UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Diseño e implementación de una aplicación Android para la organización y reserva de pistas de clubes de tenis

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Zhao , Jun Yi

Tutor/a: Andrés Martínez, David de

CURSO ACADÉMICO: 2021/2022

Resumen

La popularidad de los deportes ha venido en aumento desde los últimos años, a causa de buscar llevar una vida más saludable y provocando con ello un aumento en la cantidad de espacios deportivos como gimnasio, clubes de tenis, polideportivos, etc. En el presente trabajo se realiza el diseño e implementación de una aplicación Android con el fin de mejorar la gestión de los clubes de tenis. La aplicación permitirá a los socios de tales clubes consultar la disponibilidad y reservar las instalaciones a través de la aplicación, además de permitir a los socios poder buscar parejas y rivales del mismo nivel para completar un partido. El *backend* de la App estará gestionado en Firebase que permitirá ver los cambios de la base de datos en tiempo real.

Palabras clave: Android, deporte, Firebase, App.

Abstract

The popularity of sports has been increasing in recent years, due to seeking to lead a healthier life and thereby causing an increase in the number of sports spaces such as gyms, tennis clubs, sports centers, etc. In the present work, the design and implementation of an Android application is carried out in order to improve the management of tennis clubs. The application will allow members of such clubs to check availability and book facilities through the application, as well as allowing members to search for pairs and rivals of the same level to complete a match. The *backend* of the App will be managed in Firebase, which will allow you to see the changes in the database in real time.

Keywords : Android, sports, Firebase, App.

Índice de contenidos

Contenido

Índice de tablas

Índice de figuras

Lista acrónimos

API → Interfaz de programación de aplicaciones

REST → Transferencia de estado representacional

IU → Interfaz de usuario

ANR → Android no responde

CVS → Sistema de versiones concurrentes

MVC → Modelo-Vista-Controlador

1. Introducción

Hoy en día los dispositivos móviles y las apps móviles se han vuelto parte de la vida cotidiana de las personas. Casi todos poseen un dispositivo móvil en donde tienen instalados apps para diferentes usos. Y esto, las empresas lo tienen claro, y por ello encontramos por ejemplo apps de entidades bancarias que te facilitan la gestión de tu cuenta, así como apps de tienda que te permiten realizar las compras online.

La popularidad de las apps sumado al aumento de la cantidad de personas que busca llevar una vida más saludable y hacer deporte ha hecho que muchas empresas del sector quieran digitalizarse y facilitar el acceso de las instalaciones deportivas a sus usuarios.

Uno de los deportes que han experimento un gran auge ha sido el pádel que solo en 2021 ha aumentado más de un 22% llegando a más de 100000 licencias federadas en España. [1]

Este aumento de aficionados que practican este deporte ha llevado a que muchas empresas y clubes de tenis decidan abrir espacios destinados a este deporte. Y es por ello por lo que, para facilitar la organización y reserva de estos espacios, he decidido desarrollar una aplicación móvil que vaya a facilitar la gestión de muchas de estas empresas y clubes.

1.1 Motivación

Uno de los motivos por los que decidí desarrollar este proyecto es debido a que recientemente comencé a practicar pádel y he visto cómo varias empresas y clubes ofrecían pistas para jugar al pádel y tenis, pero ninguna de estas tenía una aplicación donde pudieras hacer gestiones. Tenía que llamar siempre por teléfono para poder reservar y con la cantidad de gente, casi siempre estaba lleno y tenía que ir preguntando la disponibilidad de las pistas, situación que se podría haber agilizado si hubiera una aplicación móvil en donde pudiera ver los horarios y disponibilidades de las pistas.

Otro motivo es que durante la carrera no he tenido ninguna asignatura en la que pudiera desarrollar una aplicación Android por lo que he decidido aprovechar este proyecto para aprender una tecnología que se encuentra en auge y desarrollar una aplicación Android que, llegado el caso, podría ofrecer servicio a determinadas empresas para agilizar su gestión de reservas, entre otras funciones.

1.2 Objetivos

El objetivo de este proyecto de Trabajo de Final de Grado es el desarrollo de una aplicación Android para la organización y reserva de pistas de pádel y tenis en clubes de tenis desde cero.

La aplicación resultante está dirigida a clubes de tenis, pero también puede estar dirigida a cualquier empresa que ofrezca como servicio pistas de tenis o pádel.

Para poder alcanzar este objetivo, se plantean los siguientes objetivos secundarios:

- Depurar la aplicación en entornos de desarrollo como Android Studio.
- Administrar bases de datos en plataformas como Firebase.
- Adquirir nociones para diseñar aplicaciones centradas en usuarios con diseños sencillos e intuitivos.

1.3 Estructura

En este apartado se va a describir los capítulos en los que se va a dividir la memoria. Está estructurada en:

- Introducción: En este primer capítulo se introduce el proyecto, así como la motivación, objetivos y estructura.
- Estado del arte: En este segundo capítulo se analizarán diversas aplicaciones similares a la que se desea desarrollar para determinar funcionalidades interesantes y aspectos de mejora que se podría incorporar a nuestra app.

- Especificación de requisitos: En el tercer capítulo se detallarán las diferentes capas que compondrá el proyecto mediante casos de uso, descripciones con tablas y mockups.
- Tecnologías y servicios utilizados: En el cuarto capítulo se describirá los entornos de desarrollo utilizados para realizar la aplicación y las tecnologías empleadas como Java, Android y Firebase.
- Implementación: En el quinto capítulo se mencionarán los procesos de diseño y desarrollo de la aplicación, así como decisiones de implementación que se han tomado.
- Evaluación: En el sexto capítulo se expondrán los resultados del desarrollo de la aplicación.
- Conclusiones: En este último capítulo se hará una reflexión sobre el trabajo realizado, analizando si se han cumplido los objetivos propuestos.

2. Estado del arte

Actualmente existen muchas aplicaciones de gestión y reserva de pistas de clubes de tenis, la mayoría son para clubes privados y están dedicadas a sus socios, pero también existen otras aplicaciones que están dedicadas al público en general en el que tiene en su catálogo varios clubes en donde se pueden reservar pistas.

A continuación, se van a analizar las características y funcionalidades de varias aplicaciones disponibles en el mercado, unas dedicada a solo socios y otros al público en general.

2.1 Aplicaciones existentes en el mercado

2.1.1 Playtomic

Playtomic [2] es una aplicación con una comunidad de usuarios dedicada a jugadores de deportes de raqueta como pádel y tenis. Además de jugadores también forman parte de ella clubes de todo el mundo, disponiendo así de más de 16 mil pistas para sus usuarios.

Entre las características principales de esta aplicación, encontramos:

- La opción que permite a los usuarios chatear y seguir a otros usuarios a modo de aplicación de red social como se muestra en la segunda imagen de la figura 1, donde se puede ver el botón de chat. Además, permite a sus usuarios crear partidos privados para el usuario y sus amigos o hacerlos público y permitir a otros usuarios unirse a los partidos.
- La facilidad para buscar clubes gracias a la gran cantidad de clubes que forman parte y la posibilidad de filtrar los clubes dependiendo de la distancia a la que están o de una ciudad en concreto.

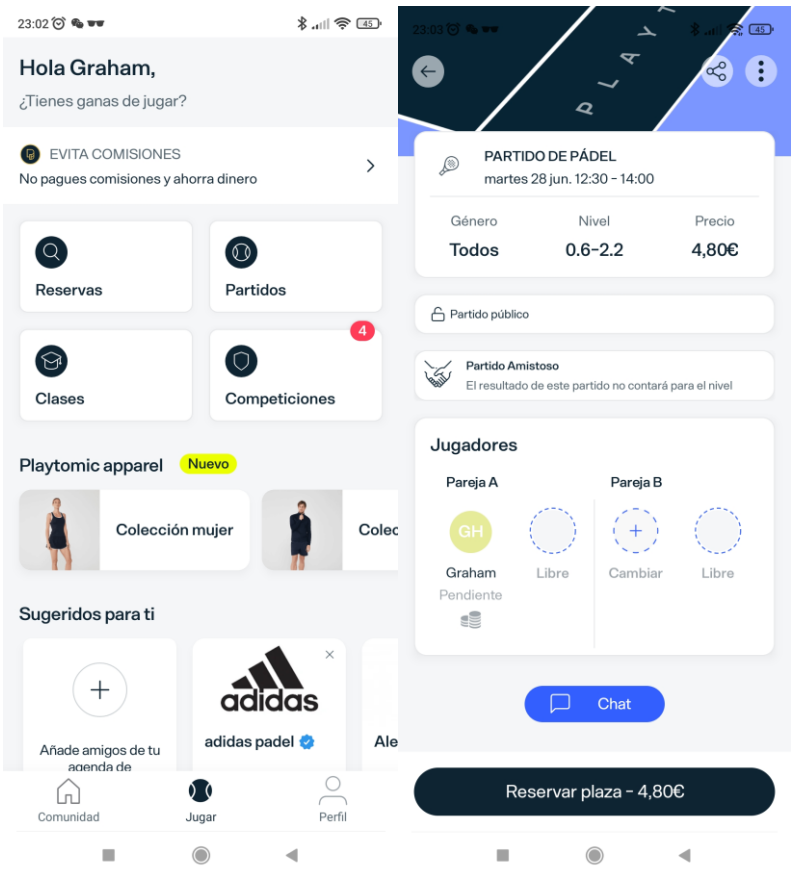


Figura 1. Aplicación Playtomic

2.1.2 ReservaPlay

ReservaPlay [3] es una aplicación para reservar pistas de diferentes deportes como pádel y tenis en los muchos clubes que están registrados. Entre las características principales que encontramos están:

- La necesidad de registrarse en cada uno de los clubes o instalaciones en los que se quiera reservar.
- La necesidad de iniciar sesión en un club para consultar las reservas hechas en este ya que la aplicación no dispone de la posibilidad de consultar todas las reservas hechas entre los diferentes clubes.
- Diseño sencillo y muy visual para hacer las reservas ya que se muestran en rojo los horarios ocupadas y en verde los disponibles (Figura 2).

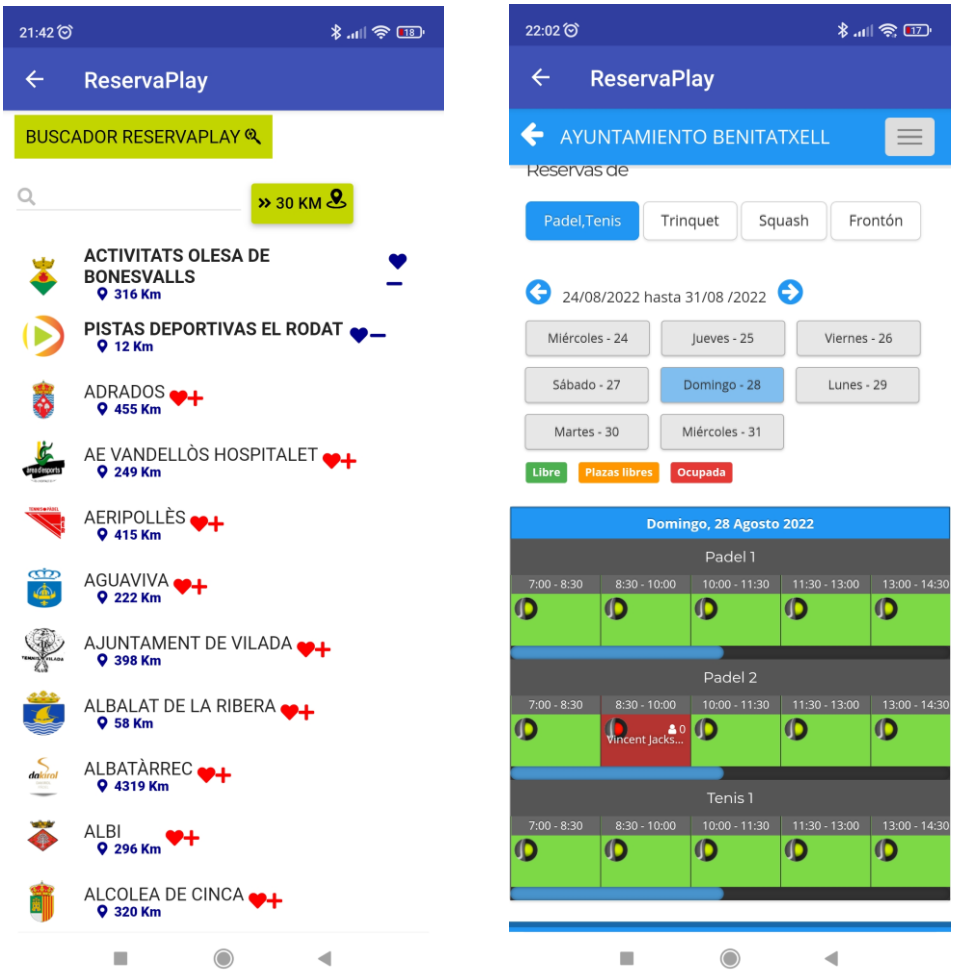


Figura 2. Aplicación ReservaPlay

2.1.3 Club Tenis Pamplona

Club Tenis Pamplona [4], como su nombre indica es el nombre de la aplicación del club de tenis de Pamplona, dedicada a sus socios. La figura 3 muestra la interfaz de la aplicación una vez se inicia sesión. Como se puede ver, la aplicación es mucho más sencilla. Entre sus características principales encontramos:

- Diseño sencillo ya que sirve principalmente para hacer gestiones y como medio informativo del club, en donde incluye información como las tarifas, las instalaciones, calendarios, etc.
- En comparación a otras aplicaciones de uso público, esta al ser de un club privado, es necesario tener el número de socio para poder registrarse y reservar.

