionListener`, que está registrado para escuchar los eventos de reconocimiento de voz. Dentro de este método, se recuperan los resultados del reconocimiento de voz, que generalmente incluyen una lista de cadenas de texto representando las palabras o frases reconocidas por el servicio.

En este método, se obtiene la lista de resultados de reconocimiento de voz a partir de la clave `SpeechRecognizer.RESULTS_RECOGNITION` del paquete de resultados. Se verifica si la lista no está vacía y se recupera el texto reconocido en la primera posición de la lista. Posteriormente, este texto reconocido puede ser procesado para determinar las acciones que la aplicación debe llevar a cabo en respuesta al comando de voz del usuario.

Es importante mencionar que este proceso de reconocimiento de voz depende en gran medida de la precisión y capacidad del servicio de reconocimiento de voz proporcionado por el sistema operativo Android. Por lo tanto, la exactitud de la transcripción del habla del usuario puede variar según diversos factores como la calidad del audio, el acento del usuario, y el ruido ambiente, entre otros.

Una vez que se reconoce el texto hablado por el usuario, la clase interpreta los resultados para determinar las acciones a tomar. Esto implica la identificación de palabras clave y la asociación de comandos con operaciones específicas dentro de la aplicación.

Además de habilitar el reconocimiento de voz, la clase VoiceRecognition incorpora la funcionalidad de TextToSpeech, que convierte texto en voz. Esto permite a la aplicación proporcionar retroalimentación auditiva al usuario, respondiendo a sus comandos de voz con instrucciones habladas y comunicando así la puntuación de ambos equipos.

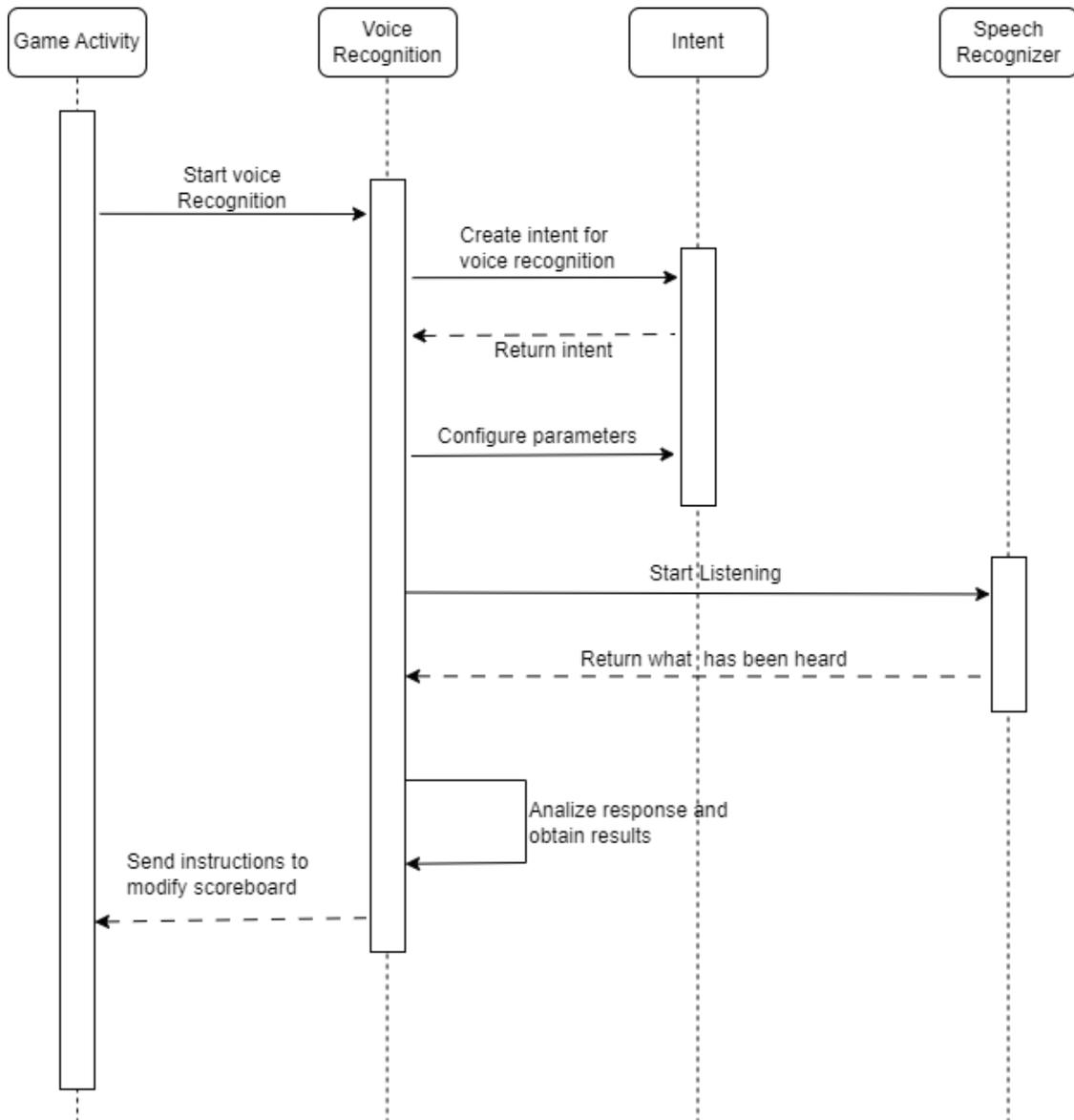


Figura 6.2.1 Diagrama de secuencia de Reconocimiento de Voz

Durante el desarrollo, el mayor desafío encontrado fue lograr que las respuestas se reprodujeran de manera secuencial, evitando que se superpusieran unas sobre otras. Este

problema requería una solución que garantizara una experiencia fluida y coherente para el usuario, sin interrupciones ni confusiones.

Para garantizar una respuesta coherente y ordenada a los comandos de voz del usuario, VoiceRecognition utiliza una cola de textos pendientes. Esto asegura que las respuestas de voz se entreguen en el orden correcto y de manera fluida, incluso cuando múltiples comandos están en espera de procesamiento.

En resumen, la clase VoiceRecognition es fundamental para mejorar la interactividad y accesibilidad de las aplicaciones Android mediante el reconocimiento de voz. Su implementación puede llevar a una experiencia del usuario más enriquecedora y satisfactoria.

6.2.1 Reconocimiento de Voz Recursivo

Para la implementación del reconocimiento de voz de manera recursiva en la aplicación, se utiliza un enfoque basado en Handler y Runnable, permitiendo la ejecución periódica de la función de reconocimiento de voz

Al utilizar un Handler, se programa la ejecución de un Runnable para que se repita a intervalos regulares. Dentro del Runnable, se llevó a cabo el proceso de reconocimiento de voz y cualquier acción asociada con los resultados obtenidos.

Este enfoque proporciona un control más granular sobre el proceso de reconocimiento de voz y permite gestionar mejor los recursos del sistema, ya que se puede ajustar fácilmente la frecuencia de la ejecución del reconocimiento de voz según sea necesario.

6.3 Interfaz

6.3.1 Pantalla Inicio de Sesión

En la pantalla de inicio de sesión, como se muestra en la figura 6.3.1.2, los usuarios tienen la oportunidad de ingresar a la aplicación móvil y acceder a todas sus funcionalidades.

Encuentran un botón que les permite acceder directamente a la pantalla de registro. Al hacer clic, son redirigidos de manera inmediata a la pantalla de registro donde pueden completar el proceso de registro para crear una nueva cuenta en la aplicación. Esta conexión directa agiliza el acceso para los nuevos usuarios y facilita el proceso de incorporación a la plataforma.

