



# PISTA & PATO

Grupo 5

Mikel Álvarez Barroso  
Elvira Baltasar Jiménez  
Chang Hong Cong Dai  
Javier Rodríguez Prudencio  
Jiayun Zhan

Noviembre 2024

# Índice

<b>1. Descripción General de la App</b>	<b>4</b>
<b>2. Requisitos Funcionales</b>	<b>4</b>
2.1. Registro . . . . .	4
2.2. Inicio de sesión . . . . .	4
2.3. Consulta de pistas . . . . .	5
2.4. Reserva de pistas . . . . .	5
2.5. Acceso y gestión del perfil de usuario . . . . .	5
2.6. Interacción entre usuarios (Comunidad) . . . . .	5
2.7. Interfaz de usuario . . . . .	5
<b>3. Arquitectura de la Aplicación</b>	<b>5</b>
3.1. Usuario Final . . . . .	5
3.2. Interfaz de Usuario . . . . .	5
3.3. API de la Comunidad de Madrid . . . . .	6
3.4. API para Envío de Correos . . . . .	6
3.5. Base de Datos (Database Server) . . . . .	6
3.6. Flujo de Comunicación . . . . .	6
<b>4. Justificación del Diseño y Consideraciones Técnicas Fundamentales</b>	<b>7</b>
4.1. Orientación al Usuario . . . . .	7
4.2. Consistencia y Estilo . . . . .	8
4.3. Adaptabilidad . . . . .	8
4.4. Funcionalidad Prioritaria . . . . .	8
4.5. Arquitectura de Software . . . . .	9
4.6. Elección de Tecnologías . . . . .	9
4.7. Seguridad . . . . .	9
4.8. Integración con Servicios Externos . . . . .	9
4.9. Escalabilidad . . . . .	10
4.10. Mantenimiento y Actualizaciones . . . . .	10
4.11. Soporte . . . . .	10
<b>5. Metodología de Trabajo y Aportación Individual</b>	<b>10</b>
5.1. Metodología . . . . .	10
5.2. Aportación Individual . . . . .	11
<b>6. Capturas de Pantalla de la App</b>	<b>12</b>
6.1. Capturas con móvil vivo v29 . . . . .	12
6.1.1. Registro . . . . .	13
6.1.2. Inicio de sesión . . . . .	14
6.1.3. Búsqueda de Pista . . . . .	15
6.1.4. Resultado búsqueda de Pista . . . . .	17
6.1.5. Errores al reservar . . . . .	18
6.1.6. Reserva . . . . .	19

6.1.7.	Perfil del Usuario . . . . .	20
6.1.8.	Consulta y cancelación de reservas . . . . .	21
6.1.9.	Pistas Conjuntas . . . . .	22
6.1.10.	Resultado de Búsqueda Antes de Crear . . . . .	23
6.1.11.	Resultado de crear . . . . .	24
6.1.12.	Resultado de búsqueda después de crear . . . . .	25
6.1.13.	Unirse a una sesión . . . . .	26
6.2.	Capturas con móvil Pixel 7 Pro . . . . .	27
6.2.1.	Registro e inicio de sesión . . . . .	27
6.2.2.	Búsqueda de Pista . . . . .	28
6.2.3.	Resultado búsqueda de Pista . . . . .	29
6.2.4.	Errores al reservar . . . . .	30
6.2.5.	Reserva . . . . .	31
6.2.6.	Perfil del Usuario . . . . .	32
6.2.7.	Consulta y cancelación de reservas . . . . .	33
6.2.8.	Pistas Conjuntas . . . . .	34
6.2.9.	Resultado de Búsqueda Antes de Crear . . . . .	35
6.2.10.	Resultado de crear . . . . .	36
6.2.11.	Resultado de búsqueda después de crear . . . . .	37
6.2.12.	Unirse a una sesión . . . . .	38
<b>7.</b>	<b>Video Demostrativo de la App</b>	<b>38</b>
<b>8.</b>	<b>Descripción del Almacenamiento Persistente</b>	<b>38</b>
8.1.	Sistema de Almacenamiento: Firebase Firestore . . . . .	38
8.2.	Estructura de Datos . . . . .	39
8.3.	Razones de la Elección . . . . .	39
8.4.	Seguridad y Privacidad . . . . .	40
<b>9.</b>	<b>Descripción del Servicio(s) Remoto(s) Utilizado(s)</b>	<b>40</b>
9.1.	API comunidad de Madrid . . . . .	40
9.2.	JavaMail API . . . . .	41
9.2.1.	Acceso a Funcionalidades Avanzadas . . . . .	41
9.3.	Funcionamiento de la Integración . . . . .	41
<b>10.</b>	<b>Diagrama de navegación</b>	<b>41</b>
10.1.	Diagrama Navegación . . . . .	44
<b>11.</b>	<b>Test unitarios</b>	<b>44</b>
11.1.	Usando Mockito . . . . .	44
11.2.	Usando Firebase directamente . . . . .	45
11.3.	Conclusión . . . . .	45
<b>12.</b>	<b>Repositorio de GitHub</b>	<b>45</b>
<b>13.</b>	<b>Conclusión</b>	<b>45</b>

# 1. Descripción General de la App

Nuestra aplicación está diseñada para facilitar el acceso y gestión de las pistas de bádminton en Madrid de manera rápida y sencilla. A través de su interfaz intuitiva, se puede consultar y reservar las pistas públicas y privadas.

Aunque la consulta y reserva de pistas es la función principal, la aplicación ofrece un conjunto de características adicionales para mejorar la experiencia del usuario. Al crear el perfil, que solo requiere un nombre de usuario, correo electrónico y contraseña, se podrá disfrutar de las siguientes opciones una vez iniciada la sesión:

- **Acceso a tu perfil:** Personalizar y gestionar la cuenta fácilmente, con información actualizada y control total sobre sus preferencias.
- **Comunidad:** Posibilidad de unión a sesiones para conectar con otros jugadores, organizar partidos y compartir la pasión por el bádminton.

La aplicación está pensada para que se pueda disfrutar de una experiencia deportiva completa, conectando con otros usuarios y aprovechando al máximo cada sesión de juego.

## 2. Requisitos Funcionales

A continuación se detallan los requisitos funcionales de la aplicación, estos definen las funcionalidades que debe ofrecer el sistema para satisfacer las necesidades del usuario.

### 2.1. Registro

La aplicación debe permitir a los nuevos usuarios crear una cuenta proporcionando la siguiente información:

- Nombre de usuario
- Apellido
- Correo electrónico
- Contraseña

Este registro solo será permitido si el correo no están ya registrados en nuestra base de datos.

### 2.2. Inicio de sesión

Los usuarios, una vez registrados, podrán iniciar sesión con su nombre de usuario y contraseña. La aplicación debe validar las credenciales y permitir acceso a los usuarios autenticados.

## **2.3. Consulta de pistas**

Los usuarios podrán consultar las pistas de bádminton existentes en Madrid, al igual que consultar su información de horario y dirección. También podrán consultar la disponibilidad de pistas por día y hora.

## **2.4. Reserva de pistas**

Los usuarios podrán ser capaces de reservar una pista. La aplicación permitirá al usuario seleccionar instalación, día y hora. El sistema garantizará que solo aparezcan en verde los huecos horarios disponibles a reservar, evitando duplicados u horas pasadas.

## **2.5. Acceso y gestión del perfil de usuario**

El usuario podrá acceder a su perfil para modificar su información, el sistema garantizará un cambio de contraseña seguro.

## **2.6. Interacción entre usuarios (Comunidad)**

Los usuarios serán capaces de crear y participar en sesiones de organización de partidos seleccionando instalación, día y hora.

## **2.7. Interfaz de usuario**

La aplicación cuenta con una interfaz fácil de usar, intuitiva y accesible que permite realizar todas las funciones anteriores de manera clara y eficiente.

# **3. Arquitectura de la Aplicación**

La arquitectura de la aplicación está diseñada para garantizar una comunicación eficiente entre los diferentes elementos que la componen, maximizando su funcionalidad y adaptabilidad. A continuación, se describen los principales componentes y cómo interactúan entre sí, tal como se muestra en el diagrama adjunto:

## **3.1. Usuario Final**

El usuario interactúa con la aplicación a través de una interfaz gráfica intuitiva y fácil de usar, diseñada para simplificar el proceso de consulta y reserva de pistas deportivas.

## **3.2. Interfaz de Usuario**

Este componente actúa como un puente entre el usuario y las funciones de la aplicación. Permite que los usuarios realicen consultas sobre las pistas disponibles, gestionen sus reservas y envíen notificaciones a otros usuarios.

- A través de la interfaz, se realizan solicitudes hacia los sistemas de almacenamiento (base de datos) y hacia las APIs externas.

### 3.3. API de la Comunidad de Madrid

- Este servicio permite acceder a información pública sobre instalaciones deportivas, como ubicaciones y características de las pistas.
- **Restricción:** Aunque la API proporciona datos valiosos, el acceso a la información de horarios de disponibilidad de las pistas ha sido restringido, lo que limita algunas funcionalidades esperadas de la aplicación. Esta limitación ha sido compensada con datos almacenados localmente en la base de datos.

### 3.4. API para Envío de Correos

Se utiliza una API para el envío de correos electrónicos con el fin de mejorar la experiencia del usuario y mantener una comunicación efectiva. Esta API permite:

- Notificar a los usuarios sobre confirmaciones de reserva.

### 3.5. Base de Datos (Database Server)

La base de datos, gestionada con Firebase, se utiliza para almacenar y recuperar información clave, como los datos de los usuarios, sus reservas y otros detalles relevantes. Este componente asegura un almacenamiento en tiempo real y sincronización eficiente.

### 3.6. Flujo de Comunicación

- El usuario interactúa con la interfaz para realizar consultas y gestionar reservas.
- Las solicitudes de consulta se envían a la API de la Comunidad de Madrid o a la base de datos, dependiendo del tipo de información requerida.
- Los datos relacionados con los horarios de disponibilidad, debido a las restricciones de la API, se gestionan directamente desde la base de datos Firebase.
- La API para el envío de correos se utiliza para generar notificaciones y mantener la comunicación con los usuarios.

En la Figura 1, se muestra el diagrama de la arquitectura de la aplicación:

---

## ARQUITECTURA DE LA APLICACIÓN

---

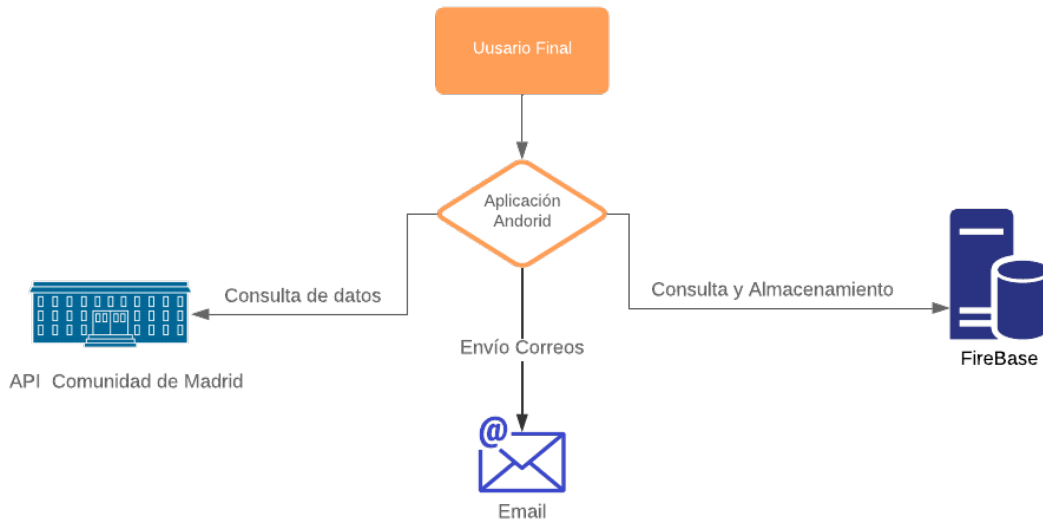


Figura 1: Arquitectura de la Aplicación

En resumen, la arquitectura de la aplicación combina una interfaz de usuario amigable, una base de datos robusta en Firebase y una integración con la API de la Comunidad de Madrid para proporcionar una experiencia fluida. Sin embargo, las restricciones en el acceso a horarios han requerido ajustes técnicos para mantener la funcionalidad esperada.

## 4. Justificación del Diseño y Consideraciones Técnicas Fundamentales

### 4.1. Orientación al Usuario

El diseño de nuestra aplicación se centra en satisfacer las necesidades y expectativas del usuario final, abordando los siguientes aspectos clave:

- **Usabilidad:** La aplicación ha sido diseñada para ser intuitiva y fácil de usar, permitiendo que los usuarios realicen reservas de pistas de bádminton sin necesidad de experiencia técnica previa. La interfaz facilita la navegación y minimiza la complejidad en la interacción.
- **Experiencia de Usuario (UX):** Hemos implementado patrones de diseño conocidos, como menús desplegables y tarjetas informativas, para garantizar una experiencia coherente y fluida. Las transiciones y animaciones han sido optimizadas para ofrecer una interacción agradable.

- **Accesibilidad:** El diseño incluye características como textos alternativos para imágenes, contraste adecuado entre texto y fondo, y soporte para fuentes escalables, asegurando la inclusión de usuarios con diferentes capacidades.

## 4.2. Consistencia y Estilo

Para reforzar la identidad visual y ofrecer una experiencia homogénea, hemos considerado:

- **Paleta de colores y tipografía:** Se seleccionaron colores suaves y neutros que transmiten tranquilidad y confianza, combinados con una tipografía legible y moderna que facilita la lectura. Estos colores pertenecen a la paleta de colores del logo, creando así un sentimiento de formidabilidad en la aplicación.
- **Componentes de diseño consistentes:** Se utilizaron elementos de interfaz uniformes en todas las pantallas, como botones, iconos y campos de entrada, lo que ayuda a los usuarios a familiarizarse rápidamente con la aplicación.
- **Diseño Visual:** Se adoptó un diseño minimalista con enfoque en la usabilidad, utilizando colores suaves y contrastes efectivos que mejoran la legibilidad y reducen la carga cognitiva del usuario.