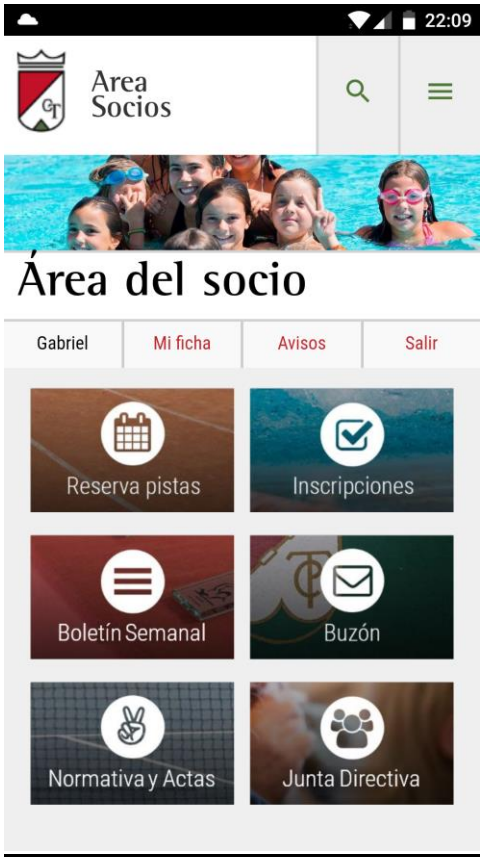


Figura 3. Aplicación Club Tenis Pamplona

2.1.4 Yecla Club Tenis

Yecla Club Tenis [5] es la aplicación utilizada por los socios del club de tenis de Yecla y es necesario ser socio para poder tener una cuenta en la aplicación. Entre las principales características que encontramos están:

- Diseño sencillo ya que sirve principalmente para hacer reservas, incluyendo alguna función secundaria como consultar actividades del club o recargar bonos como se puede ver en la figura 4.

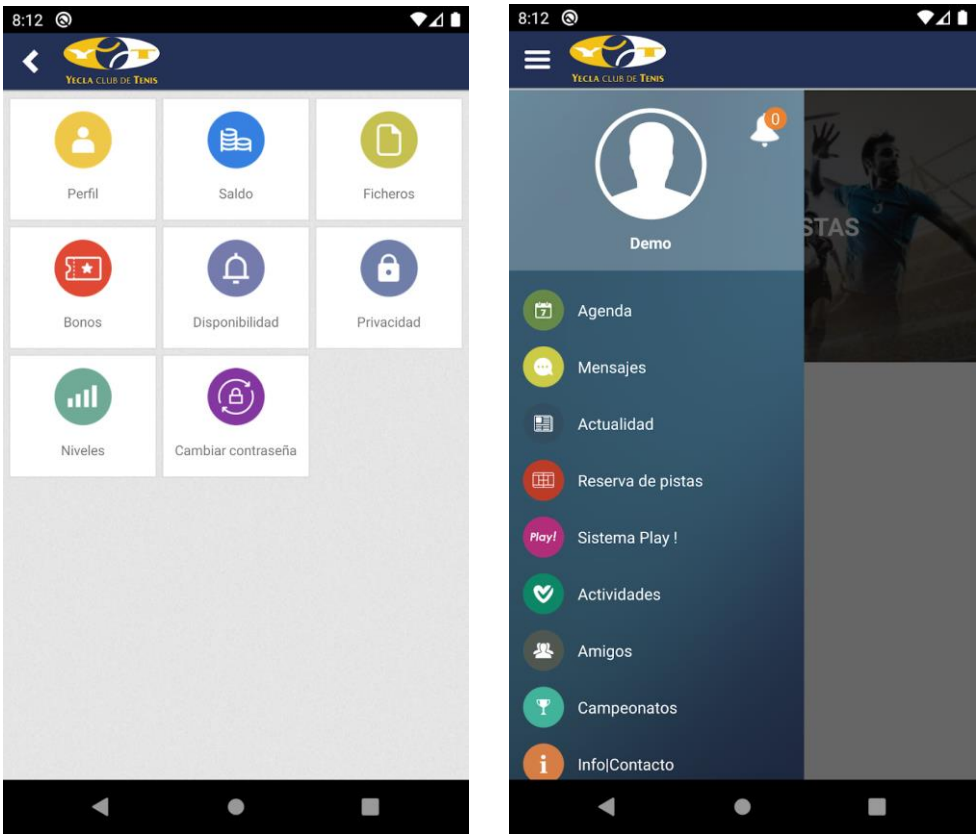


Figura 4. Aplicación Yecla Club de Tenis

2.2 Comparativa entre aplicaciones

Tabla 1. Tabla comparativa de aplicaciones existentes

Funcionalidad	Playtomic	ReservaPlay	Club Tenis Pamplona	Yecla Club Tenis
Pública/Privada	Pública	Pública	Privada	Privada
Sistema operativo	Android/IOS	Android/IOS	Android/IOS	Android/IOS
Dispone función reserva de pistas	Si	Si	Si	Si
Organizan torneos o campeonatos	Si	Si	No	Si
Funcionalidades sociales (chat, seguir usuario…)	Si	No	No	No
Recargar tarjeta virtual o bonos	Si	Si	No	Si
Ser socio para crear cuenta	No	Si/No	Si	Si
Última actualización	21 de junio 2022	24 de noviembre de 2021	19 de abril de 2017	20 de julio de 2020

En la tabla anterior, hemos analizado dos aplicaciones públicas y dos aplicaciones privadas. En el caso de las aplicaciones públicas, como Playtomic, todos los usuarios pueden crearse una cuenta en la aplicación y utilizarla para reservar pistas en uno de los clubes que están registradas en la aplicación, chartear con otros usuarios o seguir a personas como las aplicaciones de redes sociales.

También tenemos aplicaciones públicas como ReservaPlay, en el que la aplicación incluye un catálogo de clubes, pero a diferencia de los clubes en Playtomic, estas pueden ser públicas o privadas. En clubes privados, es necesario ser socio del club para poder iniciar sesión y acceder a las funciones como reservar pista o participar en torneos. Mientras que las públicas permiten el acceso a todos los usuarios.

En cuanto a las aplicaciones privadas como Club Tenis Pamplona o Yecla Club Tenis, estas están dedicadas exclusivamente a los socios de estos clubes, por lo que no son necesarias tantas funcionalidades como las aplicaciones públicas en las que tienen un mayor número de usuarios. En el caso del diseño y funcionalidad, estas son más sencillas y funcionan como un medio de comunicación del club para hacer anuncios o para hacer gestiones y reservas por parte del usuario.

En el caso de nuestra aplicación, aunque la aplicación esté dedicado a clubes, esta será pública de modo que todos los usuarios se puedan registrar en la aplicación sin tener que ser socios del club y se les permitirá reservar pistas. Pero a diferencia de aplicaciones como Playtomic, con un gran número de usuarios, nuestra app se centrará en un número limitado de usuarios que van a un club en concreto, por lo que no será necesaria funciones de tipo red social como chatear por la app, seguir a otros usuarios ni tampoco incorporación de una tienda online en la app.

Por otra parte, dependiendo de si el club organiza torneos o no, se puede valorar si crear una función para organizar torneos y permitir a los usuarios apuntarse desde la aplicación. También se podría valorar si permitir a los socios realizar gestiones del club desde la app, pero como entre los usuarios habrá muchos que no son socios del club a quienes esta funcionalidad no les serviría, por lo que se podría descartar.

3. Especificación

Este proyecto está desarrollo mediante un modelo a 3 capas [6], esta es una técnica de desarrollo de software fundamentada en la programación por capas [7], que divide los componentes de la aplicación en las capas de presentación, de lógica de negocio y de acceso a datos.

Las 3 capas de este proyecto son:

1. **Capa de presentación:** se muestra información al usuario y mediante mockups se le presenta las especificaciones del sistema.
2. **Capa o lógica de negocio:** en esta capa se definen los procesos para el correcto funcionamiento de la aplicación.
3. **Capa de datos:** es la capa que proporciona acceso simplificado a los datos almacenados en la base de datos.

3.1. Lógica de negocio

Para describir los procesos de la lógica de negocio, usaremos diagramas de casos de uso [8] y posteriormente, tablas que detallarán los procesos.

Los casos de uso que detallaremos son los que aparecen el diagrama de la figura 5, que son:

- Registrar usuario
- Iniciar sesión
- Cerrar sesión
- Consultar horarios
- Reservar pista
- Modificar reserva
- Cancelar reserva
- Ver notificaciones
- Ver historial de reservas

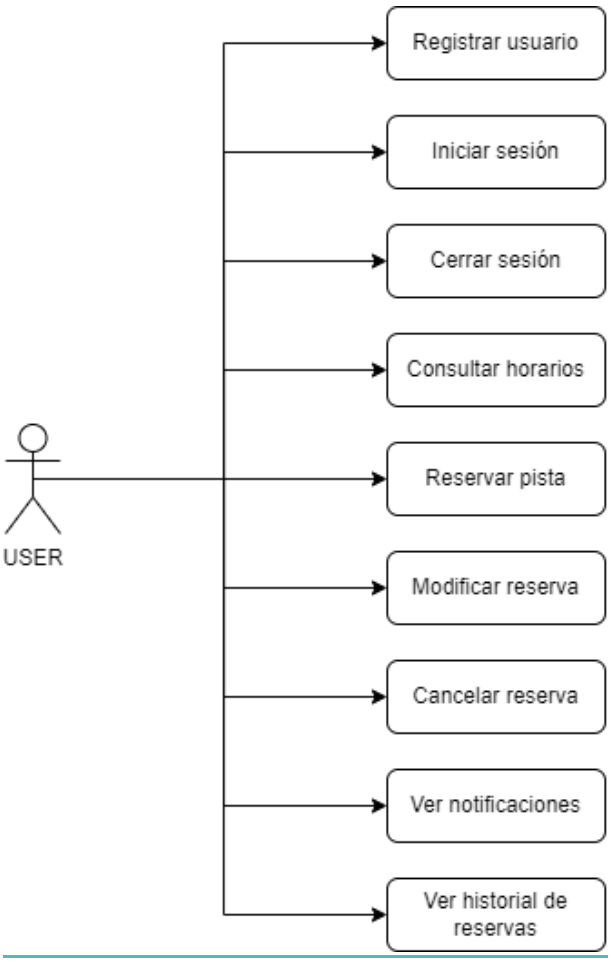


Figura 5. Diagrama casos de uso

- **Iniciar sesión**

Tabla 2. Caso 1, Iniciar sesión

Caso de uso 1	**Iniciar sesión**	
Objetivo en el contexto	El usuario quiere acceder a la aplicación.	
Precondición	El usuario tiene que haberse registrado en el sistema antes de iniciar sesión.	
Actores	Usuario	
Flujo principal	Paso	
	1	Introducir los datos
	2	Pulsar botón de iniciar sesión
Excepciones	Esta acción fallará si se introduce alguno de los datos incorrectamente o falta algún campo por rellenar.	

- **Registrar usuario**

Tabla 3. Caso 2, Registrar usuario

Caso de uso 2	**Registrar usuario**	
Objetivo en el contexto	Un usuario que no tiene cuenta en la aplicación, se quiere crear una cuenta nueva.	
Precondición	El usuario no debe estar registrado en el sistema.	
Actores	Usuario	
Flujo principal	Paso	
	1	Pulsar botón de Crear cuenta nueva
	2	Introducir los datos
	3	Pulsar botón de crear cuenta
Excepciones	1	Esta acción fallará si se introduce un email que ya existe en la base de datos
	2	Se introduce incorrectamente la contraseña a la hora de confirmar la contraseña.
	3	No se introduce la información en algún campo obligatorio.