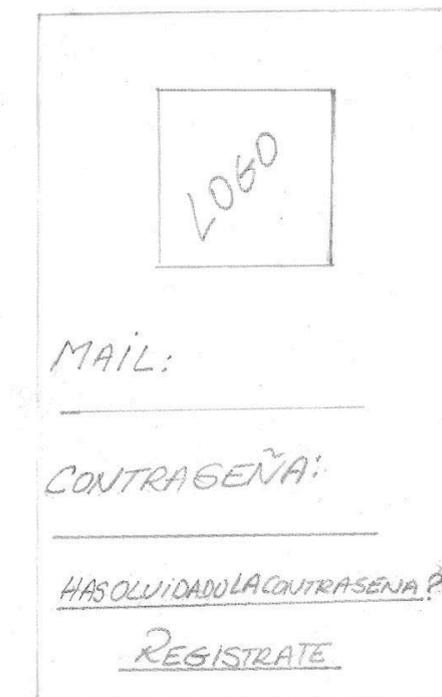


Figura 6.3.1.1 Esquema de pantalla de inicio de sesión



Figura 6.3.1.2 Pantalla de inicio de sesión final

6.3.2 Pantalla Registro

En la pantalla de registro, como se muestra en la figura 6.3.2.2, los usuarios tienen la oportunidad de crear una cuenta nueva en la aplicación. Deben proporcionar información básica, como nombre completo, dirección de correo electrónico, edad y una contraseña segura para crear su cuenta.

Además, también deben proporcionar un alias y su género.

A hand-drawn sketch of a registration screen layout. The text is written in a cursive, handwritten style. The fields are arranged vertically, each with a label and a horizontal line below it. The labels are: REGISTRARSE, NOMBRE Y APELLIDO, MAIL, CONTRASEÑA, EDAD, GÉNERO, and ALIAS.

Figura 6.5.2.1 Esquema de pantalla de registro

A final registration screen mockup. At the top, a status bar shows the time 9:51, signal strength, Wi-Fi, and 88% battery. Below the status bar, the title "REGISTRO" is centered in bold. The screen is divided into sections by horizontal lines. Each section has a title in bold and a corresponding input field with an icon and placeholder text. The sections are: "NOMBRE Y APELLIDO" with a person icon and "Escribe tu nombre y apellido"; "CORREO" with an envelope icon and "ejemplo@gmail.com"; "CONTRASEÑA" with a lock icon and a toggle eye icon; "EDAD" with a calendar icon and "Escribe tu edad"; "GÉNERO" with a dropdown menu showing "Masculino"; and "ALIAS" with a clipboard icon and "Cómo te verán los demás".

Figura 6.5.2.2 Pantalla de registro final

6.3.3 Pantalla Menú Principal

En la pantalla del menú principal, ver figura 6.3.3.2, los usuarios tienen acceso a las pantallas de ajustes, de datos de usuario, de inicio de partido y de historial de partidos.

En esta pantalla, cada usuario puede acceder fácilmente a un historial completo de sus partidos anteriores de pádel. Al poder ver la fecha en la que se jugaron, en qué equipo jugaron y el resultado de cada uno de ellos, los usuarios pueden realizar un seguimiento de su progreso, analizar su rendimiento y revivir experiencias pasadas.

Desde esta pantalla, los usuarios pueden acceder a sus datos de cuenta, a los ajustes de la aplicación y pueden iniciar un nuevo partido.



Figura 6.3.3.1 Esquema de pantalla de menú principal



Figura 6.3.3.2 Pantalla de menú principal final

6.3.4 Pantalla Ajustes

En la sección de ajustes de la aplicación, ver figura 6.3.4.2, los usuarios tienen la capacidad de personalizar diversas configuraciones para adaptar la experiencia de uso según sus preferencias individuales. En esta sección, se puede realizar las siguientes acciones:

Ajustar la velocidad del narrador: Se permite modificar la velocidad a la que el narrador de la aplicación proporciona información auditiva sobre el desarrollo del partido de pádel. Esta opción permite ajustar la velocidad de reproducción del narrador según sus preferencias de escucha.

Personalizar la cantidad de información proporcionada: Se permite escoger la cantidad de detalles que desean recibir del narrador durante el partido de pádel. Se podrá

seleccionar entre opciones que van desde una narración detallada de cada punto hasta un resumen más conciso del progreso del juego.

Acceder a los datos del usuario: En la pantalla de ajustes se incluye un botón claramente identificado que permite a los usuarios acceder fácilmente a la pantalla de datos de usuario. Al hacer clic en este botón, se puede acceder de manera rápida y conveniente a su perfil, garantizando un acceso sencillo a la gestión de su información personal en la aplicación.

Cambiar el idioma de la aplicación: Se permite seleccionar el idioma de preferencia para la interfaz de la aplicación. Esta opción permite adaptar la experiencia de uso a la lengua con la que se sientan más cómodos, facilitando la navegación y el acceso a las funcionalidades de la aplicación.

Seleccionar la voz del narrador: Se permite escoger si prefieren que el narrador sea una voz masculina o femenina. Esta opción permite personalizar aún más la experiencia auditiva según sus preferencias.

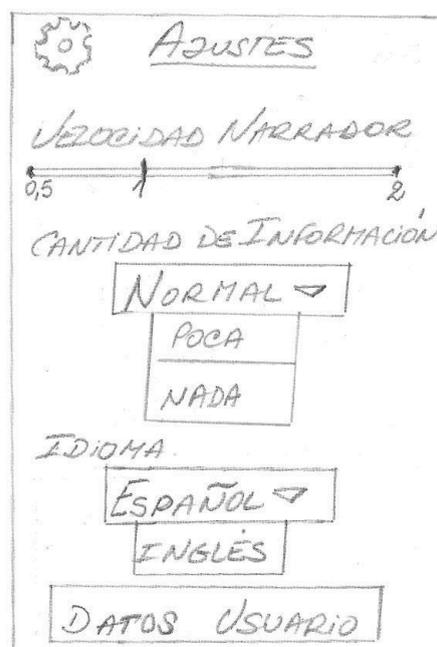


Figura 6.3.4.1 Esquema de pantalla de ajustes

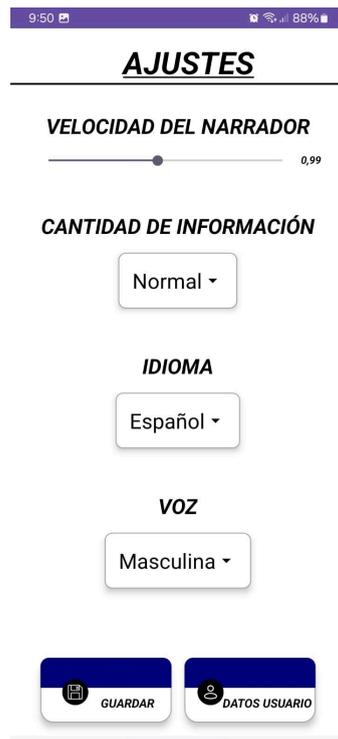


Figura 6.3.4.2 Pantalla de ajustes final

6.3.5 Desarrollo Ajustes

Para habilitar la sección de ajustes, se desarrolla una clase en Java que emplea el patrón Singleton. Esta elección se fundamenta en la necesidad de mantener una sola instancia de los ajustes en toda la aplicación, evitando así el desperdicio de recursos.

La clase en cuestión actúa como un único contenedor para todos los ajustes relevantes de la aplicación. Al seguir el patrón Singleton, se asegura que esta clase tenga una sola instancia durante la ejecución del programa. Por lo tanto, cualquier solicitud de la instancia de esta, devuelve siempre la misma instancia, con los mismos valores de configuración.

Este enfoque proporciona varias ventajas. En primer lugar, garantiza la coherencia de los ajustes en toda la aplicación, ya que cualquier cambio realizado se refleja de manera consistente en todas las partes que acceden a la instancia única. Además, al limitar la

creación a una única instancia, se minimiza el uso de memoria y se optimiza el rendimiento general de la aplicación.