## 4.3. Adaptabilidad

El diseño ha sido optimizado para adaptarse a diferentes dispositivos y tamaños de pantalla:

- **Enfoque responsivo:** La aplicación se adapta automáticamente a distintos tamaños de pantalla, garantizando una experiencia óptima en smartphones y tablets.
- **Compatibilidad:** Se priorizó la compatibilidad con versiones actuales y anteriores de Android, ampliando el alcance a un mayor número de usuarios.

## 4.4. Funcionalidad Prioritaria

Nos enfocamos en las funcionalidades clave identificadas durante el análisis de requisitos:

- **Reserva de pistas:** Proceso simplificado para buscar, identificar disponibilidad y reservar pistas disponibles en tiempo real.
- **Información detallada:** Acceso a detalles de las instalaciones, incluyendo ubicación, horarios y servicios adicionales.



## 4.5. Arquitectura de Software

La arquitectura de la aplicación ha sido diseñada para ser escalable y mantenible:

- **Modularidad:** El código se organizó en módulos independientes para mejorar la legibilidad y permitir futuras expansiones.
- **Backend robusto:** Aunque la lógica principal reside en el cliente, se planificó una estructura que permite la integración con servicios backend si es necesario.

## 4.6. Elección de Tecnologías

Las tecnologías seleccionadas se alinean con los objetivos del proyecto:

- **Android Studio con Java:** Se optó por Android Studio como entorno de desarrollo integrado (IDE) utilizando Java, aprovechando sus características y soporte nativo para Android.
- **Firebase:** Se utilizó Firebase como base de datos en tiempo real y para autenticación, lo que simplifica la gestión de datos y usuarios sin necesidad de mantener un servidor propio.

## 4.7. Seguridad

La seguridad de los datos y la protección de la información del usuario son primordiales:

- **Autenticación segura:** Implementación de autenticación mediante Firebase Authentication, que proporciona métodos seguros como email/contraseña y autenticación por terceros.
- **Reglas de seguridad:** Configuración de reglas de seguridad en Firebase Firestore para controlar el acceso y las operaciones permitidas en la base de datos.

## 4.8. Integración con Servicios Externos

La aplicación se beneficia de la integración con servicios externos para ampliar sus funcionalidades:

- **Firebase**
- **API de la Comunidad de Madrid:** Integración con la API pública para obtener información actualizada sobre las instalaciones deportivas, en este caso la comunidad de Madrid no da acceso a estos datos.
- **JavaMail API:** La aplicación incorpora la funcionalidad de envío de correos electrónicos utilizando la **JavaMail API**. Este servicio se integra con la infraestructura de la aplicación para gestionar la confirmación de reservas, notificando automáticamente a los usuarios una vez que su reserva ha sido procesada exitosamente.

## 4.9. Escalabilidad

El diseño y la elección de tecnologías permiten que la aplicación pueda crecer y adaptarse a demandas mayores:

- **Servicios en la nube:** Uso de Firebase, que es altamente escalable y puede manejar incrementos en el tráfico y en la cantidad de datos sin afectar el rendimiento.
- **Arquitectura preparada para expansión:** La modularidad del código facilita la adición de nuevas funcionalidades y la integración con otros servicios en el futuro.

## 4.10. Mantenimiento y Actualizaciones

Para facilitar el mantenimiento y las futuras actualizaciones, se adoptaron las siguientes prácticas:

- **Código bien documentado:** Se incluyeron comentarios y documentación interna para explicar funcionalidades y flujos de trabajo.
- **Control de versiones:** Uso de Git y repositorio en GitHub para gestionar cambios en el código y colaborar eficientemente.

## 4.11. Soporte

- **Tecnología:** Elegimos Android Studio y Firebase para aprovechar herramientas potentes y servicios en la nube que aceleran el desarrollo y simplifican la gestión de datos.

# 5. Metodología de Trabajo y Aportación Individual

## 5.1. Metodología

En el marco de desarrollo de la aplicación de reserva de pistas de bádminton, Trello se utilizó como una herramienta complementaria para la planificación visual y seguimiento de tareas. Durante la fase inicial, el equipo utilizó Trello para crear un tablero inicial con tareas relacionadas con la conceptualización de la aplicación y primeras decisiones de diseño y funcionalidad. A lo largo de los siguientes meses el equipo mantuvo tableros actualizados para supervisar las tareas diarias, brindando a todos los miembros del equipo una visión clara de lo que estaba ocurriendo en tiempo real.

Una de las características a la que dimos más uso fueron las etiquetas y colones, permitiendo a los miembros del equipo identificar rápidamente la categoría y urgencia de cada tarea. Se asignaron colores específicos a las tareas según su prioridad:

- Rojo: Alta prioridad
- Amarillo: Media prioridad
- Verde: Baja prioridad

También se etiquetaron las tareas según las áreas de trabajo (backend, diseño UI/UX, Pruebas...), lo que permitió una fácil identificación de qué miembro del equipo estaba trabajando en qué tipo de tarea.

Esta organización visual fue clave para que el equipo estuviera enfocado en las actividades más críticas.

## 5.2. Aportación Individual

### ■ Mikel Álvarez Barroso:

- Conexión con la Base de datos
- Conexión/carga API en la base de datos
- Pantalla de cancelar reserva
- Funcionalidad de pantalla de cancelar reserva
- Clase Usuario
- Clase Pista
- Memoria

### ■ Elvira Baltasar Jiménez:

- Decisiones de diseño y planificación
- Creación de esqueleto para la aplicación y su funcionalidad
- Conexión con la API de la comunidad de Madrid y con la de emails
- Pantalla de búsqueda
- Funcionalidad de pantalla de búsqueda
- Pantalla de resultado para reservar
- Funcionalidad de pantalla de resultado (reserva)
- Conexión con la base de datos
- Clase instalación y pista
- Memoria
- Capturas de la aplicación

### ■ Javier Rodríguez Prudencio:

- Pantalla de perfil
- Pantalla de password antes de modificar perfil
- Pantalla de modificar perfil
- Implementación de perfil
- Implementación de password antes de modificar perfil
- Implementación de modificar perfil
- Vídeo de todas las funcionalidades de la app

- Jiayun Zhan:
  - Conexión con la Base de datos
  - Conexión con la API de envío de emails
  - Pantalla de solitario/Chat
  - Funcionalidad de pantalla de listar/solitario/Chat
  - Pantalla perfil resultado de reservas
  - Clase reserva
  - Diagrama de navegación
  - Test unitario
- Chang Hong Cong Dai:
  - Pantalla de iniciar sesión
  - Pantalla de registro
  - Funcionalidad de iniciar sesión
  - Funcionalidad de registro
  - Test unitario
  - Conexión con la base de datos

## 6. Capturas de Pantalla de la App

En esta sección se presentan las capturas de pantalla que reflejan el flujo de uso y las funcionalidades principales de la aplicación.

### 6.1. Capturas con móvil vivo v29

En nuestra aplicación, hemos decidido no implementar la orientación vertical porque el diseño y la experiencia de usuario están optimizados exclusivamente para el modo horizontal. Esto garantiza que todos los elementos de la interfaz, como botones y vistas, se mantengan bien distribuidos y funcionales, evitando desajustes o escalas inadecuadas que afectarían la usabilidad. Además, bloquear la orientación en horizontal nos permite ofrecer una experiencia más consistente y simplificar el desarrollo.

El cambio de idioma a inglés y el modo oscuro si están implementados pero, este último, sin muchas diferencias ya que nuestra paleta de color cuadra con ambos modos.

### 6.1.1. Registro



12:24 100% 27.13 KB/s 5G



**Usuario**

Usuario

**Apellido**

Apellido

**Email**

Email

**Contraseña**

Contraseña

**Confirmar Contraseña**

Confirmar Contraseña

**Registrarse**

Figura 2: Pantalla de registro.



17:26 100% 0.16 KB/s



**Usuario**

Usuario

**Apellido**

Apellido

**Email**

Email

**Contraseña**

Contraseña

**Confirmar Contraseña**

Confirmar Contraseña

**Registrarse**

Figura 3: Pantalla de registro en modo oscuro.

### 6.1.2. Inicio de sesión

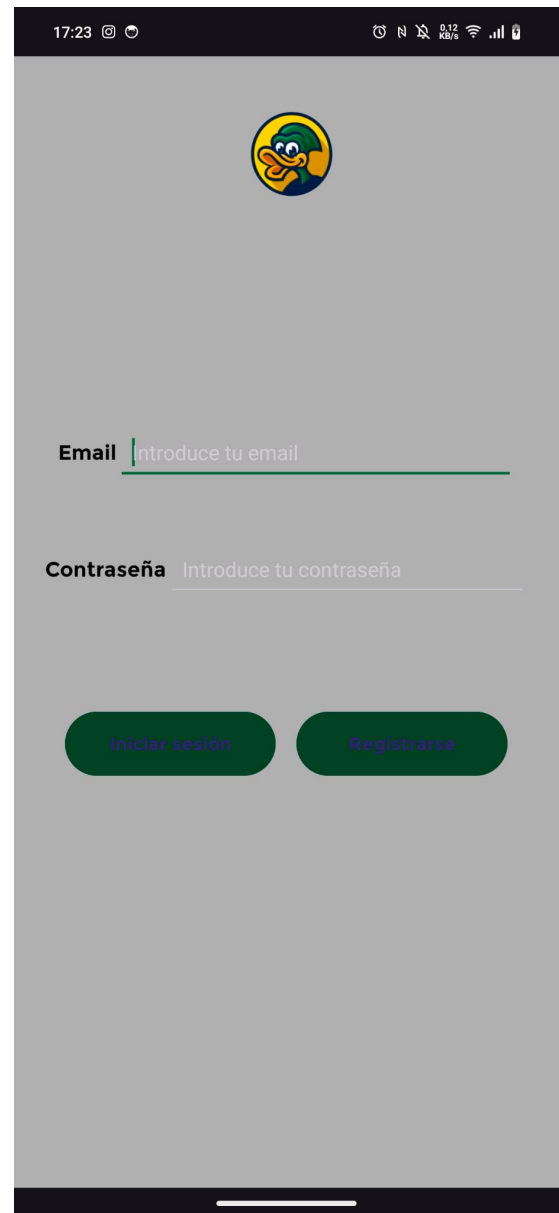
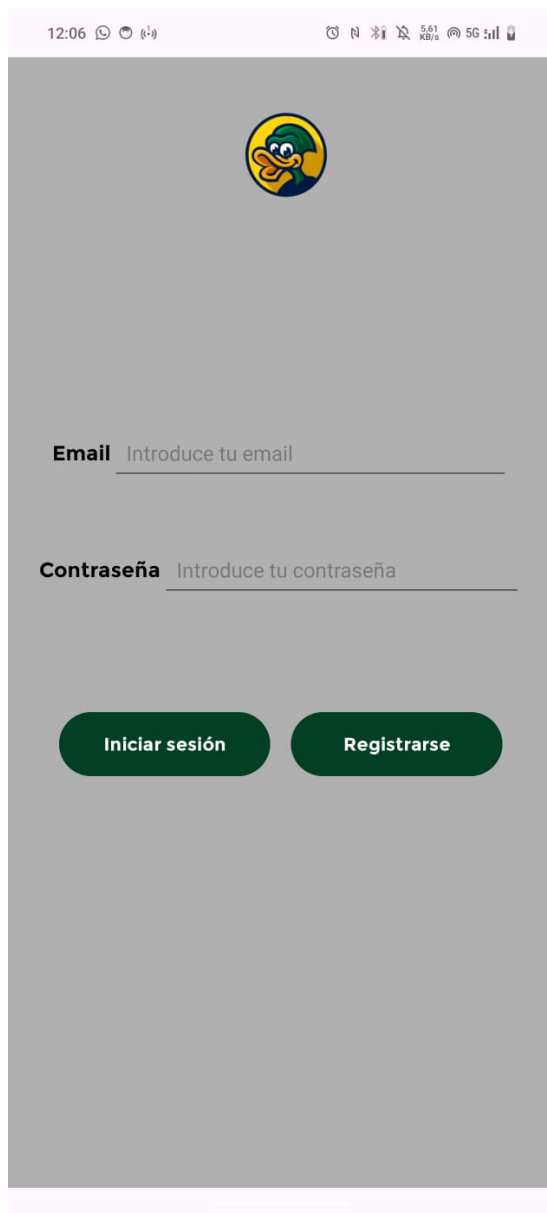


Figura 4: Pantalla de inicio de sesiónFigura 5: Pantalla de inicio de sesión en modo oscuro.