- **Cerrar sesión**

Tabla 4. Caso 3, Cerrar sesión

Caso de uso 3	**Cerrar sesión**	
Objetivo en el contexto	El usuario quiere cerrar su cesión de la aplicación.	
Precondición	El usuario ha de haber iniciado sesión en la aplicación.	
Actores	Usuario	
Flujo principal	Paso	
	1	Acceder a Mi perfil
	2	Pulsar el botón de cerrar sesión.
Excepciones		

- **Consultar horarios**

Tabla 5. Caso 4, Consultar horarios

Caso de uso 4	**Consultar horarios**	
Objetivo en el contexto	Se quiere consultar las pistas y cuando están disponibles.	
Precondición	El usuario ha de haber iniciado sesión en el sistema para poder consultar la disponibilidad de las pistas.	
Actores	Usuario	
Flujo principal	Paso	
	1	Pulsar el icono de la lupa de buscar
	2	Introducir la fecha y hora que se quiera consultar
	3	Pulsar botón de buscar
Excepciones		

- **Reservar pista**

Tabla 6. Caso 5, Reservar pista

Caso de uso 5	**Reservar pista**	
Objetivo en el contexto	Se ha encontrado una pista en un horario que está disponible y se quiere reservar la pista en ese horario.	
Precondición	El usuario tiene que estar autenticado para poder realizar la reserva de una pista.	
Actores	Usuario	
Flujo principal	Paso	
	1	Pulsar el icono de la lupa de buscar
	2	Introducir la fecha y hora que se quiera consultar
	3	Pulsar botón de buscar
	4	Seleccionar una de las pistas disponibles
	5	Pulsar botón de Reservar
	6	Pulsar botón de Confirmar
Excepciones	No se podrá reservar una pista si en el horario elegido no pistas disponibles.	

- **Modificar reserva**

Tabla 7. Caso 6, Modificar reserva

Caso de uso 6	**Modificar reserva**	
Objetivo en el contexto	Se quiere cambiar la fecha de la reserva.	
Precondición	El usuario debe tener una reserva.	
Actores	Usuario	
Flujo principal	Paso	
	1	Pulsar el icono de los 3 puntos al lado de la reserva
	2	Pulsar la opción de Modificar la reserva
	3	Introducir nueva fecha y hora
	4	Pulsar el botón de buscar
	5	Seleccionar una de las pistas disponibles
	6	Pulsar Modificar reserva
Excepciones		

- **Cancelar reserva**

Tabla 8. Caso 7, Cancelar reserva

Caso de uso 7	**Cancelar reserva**	
Objetivo en el contexto	Se quiere cancelar la reserva por alguna razón del usuario.	
Precondición	El usuario debe tener una reserva.	
Actores	Usuario	
Flujo principal	Paso	
	1	Pulsar el icono de los 3 puntos al lado de la reserva
	2	Pulsar la opción de Cancelar la reserva
	3	Pulsar botón de Confirmar
Excepciones		

- **Ver notificaciones**

Tabla 9. Caso 8, Ver notificaciones

Caso de uso 8	**Ver notificaciones**	
Objetivo en el contexto	Se quiere consultar notificaciones recibidas.	
Precondición	El usuario debe estar autenticado.	
Actores	Usuario	
Flujo principal	Paso	
	1	Acceder a la barra de navegación.
	2	Pulsar la opción de Notificaciones
	3	Pulsar la notificación deseada.
Excepciones		

- **Ver historial de reservas**

Tabla 10. Caso 9, consultar historial de reservas

Caso de uso 9	**Ver historial de reservas**	
Objetivo en el contexto	Se quiere consultar el historial de reservas hechas.	
Precondición	El usuario debe estar autenticado.	
Actores	Usuario	
Flujo principal	Paso	
	1	Acceder a la barra de navegación.
	2	Pulsar la opción de Mis reservas
Excepciones		

3.2. Capa de presentación

Para describir los casos de uso anteriores se han creado varios bocetos exponiendo el diseño de la aplicación y exponiendo el funcionamiento de esta dependiendo de la situación de cada caso de uso.

3.2.1. Iniciar sesión

La figura 6 es la pantalla de inicio de sesión. Es la primera interfaz que aparece al encender la aplicación y coincide con el Caso de uso 1 descrito en el apartado anterior. Como se puede ver en la siguiente figura, la interfaz consta de dos campos de texto donde el usuario debe introducir sus datos de email y contraseña correctamente para iniciar sesión. También incluye las opciones para recuperar la contraseña y la de registrarse para usuarios nuevos.

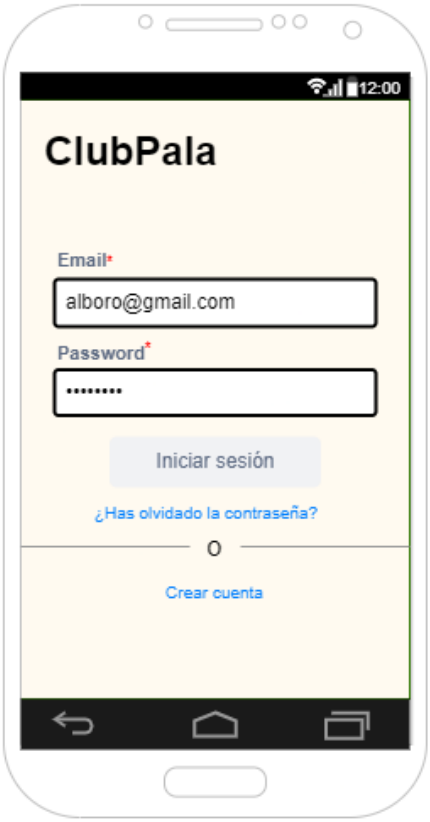


Figura 6. Pantalla inicio de sesión

3.2.2. Registrar usuario

La interfaz de la siguiente figura es a la que accede el usuario para registrarse en el sistema. La interfaz incluye seis campos de texto obligatorios, donde el usuario debe introducir sus datos para poder crearse la cuenta. Si los datos introducidos son correctos, saltará un *pop-up* que indicará que el registro fue exitoso (Figura 7).

Figura 7. Interfaz registro nuevo usuario

3.2.3. Interfaz principal

La figura 8 representa la interfaz principal, que es la interfaz de inicio a la que se accede una vez se inicia sesión en la aplicación. En esta interfaz se pueden consultar las reservas ya hechas e incluye la opción de poder cancelar o modificar la reserva (Figura 9).

Si se pulsa en la opción de cancelar reserva, aparecerá un *pop-up* como el de la figura 10 para confirmar la cancelación. En caso de querer modificar la reserva, esta redirigirá el usuario a la interfaz para consultar horarios de pistas disponibles.

Además de las opciones de reserva también se puede acceder a la barra desplegable de navegación que se encuentra en la parte superior izquierda.

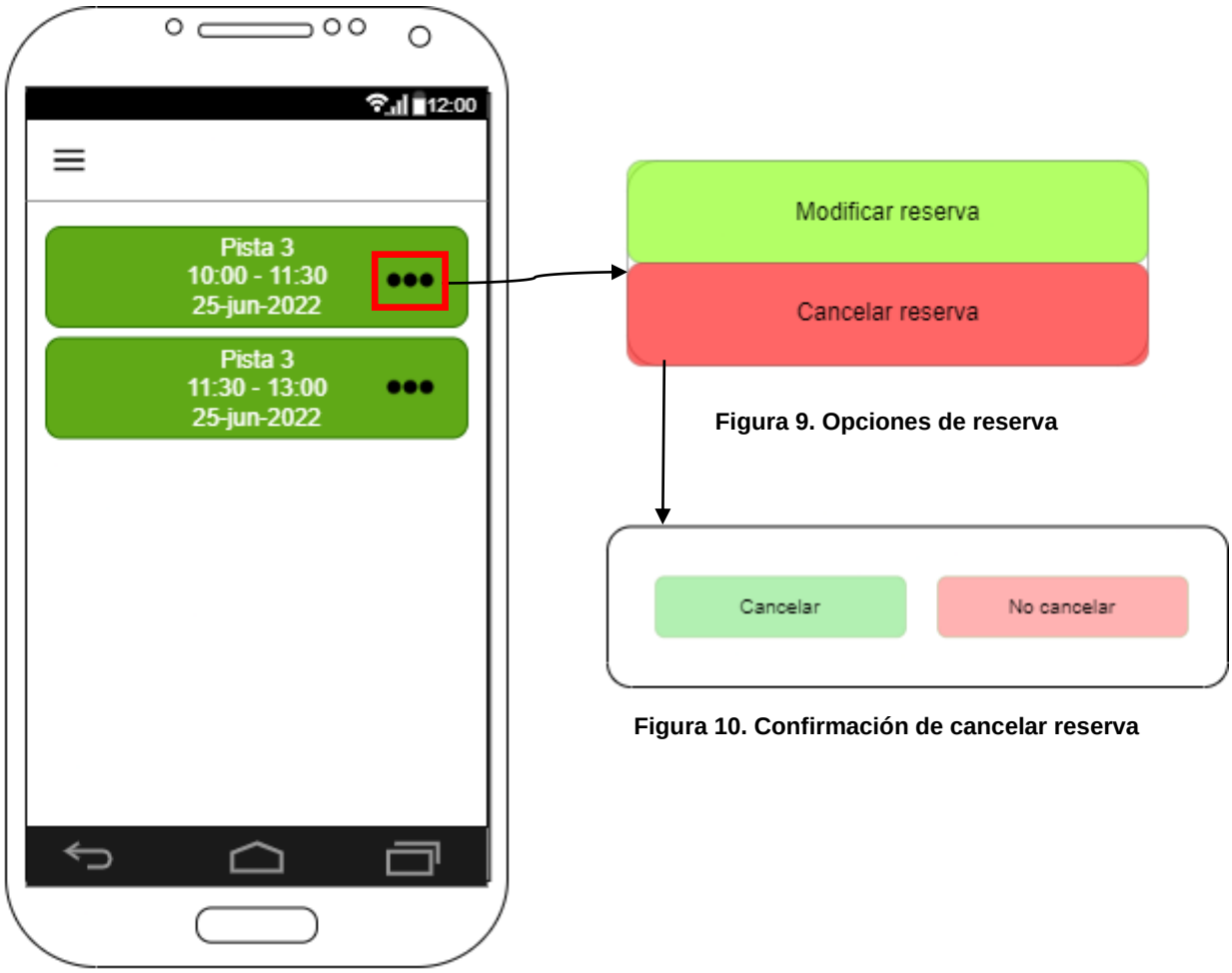


Figura 8. Interfaz de pantalla principal

Figura 9. Opciones de reserva

Figura 10. Confirmación de cancelar reserva

3.2.4. Barra de navegación

El panel lateral de navegación se puede acceder desde la interfaz principal y en ella encontramos las funciones mostradas en la figura 11, como acceder al perfil del usuario, consultar notificaciones, reservas o buscar horarios de pistas disponibles.

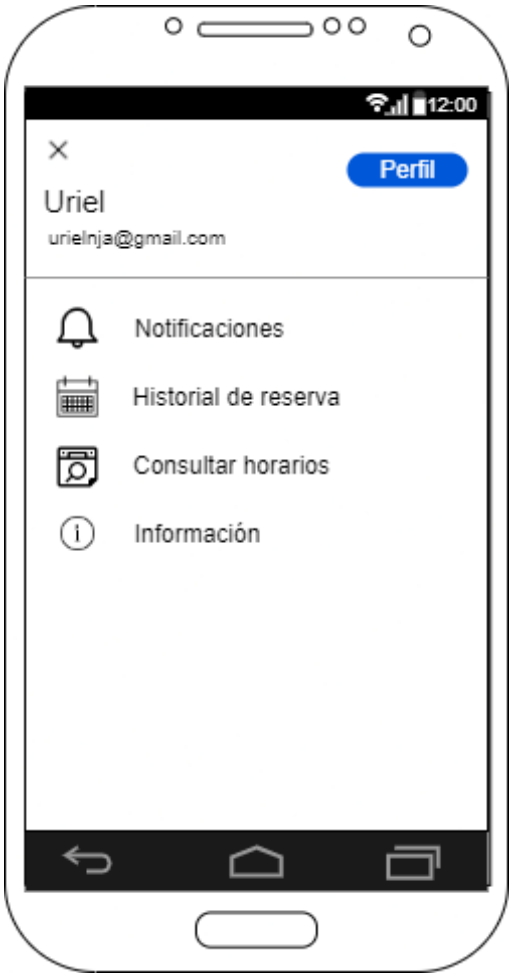


Figura 11. Barra de navegación

3.2.5. Cerrar sesión

Para poder cerrar sesión, se tiene que acceder al del perfil del usuario desde la barra de navegación. En el perfil del usuario encontramos la información del usuario y la opción de cerrar sesión como se muestra en la figura 12.