Sin embargo, a pesar de la eficacia del patrón Singleton en la gestión de los ajustes de la aplicación, persiste un problema crucial: la posibilidad de que los ajustes se reinicien cada vez que el usuario cierra la aplicación. Para abordar esta cuestión y asegurar que las preferencias del usuario se mantengan persistentes a lo largo del tiempo, se incorpora el uso de SharedPreferences.

SharedPreferences es una herramienta clave en el desarrollo de aplicaciones Android, diseñada específicamente para almacenar datos de forma persistente mediante pares clave-valor. Al utilizar SharedPreferences en conjunto con el patrón Singleton, se puede garantizar que los ajustes de la aplicación se conserven incluso después de cerrarla y volver a abrirla.

Mediante SharedPreferences, los ajustes se almacenan en un archivo persistente en el dispositivo del usuario, lo que permite recuperarlos fácilmente en futuras sesiones de la aplicación. Esto significa que, una vez que el usuario configure sus preferencias, estas se mantendrán intactas hasta que decida cambiarlas, independientemente de cuántas veces abra o cierre la aplicación.

6.3.6 Pantalla Datos de Usuario

En la pantalla de datos del usuario, ver figura 6.3.6.2, se permite visualizar y gestionar la información asociada a su cuenta, como su nombre, dirección de correo electrónico, preferencias de comunicación y otros detalles relevantes. También se puede realizar cambios en su perfil, actualizar su contraseña o modificar cualquier otra información personal según sea necesario.

A hand-drawn wireframe of a user data form. The title is "DATOS USUARIO". Below it are six input fields, each with a pencil icon to its right, indicating they are for editing. The fields are labeled: "NOMBRE Y APELLIDO", "EDAD", "GÉNERO", "MAIL", "CONTRASEÑA", and "ALIAS".

Figura 6.3.6.1 Esquema de pantalla de datos de usuario

A screenshot of the final user data form on a mobile device. The title is "USER DATA". The form contains the following fields and values:

- NAME AND SURNAME:** Daniel Salinas
- E-MAIL:** daniel.salinas.a17@gmail.com
- PASSWORD:** (with a lock icon and a visibility toggle icon)
- AGE:** 24 (with a calendar icon)
- GENDER:** (empty)
- ALIAS:** Dani (with a tag icon)

At the bottom, there are two buttons: "CANCEL" (with a close icon) and "ACCEPT" (with a play icon).

Figura 6.3.6.2 Pantalla de datos de usuario final

6.3.7 Pantalla Iniciar Partido

En la pantalla de inicio de partido, según se muestra en la figura 6.3.7.2, se tiene la capacidad de personalizar varios aspectos del juego antes de comenzar. En este lugar, se puede realizar las siguientes acciones:

- **Seleccionar quién empieza:** Se les permite a los jugadores escoger qué jugador dará inicio al partido, lo que les permite establecer una dinámica justa desde el principio.
- **Definir la cantidad de sets:** Se permite a los jugadores elegir la cantidad de sets que se jugarán en el partido, lo que les permite adaptar la duración del juego según sus preferencias o el tiempo disponible.
- **Escoger el tipo de ventaja:** Se ofrece a los jugadores la opción de escoger entre dos tipos de ventaja durante el juego. La primera opción es la clásica, que sigue las reglas tradicionales del pádel. La segunda opción es conocida como "punto de oro", en la cual el primer jugador o equipo que consiga un punto gana automáticamente el punto de set, lo que añade emoción y rapidez al desenlace de cada set. Esta variedad en las reglas brinda a los jugadores la oportunidad de adaptar su estrategia de juego según sus preferencias y habilidades, añadiendo un elemento de elección y emoción al enfrentamiento en la pista.
- **Escoger desempate de sets:** Se ofrece a los jugadores la opción de escoger entre dos formas de desempatar los sets durante el juego. La primera opción es la forma clásica, en la cual un equipo debe ganar por al menos dos puntos de diferencia para asegurar la victoria del set. La segunda opción es el tie-break. En esta, cuando ambos equipos llegan a seis puntos, se juega una cantidad de puntos adicionales.

En este caso, el equipo que consigue siete puntos se lleva la victoria del set. Estas dos alternativas ofrecen a los jugadores la flexibilidad para elegir el método que mejor se adapte a su estilo de juego y estrategia en la cancha.

Además, en esta pantalla se incluye un botón claramente identificado que permite a los jugadores iniciar el partido una vez que han configurado todas las opciones según sus preferencias. Este botón facilita el inicio del juego y garantiza una experiencia fluida y sin complicaciones para los usuarios.

Cuando el usuario pulsa el botón de inicio de partido, se abre un diálogo que permite añadir usuarios al partido y asignarlos a los equipos correspondientes, asegurando así una organización eficiente y personalizada del juego.

Diagrama de la pantalla de inicio de partido con los siguientes elementos:

- INICIO DE PARTIDO**
- EQUIPO CON 1º SAQUE**
 - EQUIPO A ▼
 - EQUIPO B
- TIPO DE VENTAJA**
 - NORMAL ▼
 - PUNTO DE ORD
- DESEMPATE DE SET**
 - GANADOR DE DOS ▼
 - LLEGAR A 7
- CANTIDAD DE SETS**
 - 1 ▼
 - 3
 - 5

Figura 6.3.7.1 Esquema de pantalla de inicio de partido



Figura 6.3.7.2 Pantalla de inicio de partido final



Figura 6.3.7.3 Pantalla añadir jugadores al partido

6.3.8 Pantalla Marcador

En la pantalla de marcador, como se muestra en la figura 6.3.8.2, se brinda a los jugadores acceso al marcador actual del partido, lo que les permite mantenerse al tanto de la puntuación en todo momento. Además, se incluyen botones intuitivos que les permiten aumentar o disminuir el marcador según corresponda, facilitando la actualización rápida y precisa de la puntuación durante el juego.

Por otro lado, la aplicación está equipada con un sistema de reconocimiento de comandos de voz. Esto significa que los jugadores pueden dar instrucciones verbales, como "punto equipo Rojo", y la aplicación reconoce estas órdenes para modificar el marcador en consecuencia. Además, la aplicación también proporciona información relevante sobre el partido, como el marcador actual, en respuesta a comandos específicos.

En la propia pantalla de marcador, también se puede visualizar a los diferentes participantes del partido si los jugadores han añadido a alguien. Esto permite identificar claramente a los jugadores que están participando, ofreciendo una visión completa y detallada del desarrollo del partido.

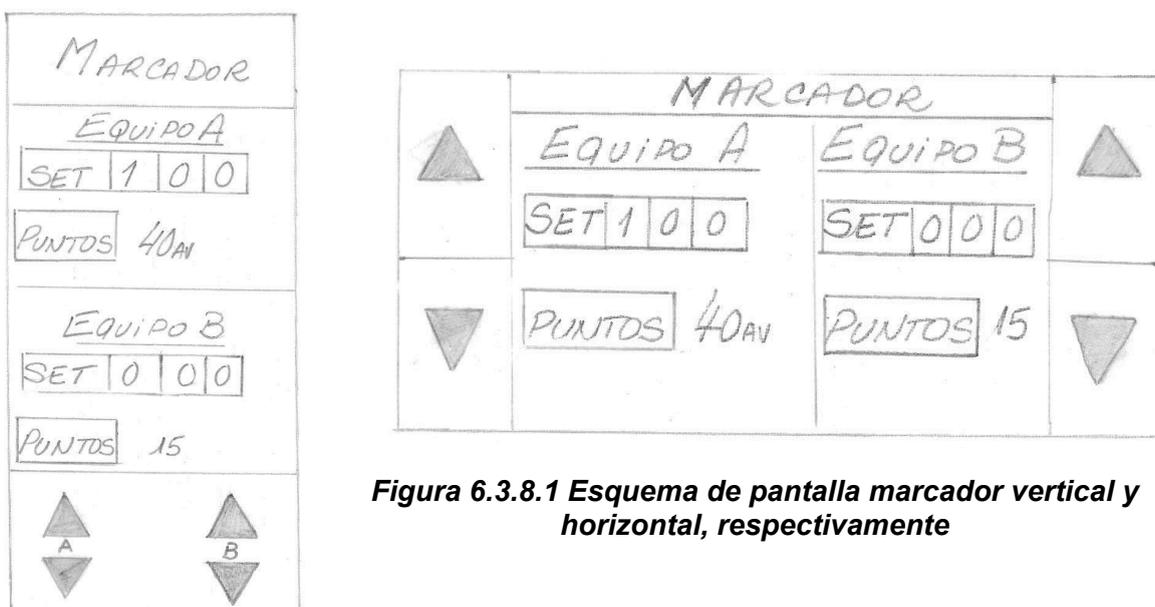


Figura 6.3.8.1 Esquema de pantalla marcador vertical y horizontal, respectivamente