### 6.1.3. Búsqueda de Pista

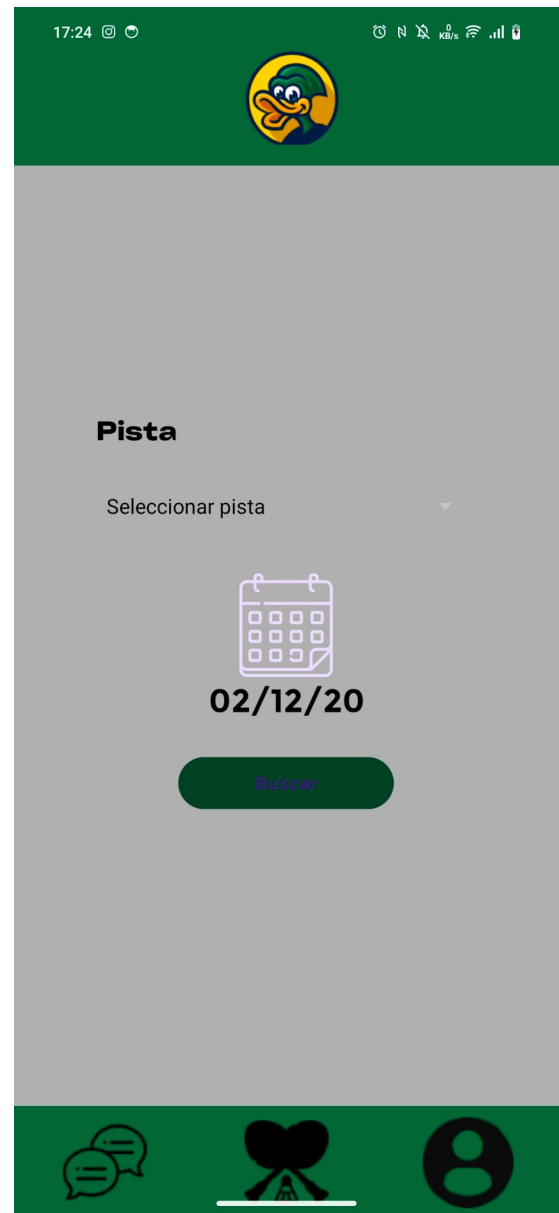
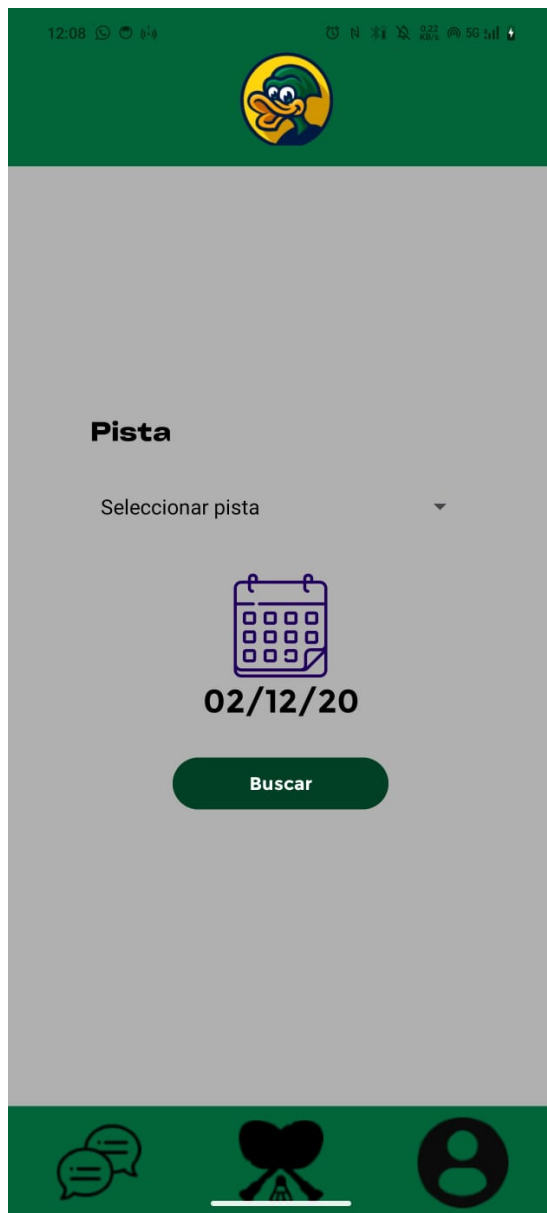


Figura 6: Pantalla de búsqueda de pista. Figura 7: Pantalla de búsqueda de pista en modo oscuro.



Figura 8: Desplegable de pistas disponibles.



#### 6.1.4. Resultado búsqueda de Pista

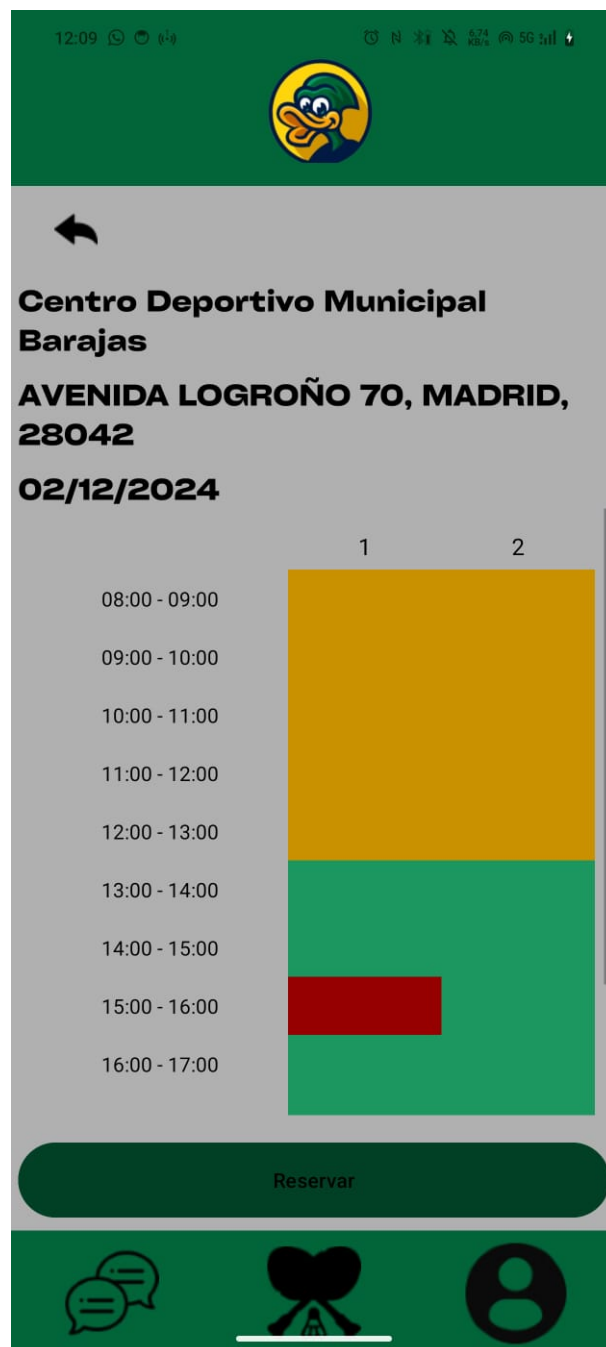


Figura 9: Pantalla de selección de pista libre

### 6.1.5. Errores al reservar

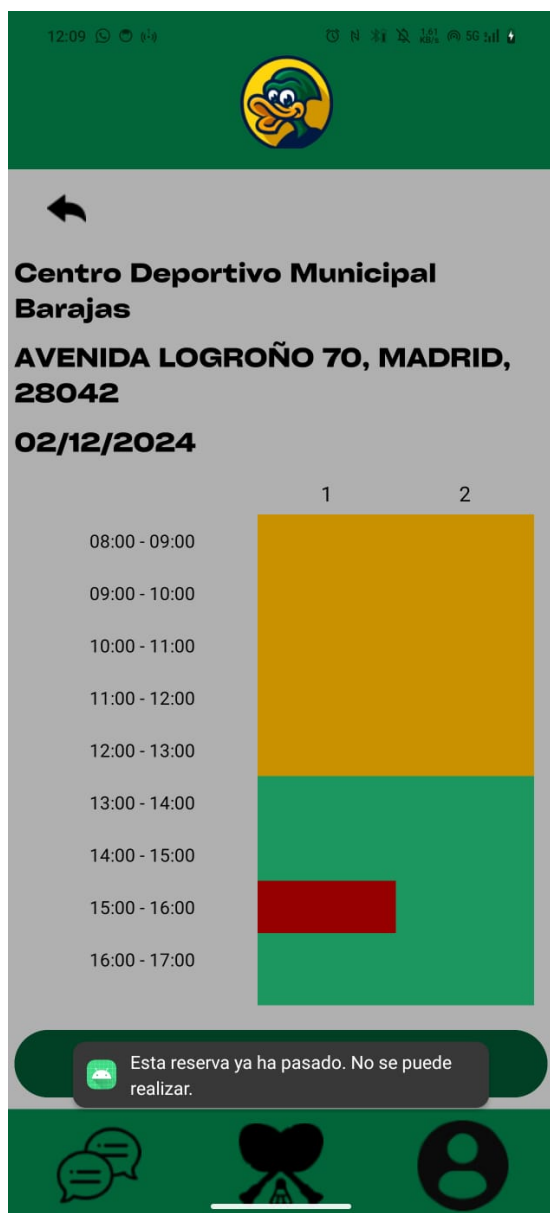


Figura 10: Hora pasada.

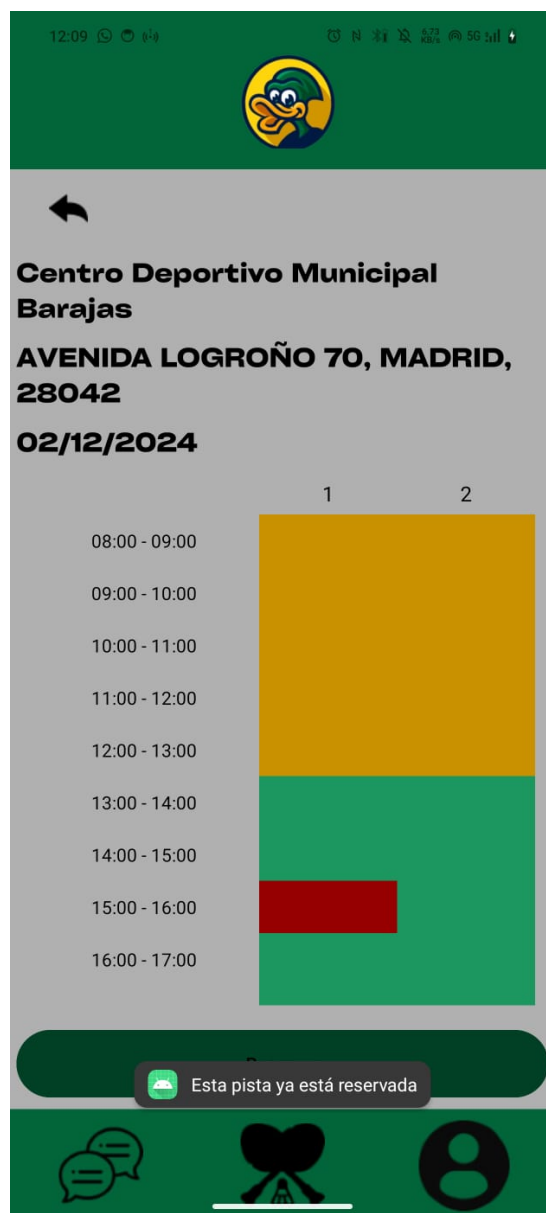


Figura 11: Pista ya reservada.

### 6.1.6. Reserva

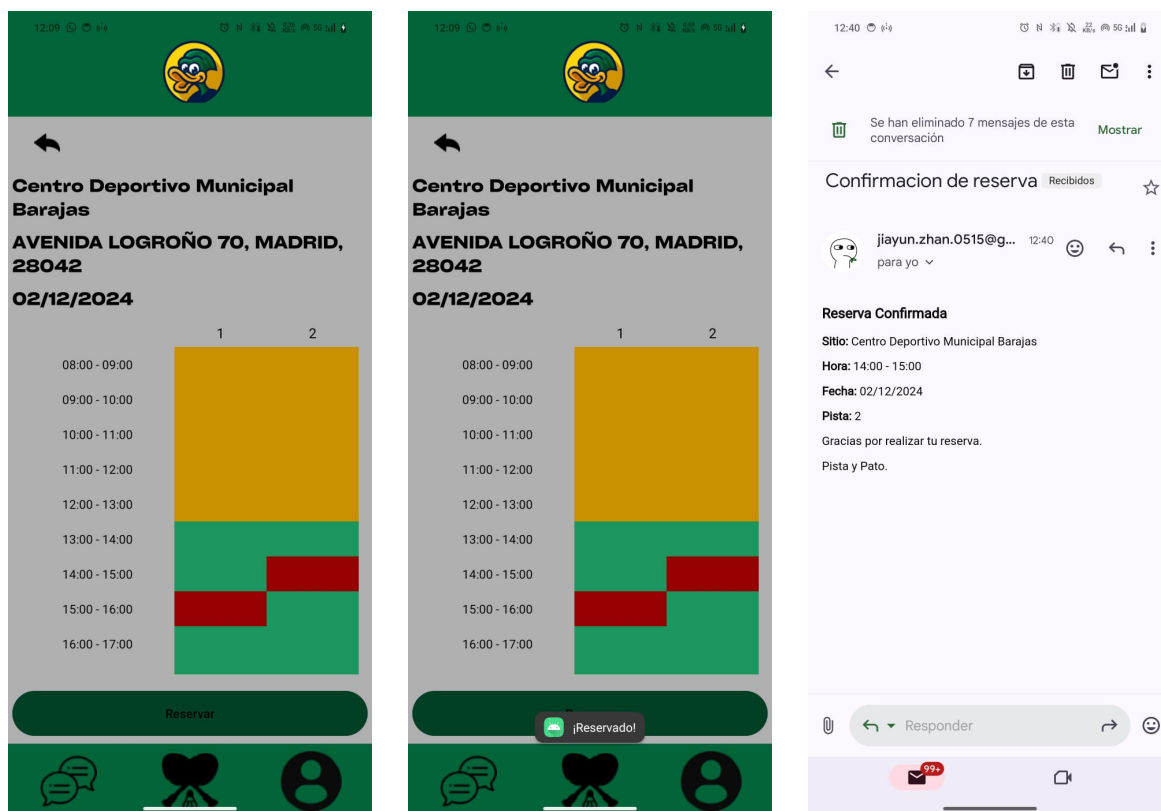


Figura 12: Pista libre seleccionada, reservada e email de confirmación.