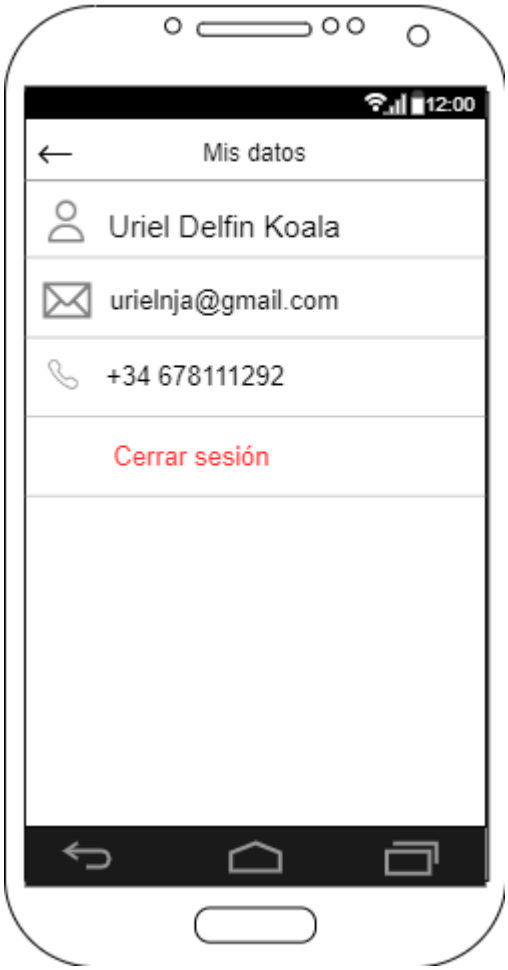


Figura 12. Interfaz perfil usuario

3.2.6. Consultar horarios

Desde la barra de navegación también se puede acceder a la interfaz de consultar horarios, para consultar si hay pistas disponibles en determinadas fechas y horas.

La figura 13 representa esta interfaz, que incluye la opción de elegir fecha pulsando el icono del calendario, lo que despliega un calendario que permite elegir la fecha que se quiera (Figura 14). Para elegir la hora, este despliega un *dropdown* que muestra las franjas de horario en las que se puede reservar (Figura 15).

Una vez se tenga fecha y hora seleccionada, se le pulsa el botón de buscar, y esta mostrará las pistas disponibles. Las disponibles estarán a color y las no disponibles, estarán transparentes (Figura 17).

A continuación, si se pulsa una de las pistas disponibles, se mostrará en pantalla un *pop-up* con la información de la reserva y las opciones de reservar o cancelar (Figura 16).

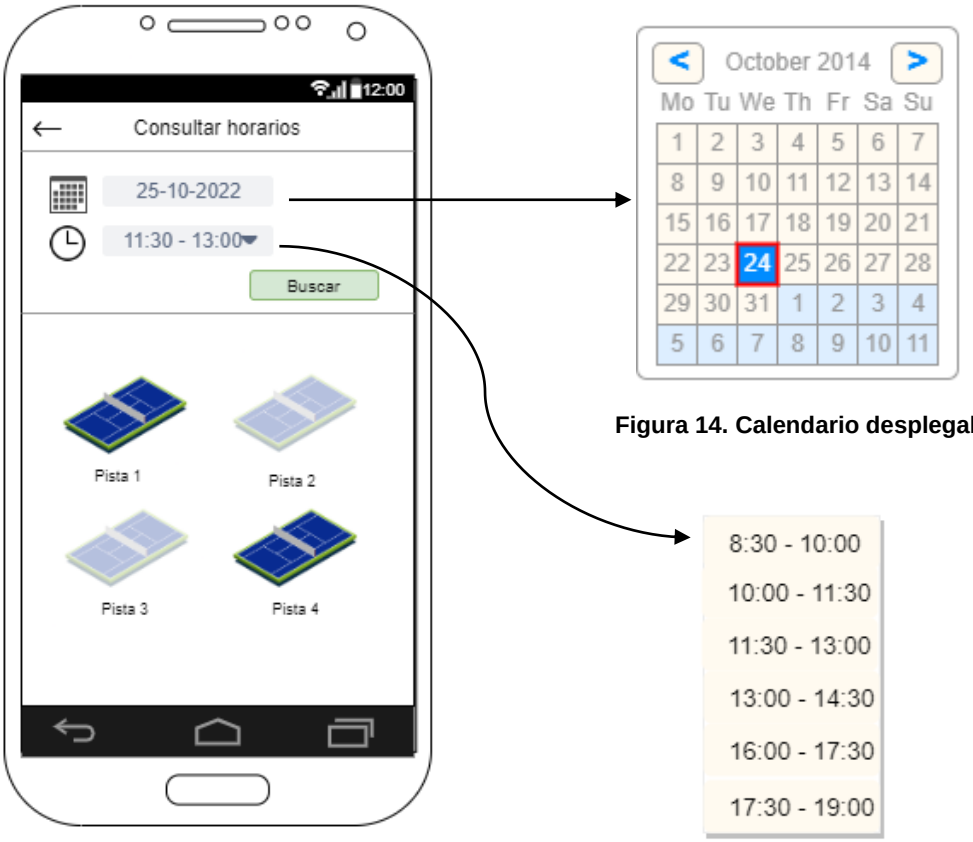


Figura 13. Interfaz consultar disponibilidad

Figura 14. Calendario desplegable

Figura 15. Horarios

Figura 16. Confirmación reserva

Figura 17. Disponibilidad pistas

3.2.7. Ver notificaciones

A esta interfaz se accede desde la barra de navegación. Como se puede ver en la figura 18, dependiendo de si el usuario tiene notificaciones o no, en la interfaz aparecerán las notificaciones recibidas, o en caso de no haber ninguna notificación, aparecerá de fondo el texto “Sin notificaciones”.

Figura 18. Interfaz de notificaciones

3.2.8. Ver historial de reservas

La figura 19 es la interfaz de historial de reservas. Esta interfaz solo permite consultar reservas hechas anteriormente por el usuario.

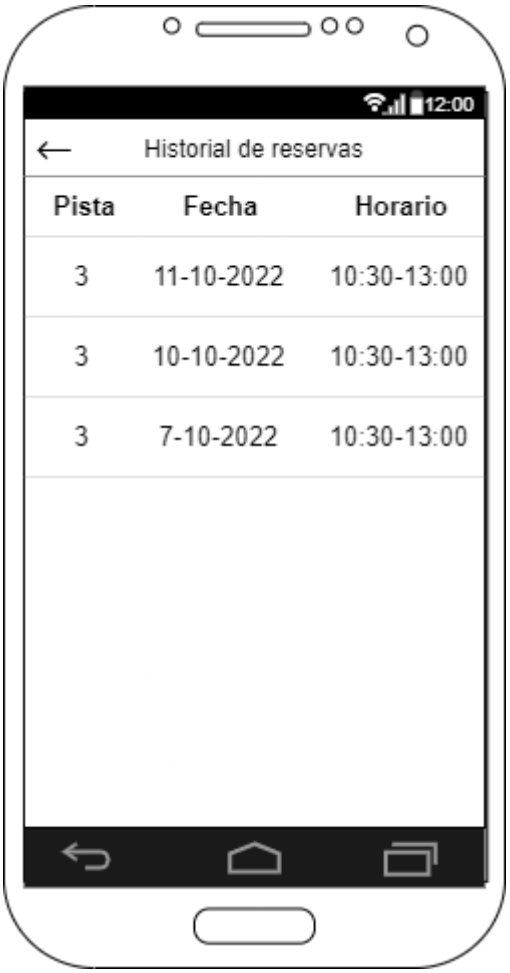


Figura 19. Interfaz historial de reservas

3.3. Capa de persistencia

Tras analizar la base de datos y su flujo, se ha obtenido el siguiente esquema de datos:

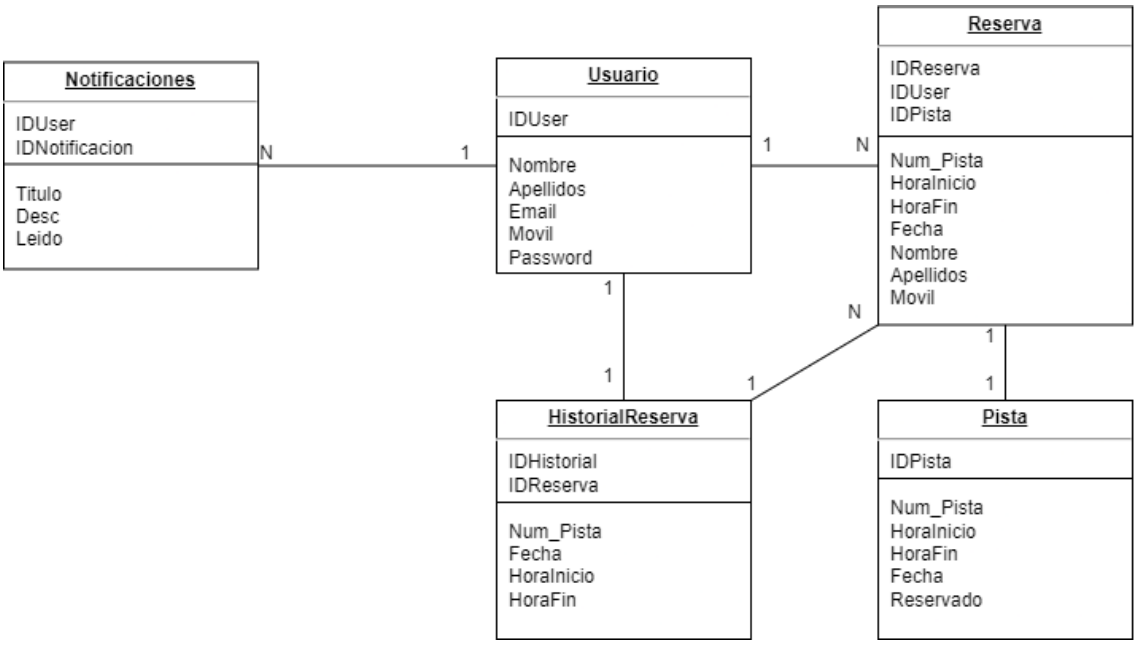


Figura 20. Esquema de la Base de Datos

Como se puede ver en el esquema, la base de datos estará dividida en 5 tablas y se almacenarán en el servidor de Firebase Realtime Database en formato JSON. La tabla de Usuario (figura 23) es donde se añadirán todos los nuevos usuarios que se registren en el sistema y la tabla de Pista (figura 21) contendrá la información del calendario, horario, numero de pista y el valor Reservado que indicará si durante la fecha y hora está disponible una determinada pista.

Por otro lado, tenemos la tabla Reserva (figura 22) que es donde se guardarán todas las reservas hechas por el usuario con IDUser. Estas reservas se copiarán a la tabla historial reserva (figura 24) que el usuario podrá acceder para consultar las reservas pasadas.

Por último, tenemos la tabla de notificaciones (figura 25) en donde se almacena los anuncios u otros mensajes que se le envía al usuario.