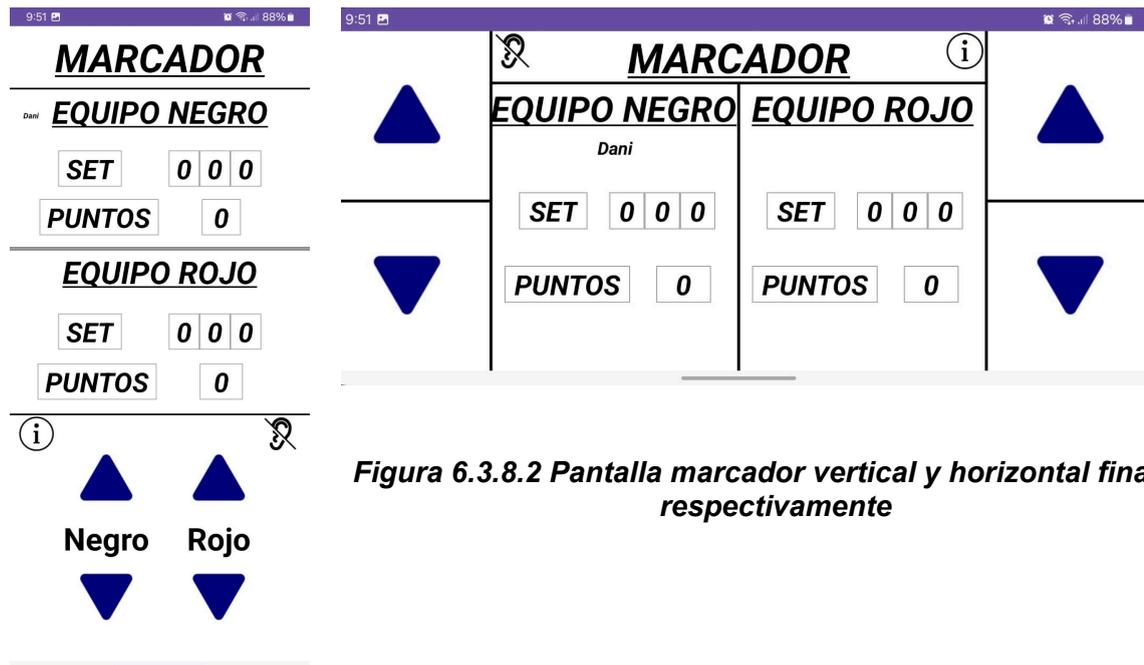


Figura 6.3.8.2 Pantalla marcador vertical y horizontal finales, respectivamente

Se ha identificado un problema al cambiar entre los modos de pantalla portrait y landscape en la aplicación. Se observa que al realizar esta transición, la actividad se reinicia por completo, lo que puede resultar en una interrupción significativa, especialmente si el usuario se encuentra en medio de una partida u otra actividad crítica.

Para abordar este problema y garantizar una transición suave entre los modos de pantalla, se implementarán los métodos `onSaveInstanceState()` y `onRestoreInstanceState()`. Estos métodos son fundamentales en el ciclo de vida de una actividad en Android, ya que permiten guardar y restaurar el estado de la actividad cuando se produce un cambio de configuración, como la rotación de la pantalla.

Al utilizar `onSaveInstanceState()`, la actividad puede almacenar información sobre su estado actual, como los datos de juego o el progreso del usuario. Luego, cuando la actividad se reinicie debido a un cambio de configuración, como la rotación de la pantalla, `onRestoreInstanceState()` se utilizará para recuperar y restaurar estos datos, asegurando que la actividad se reanude desde donde se dejó, sin perder ningún progreso.

No se han usado las SharedPreferences, como en los ajustes, porque el uso de SharedPreferences no es adecuado para resolver este problema. SharedPreferences está diseñado para almacenar preferencias de usuario, no para mantener el estado dinámico de una actividad durante cambios de configuración, como la rotación de la pantalla. Utilizar onSaveInstanceState() y onRestoreInstanceState() proporciona una solución más precisa y eficiente para preservar el estado de la actividad en tales circunstancias.

6.3.9 Restablecimiento de Contraseña

Se ha desarrollado una nueva actividad a la que se puede acceder desde la pantalla de inicio de sesión. Esta actividad permite a los usuarios restablecer su contraseña si así lo desean. Al ingresar a esta actividad, se solicita al usuario que ingrese el correo electrónico asociado a su cuenta.

Si el correo electrónico existe en la base de datos, se envía un correo electrónico al usuario con un enlace único que le permitirá restablecer su contraseña de manera segura y sencilla.

Este proceso asegura que los usuarios tengan una opción conveniente y segura para recuperar el acceso a sus cuentas en caso de olvidar su contraseña.

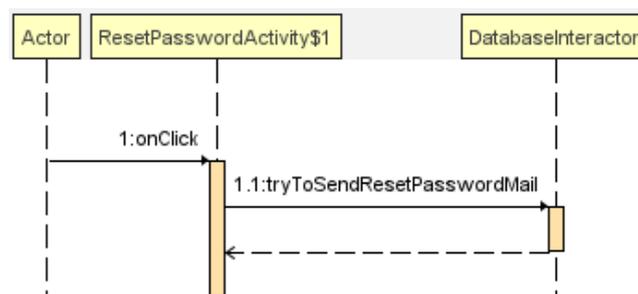


Figura 6.3.9.1 Diagrama de secuencia de restablecimiento de Contraseña

6.3.10 Personalización Componentes

Se ha identificado un desafío en la personalización de los diversos componentes en las diferentes actividades de la aplicación, particularmente en los elementos como spinner y numberPicker. Este problema radica en la limitada disponibilidad de atributos estándar para la personalización de estos componentes.

Para solucionar esta limitación, se han utilizado los archivos "Drawable". Estos archivos, en formato XML, ofrecen una mayor flexibilidad para la personalización de los componentes. Por ejemplo, permiten la configuración de bordes alrededor de los elementos y la especificación de colores personalizados.

Al utilizar archivos "Drawable", se ha mejorado significativamente la capacidad de adaptación y la estética visual de la aplicación.

6.4 Base de Datos

6.4.1 Firebase

Implementar Firebase como base de datos para una aplicación móvil ofrece una serie de beneficios significativos que pueden mejorar tanto el desarrollo como la experiencia del usuario. En primer lugar, Firebase proporciona una base de datos en tiempo real que permite la sincronización automática de datos entre los dispositivos de los usuarios, lo que resulta en una experiencia colaborativa y en tiempo real. Esta característica es especialmente útil para aplicaciones que requieren interacción en tiempo real, como aplicaciones de mensajería, aplicaciones de colaboración y juegos multijugador.

Además, Firebase ofrece una integración sin problemas con otras funcionalidades de la plataforma, como la autenticación de usuarios, el almacenamiento en la nube, las notificaciones push y las analíticas. Esto significa que los desarrolladores pueden aprovechar una gama completa de servicios que abarcan todos los aspectos del desarrollo de aplicaciones móviles sin tener que integrar múltiples proveedores de servicios.