### 6.1.7. Perfil del Usuario

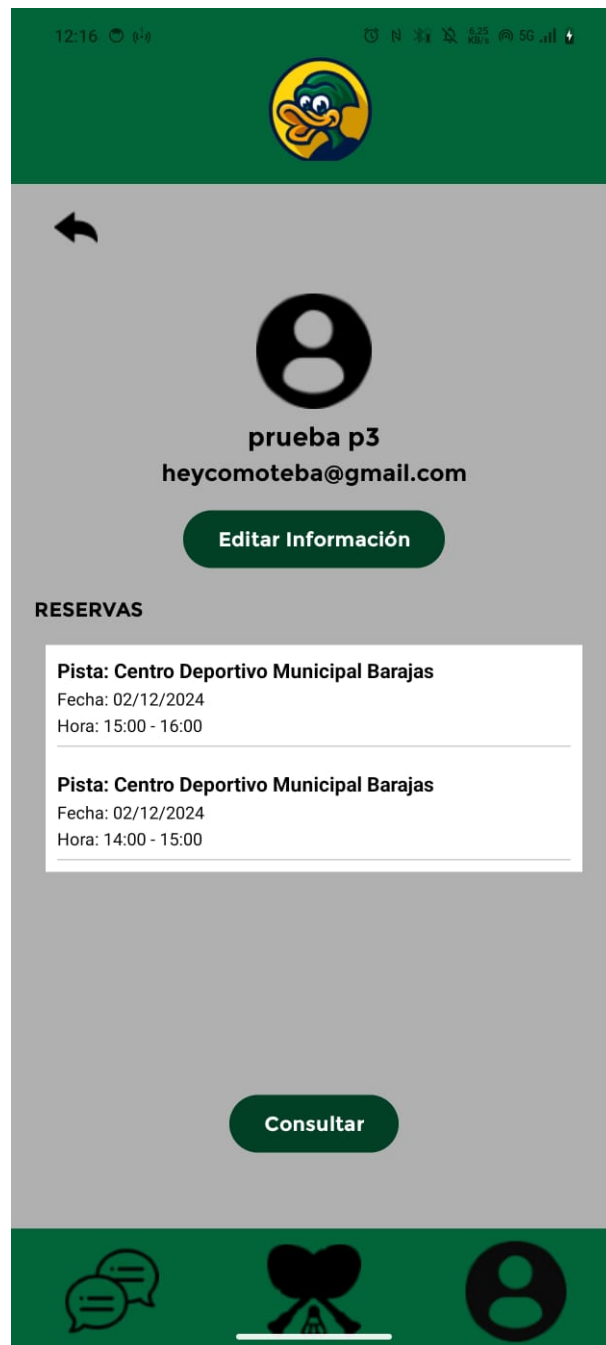


Figura 13: Pantalla del perfil del usuario

### 6.1.8. Consulta y cancelación de reservas

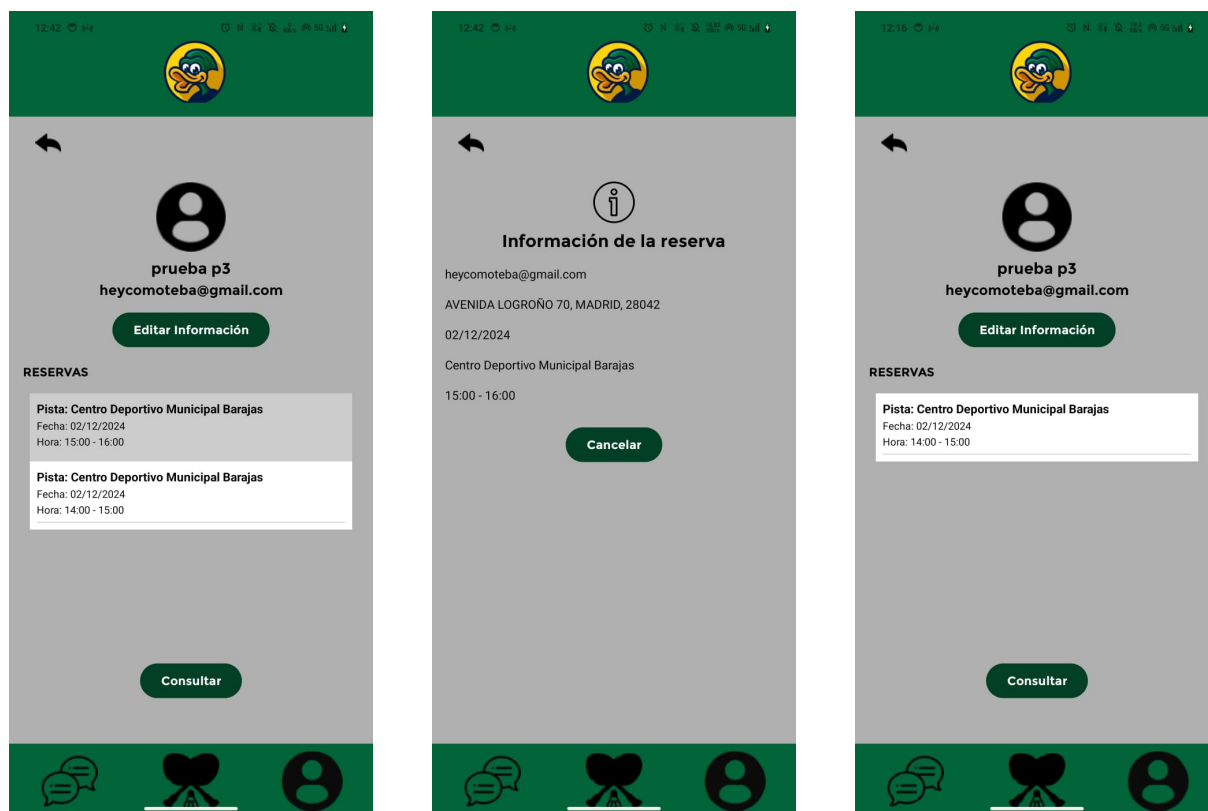


Figura 14: Pista seleccionada, su información y resultado tras cancelar.

### 6.1.9. Pistas Conjuntas

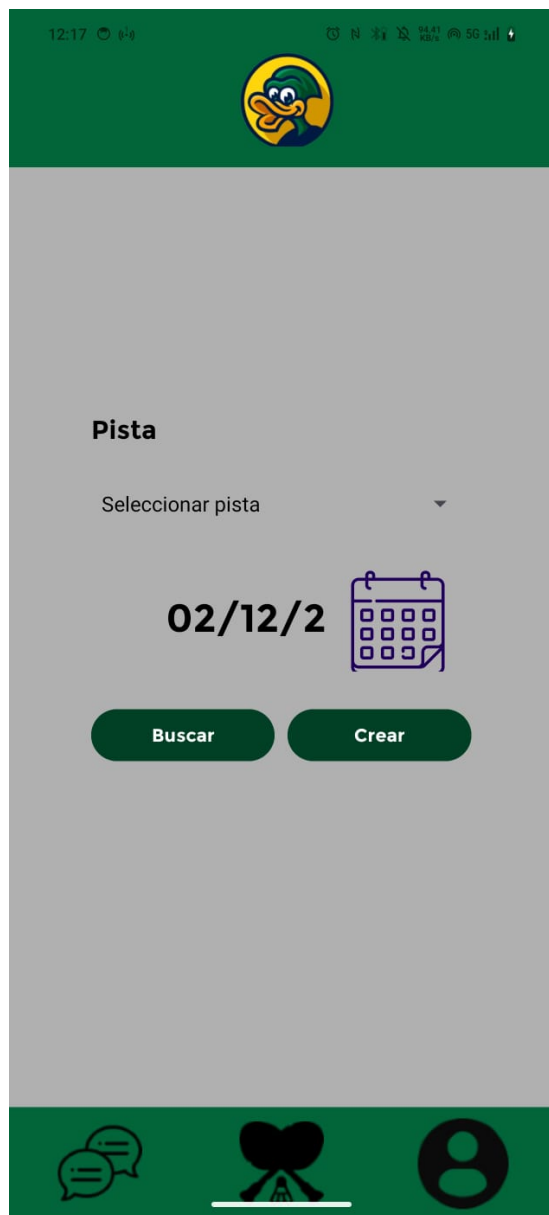


Figura 15: Pantalla de búsqueda de pistas conjuntas.

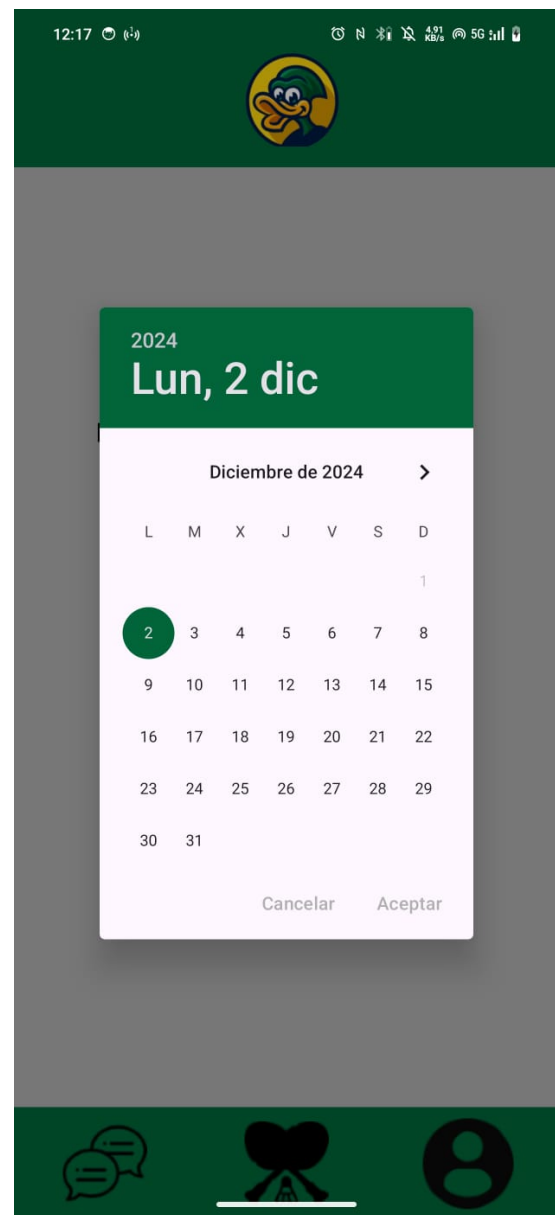


Figura 16: Desplegable de calendario.