Figura 21. Tabla Pista

Figura 22. Tabla Reservas

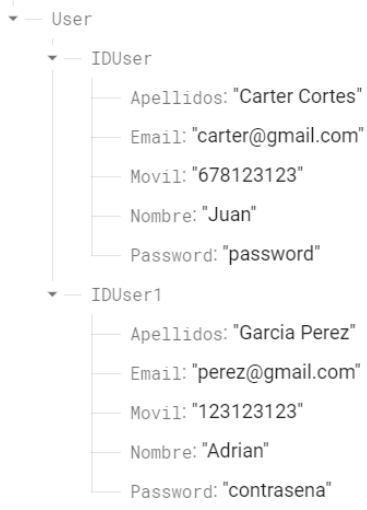


Figura 23. Tabla Usuarios

Figura 24. Tabla Historial de reservas

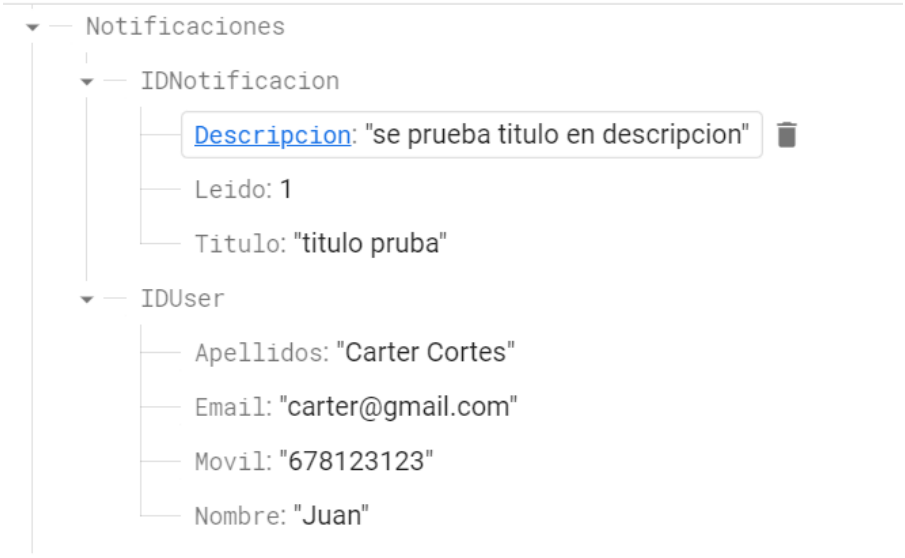


Figura 25. Tabla Notificaciones

4. Tecnologías y servicios utilizados

En este apartado se van a exponer los entornos de desarrollo, lenguajes y tecnologías utilizadas para el desarrollo de este proyecto.

4.1 Android Studio

Android Studio [9] es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones para Android y está basado en IntelliJ IDEA [10]. Este IDE incluye, además de las herramientas de IntelliJ, otras funciones que mejoran la productividad a la hora de desarrollar aplicaciones. En la figura 26 tenemos el logotipo del IDE. Entre las funciones adicionales tenemos:

- Un sistema de compilación flexible basado en Gradle
- Un entorno unificado donde se puede desarrollar aplicaciones para todos los dispositivos Android
- Integración con GitHub y plantillas de código de funciones comunes
- Variedad de marcos de trabajo y herramientas de prueba
- Herramientas de Lint para identificar problemas de rendimiento, usabilidad y compatibilidad de versiones, entre otros
- Administrador de dispositivos que permite emular un dispositivo Android o conectar un dispositivo físico para ejecutar y depurar la aplicación en desarrollo

Figura 26. Android studio

4.2 Java

Java [11] es un lenguaje de programación orientado a objetos y concurrente que produce software para muchas plataformas. Las aplicaciones compiladas en este lenguaje funcionan en la mayoría de los sistemas operativos, incluidos Windows, Linux y MAC OS.

Cabe destacar que este lenguaje sigue siendo uno de los más populares para el desarrollo de aplicaciones Android actualmente junto a lenguajes como Kotlin. La figura 27 representa el logotipo de Java.

Figura 27. Java

4.3 Android

Android [12] es un sistema operativo diseñado para dispositivos móviles con pantallas táctiles y basado en el núcleo de Linux como se puede ver en la arquitectura de Android de la figura 29. La figura 28 representa el logotipo de Android, también conocido como Andy.

Figura 28. Android

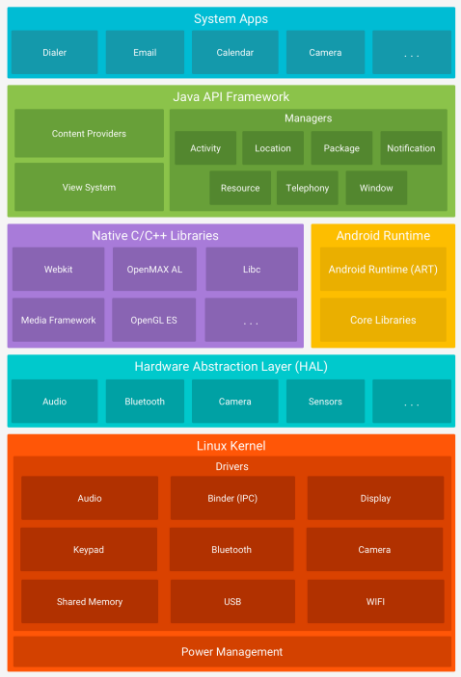


Figura 29. Arquitectura Android

4.3.1 XML

XML [13] (Extensible Markup Language) es un lenguaje de marcado de propósito general, definido por etiquetas. En el caso de Android, toda la jerarquía de la interfaz de usuario está definida mediante XML, la cual organiza las vistas de la aplicación. En la siguiente figura, tenemos el logo de XML.

Figura 30. Logotipo XML

4.3.2 Google Play Developer

La API de Google Play Developer [14] es un servicio web basado en REST [15] que permite realizar una serie de tareas de publicación y administración de aplicaciones. Esta API también permite automatizar tareas de administración de aplicaciones entre las que se encuentran:

- Carga y lanzamiento de nuevas versiones.
- Creación y modificación de las fichas de las aplicaciones en Google Play Store, incluyendo tanto gráficos como texto localizado y capturas de pantalla.
- Administración del catálogo de productos y verificación del estado compra de los mismos.
- Modificación y cancelación de compras de subscripciones.

Figura 31. Google Play

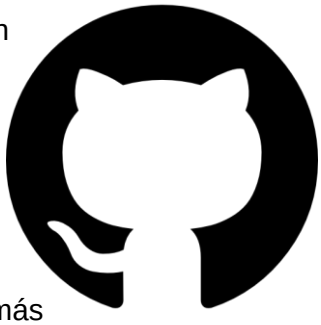
4.3.3 Técnicas de *debugging*

A la hora de desarrollar una aplicación hay que hacer una gran cantidad de tareas de *debugging* (depuración), por lo que el entorno de desarrollo es muy importante para estas tareas ya que, si el entorno ofrece mucha información sobre la causa de los posibles errores durante la ejecución de la app, esto nos puede ahorra mucho tiempo a la hora de encontrar donde se encuentran los fallos y arreglarlos.

En el caso de este proyecto, Android Studio incluye además de herramientas para la depuración de código, un emulador de dispositivo Android y la opción de conectar un dispositivo físico donde se puede ejecutar la aplicación y depurarla mientras se están ejecutando.

4.3.4 Control de versiones

El control de versiones es la práctica de rastrear y gestionar cambios en el código del software. Los sistemas de control de versiones son herramientas de software que permiten al usuario o equipo hacer seguimiento y gestionar los cambios realizados sobre los archivos del programa a lo largo del tiempo.



Para este proyecto utilizaré Github [16] (figura 32), que está basado en el CVS Git y es con el que estoy más familiarizado.

Figura 32. Github

4.4 Firebase

Firebase [17] se trata de una plataforma ubicada en la nube creada por Google. Esta plataforma sirve principalmente para facilitar el desarrollo de aplicaciones calidad de forma rápida y está disponible para diferentes plataformas como IOS, Android y web. También incluye funciones que permite a los desarrolladores combinar y adaptar la plataforma a sus necesidades. La figura 33 representa el logotipo de Firebase.

Figura 33. Logotipo firebase

Entre las principales características de Firebase encontramos:

- Optimización del proceso desarrollo de aplicaciones, mediante diferentes funciones como la detección de errores y de testeo. También permite almacenar todo en la nube, testear la aplicación o configurarla de manera remota.
- Desde la plataforma de Firebase se puede analizar y controlar el rendimiento de la aplicación.
- Gestión de usuarios con servicios como *authentication* que se encargan de gestionar a los usuarios autorizados para que tenga acceso a la aplicación.
- Fácil y rápido de implementar con la API de Firebase.

4.4.1 Autenticación

La mayoría de las aplicaciones necesitan identificar los usuarios para proporcionar el mismo servicio personalizado en los distintos dispositivos del usuario.

Firebase Authentication [18] proporciona servicios de *backend*, SDK fáciles de usar y bibliotecas de IU ya elaboradas para autenticar a los usuarios de las aplicaciones. La autenticación de los usuarios se puede realizar mediante contraseñas, números de teléfono, proveedores de identidad federada, como Google, Facebook y Twitter, entre otros.

Firebase Authentication se integra con otros servicios de Firebase y aprovecha estándares como OAuth 2.0 [19] y OpenID Connect [20] para facilitar la integración con los *backend* personalizados.

Los usuarios se pueden autenticar en la app mediante FirebaseUI como método de autenticación directa o con el SDK de Firebase Authentication, el cual permite integrar uno o más proveedores de acceso a la aplicación.

4.4.2 Base de datos en tiempo real

Firebase Realtime Database [21] es una base de datos NoSQL alojada en la nube que almacena los datos en formato JSON y los sincronizan en tiempo real con cada cliente conectado. Al permanecer la base de datos en la nube, los datos se mantienen disponibles aun cuando la aplicación no tiene conexión. A la hora de compilar la aplicación, todos los clientes comparten una instancia de Realtime Database, lo que les permite recibir actualizaciones automáticamente con los datos más recientes.

4.5 Draw.io

Draw.io [22] es un software de diagramación online totalmente gratuita y de código libre en el que no es necesario crearse ninguna cuenta. En cuanto a la apertura y almacenamiento de los archivos, se pueden guardar en el disco duro local o almacenes en la nuble como Google drive. Esta es la herramienta que usaremos para diseñar los esquemas y diagramas de la memoria. El logotipo de la herramienta es el de la figura 34.

Figura 34. Draw.io

5. Implementación

En este apartado se va a describir el proceso de desarrollo de la arquitectura de la aplicación (Figura 35) y sus funcionalidades. La arquitectura software de la aplicación está basado en modelo MVC [23] separando así los datos de la aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos.

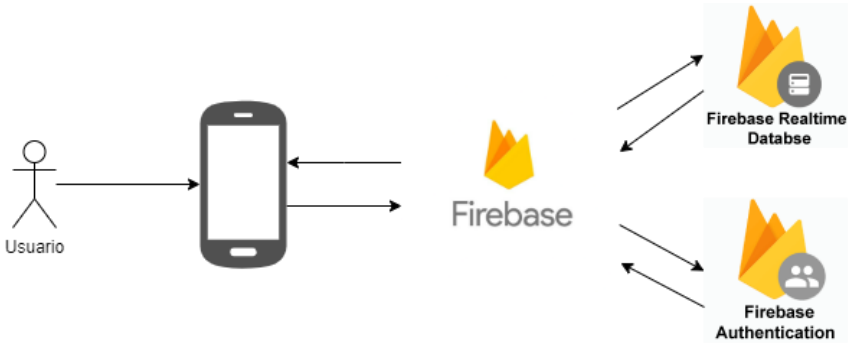


Figura 35. Esquema arquitectura de la aplicación

5.1 Interfaces de navegación

Comenzaremos comentado la implementación de las diferentes interfaces de la aplicación.

5.1.1 Interfaz de inicio de sesión

Una vez se inicia la aplicación, la interfaz que aparece por defecto es la de inicio de sesión que corresponde con la figura 36. En esta ventana, el usuario debe introducir su email y contraseña para poder iniciar sesión. También puede hacer click en “Registrarse” para pasar a la interfaz de registro.

Figura 36. Ventana de inicio sesión

En caso de error al intentar iniciar sesión, se mostrará un *toast* indicando al usuario de que ha habido un error al intentar iniciar sesión. El toast es un mensaje emergente relacionado con alguna interacción realizada por el usuario que desaparece tras unos segundos.

5.1.2 Interfaz de registro

Desde la ventana de registro el usuario debe rellenar todos los campos obligatorios para poder registrarse. En caso de error porque el formato de email o un campo obligatorio esté vacío, se mostrará un icono de error en el *textfield.* Los textfields son los cuadros de texto presentes en el formulario como se muestra en la figura 37.

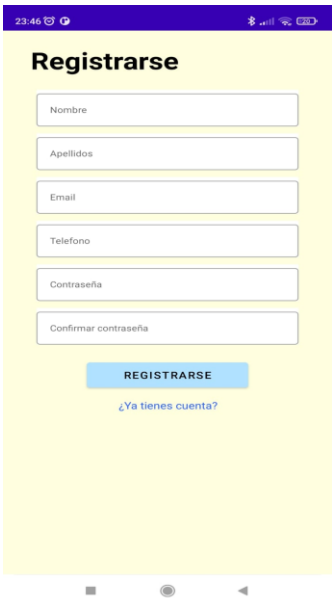


Figura 37. Ventana de registro

Para simplificar el diseño de la interfaz, se ha utilizado *outlined textfields* de la guía de Material Design de Google [24], el cual permite agregar bordes al cuadro de texto, también añade otras funcionalidades como por ejemplo al seleccionar un campo, se muestra el nombre del campo en el borde del cuadro de texto. Esto se ha conseguido añadiendo com.google.android.material.textfield.TextInputEditText dentro de un com.google.android.material.textfield.TextInputInputLayout.