La escalabilidad es otro beneficio clave de Firebase como base de datos. La plataforma está diseñada para manejar cargas de trabajo de cualquier tamaño, lo que significa que las aplicaciones pueden crecer y expandirse sin preocuparse por problemas de rendimiento o escalabilidad. Esto es particularmente importante para aplicaciones móviles que pueden experimentar un rápido crecimiento en el número de usuarios.

Además, Firebase ofrece una API simple y fácil de usar que facilita el desarrollo y la administración de la base de datos. Los desarrolladores pueden acceder y manipular los datos de manera eficiente utilizando una variedad de lenguajes de programación y marcos, lo que les permite crear aplicaciones móviles robustas y funcionales en menos tiempo.

En resumen, implementar Firebase como base de datos para una aplicación móvil es una buena idea debido a su capacidad para proporcionar una experiencia colaborativa y en tiempo real, su integración sin problemas con otras funcionalidades de la plataforma, su escalabilidad y su API fácil de usar. Estos beneficios hacen que Firebase sea una opción atractiva para desarrolladores que buscan crear aplicaciones móviles modernas y eficientes.

6.4.2 Estructura

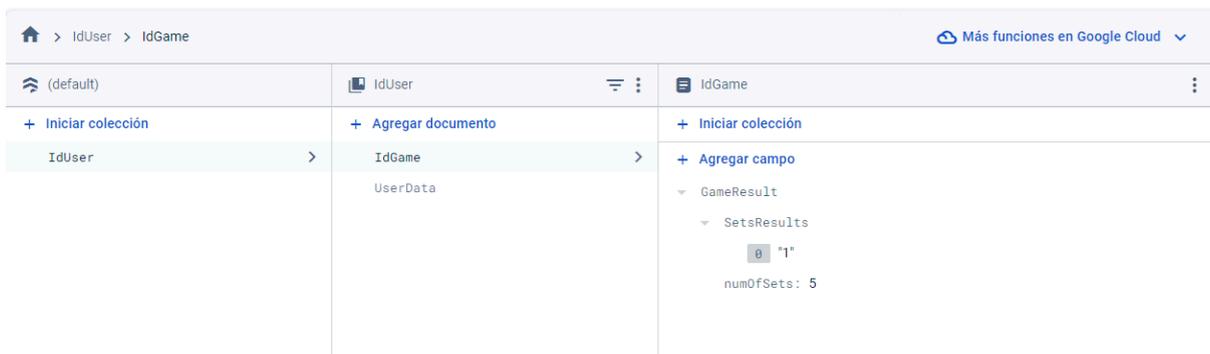


The screenshot shows a user registration form with the following elements:

- Buttons: "Agregar usuario" (top right), "Cancelar", and "Agregar usuario" (bottom right).
- Fields: "Correo electrónico" (with a red border and an error message "Debes ingresar un correo electrónico válido.") and "Contraseña".
- Text: "Agrega un usuario con correo electrónico y contraseña" and "Este proyecto todavía no tiene usuarios".
- Table Headers: "Identificador", "Proveedores", "Fecha de creación ↓", "Fecha de acceso", and "UID de usuario".

Figura 6.4.2.1 Estructura Base de datos Autenticación

Para guardar y recuperar los datos de inicio de sesión del usuario (únicamente el correo y la contraseña) se utilizará Firebase Authenticator.



The screenshot shows the Firebase console structure for the database:

- Path: IdUser > IdGame
- IdUser collection: IdGame (containing UserData)
- IdGame collection: GameResult (containing SetsResults, which contains numOfSets: 5)

Figura 6.4.2.2 Estructura General Base de datos

Para guardar los demás datos de inicio de sesión y los partidos guardados se utilizará Firebase Firestore Database.

Por cada usuario se creará una colección cuyo valor será el correo del usuario y dentro de esta colección se creará, mínimo un documento llamado UserData, que será el que contenga los datos de usuario.

A medida que el usuario vaya jugando partidos, se crearán diferentes documentos, cuyos identificadores serán la fecha en la que se jugaron.

Por tanto, cada usuario tendrá mínimo un documento, el UserData, que como bien indica su nombre, contendrá todos los datos de inicio de sesión de este.

Por cada partido que juegue un usuario, se creará un documento adicional dentro de su colección. El nombre que tendrá este documento será el ID del partido. Dentro de este documento se encontrará todos los datos relacionados con el partido.

ID de documento ⓘ

matchDate

Campo	Tipo	Valor
numOfSets	= string	
setsResults	= array	
0	string	
⊕ Agregar campo		
team	= string	
⊕ Agregar campo		

Figura 6.4.2.3 Estructura base de datos Resultados de partidos

Para poder guardar los datos de los partidos en Firebase se seguirá la estructura de la Figura 6.4.2.3.

Los documentos de los partidos tendrán dos campos. Un campo de tipo number llamado “numOfSets”, que serán la cantidad de sets que se han jugado durante el partido.

El segundo, es un campo de tipo array, cuya longitud será la cantidad de sets multiplicado por dos, en el que se guardará, en cada elemento, la puntuación de los equipos en los diferentes sets, siendo los pares, la puntuación del equipo uno (equipo que empieza sacando), y los impares, la puntuación del equipo dos.

Por último, un campo llamado “team” en el que se guardará el equipo del jugador.

ID de documento [?](#)

UserData

Campo	Tipo	Valor
Age	= string	<input type="text"/> ⊖
Alias	= string	<input type="text"/> ⊖
Gender	= string	<input type="text"/> ⊖
Mail	= string	<input type="text"/> ⊖
Name	= string	<input type="text"/> ⊖

[⊕](#) Agregar campo

Figura 6.4.2.4 Estructura base de datos Datos de Usuario

Por otro lado, para guardar los datos introducidos por parte del usuario, se seguirá la estructura de la figura 6.4.2.4. Se creará otro documento dentro de la colección del usuario llamado "UserData" en la que se crearán cinco campos, que serán los que se pedirán a los usuarios para poder crear una cuenta.

6.4.3 Conexión

Para conectar la base de datos, se deben seguir unos pasos por orden. Primero, se debe descargar un archivo JSON proporcionado por la propia base de datos, que contiene la información de configuración necesaria. Este archivo se coloca dentro de la carpeta app del proyecto.

Luego, en el archivo build.gradle a nivel de proyecto, se añade la siguiente línea en el apartado de plugins: `id("com.google.gms.google-services")` versión "4.4.1" `apply false`. Esto se encarga de configurar los servicios de Google necesarios para la conexión.

Seguidamente, en el archivo build.gradle a nivel de la aplicación, se agrega la siguiente línea en el apartado de plugins: `id("com.google.gms.google-services")`. Esto habilita los servicios de Google en la aplicación.

Además, en el mismo archivo, en el apartado de dependencias, se añade la implementación de Firebase BOM (Bill of Materials) para mantener las versiones de todas las bibliotecas de Firebase consistentes. Esto se logra con la línea: `implementation(platform("com.google.firebase:firebase-bom:33.0.0"))`.

Dentro del mismo apartado de dependencias, se incluyen las implementaciones específicas necesarias para las funcionalidades deseadas. Por ejemplo, para utilizar Firebase Authentication se agrega `implementation("com.google.firebase:firebase-auth")`, mientras que

para Firebase Firestore se agregan
implementation("com.google.firebase:firebase-analytics") y
implementation("com.google.firebase:firebase-firestore").

Con estos pasos, la aplicación queda correctamente configurada para conectarse y utilizar la base de datos Firebase.

6.4.3 Uso

Se ha creado una clase dedicada exclusivamente a interactuar con la base de datos. Esta clase despliega un conjunto de funciones diseñadas para gestionar la información de los usuarios y los resultados de los partidos en Firebase.

Una de las principales funciones de esta clase es facilitar la creación de nuevos usuarios y almacenarlos en Firebase Authentication. Además, se encarga de crear una nueva colección en Firebase Firestore para cada usuario registrado, donde se almacena un documento con toda la información proporcionada por el jugador durante el proceso de registro.