#### 6.1.10. Resultado de Búsqueda Antes de Crear

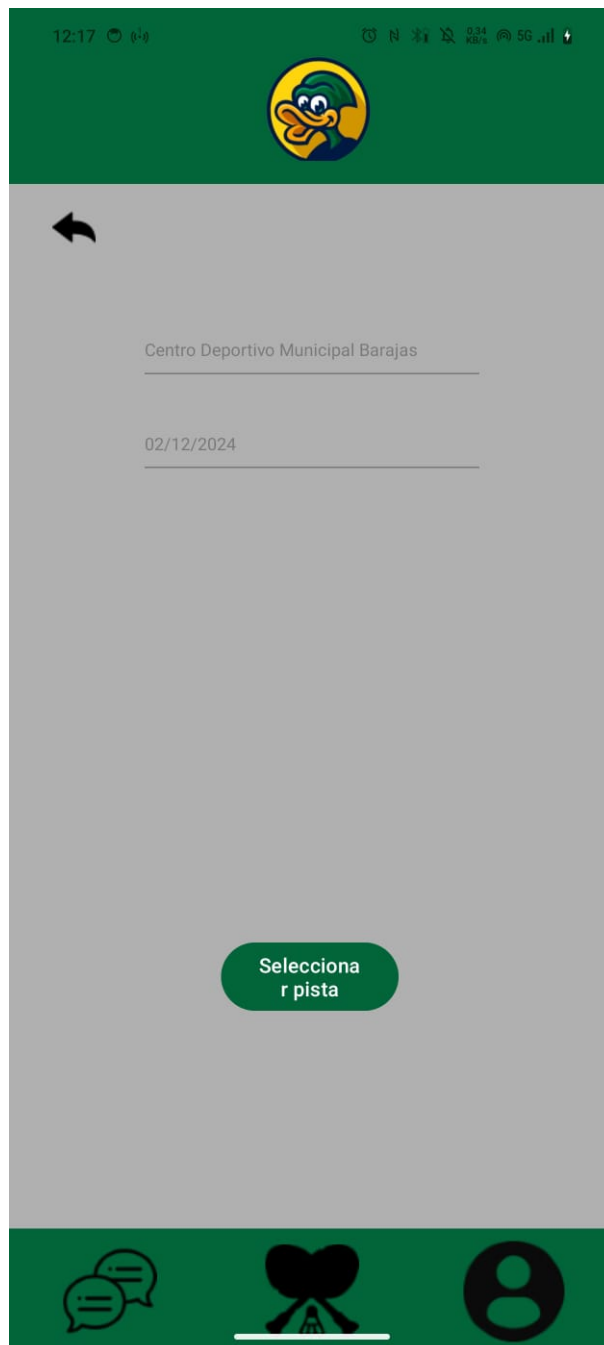


Figura 17: Pantalla con el resultado de búsqueda antes de crear

#### 6.1.11. Resultado de crear

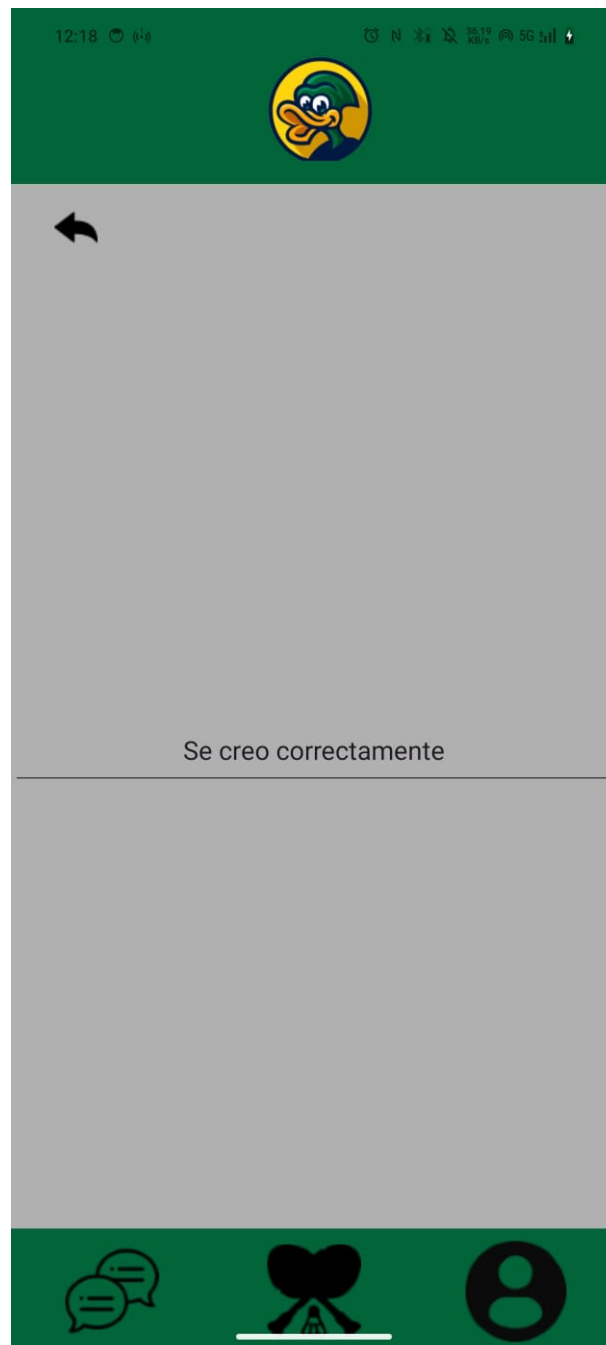


Figura 18: Pantalla de creación de pista



### 6.1.12. Resultado de búsqueda después de crear



Figura 19: Pantalla con el resultado de búsqueda después de crear

### 6.1.13. Unirse a una sesión

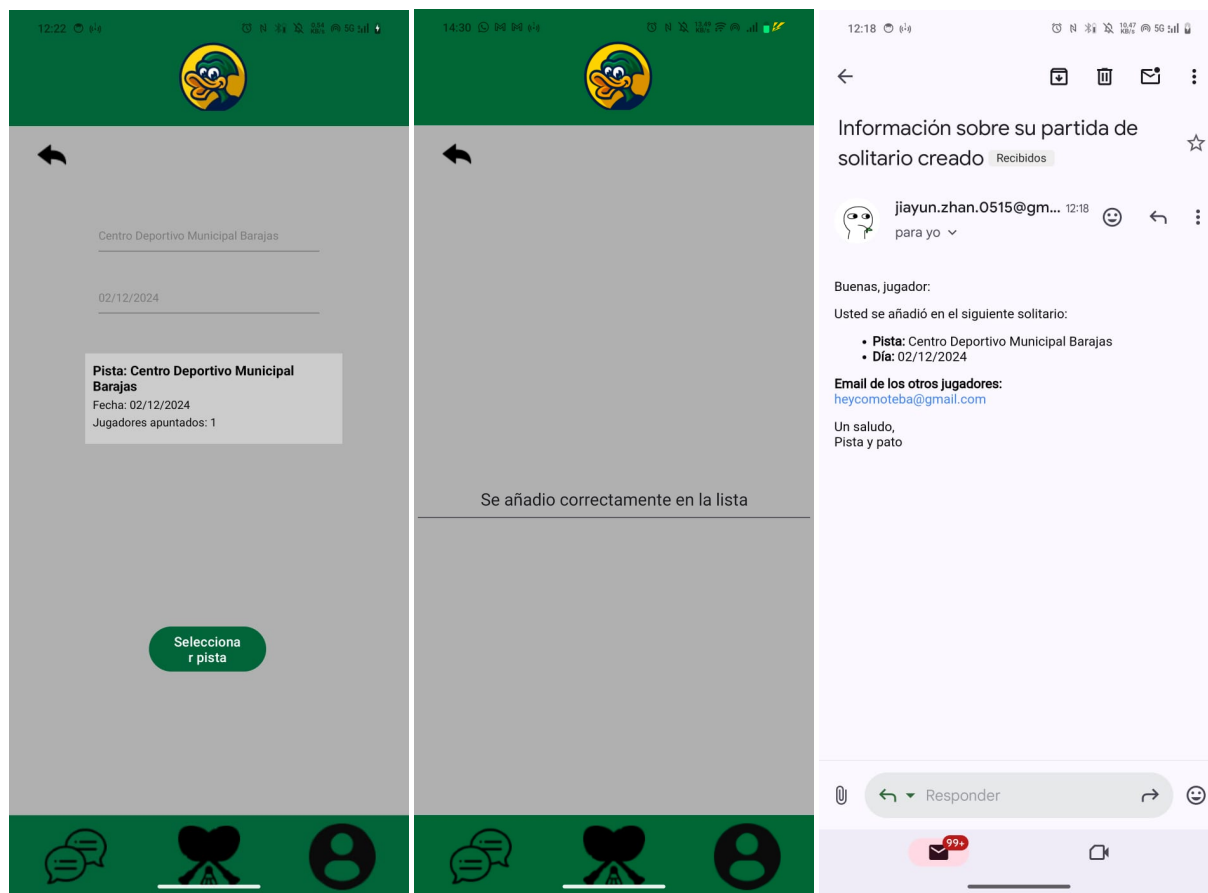
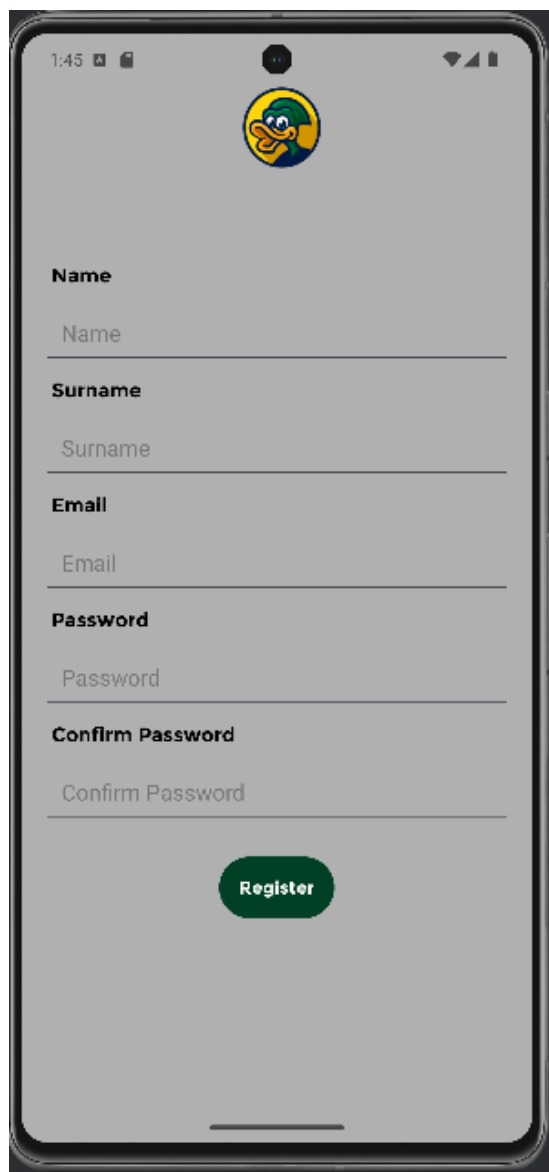



Figura 20: Selección de pista, unión e email de confirmación.

## 6.2. Capturas con móvil Pixel 7 Pro

### 6.2.1. Registro e inicio de sesión



1:45



**Name**

Name

**Surname**

Surname

**Email**

Email

**Password**

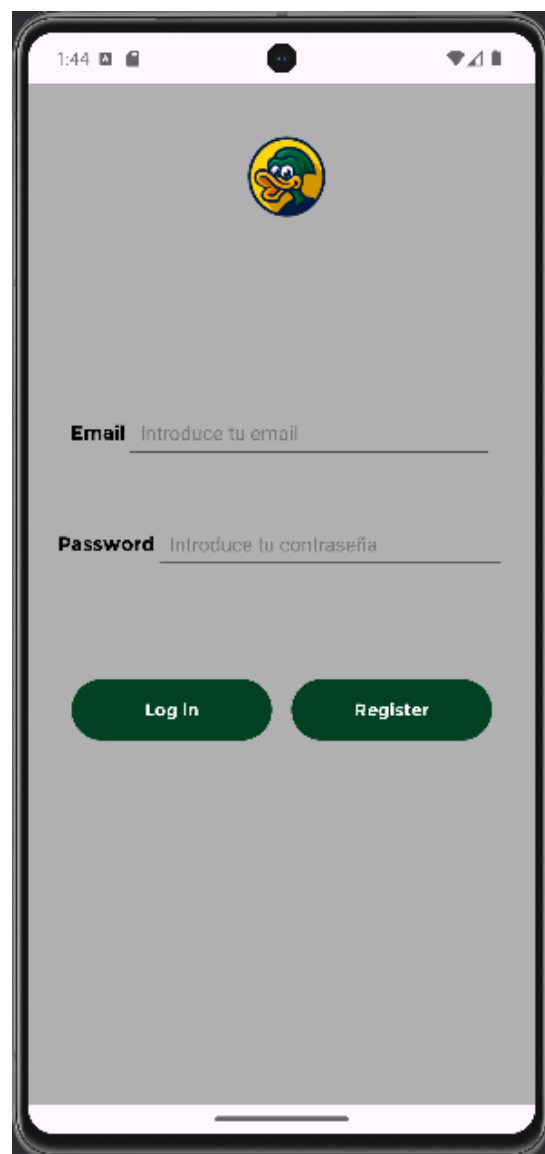
Password

**Confirm Password**


Confirm Password

**Register**

Figura 21: Pantalla de registro.



1:44



**Email** Introduce tu email

**Password** Introduce tu contraseña

**Log In** **Register**

Figura 22: Pantalla de inicio de sesión.

### 6.2.2. Búsqueda de Pista

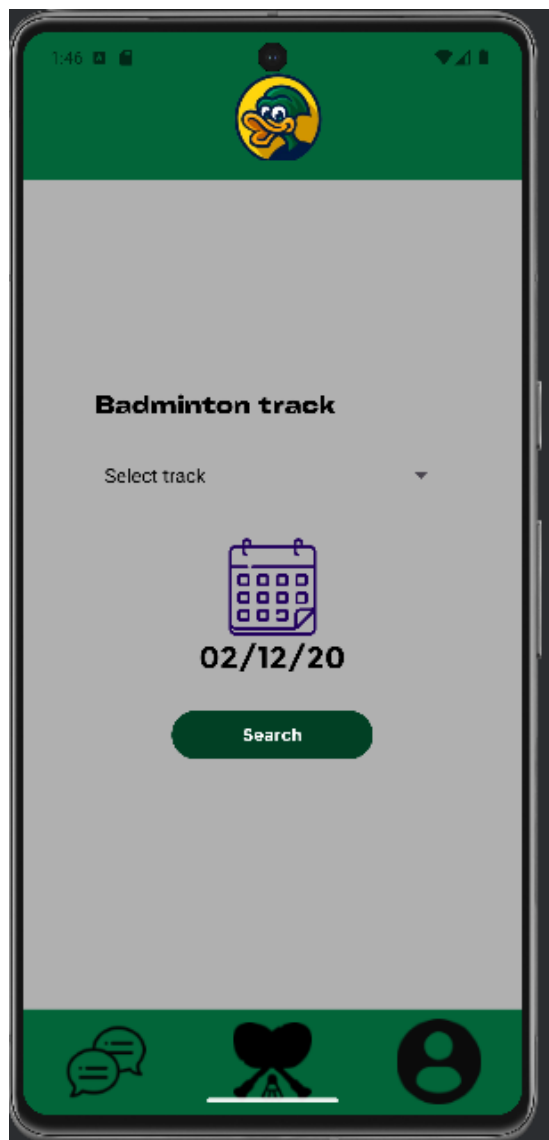


Figura 23: Pantalla de búsqueda de pista.

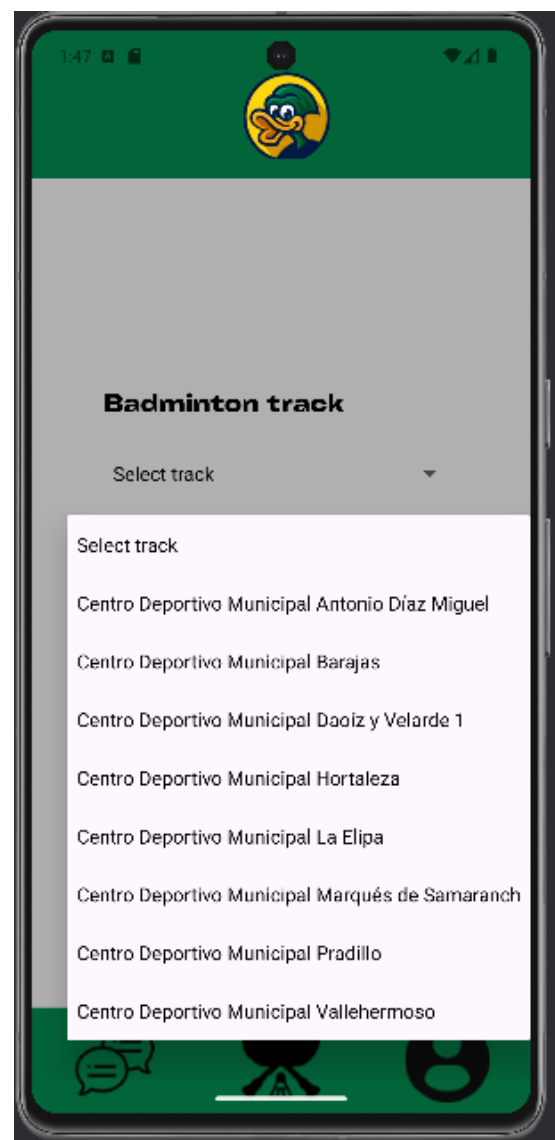


Figura 24: Desplegable de pistas disponibles.