5.1.3 Interfaz principal

Esta interfaz corresponde a la figura 38 e incluye un NavigationDrawer, que es un panel lateral desplegable para proveer a la aplicación una mayor funcionalidad y acceso a destinos.

Como se puede ver en la figura 39, el panel de navegación está dividido por 2 partes:

- La cabecera en donde incluye el nombre y mail del usuario y un botón para ir a la vista del perfil del usuario.
- El menú que se ha creado a partir de un XML agrupando los cuatro ítems en un grupo.

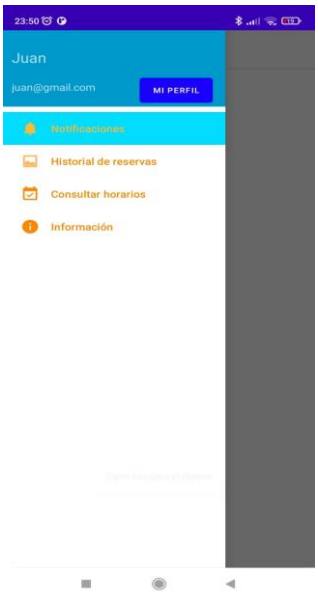


Figura 38. NavigationDrawer

Figura 39. Venta principal

En cuanto a los cuadros de las reservas, para que estas se muestren en una misma columna y que se pueda deslizar la pantalla hacia abajo para ver más reservas de la manera más eficiente, se ha utilizado un *RecyclerView*. Este recicla los elementos de modo que creamos un diseño con los elementos que deben crear de forma dinámica, que en este caso es un *textview* en donde aparecen los datos y un botón para mostrar opciones. Para crear estos objetos de forma dinámica, se utiliza un *ViewHolder* como contenedor de vistas de los elementos individuales que se van a cerar. Luego, con un adaptador se solicitan las vistas y vinculan lo datos al *RecycleView*. Por último, mediante un *LayoutManager* o administrador de diseño organizamos los elementos de la lista.

Si se hace click en el icono de más opciones, se abrirá un cuadro de diálogo con las opciones de cancelar la reserva y modificar la reserva. Al cancelar la reserva, el cuadro de la reserva desaparece de la interfaz y si se hace click en modificar reserva, este se pasará a la ventana de buscar horarios.

Para que en la vista de buscar horarios se pueda modificar la reserva en vez de reservar una nueva, se ha establecido dos variables globas.

```
public static int modifyID = 0;
public static String reservaID;
```

5.1.4 Interfaz perfil del usuario

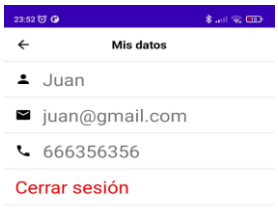


Figura 40. Ventana perfil

Para la vista del perfil del usuario, como se puede ver en la figura 40, la vista muestra los datos del usuario y es donde se encuentra el botón de cerrar sesión.

5.1.5 Interfaz notificaciones

Las notificaciones de la vista como se puede ver en las pantallas de la figura 41, aparece el texto “Sin notificaciones”, cuando no tiene notificaciones. El usuario puede hacer click en la notificación y aparecerá un cuadro de diálogo que mostrará la información de la notificación.

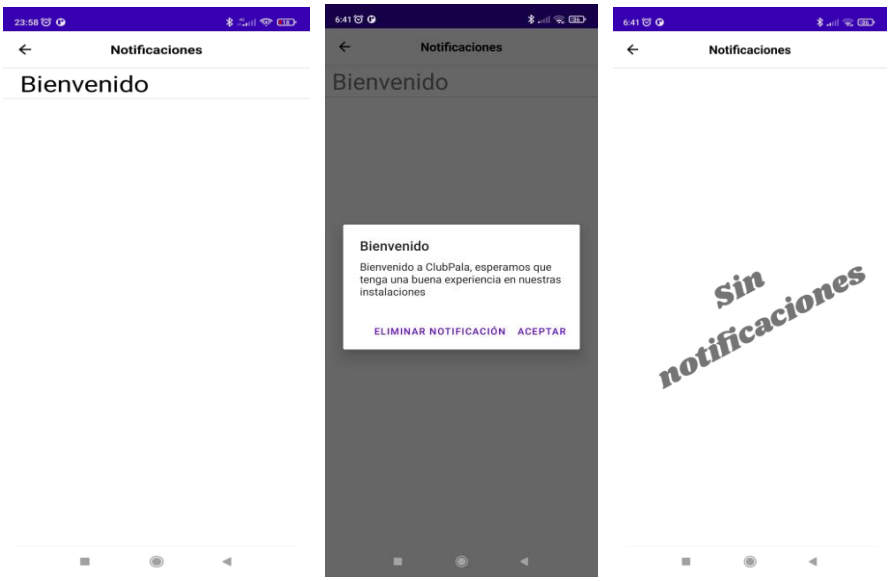


Figura 41. Ventana notificaciones

5.1.6 Interfaz historial de reservas

En esta vista se muestran todas las reservas anteriores al día de hoy que ha tenido el usuario. Se muestran como en la figura 42.

Figura 42. Ventana historial reservas

5.1.7 Interfaz buscar horarios

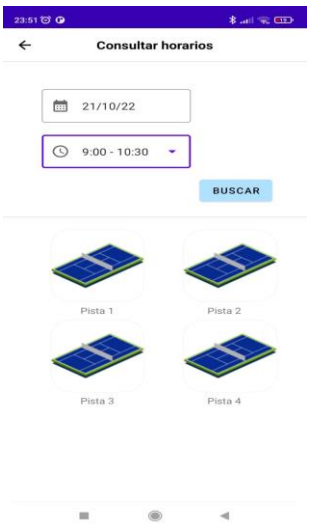


Figura 43. Ventana consultar horarios

Esta es la vista en la que se van a hacer todas las reservas y modificaciones de reservas. Como se puede ver en la figura 43, la vista contiene dos desplegables. Al hacer click en el primero, este abrirá un calendario para que el usuario elija la fecha de la reserva. El segundo desplegable contiene la franja horaria de la reserva. Una vez seleccionado estos dos campos, se pulsa el botón de buscar y aparecerá en la parte inferior cuatro imágenes de pistas. En caso de que esté disponible, esta aparecerá en color y si alguna no está disponible, aparecerá transparente.

Para reservar, solo hay que pulsar en una de las imágenes, se abrirá un cuadro de diálogo mostrando los datos de la reserva donde hay que pulsar confirmar.

En caso de que la variable global “modifyID” esté a 1, en vez de reservar, aparecerá la opción de modificar siguiendo el mismo proceso que reservar.

5.2 Firebase

En este apartado se va a comentar el método utilizado para autenticar a los usuarios y las clases utilizadas para actualizar la base de datos de Firebase.

5.2.1 Autenticación

Para la autenticación, primero se ha habilitado el método de registro con email y contraseña en la consola de Firebase Authentication. Una vez se ha tenido las configuraciones finalizadas, se establece la conexión con Firebase en el método onCreate().

```
        //Variable Firebase
FirebaseDatabase database;
DatabaseReference myRef;
        @Override
protected void onCreate(Bundle savedInstanceState) {
      …
      mAuth = FirebaseAuth.getInstance();
      database = FirebaseDatabase.getInstance();
        …
}
```

Una vez se han establecido la conexión y realizado las comprobaciones sobre el formulario de registro se llama a la funciones registerUserData() y registerNewUser().

El método registerUserData() sirve para guardar los datos del nuevo usuario en Firebase Realtime Database.

```
//Register user data to realtime database
private void registerUserData(){
    database = FirebaseDatabase.getInstance();
    myRef = database.getReference("user");

    BD_user newUser = new BD_user(name_text, lastname_text, email_text, tel_text,
pass_text);

    myRef.child(mAuth.getCurrentUser().getUid()).setValue(newUser);
}
```

El método registerNewUser() crea un nuevo usuario en Firebase y lanza la pantalla Main que corresponde a la pantalla de inicio.

```
        //Register new user in firebase Authentication
private void registerNewUser(){
    mAuth.createUserWithEmailAndPassword(email_text, pass_text)
            .addOnCompleteListener(new OnCompleteListener<AuthResult>() {

                @Override
                public void onComplete(@NonNull Task<AuthResult> task)
                {
                    if (task.isSuccessful()) {
                        Toast.makeText(
                                getApplicationContext(), R.string.registro_correcto,
                                Toast.LENGTH_LONG).show();
                        registerUserData();

                        // if the user created intent to login activity
                        Intent intent = new Intent(registro.this, MainActivity.class);
                        startActivity(intent);
                    }
                    else {

                        // Registration failed
                        Toast.makeText(
                                getApplicationContext(), R.string.registro_fallido,
                                Toast.LENGTH_LONG).show();
                    }
                }
            });
}
```

En cuanto al inicio de sesión, se ha utilizado el método checkUserAndLogin() que comprueba si existe un usuario autenticado con el email y la contraseña introducido. En caso positivo, se lanza la actividad Home que corresponde a la pantalla principal.

```
    private void checkUserAndLogin(){
    …
  // signin existing user
  mAuth.signInWithEmailAndPassword(email_text, password_text)
    .addOnCompleteListener(
      new OnCompleteListener<AuthResult>() {
        @Override
        public void onComplete(NonNull Task<AuthResult> task)
        {
          if (task.isSuccessful()) {
            Toast.makeText(
                    getApplicationContext(), R.string.inicio_exitoso,
                    Toast.LENGTH_LONG).show();
            // if sign-in is successful
            Intent intent = new Intent(MainActivity.this,
                HomeActivity.class);
            startActivity(intent);
          }  else {
            // sign-in failed
            Toast.makeText(
                    getApplicationContext(), R.string.inicio_fallido,
                    Toast.LENGTH_LONG).show();
          }
        }
      });
}
```

5.3 Modelo de datos

El modelo de datos de la aplicación, a diferencia de lo analizado en el apartado 3.3 de especificación de la capa de persistencia, la app se va a quedar finalmente con 3 tablas que vendrán definidas por las clases BD_user, BD_notificaciones y BD_reserva.

5.3.1 Clase BD_user

La clase BD_user se va a utilizar para enviar y recibir datos de usuarios en Firebase Realtime Database.

La clase incluye métodos getter y setter las variables nombre, apellidos, email, número de teléfono y contraseña.

```
public class BD_user {

    String name, lastname, email, tel, pass;
    public BD_user(String name, String lastname, String email, String tel, String pass) {
        this.name = name;
        this.lastname = lastname;
        this.email = email;
        this.tel = tel;
        this.pass = pass;
    }

    public BD_user(){    }

    public String getName() {return name; }
    public void setName(String name) {this.name = name; }

    public String getLastname() {return lastname; }
    public void setLastname(String lastname) {this.lastname = lastname; }

    public String getEmail() {return email; }
    public void setEmail(String email) {this.email = email; }

    public String getTel() {return tel; }
    public void setTel(String tel) {this.tel = tel; }

    public String getPass() {return pass; }
    public void setPass(String pass) {this.pass = pass; }
}
```

5.3.2 Clase BD_reserva

La clase BD_reserva se va a utilizar para enviar y recibir datos de reservas de usuarios en Firebase Realtime Database.

La clase incluye métodos getter y setter las variables fecha, hora inicio, hora fin y pista.

```
public class BD_reserva {

    String userID, fecha, hora_ini, hora_fin, pista;

    public BD_reserva(){   }

    public BD_reserva(String userID, String fecha, String hora_ini, String hora_fin, String
pista) {
        this.userID = userID;
        this.fecha = fecha;
        this.hora_ini = hora_ini;
        this.hora_fin = hora_fin;
        this.pista = pista;
    }

    public String getUserID() {        return userID;    }
    public void setUserID(String userID) {        this.userID = userID;    }

    public String getFecha() {        return fecha;    }
    public void setFecha(String fecha) {        this.fecha = fecha;    }

    public String getHora_ini() {        return hora_ini;    }
    public void setHora_ini(String hora_ini) {        this.hora_ini = hora_ini;    }

    public String getHora_fin() {        return hora_fin;    }
    public void setHora_fin(String hora_fin) {        this.hora_fin = hora_fin;    }

    public String getPista() {       return pista;    }
    public void setPista(String pista) {        this.pista = pista;    }
}
```

5.3.3 Clase BD_notificaciones

La clase BD_notificacioens se va a utilizar para enviar y recibir datos de las notificaciones de los usuarios en Firebase Realtime Database.