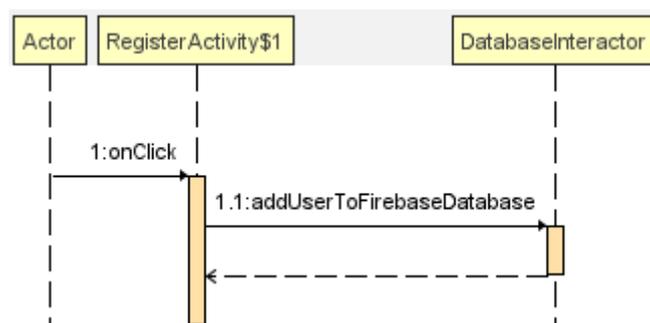


Figura 6.4.3.1 Diagrama de secuencia de registro de usuario

Otra funcionalidad de la clase es permitir a los usuarios iniciar sesión de manera eficiente y segura. Esta clase se encarga de autenticar las credenciales proporcionadas por los usuarios, verificando su validez a través de Firebase Authentication. Además, se gestiona la sesión activa del usuario, asegurando que puedan acceder a sus datos personales y actualizarlos.

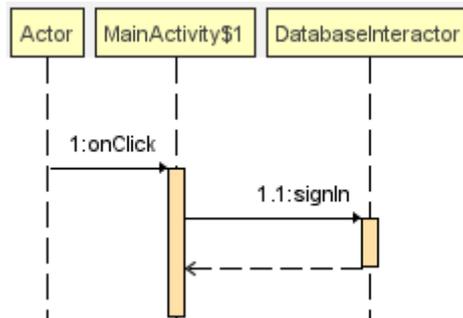


Figura 6.4.3.2 Diagrama de secuencia de inicio de sesión

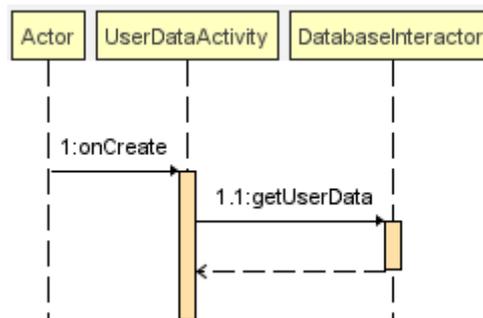


Figura 6.4.3.3 Diagrama de secuencia de obtención de datos de usuario

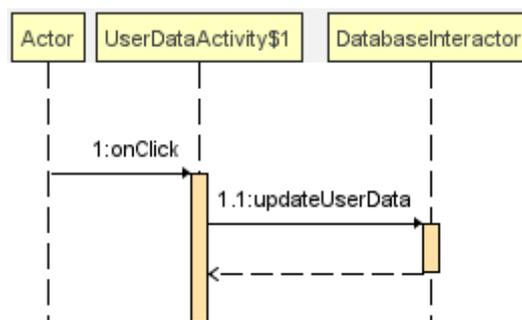


Figura 6.4.3.4 Diagrama de secuencia de actualización de datos de usuario

Esta clase también juega un papel fundamental en la recuperación de los resultados de los partidos anteriores jugados por el usuario. Utilizando consultas a Firebase Firestore, se pueden obtener los datos históricos de los encuentros anteriores y presentarlos al jugador de manera clara y organizada.

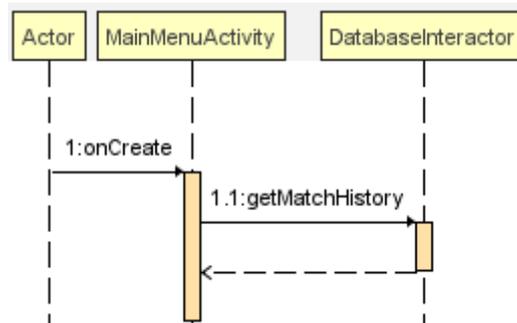


Figura 6.4.3.5 Diagrama de secuencia obtención historial de partidos

Otra función crucial de esta clase es la capacidad de guardar los resultados de los partidos. Esta función añade un nuevo documento dentro de la colección del usuario actual en Firebase Firestore, registrando así los detalles y el desempeño del jugador en cada encuentro disputado.

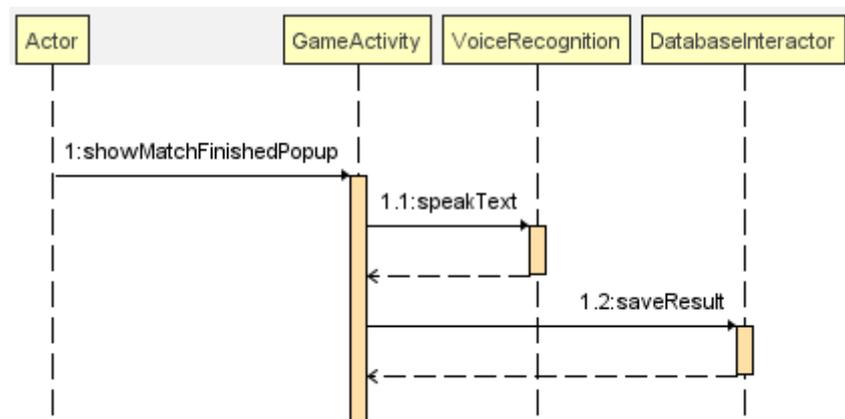


Figura 6.4.3.6 Diagrama de secuencia de guardado de resultados

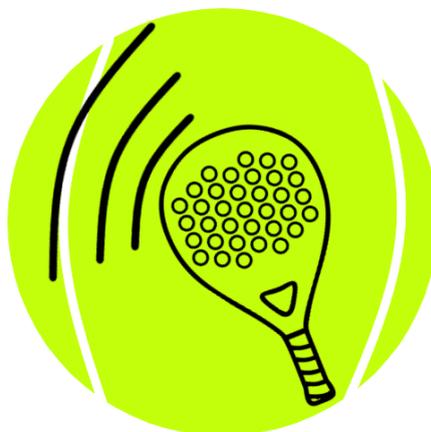
6.5 Icono de la aplicación

Se busca crear un icono que esté completamente relacionado con la aplicación, que la represente y, a su vez, sea minimalista. En este proceso, se pretende que el pilar central del icono sean las palas de pádel, debido a su esencial importancia en este deporte.

Inicialmente, se contempló la idea de colocar dos palas entrecruzadas, ya que esta disposición representa visualmente el espíritu del pádel. Sin embargo, durante el desarrollo, se observa que para que ambas palas cupieran completamente en el espacio disponible, su tamaño tendría que ser muy reducido, lo que las haría difíciles de distinguir y restaría claridad al icono.

Además de la pala, se incluyen ondas que simulan el acto de hablar. En este caso, estas representan la función de reconocimiento de voz de la aplicación. La idea es que parezca que es la propia pala la que está hablando, lo cual establece una conexión directa con la funcionalidad de la aplicación y añade un elemento de originalidad al diseño.

Para complementar estos elementos y añadir un fondo que proporcione contexto, se decide incluir una pelota de pádel. La pelota contextualiza el icono dentro del ámbito del pádel haciendo que el diseño sea más dinámico y atractivo.



7. Testing

En el apartado de testing, se lleva a cabo un proceso exhaustivo para evaluar el funcionamiento y la usabilidad de la aplicación. Se realizan pruebas rigurosas en diferentes dispositivos y entornos para garantizar que la aplicación funcione correctamente en diversas condiciones. Además, se recopila feedback de los usuarios durante el proceso para identificar posibles problemas y áreas de mejora. Este enfoque permite detectar y corregir cualquier fallo o deficiencia antes de la finalización de la aplicación, asegurando así una experiencia óptima para los usuarios finales.

6.5.1 Primera Iteración: Entrevistas Interfaces

Para la primera iteración del proceso de testing, se realizó una consulta a un grupo de personas con el objetivo de recopilar feedback sobre los diversos croquis de interfaces desarrollados para la aplicación.