### 6.2.3. Resultado búsqueda de Pista

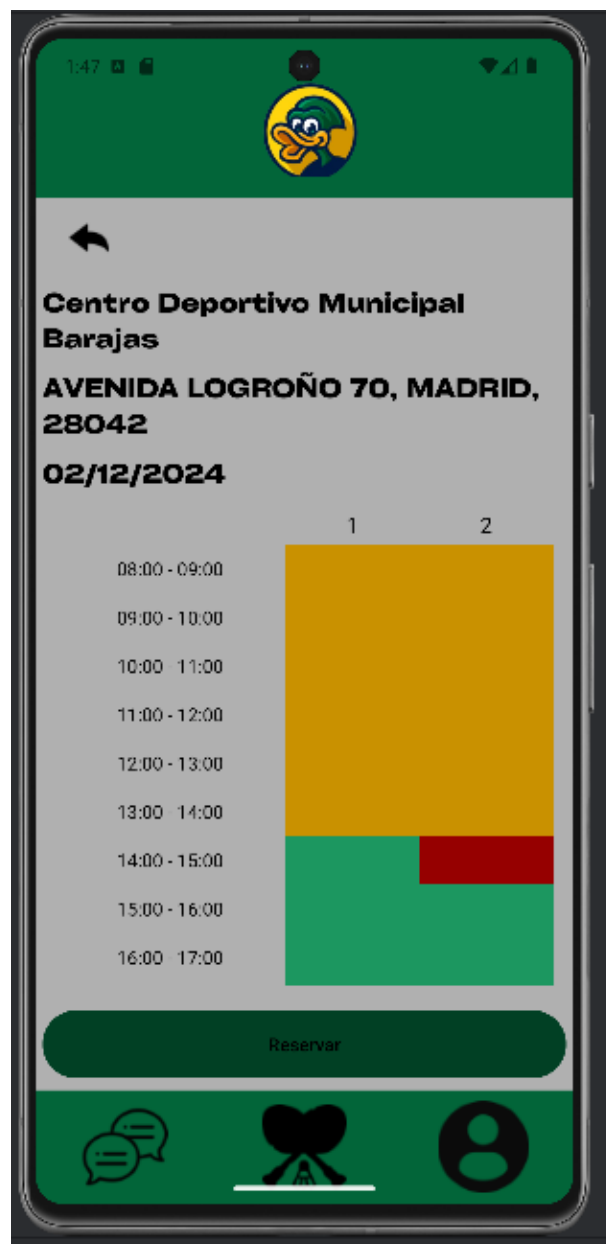


Figura 25: Pantalla de selección de pista libre

#### 6.2.4. Errores al reservar

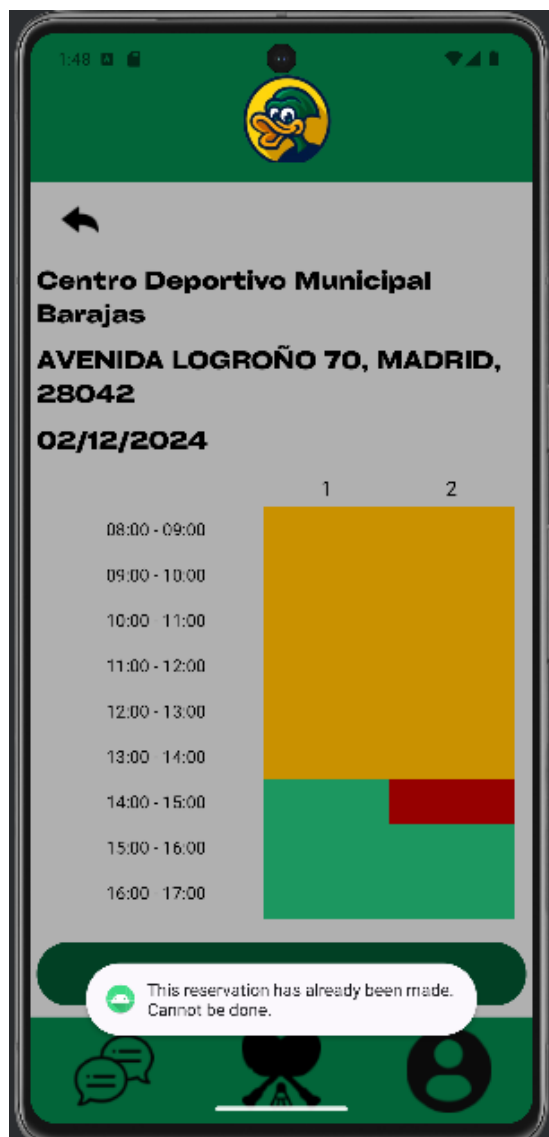


Figura 26: Hora pasada.

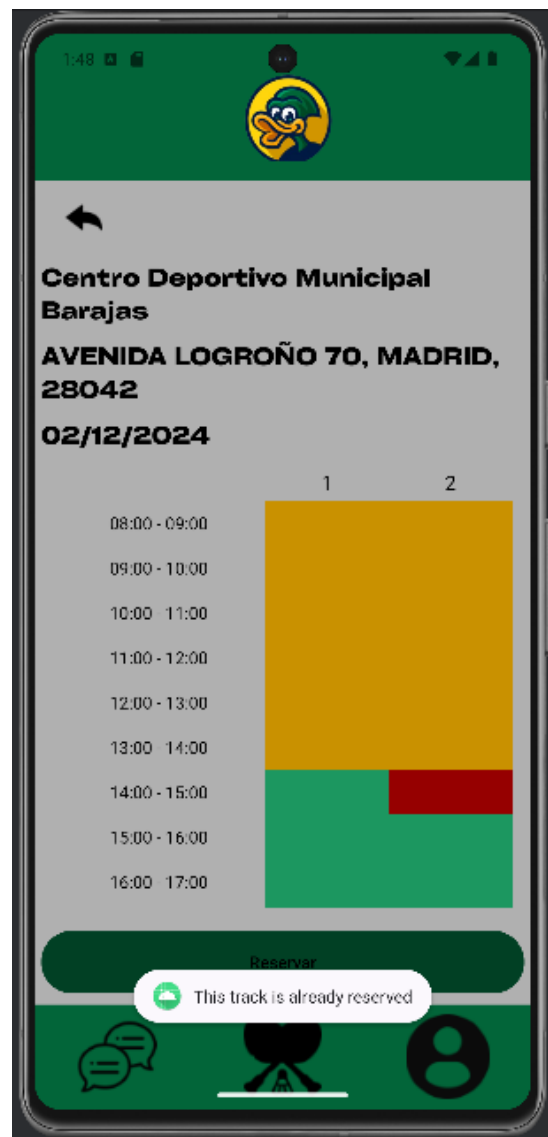


Figura 27: Pista ya reservada.

### 6.2.5. Reserva

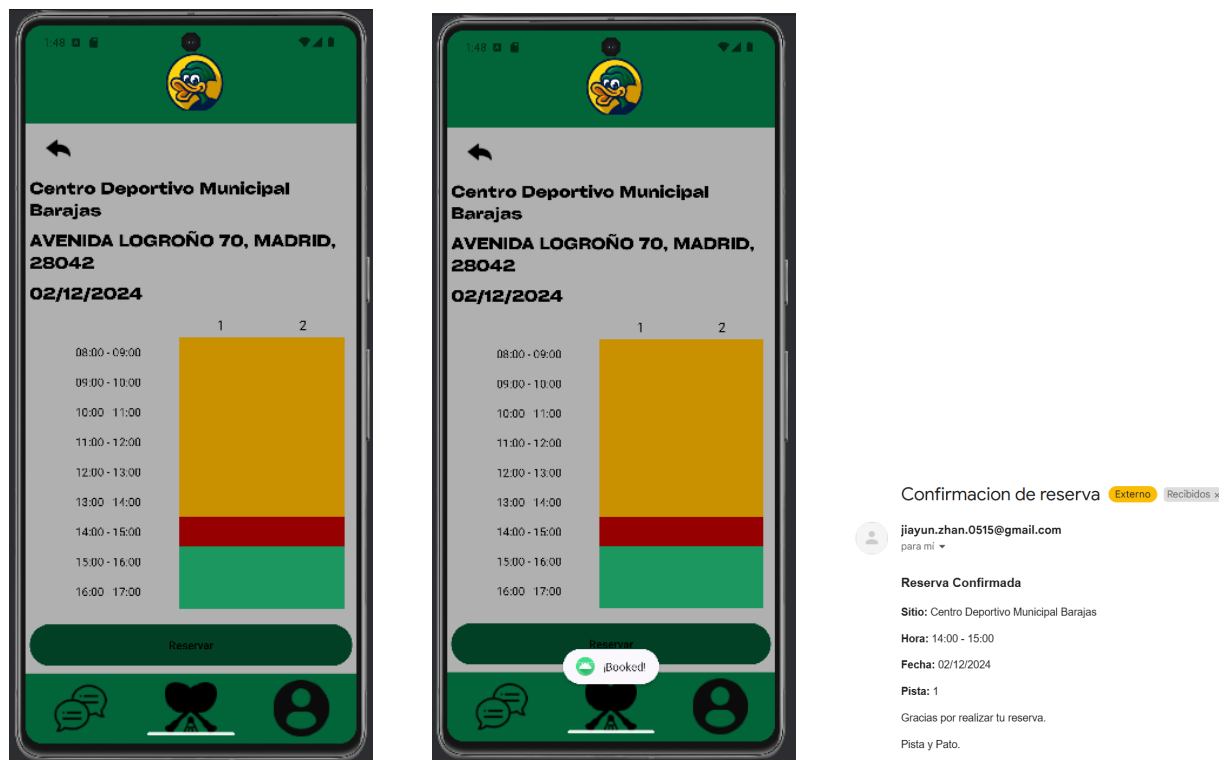


Figura 28: Pista libre seleccionada, reservada e email de confirmación.

### 6.2.6. Perfil del Usuario

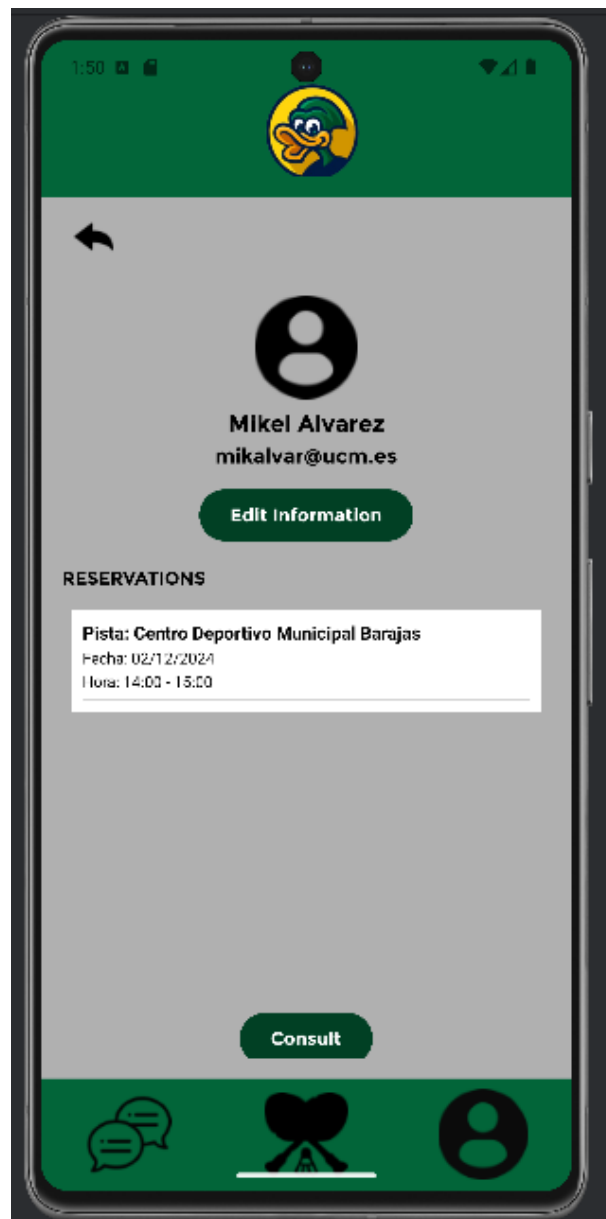


Figura 29: Pantalla del perfil del usuario



### 6.2.7. Consulta y cancelación de reservas

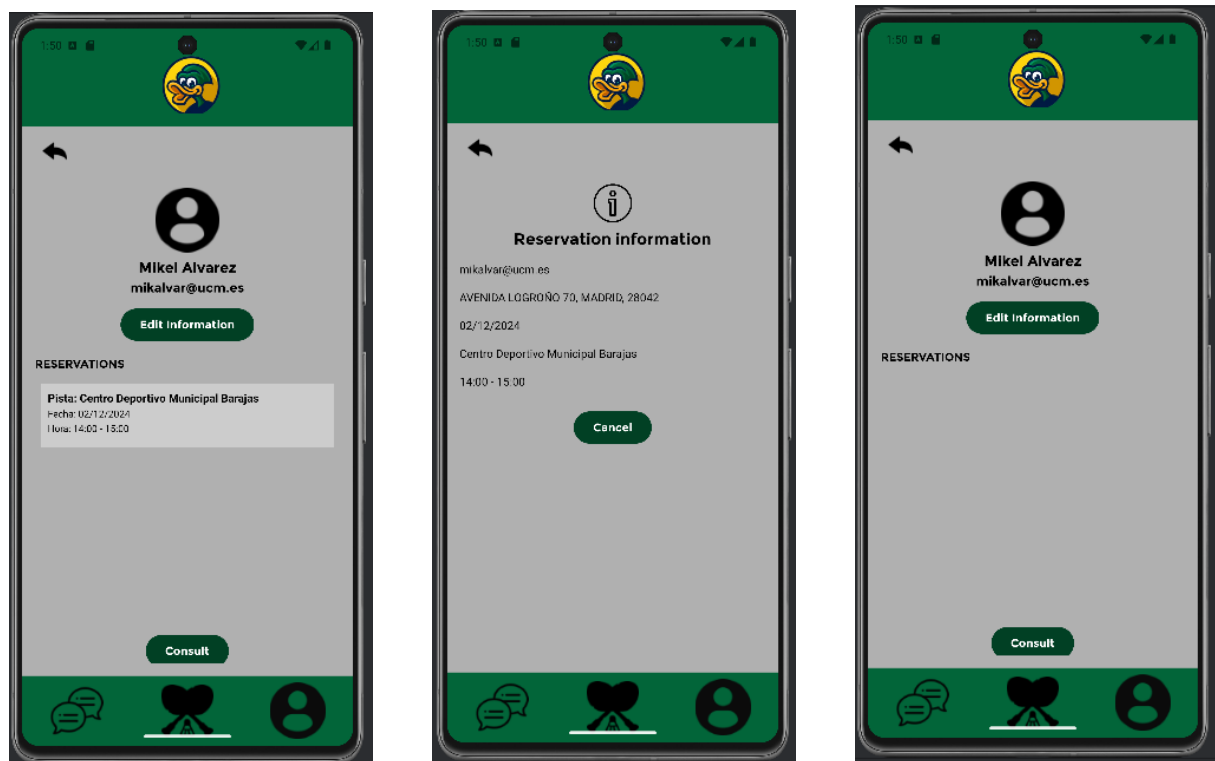


Figura 30: Pista seleccionada, su información y resultado tras cancelar.