La clase incluye métodos getter y setter las variables descripción, título y userID.

```
public class BD_notificaciones {

    String description, title, userID;
    long read;

    public BD_notificaciones(){     }

    public BD_notificaciones(String description, String title, long read, String userID) {
        this.description = description;
        this.title = title;
        this.read = read;
        this.userID = userID;
    }

    public String getDescription() {        return description;    }
    public void setDescription(String description) {        this.description = description;    }

    public String getTitle() {        return title;    }
    public void setTitle(String title) {        this.title = title;    }

    public long getRead() {        return read;    }
    public void setRead(long read) {        this.read = read;    }

    public String getUserID() {        return userID;    }
    public void setUserID(String userID) {        this.userID = userID;    }

}
```

6. Evaluación

Una finalizada la implementación, se han hecho tareas de depuración para solucionar los fallos de la aplicación. Para ello se ha conectado un dispositivo físico a Android Studio mediante wifi con el administrador de dispositivos. Una vez conectada se ha instalado la aplicación y probado el funcionamiento. Para mostrar el funcionamiento, se han diseñado los siguientes contextos de uso para la usuaria Ana que utiliza el dispositivo Redmi Note 9 Pro con Android 11.

La usuaria Ana se acaba instalar la aplicación a recomendación de un amigo para reservar una pista y jugar un partido. Así que Ana se registra, inicia sesión y reserva la pista 4 para el día 25/10/2022 a las 15:00, pero uno los jugadores no pueden ir ese día, así que Ana decide cambiar la reserva para el día siguiente a las 18:00. Pasado un tiempo usando la aplicación, Ana quiere ver su historial de reservas para ver cuántas reservas ha hecho y también, ver si tiene alguna notificación nueva del club. Después de ver las notificaciones, Ana accede a su perfil y cierra sesión. (Véase las pantallas de las figuras 44, 45, 46, 47 y 48)

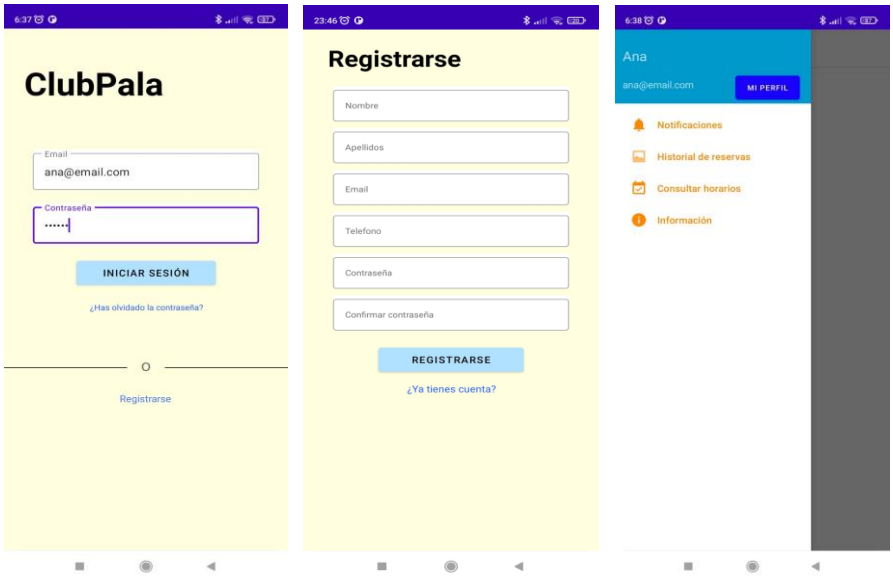


Figura 44. Pantalla de inicio y principal

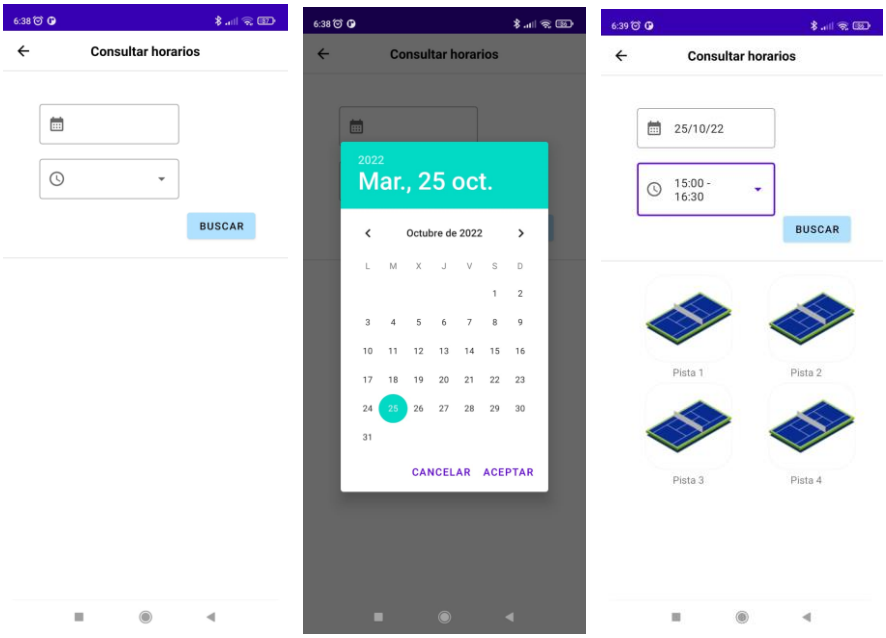


Figura 45. Pantalla de consultar horarios y proceso de reserva de pista

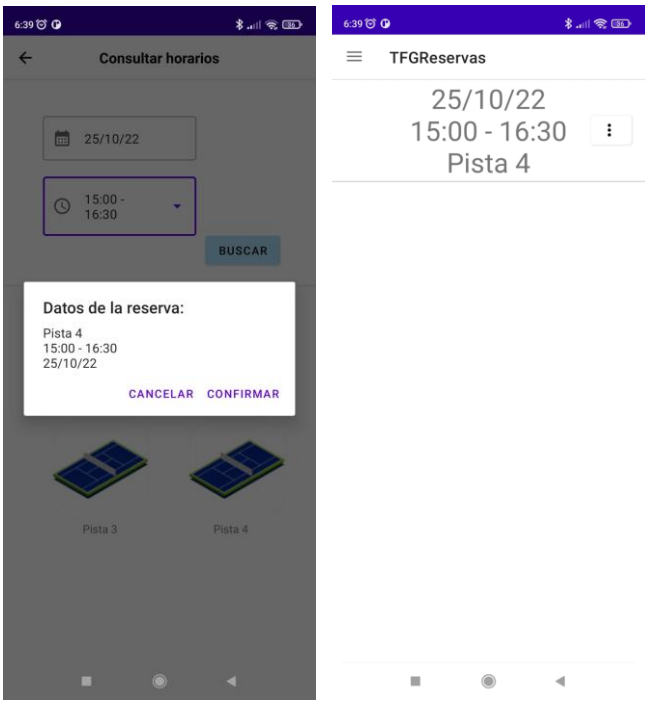


Figura 46. Confirmar datos de la reserva

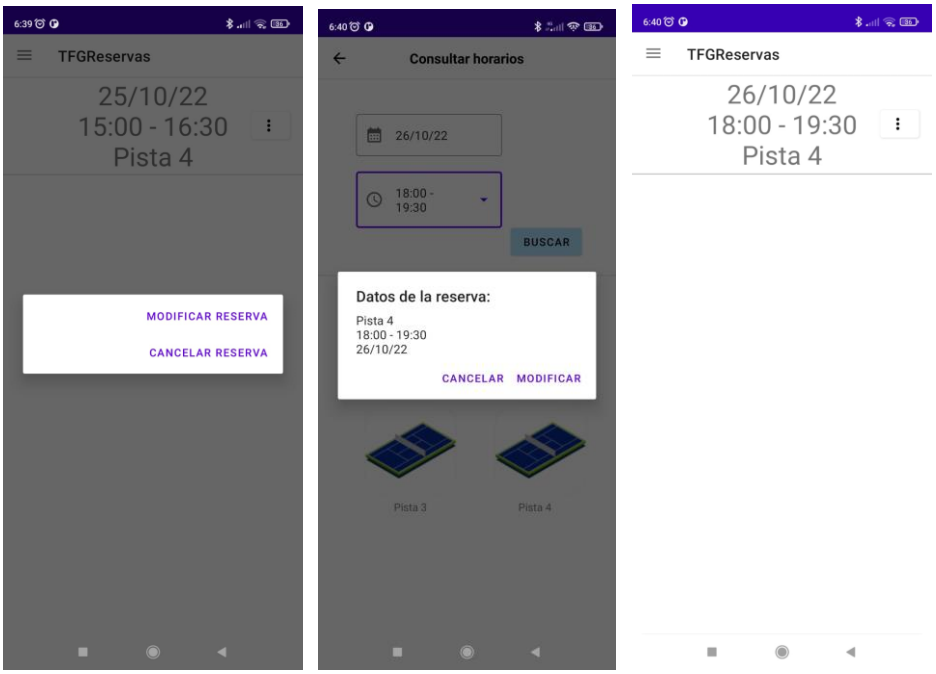


Figura 47. Modificar reserva

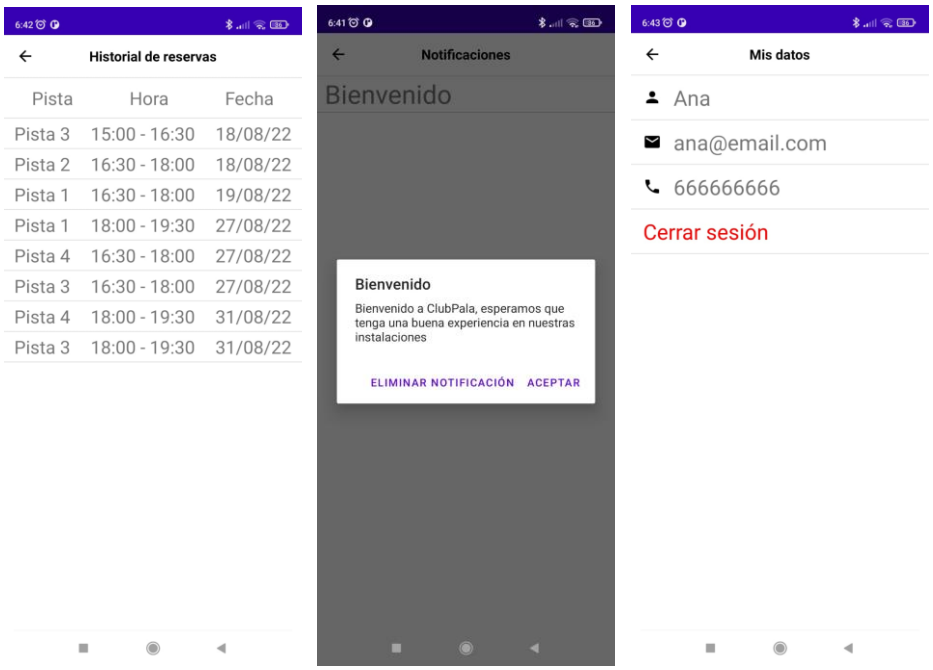


Figura 48. Pantalla de historial de reservas, notificaciones y perfil

Para probar el funcionamiento en distintos dispositivos, se ha, creado varios dispositivos virtuales en el SDK de Android. Los dispositivos son Redmi Note 9 Pro, Pixel, Pixel 2 y Nexus 5. En ellas hemos hechos pruebas de arranque de la aplicación y han tardado entre 1.5 y 3 segundos. Y en cuanto al inicio de sesión, han tardado de 1.5 a 2.5 segundos.

Siguiendo la guía de calidad de Google [25] se han probado y validados criterios como el rendimiento, la estabilidad, la navegación y el uso de recursos obteniendo como resultado las tablas 11, 12 y 13.