Este enfoque se lleva a cabo con la intención de obtener información valiosa sobre la usabilidad, la claridad de diseño y la eficacia de las interfaces propuestas. El feedback recopilado, ver ANEXO I., subapartado 1.1 ENTREVISTAS INTERFACES, proporciona una base sólida para iterar y mejorar aún más las diferentes interfaces de la aplicación.

En general, los entrevistados destacan aspectos positivos de la interfaz de usuario, como su funcionalidad y facilidad de uso. Sin embargo, también ofrecen sugerencias para mejorar la experiencia del usuario y la claridad de la aplicación.

Las principales áreas de mejora identificadas son:

Priorización visual de elementos: Los participantes sugieren resaltar visualmente los puntos del set actual en la ventana del marcador para facilitar su identificación durante el juego.

Indicadores visuales de progreso: Hay recomendaciones para agregar indicadores visuales en la pantalla para mostrar el set actual en el que están jugando los participantes, lo que ayuda a los jugadores a mantenerse informados sobre el estado del partido y evitar confusiones.

Personalización de nombres de usuarios: Se propone incluir los nombres de los usuarios individuales en las interfaces en lugar de solo el nombre del equipo, lo que contribuiría a una experiencia más personalizada y organizada para los usuarios.

Coherencia en la estructura de las pantallas: Se señaló la importancia de mantener la coherencia en la estructura de las pantallas para garantizar una experiencia de usuario uniforme y cohesiva en toda la aplicación.

Mediante esta práctica, se busca garantizar que las interfaces finales satisfagan las necesidades y expectativas de los usuarios, mejorando así la experiencia general de la aplicación.

6.5.2 Segunda Iteración: Entrevistas Interfaces

Durante la segunda iteración del proceso de desarrollo, se lleva a cabo una cuidadosa revisión del feedback recibido para identificar áreas de mejora en los diferentes croquis de

la aplicación. A partir de este análisis, se realizan diversas modificaciones que buscan abordar las sugerencias y comentarios recibidos por parte de los usuarios. Estas modificaciones están diseñadas para mejorar la claridad del diseño y la eficacia de la interfaz de usuario en general.

Una vez completadas las modificaciones en los croquis, se inicia el proceso de implementación en Android Studio.

Durante este proceso, se aplican las mejores prácticas de desarrollo para garantizar un código limpio, eficiente y fácilmente mantenible.

La integración de las sugerencias de los usuarios en la segunda iteración del desarrollo es fundamental para asegurar que la aplicación evolucione de acuerdo con las necesidades y expectativas de su público objetivo. Al priorizar la retroalimentación de los usuarios, se fomenta una cultura de mejora continua que contribuye a la creación de una aplicación más sólida y satisfactoria para los usuarios finales.

6.5.3 Tercera Iteración: Entrevistas Interfaces

En la tercera iteración, se ha llevado a cabo una evaluación exhaustiva del funcionamiento de las interfaces de la aplicación en una variedad de dispositivos móviles, ver ANEXO I subapartado 1.3 TERCERA ITERACIÓN. Durante estas pruebas, se ha observado que, en algunos dispositivos, las interfaces no se visualizan correctamente, lo que puede comprometer la experiencia del usuario.

Dado que la visualización incorrecta afecta la funcionalidad y la satisfacción del usuario, se ha concluido que es necesario volver a iterar sobre el diseño y la implementación de las interfaces. Este proceso incluirá ajustes y optimizaciones para garantizar que la aplicación se vea y funcione de manera óptima en todos los dispositivos móviles. La identificación y

corrección de estos problemas son esenciales para cumplir con los estándares de calidad y mejorar la experiencia general del usuario.

6.5.4 Cuarta Iteración: Entrevistas Interfaces

Tras recibir el feedback detallado, ver ANEXO I. subapartado 1.4 CUARTA ITERACIÓN, se ha llegado a la conclusión de que no es necesario llevar a cabo ninguna iteración adicional de testing de las interfaces. Las evaluaciones realizadas proporcionaron una perspectiva integral de la experiencia del usuario y confirmaron que las interfaces cumplen con los requisitos establecidos. Este análisis exhaustivo y la validación de los resultados obtenidos en las pruebas anteriores han respaldado la decisión de avanzar con el desarrollo sin necesidad de realizar más pruebas de interfaz en este momento.

6.5.5 Primera Iteración: Entrevistas Funcionales

Tras recibir feedback de los diferentes móviles, según se describe en el ANEXO II. subapartado 2.1 PRIMERA ITERACIÓN, se inicia una investigación sobre el incorrecto funcionamiento del reconocimiento de voz y de los distintos modos de juego.

Este análisis exhaustivo tiene como objetivo identificar las causas subyacentes del problema y desarrollar soluciones efectivas que restauren la funcionalidad adecuada del reconocimiento de voz en todas las situaciones de uso. La información recopilada a través de las pruebas y el feedback recibido desempeña un papel crucial en este proceso, proporcionando una guía valiosa para la resolución de problemas y la mejora continua de la aplicación.

Estas soluciones se implementan y se prueban exitosamente en el último sprint del proyecto.

8. Conclusiones

La aplicación de pádel cumple con el objetivo principal de proporcionar una herramienta a los jugadores que facilite en todo momento el conocimiento del marcador actual del partido. Se logra ofrecer una plataforma intuitiva y accesible que permite a los usuarios organizar y gestionar partidos de manera eficiente. La aplicación proporciona un sistema eficaz para registrar y seguir los resultados de los partidos, ofreciendo un historial detallado y preciso de los juegos jugados.

Entre los beneficios más destacados para los usuarios se encuentran la gestión simplificada de partidos y la capacidad de programar juegos e invitar a otros jugadores. Además, la aplicación permite registrar los resultados de los partidos y almacenar esta información en una base de datos, lo que facilita el seguimiento del rendimiento de los jugadores a lo largo del tiempo.

El uso de Firebase como base de datos ha sido un acierto significativo en el desarrollo de la aplicación. Firebase destaca por su facilidad de conexión e implementación, permitiendo una gestión de datos en tiempo real y sin complicaciones. La capacidad de Firebase para sincronizar datos instantáneamente entre todos los dispositivos conectados ha sido fundamental para mantener la información actualizada y accesible para todos los usuarios, independientemente de su ubicación. Además, la seguridad y la escalabilidad ofrecidas por Firebase aseguran que la aplicación pueda crecer y adaptarse a un número creciente de usuarios sin comprometer la integridad de los datos.

Un problema que se ha tenido es que no ha habido mucho tiempo para poder hacer testing en diferentes dispositivos móviles y que, en general, no se ha podido hacer tanto testing como se quería en un inicio. Este aspecto limita la capacidad de identificar y resolver

problemas específicos de ciertos dispositivos o versiones del sistema operativo, lo que podría afectar la experiencia del usuario en algunos casos.

En general, se valora positivamente el resultado de la aplicación ya que cumple con los objetivos iniciales planteados. Desde su concepción, la aplicación tenía como propósito principal proporcionar a los jugadores de pádel una herramienta eficaz que les permitiera conocer y gestionar el marcador de sus partidos de manera sencilla y precisa.

El desarrollo de la aplicación ha demostrado ser exitoso en alcanzar este objetivo, ofreciendo una plataforma intuitiva y funcional que mejora significativamente la experiencia de los usuarios. La implementación de Firebase como base de datos ha sido una decisión acertada, facilitando la conexión y el uso de la aplicación, así como garantizando la seguridad y accesibilidad de los datos en tiempo real.

9. Posibles Ampliaciones

9.1 Funcionamiento completo mediante el uso de la voz

Para posibles ampliaciones, se considera la implementación de una funcionalidad completa basada en comandos de voz en toda la aplicación, lo que implica extender el reconocimiento de voz más allá de la gestión del marcador de pádel. Esta estrategia busca mejorar la experiencia del usuario y agregar valor a la aplicación al permitir que todas las interacciones y acciones estén disponibles mediante comandos de voz.