### 6.2.8. Pistas Conjuntas

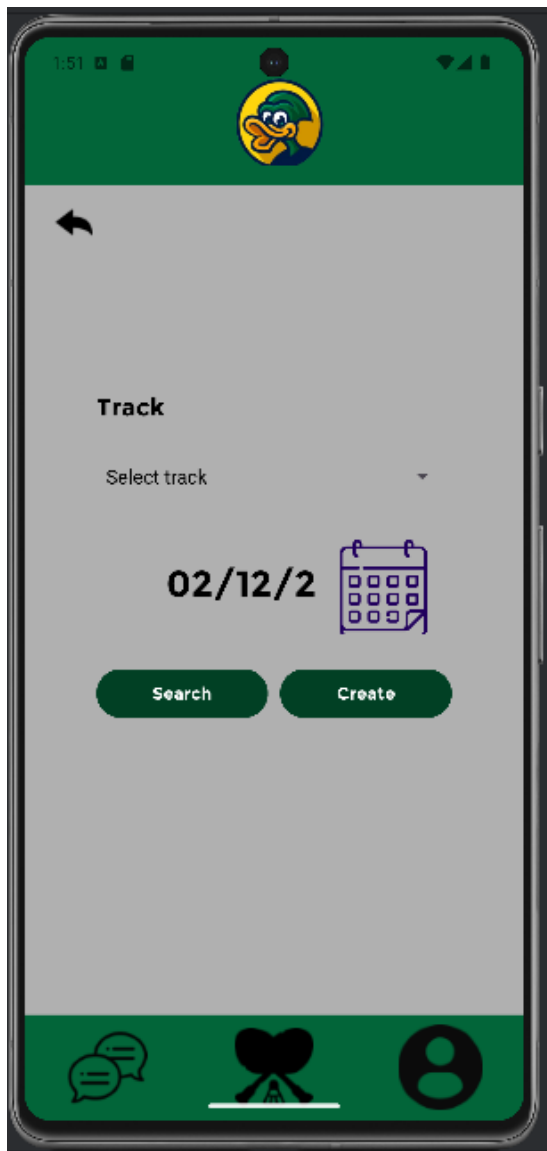


Figura 31: Pantalla de búsqueda de pistas conjuntas.



Figura 32: Desplegable de calendario.

### 6.2.9. Resultado de Búsqueda Antes de Crear

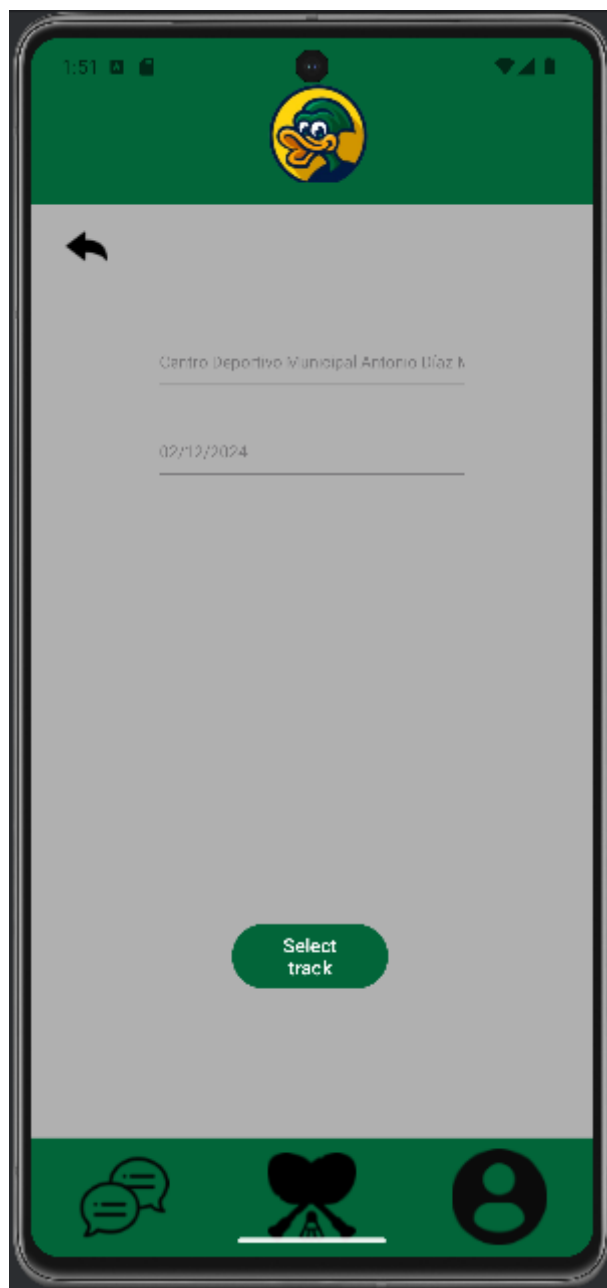


Figura 33: Pantalla con el resultado de búsqueda antes de crear

#### 6.2.10. Resultado de crear



Figura 34: Pantalla de creación de pista

### 6.2.11. Resultado de búsqueda después de crear

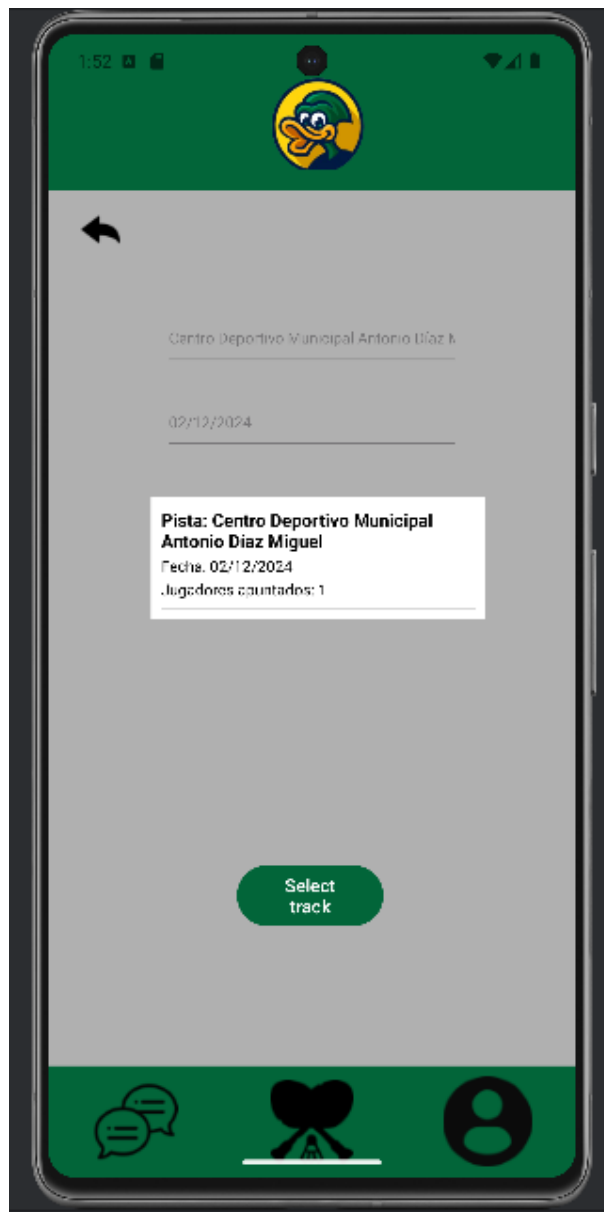


Figura 35: Pantalla con el resultado de búsqueda después de crear

### 6.2.12. Unirse a una sesión

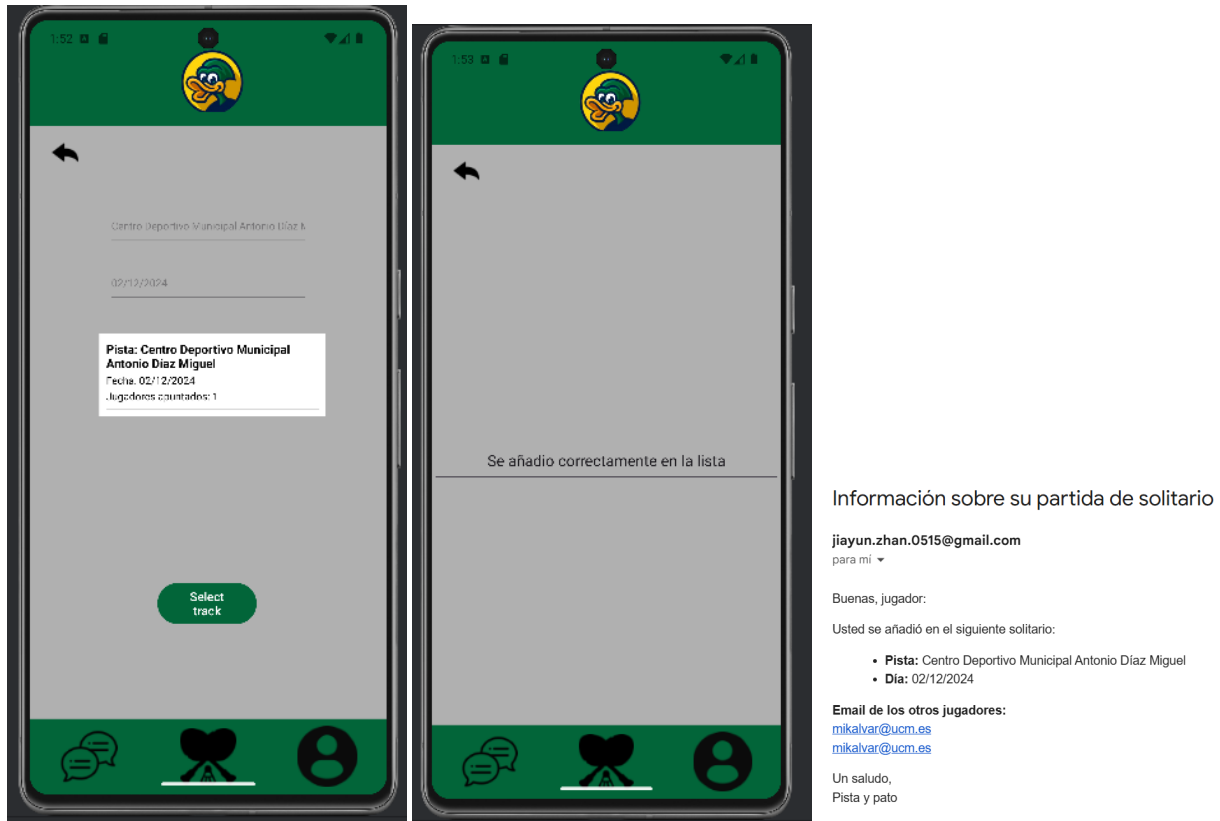


Figura 36: Selección de pista, unión e email de confirmación.

## 7. Video Demostrativo de la App

URL al vídeo: [Clic aquí](#)

## 8. Descripción del Almacenamiento Persistente

El sistema de almacenamiento persistente utilizado en la aplicación está basado en Firebase, una plataforma en la nube que ofrece una solución integral para la gestión de datos en tiempo real y de forma escalable. A continuación, se describen los elementos clave del almacenamiento y las razones que justifican su elección:

### 8.1. Sistema de Almacenamiento: Firebase Firestore

Firebase Firestore es una base de datos NoSQL en tiempo real que permite almacenar y sincronizar datos de forma eficiente entre los dispositivos de los usuarios y la nube. Este sistema se caracteriza por su flexibilidad y facilidad de integración con aplicaciones desarrolladas en Android Studio.

## 8.2. Estructura de Datos

La estructura de datos de Firestore está diseñada para reflejar las necesidades específicas de la aplicación:

- **Usuarios:**

- Se almacena información básica de los usuarios, como su nombre, apellido, correo electrónico y un identificador único.
- También se registran las reservas realizadas por cada usuario, vinculándolas mediante su identificador.

- **Reservas:**

- Cada reserva contiene información sobre la Instalación, la dirección de esta, pista, fecha y hora.

- **Pistas deportivas:**

- Se almacenan datos sobre las instalaciones de la comunidad de Madrid que ofrecen pistas de Bádminton, como nombre, ubicación, y una lista de pistas disponibles para bádminton.
- Dado que el acceso a los horarios de las pistas y su disponibilidad dejó de ser accesible desde la API de la Comunidad de Madrid recientemente, se gestionan horarios ficticios desde Firebase para suplir dicha limitación.

## 8.3. Razones de la Elección

Se eligió Firebase Firestore como base de datos para la aplicación debido a las siguientes ventajas:

- **Sincronización en tiempo real:** Firestore permite que los datos se actualicen automáticamente en los dispositivos de los usuarios, asegurando una experiencia fluida y consistente.
- **Escalabilidad:** La arquitectura de Firebase se adapta al crecimiento de la aplicación, gestionando eficientemente grandes volúmenes de datos.
- **Facilidad de uso:** La integración con Android Studio y el SDK de Firebase simplifican enormemente el desarrollo y la implementación.
- **Reglas de seguridad personalizables:** Firebase permite definir reglas de acceso basadas en el rol del usuario, asegurando que los datos sensibles estén protegidos.
- **Coste efectivo:** Firebase ofrece un modelo de precios flexible, permitiendo que el proyecto crezca de manera sostenible.

## 8.4. Seguridad y Privacidad

Para proteger la información almacenada en Firebase, se han implementado las siguientes medidas:

- Uso de **reglas de seguridad** para restringir el acceso a los datos según la autenticación del usuario.

En resumen, Firebase Firestore proporciona una solución robusta y escalable para gestionar el almacenamiento persistente en la aplicación, abordando tanto los desafíos técnicos como las necesidades funcionales del proyecto.