Tabla 11. Experiencia visual

Área	Prueba	Descripción	Cumple
Navegación	CR-1, CR-3, CR-5	La app admite el uso del botón atrás en todas las pantallas y preserva y restaura correctamente el estado del usuario o la app.	Si
Notificaciones	CR-9	Las notificaciones siguen los lineamientos de Material design.	No se envían notificaciones
IU y gráficos	CR-5	La app soporta la orientación horizontal	No
Calidad visual	CR-all	La app muestra gráficos, texto, imágenes y otros elementos de la IU sin distorsión, esfumado ni pixelado.	SI

Tabla 12. Rendimiento y establidad

Área	Prueba	Descripción	Cumple
Estabilidad	CR-all	La app no falla ni bloquea el subproceso de IU que provoca errores ANR.	Si
Rendimiento	CR-all	La app carga rápidamente.	Si
SDK	CR-0, SP-1	La app se orienta al SDK de Android más reciente y se compila con el último SDK estableciendo el valor compileSdk.	No
Batería	BA-1	La app admite las funciones de administración de energía de Android 6.0 (Descanso y App Standby)	No

Tabla 13. Privacidad y seguridad

Área	Prueba	Descripción	Cumple
Permisos	CR-0, SC-4	La app solicita la cantidad mínima de permisos necesarios.	Si
Datos y archivos	SC-1	Todos los datos sensibles se almacenan en el almacenamiento interno de la app.	No
Identidad	CR-0	Brinda sugerencias para autocompletar las credenciales de la cuenta y otra información sensible.	No
Componentes de la app	SC-5	Solo se exportan los componentes de la aplicación que comparten datos con otras apps.	Si
	CR-0, SC-4	Todos los intents y transmisiones siguen las prácticas recomendadas.	Si
Redes	SC-9	El tráfico de red se envía mediante SSL.	Si
Bibliotecas	SP-2	Todas las bibliotecas, los SDK y las dependencias están actualizadas.	Si
WebViews	SC-6	Se utiliza WebViewAssetLoader	No implementado
Ejecución		La app utiliza Android App Bundles.	Si

Existen más pruebas de calidad relacionadas con otros criterios como la funcionalidad (audio, multimedia…) que hemos omitido ya que la aplicación no tiene ninguna funcionalidad relacionada a estas. Otras pruebas que también hemos omitido son los relacionados con la publicación en Google Play, debido a que por el momento se tiene la intención de publicar la app.

Por último, cabe destacar que la app ocupa 20.27 MB y como los datos se almacenan en el servidor de Firebase Realtime Database, la aplicación no va a ocupar mucho más espacio. Por lo que podemos decir que el tamaño de la aplicación comparado a la capacidad de almacenamiento que tienen los dispositivos Android hoy en día es despreciable.

7. Conclusiones

El reciente aumento del interés en los deportes de raqueta ha llevado a que se abran muchos espacios dedicados a deportes como el pádel. Por esta razón decidí desarrollar una aplicación Android para facilitar el proceso de reserva en estos espacios.

El proyecto ha sido desarrollar una aplicación en Android Studio desde cero y usando productos ofrecidos por Firebase como *backend,* almacenando así toda la base datos en la nube sin necesidad de almacenamiento de datos en local y reduciendo la complejidad de la implementación utilizando el servicio de autenticación ofrecido. El resultado obtenido ha sido una aplicación funcional en el que los usuarios se pueden registrar y reservar una pista en un determinado club.

Entre las tareas que más tiempo han llevado, hay que destacar el diseño e implementación de las ventanas, ya que no teníamos conocimiento previo de Android Studio, hemos tenido que buscar por internet posibles maneras de implementar algunos detalles de las vistas. Aunque con la existencia de la página Material Design creada por Google donde se documentan guías para diseñar las características de la ventana, se ha reducido bastante el tiempo que originalmente se habría necesitado.

Para terminar, podemos decir que la aplicación se podría mejorar bastante añadiendo más funcionalidades como la posibilidad de recargar o pagar una reserva. Pero como de momento no existe la intención de publicar la aplicación y ponerla al mercado, se ha decido descartar la idea de añadir este tipo de funcionalidades a la aplicación. Aunque eso no quita la posibilidad de mejorar la aplicación una vez se hayan obtenido más conocimiento y experiencia en el entorno de Android.

8. Bibliografía

[1] Marca.es, «El pádel es el deporte que más ha crecido a nivel mundial en 2021,» 2022. [En línea]. Available: https://www.marca.com/blogs/espanasemueve/2022/06/03/el-padel-se-convierte-en-el-deporte-que.html#:~:text=Las%20visualizaciones%20del%20World%20Padel,%25%20hasta%20casi%20el%20500%25.. [Último acceso: 25 Abril 2022].

[2] «Playtomic,» 2022. [En línea]. Available: https://playtomic.com/. [Último acceso: 26 abril 2022].

[3] «ReservaPlay,» 2022. [En línea]. Available: https://www.reservaplay.com/. [Último acceso: 26 abril 2022].

[4] «Club Tenis Pamplona,» 2022. [En línea]. Available: https://www.club-tenis.com/portal/default.aspx. [Último acceso: 26 abril 2022].

[5] «Club Tenis Yecla,» 2022. [En línea]. Available: http://www.yct.es/#1. [Último acceso: 26 abril 2022].

[6] Zona Coder, «Arquitectura en Tres Capas (Programación por capas),» 3 marzo 2018. [En línea]. Available: https://zonacoder.wordpress.com/category/java-ee/arquitectura-en-tres-capas-programacion-por-capas/. [Último acceso: 1 julio 2022].

[7] M. Fowler, D. Rice, M. Foemmel, E. Hieatt, R. Mee y R. Stafford, Patterns of Enterprise Application Architecture, Primera ed., Addison Wesley, 2002.

[8] A. Cockburn, Writing effective use cases, Primera ed., Addison-Wesley Professional, 2000.

[9] Google, «Android Studio,» 2022. [En línea]. Available: https://developer.android.com/studio. [Último acceso: 15 junio 2022].

[10] JET BRAINS, «IntelliJ IDEA,» 2022. [En línea]. Available: https://www.jetbrains.com/es-es/idea/. [Último acceso: 15 junio 2022].

[11] ORACLE, «JAVA,» 2022. [En línea]. Available: https://www.java.com/. [Último acceso: 15 junio 2022].

[12] Android, «Android,» 2022. [En línea]. Available: https://www.android.com/intl/es_es/. [Último acceso: 16 junio 2022].

[13] Academia Android, «Tratamiento ded xml en Android introducción,» 9 Diciembre 2015. [En línea]. Available: https://academiaandroid.com/tratamiento-de-xml-en-android-introduccion/. [Último acceso: 16 junio 2022].

[14] «Google Play,» Google, 2022. [En línea]. Available: https://developer.android.com/distribute/console?hl=es-419. [Último acceso: 16 Junio 2022].

[15] Red Hat, «What is a rest api,» 2022. [En línea]. Available: https://www.redhat.com/es/topics/api/what-is-a-rest-api. [Último acceso: 30 julio 2022].

[16] Github, «Github,» 2022. [En línea]. Available: https://github.com/. [Último acceso: 30 Agosto 2022].

[17] Google, «Firebase,» 2022. [En línea]. Available: https://firebase.google.com/. [Último acceso: 30 Agosto 2022].

[18] Google, «Firebase Authentication,» 2022. [En línea]. Available: https://firebase.google.com/products/auth. [Último acceso: 30 Agosto 2022].

[19] «OAuth 2.0,» 2022. [En línea]. Available: https://oauth.net/2/. [Último acceso: 30 julio 2022].

[20] OpenID, «OpenID Connect,» 2022. [En línea]. Available:

https://openid.net/connect/. [Último acceso: 30 julio 2022].

[21] Google, «Firebase Realtime Databse,» 2022. [En línea]. Available: https://firebase.google.com/products/realtime-database. [Último acceso: 30 Agosto 2022].

[22] «Draw.io,» 2022. [En línea]. Available: https://drawio-app.com/. [Último acceso: 20 julio 2022].

[23] M. Fowler, «GUI Architectures,» 2006. [En línea]. Available: https://martinfowler.com/eaaDev/uiArchs.html. [Último acceso: 31 Agosto 2022].

[24] «Material Design,» 2022. [En línea]. Available: https://material.io/design. [Último acceso: 30 Agosto 2022].

[25] Google, «Calidad básica de las apps,» 2022. [En línea]. Available: https://developer.android.com/docs/quality-guidelines/core-app-quality?hl=es-419. [Último acceso: 31 Agosto 2022].

[26] Naciones Unidas, «Objetivos de desarrollo sostenible,» 2022. [En línea]. Available: https://www.un.org/sustainabledevelopment/es/. [Último acceso: 1 Septiembre 2022].

[27] A. Summer, C. Hoy y E. Ortiz-Juarez, «Estimates of the impact of COVID-19 on global poverty,» 2020. [En línea]. Available: https://www.wider.unu.edu/publication/estimates-impact-covid-19-global-poverty. [Último acceso: 1 Septiembre 2022].

ANEXO

OBJETIVOS DE DESARROLLO SOSTENIBLE

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenibles	**Alto**	**Medio**	**Bajo**	**No Procede**
ODS 1. **Fin de la pobreza.**				X
ODS 2. **Hambre cero.**				X
ODS 3. **Salud y bienestar.**		X		
ODS 4. **Educación de calidad.**				X
ODS 5. **Igualdad de género.**				X
ODS 6. **Agua limpia y saneamiento.**				X
ODS 7. **Energía asequible y no contaminante.**				X
ODS 8. **Trabajo decente y crecimiento económico.**				X
ODS 9. **Industria, innovación e infraestructuras.**				X
ODS 10. **Reducción de las desigualdades**.				X
ODS 11. **Ciudades y comunidades sostenibles**.				X
ODS 12. **Producción y consumo responsables.**				X
ODS 13. **Acción por el clima.**				X
ODS 14. **Vida submarina.**				X
ODS 15. **Vida de ecosistemas terrestres.**				X
ODS 16. **Paz, justicia e instituciones sólidas.**				X
ODS 17. **Alianzas para lograr objetivos.**				X

Los Objetivos de Desarrollo Sostenible (ODS) [26] son 17 objetivos establecidos por la Asamblea General de las Naciones Unidas y diseñados para lograr un futuro mejor y más sostenible para todos. Estos objetivos abarcan desde la eliminación de la pobreza hasta el combate contra el cambio climático, la educación, la igualdad, la defensa del medio ambiente, la salud o el diseño de las ciudades.

Cada uno de estos objetivos tienen una serie de metas que se deben alcanzar para considerar a estos objetivos como cumplidos. Todas estas metas deben de alcanzarse para el año 2030, pero debido a la crisis social y económica causada por el COVID-19 se han perdido muchos de los esfuerzos realizados y dependiendo del ámbito, podría significar una reversión de aproximadamente una década en el progreso mundial. Tenemos por ejemplo el caso del aumento de la pobreza que puede llegar a afectar a 500 millones de personas debido a las consecuencias económicas de la pandemia [27], poniendo en desafío el objetivo de poner fin a la pobreza para el año 2030.

Entre todos los objetivos de desarrollos sostenible, se encuentra el objetivo de salud y bienestar, cuya meta es garantizar una vida sana y promover el bienestar para todas las edades. Esto es algo cada vez más importante debido a la gran cantidad de personas que llevan un estilo de vida no saludable, caracterizada por ser muy sedentaria. El llevar una vida sedentaria puede llegar a causar muchos problemas de salud entre los jóvenes y no tan jóvenes como la obesidad, pérdida de masa muscular, mala circulación sanguínea entre otros problemas de salud, por lo debería ser importante la promoción de actividades, así como la posibilidad de ofrecer herramientas para reducir este problema cada vez más generalizado de salud. Una de las posibles soluciones a este problema es llevar un estilo de vida saludable en el que el deporte forme parte del día a día, y esto es lo que ofrece nuestro proyecto, una plataforma que permita conectar el deporte de raqueta con los usuarios.

Nuestro proyecto se puede relacionar con el objetivo de desarrollo sostenible promovido por las Naciones Unidas de salud y bienestar ya que la app desarrollada ofrece a los usuarios una herramienta que permite reservar pistas en espacios deportivos desde cualquier lugar y momento.

El poder reservar permite a los usuarios elegir el horario que mejor les convenga así que si un día no existen pistas disponibles, el usuario podrá reservar una pista para

otro día y no tener que perder el tiempo llamado por teléfono y preguntando o tener que desplazarse hasta el lugar y preguntar y reservar en persona.

En cuanto a los otros ODS promovidos por la agenda 2030 no se pueden relacionar con este proyecto puesto que nuestra aplicación tiene un público concreto y por ende está dirigida a la población en general, únicamente nos centramos en aquellos interesados en los deportes de raqueta. El proyecto no tiene relación alguna con la mejora de los niveles de pobreza de la sociedad ni tampoco con la mejora medioambiental.