Al hacer que toda la aplicación sea completamente funcional utilizando la voz, se pueden obtener diversos beneficios. Primero, se mejora la accesibilidad al permitir que los usuarios interactúen con la aplicación sin depender exclusivamente de la interacción táctil. Esto puede ser especialmente útil para aquellos con discapacidades visuales. Además, la experiencia del usuario se ve mejorada al hacer que la navegación y la realización de acciones sean más intuitivas y fluidas. Esto puede conducir a una mayor satisfacción del usuario y a una reducción de la fricción durante el uso de la aplicación.

Además, la implementación de una funcionalidad completa basada en comandos de voz puede diferenciar la aplicación de otras en el mercado, proporcionando un valor único y atrayendo a usuarios que buscan una experiencia innovadora.

Es importante tener en cuenta aspectos como la precisión y la confiabilidad del reconocimiento de voz, así como proporcionar retroalimentación clara al usuario sobre los comandos reconocidos y las acciones realizadas. También es fundamental permitir a los usuarios alternar entre la interacción por voz y la interacción táctil, brindando flexibilidad para adaptarse a las preferencias individuales de cada usuario.

En conclusión, la implementación de una funcionalidad completa basada en comandos de voz puede ser una estrategia valiosa para mejorar la accesibilidad, la experiencia del usuario y la eficiencia de la aplicación, así como para diferenciarla en el mercado competitivo de aplicaciones móviles.

9.2 Posibilidad de guardar estadísticas del partido

Una posible ampliación futura consiste en la incorporación de la capacidad de guardar estadísticas durante el partido y almacenarlas en la base de datos. De esta manera, si un usuario desea obtener más información sobre un partido, puede acceder al historial y visualizar las estadísticas previamente guardadas. Estas estadísticas incluyen detalles como los jugadores que participaron, el porcentaje de puntos ganados, la cantidad de break points realizados y cuántos de estos fueron ganados, entre otros datos relevantes. Esta funcionalidad adicional proporciona a los usuarios una visión más completa y detallada de los partidos, mejorando así la experiencia y el análisis del juego.

9.3 Creación de Modo de Campeonato

Como ampliación futura, se contempla el desarrollo de un modo de campeonato donde los jugadores se inscriben en un torneo, ya sea por equipos o de manera individual. La propia aplicación se encarga de establecer los partidos a jugar, basándose en los resultados de los encuentros previos dentro del torneo. Esta funcionalidad permite una gestión automatizada y eficiente de los torneos, facilitando la organización y el seguimiento de los mismos, y mejorando la experiencia competitiva para los jugadores.

10. Bibliografía

- [1] REGLAMENTO DE JUEGO DEL PÁDEL, FEDERACIÓN INTERNACIONAL PÁDEL, España, 2021. Disponible: <https://www.padelfederacion.es/refs/docs/Reglamento-Juego%202021.pdf>
- [2] T. Ayers and B. Nolan, "Developing a Distributed Java-based Speech Recognition Engine ." The ITB Journal, Dublin, Ireland, 2004
- [3] R. D. Martin, "El Pádel está de Moda y en Auge | PadelStar," padelstar.es, Oct. 29, 2010. <https://padelstar.es/articulos-de-padel/el-padel-esta-de-moda-y-en-auge/>
- [4] J. L. Oropeza Rodríguez, "Algoritmos y Métodos para el Reconocimiento de Voz en Español Mediante Sílabas," Centro de Investigación en Computación-IPN, México, 2006.
- [5] M. Windows, "What is Cortana?," Microsoft Support, <https://support.microsoft.com/en-us/topic/what-is-cortana-953e648d-5668-e017-1341-7f26f7d0f825> (accessed Nov. 27, 2023).
- [6] S. T. Apple, "Usar Siri en el iphone," Apple Support, <https://support.apple.com/es-us/guide/iphone/iph83aad8922/ios#:~:text=Realiza%20tareas%20cotidianas%20s%C3%B3lo%20con.consultar%20el%20clima%20y%20m%C3%A1s.> (accessed Nov. 27, 2023).
- [7] Amazon, "¿Por qué Alexa?," Amazon, <https://developer.amazon.com/es-ES/alexa> (accessed Nov. 27, 2023).
- [8] M. Support, "Cortana Voice Support," Microsoft Learn, <https://learn.microsoft.com/en-us/windows-hardware/customize/desktop/cortana-voice-support> (accessed Dec. 17, 2023).
- [9] Tech Bazar, "How does Siri work? how does it recognize our voice and understand what we're saying?," Quora, <https://www.quora.com/How-does-Siri-work-How-does-it-recognize-our-voice-and-understand-what-were-saying> (accessed Dec. 17, 2023).
- [10] J. Wallis, "The Tech behind Amazon alexa," Intuji, <https://intuji.com/the-tech-behind-amazon-alexa/> (accessed Dec. 17, 2023).

- [11] B. Sport, "SCOREPAD, el pupitre táctil y multideportivo," Pupitre táctil para controlar diferentes tipos de pantallas deportivas, <https://www.bodet-sport.com/es/productos/control-de-visualizacion-deportiva/pupitre-d-e-control.html> (accessed Nov. 27, 2023).
- [12] B. Sport, "SCOREAPP, El autoarbitraje al alcance de todos," Una aplicación conectada para gestionar su puntuación desde su smartphone., <https://www.bodet-sport.com/es/productos/control-de-visualizacion-deportiva/aplicacion-movil.html> (accessed Nov. 27, 2023).
- [13] P. Garcia De Palau Roca, "Padel Watch: Padel scorekeeper," App Store, <https://apps.apple.com/us/app/padel-watch-padel-scorekeeper/id6443518532> (accessed Nov. 27, 2023).
- [14] Google, "Meet Google Play's target API level requirement," Dec. 14, 2023. <https://developer.android.com/google/play/requirements/target-sdk> (accessed Jan. 07, 2024).
- [15] "Padel (Marcador por Voz) - Apps en Google Play," play.google.com. https://play.google.com/store/apps/details?id=com.mota.padel&hl=es_GT&pli=1 (accessed Jan. 07, 2024).
- [16] "Meet Android Studio | Android Developers". Android Developers. Accedido el 13 de marzo de 2024. <https://developer.android.com/studio/intro> (accessed Mar. 13, 2024).
- [17] "FAQ". Flutter documentation | Flutter. Disponible: <https://docs.flutter.dev/resources/faq> (accessed Mar. 13, 2024).
- [18] "Agregue Firebase a su aplicación Flutter". Firebase. Accedido el 13 de marzo de 2024. <https://firebase.google.com/docs/flutter/setup?hl=es&platform=android> (accessed Mar. 13, 2024).
- [19] "Introduction to Ionic | Ionic Documentation". Ionic Framework - The Cross-Platform App Development Leader. <https://ionicframework.com/docs> (accessed Mar. 13, 2024).
- [20] "Firebase Realtime Database". Firebase. <https://firebase.google.com/docs/database?hl=es> (accessed Mar. 13, 2024).

- [21] "Introduction to the Realm SDK for Android | MongoDB". MongoDB: The Developer Data Platform | MongoDB. <https://www.mongodb.com/developer/products/realm/introduction-realm-sdk-android/> (accessed Mar. 13, 2024).
- [22] "Access SQL: conceptos básicos, vocabulario y sintaxis - Soporte técnico de Microsoft". Microsoft Support. <https://support.microsoft.com/es-es/topic/access-sql-conceptos-básicos-vocabulario-y-sintaxis-444d0303-cde1-424e-9a74-e8dc3e460671> (accessed Mar. 13, 2024).
- [23] "Firebase Authentication". Firebase. <https://firebase.google.com/docs/auth?hl=es> (accessed Mar. 13, 2024).
- [24] "Auth0.Android". Auth0 Docs. <https://auth0.com/docs/libraries/auth0-android> (accessed Mar. 13, 2024).