## 9. Descripción del Servicio(s) Remoto(s) Utilizado(s)

### 9.1. API comunidad de Madrid

En el desarrollo de la aplicación de reserva de pistas de bádminton, se ha integrado un servicio remoto clave proporcionado por la API de la Comunidad de Madrid. Este servicio ha permitido llevar a cabo las funcionalidades de la aplicación.

La API de la Comunidad de Madrid es un servicio web proporcionado por el gobierno regional que permite acceder a diversos datos públicos sobre instalaciones deportivas, eventos, servicios y recursos disponibles para la ciudadanía de Madrid. Esta API es gratuita y ofrece una gran cantidad de información que se puede utilizar en aplicaciones que buscan proporcionar a los usuarios datos actualizados sobre el acceso a servicios públicos. La integración de la API de la Comunidad de Madrid en la aplicación tiene como propósito consultar la disponibilidad de las pistas de bádminton y ofrecer a los usuarios la información más precisa sobre las instalaciones. Con esta integración, la app se beneficia de datos oficiales sobre las instalaciones deportivas en Madrid, lo que asegura que los usuarios puedan acceder a información confiable y actualizada.

- **Consulta de Disponibilidad de Pistas:** A través de esta API, la aplicación puede consultar la disponibilidad de las pistas de bádminton en diferentes centros deportivos de Madrid.
- **Acceso a Datos de Instalaciones:** Además de la consulta de disponibilidad, la API también permite obtener información sobre las características de las instalaciones (como la ubicación, el tipo de pista, los servicios disponibles, etc.), lo que ayuda a los usuarios a elegir la pista que mejor se adapta a sus necesidades.

La integración de la API de la Comunidad de Madrid se realiza mediante solicitudes HTTP a los endpoints específicos proporcionados por la API. Los datos se reciben en formato JSON, lo que permite que la aplicación los procese y los muestre en la interfaz de usuario de manera eficiente.

En el desarrollo de la aplicación de reserva de pistas de bádminton, se ha integrado un servicio clave para la gestión de comunicaciones, proporcionado por la API de JavaMail. Este servicio ha permitido automatizar el envío de correos electrónicos, mejorando la interacción con los usuarios y garantizando la confirmación de sus reservas de manera eficiente.



## 9.2. JavaMail API

La **JavaMail API** es una biblioteca estándar de Java que facilita el envío de correos electrónicos desde una aplicación. En este caso, se ha utilizado para enviar correos de confirmación de reserva a los usuarios, asegurando que reciban notificaciones claras y rápidas tras completar su reserva. La API permite gestionar la creación y el envío de correos electrónicos, y se integra perfectamente con servidores de correo, como el servicio SMTP de Gmail.

Este correo incluye detalles como el nombre de la pista reservada, la fecha y hora de la reserva, y un mensaje agradeciendo al usuario por usar la aplicación.

### 9.2.1. Acceso a Funcionalidades Avanzadas

Además de la confirmación de reservas, la JavaMail API también permite personalizar los correos, lo que ofrece la posibilidad de incluir formatos HTML, añadir archivos adjuntos y enviar correos a múltiples destinatarios.

## 9.3. Funcionamiento de la Integración

La integración de JavaMail API en la aplicación se realiza mediante el envío de correos electrónicos a través de un servidor SMTP. A continuación, se describe el flujo de trabajo:

- El usuario realiza una reserva de pista a través de la aplicación.
- La aplicación genera automáticamente el correo de confirmación, utilizando los detalles de la reserva (pista, fecha, hora, etc.).
- El correo se envía al correo electrónico del usuario utilizando el servidor SMTP configurado, Gmail, a través de la **JavaMail API**.

## 10. Diagrama de navegación

- Pantalla de inicio
  - Opciones disponibles:
    - Iniciar sesión: Permite al usuario acceder a la aplicación con una cuenta existente.
    - Registro: Redirige a una pantalla de creación de cuenta
  - Acción al pulsar:
    - Si se inicia sesión, la aplicación se redirige a la pantalla de búsqueda.
    - Si se registra, se redirige a la pantalla de inicio de sesión.
- Pantalla de registro
  - Campos solicitados:
    - Nombre de usuario
    - Correo electrónico

- Contraseña
- Acción al pulsar:
  - Si se registra, se redirige a la pantalla de inicio de sesión.
  - EN caso de error en los datos, aparece un mensaje de advertencia.
- Pantalla de inicio de sesión
  - Campos solicitados:
    - Nombre de usuario
    - Contraseña
  - Acción al pulsar
    - Una vez ingresadas las credenciales y autenticadas, la aplicación se redirige a la pantalla de búsqueda.
    - Si las credenciales son incorrectas, aparece un mensaje de error y se permite intentar de nuevo.
- Pantalla principal (Esqueleto)
  - Opciones disponibles
    - Buscar pistas: Redirige al usuario a la pantalla inicial de búsqueda de pistas.
    - Acceso al perfil: Redirige a la pantalla del perfil de usuario, donde se puede gestionar la información del usuario.
    - Comunidad: Redirige al usuario a la pantalla inicial de búsqueda de partidos.
- Pantalla inicial de búsqueda
  - Campos solicitados:
    - Pista
    - Fecha
  - Acción al pulsar
    - Una vez ingresada la pista y fecha se redirige la aplicación a la pantalla de búsqueda-resultado.
    - Si no se ha seleccionado pista, sale un mensaje de error y se permite intentar de nuevo.
- Pantalla de búsqueda-resultado
  - Opciones disponibles:
    - Reservar un hueco horario
  - Acción al pulsar
    - Pulsando en un hueco del horario en verde y, tras clickar en reservar, la reserva se guarda.

- Si se clicka en un hueco en rojo en el horario aparece un mensaje de error y se permite intentar de nuevo.
- Pantalla de perfil
  - Opciones disponibles:
    - Editar información
    - Consultar una reserva
  - Acción al pulsar
    - Pulsando el botón de editar información se navega a otro fragmento donde se debe introducir la contraseña correctamente, una vez hecho se navegará a la pantalla de modificar perfil.
    - Pulsando una de las reservas y después el botón consultar se navega la pantalla de búsqueda
- Pantalla de modificar perfil
  - Opciones disponibles:
    - Confirmar
  - Acción al pulsar
    - Una vez rellenados todos los campos (nombre, apellido, correo y contraseña) clickando en el botón se cambiarán los datos.
- Pantalla de comunidad
  - Campos solicitados:
    - Pista
    - Fecha
  - Opciones disponibles:
    - Buscar
    - Crear
  - Acción al pulsar
    - Una vez ingresada la pista y fecha, clickando crear se redirige la aplicación se redirige a la pantalla éxito de creación.
    - Una vez ingresada la pista y fecha, clickando buscar, si ya hay una sesión creada se redirige la aplicación a la pantalla de comunidad-resultado.
    - Si no se ha seleccionado pista, sale un mensaje de error y se permite intentar de nuevo.
- Pantalla de comunidad-resultado
  - Opciones disponibles:
    - Unirse a la sesión
  - Acción al pulsar
    - Pulsando una de las sesiones creadas y el botón de unirse la aplicación se redirige a la pantalla éxito de unión.

## 10.1. Diagrama Navegación

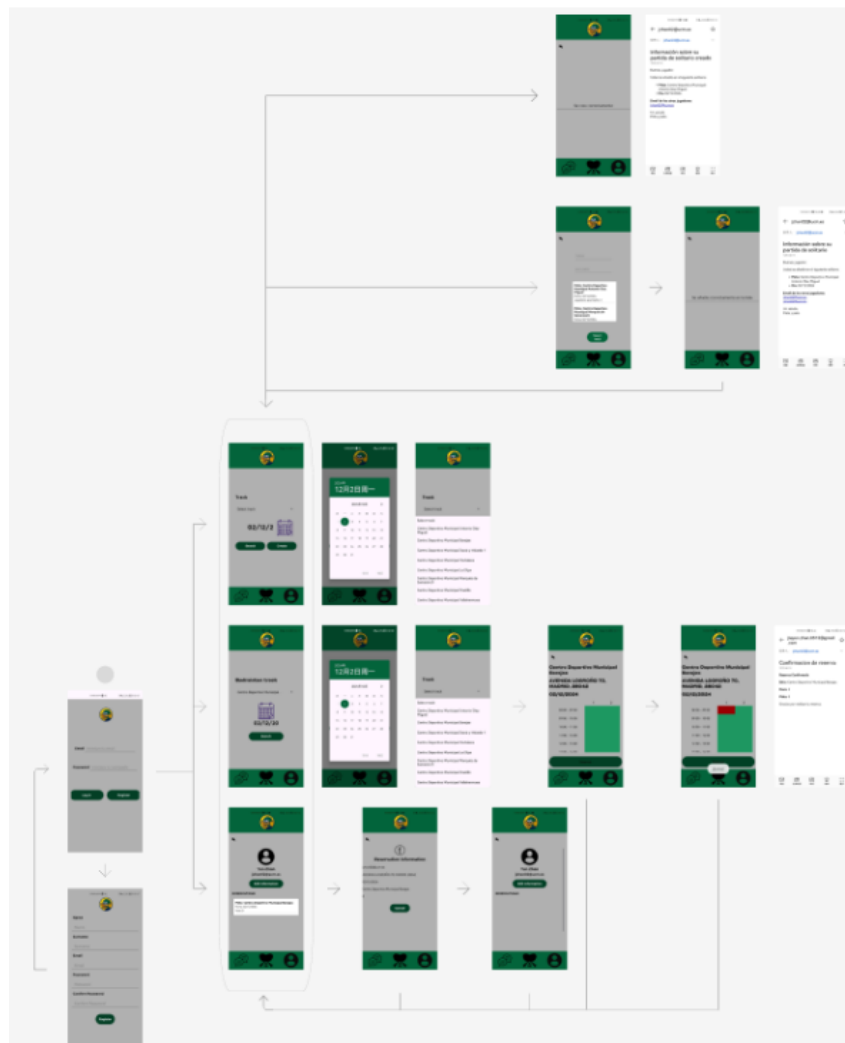


Figura 37: Diagrama Navegación de la Aplicación

## 11. Test unitarios

Se intentaron implementar las pruebas mediante dos enfoques distintos, pero ambos presentaron desafíos técnicos que no pudimos resolver de manera satisfactoria:

### 11.1. Usando Mockito

Inicialmente, intentamos simular la interacción con Firebase utilizando Mockito, con el objetivo de evitar llamadas reales a la base de datos. Este enfoque nos permitiría aislar las pruebas y eliminar dependencias externas. Para ello, empleamos un código como el siguiente:

```
doNothing().when(mockPushReference).setValue(any(Solitario.class));
```

Sin embargo, nos enfrentamos a problemas persistentes al configurar correctamente el comportamiento simulado. A pesar de investigar y realizar múltiples ajustes, no logramos que la funcionalidad se comportara según lo esperado. Como resultado, descartamos este enfoque.

## 11.2. Usando Firebase directamente

Dado que no pudimos avanzar con Mockito, optamos por realizar pruebas conectándonos directamente a Firebase Real. Sin embargo, este enfoque también presentó dificultades, ya que el código

```
FirebaseDatabase.getInstance("https://pistaypato-default-rtdb.europe-west1.firebaseio.com/");
```

fallaba de forma recurrente.

En un intento por solventar este problema, investigamos el uso de Firebase Local Emulator para realizar las pruebas en un entorno local. Esto implicaba la descarga y configuración de la herramienta en nuestro entorno de desarrollo. A pesar de nuestros esfuerzos, no logramos que esta opción fuera viable dentro del tiempo disponible, por lo que recurrimos nuevamente a la versión en la nube, aunque sin éxito.

## 11.3. Conclusión

Debido a estas dificultades, los tests creados no alcanzaron un estado funcional y, en consecuencia, decidimos no subirlos al repositorio de GitHub. Tomamos esta decisión con el objetivo de evitar que código incompleto o defectuoso fuera integrado en el proyecto. Somos conscientes de la importancia de contar con tests funcionales, pero priorizamos mantener la calidad y estabilidad del código entregado.

## 12. Repositorio de GitHub

URL del repositorio:

<https://github.com/javiropruden/PAD-2024-25-PistaYPato>

## 13. Conclusión

El desarrollo de la aplicación de reserva de pistas de bádminton ha sido una experiencia sumamente enriquecedora, en la que se integraron diversas herramientas y tecnologías con el propósito de crear una solución útil y accesible para los usuarios. A lo largo del proyecto, se abordaron tanto los aspectos técnicos como los de diseño, siempre con el objetivo de ofrecer una plataforma sencilla, eficiente y fácil de usar que permita realizar reservas de pistas deportivas en Madrid de manera rápida y efectiva.

La aplicación no solo cumple con las expectativas iniciales, sino que también ofrece una solución escalable y adaptable para el futuro. El sistema de reservas, combinado con la capacidad de interactuar con otros usuarios mediante una comunidad integrada, fomenta un sentido de pertenencia y crea una red deportiva dentro de la aplicación. Además, la

integración con la API de la Comunidad de Madrid garantiza que la aplicación se mantenga actualizada automáticamente con información relevante y confiable, optimizando la experiencia del usuario.

En términos de impacto, la aplicación no solo facilita el acceso a las instalaciones deportivas en Madrid, sino que también contribuye a la organización y optimización de los recursos disponibles, ayudando a los usuarios a gestionar mejor su tiempo y participación en actividades deportivas.

Desde una perspectiva técnica, este proyecto ha representado una oportunidad invaluable de aprendizaje, permitiendo a los desarrolladores experimentar con tecnologías modernas como Firebase y Android Studio, así como aplicar metodologías ágiles para gestionar el trabajo en equipo. Estas herramientas y enfoques han permitido el desarrollo de una aplicación robusta y eficiente, centrada en las necesidades de los usuarios.

En resumen, la aplicación de reserva de pistas de bádminton es una solución práctica que resuelve problemas reales de los usuarios mientras establece una base sólida para futuras mejoras. Con el feedback de los usuarios y el avance en tecnologías, esta aplicación tiene un gran potencial para evolucionar con nuevas funcionalidades, consolidándose como una herramienta indispensable para los amantes del deporte. Este proyecto destaca no solo por su utilidad, sino también por la experiencia de aprendizaje y crecimiento profesional que ha aportado al equipo de desarrollo.