



App de quedadas deportiva + API REST

05/03/2025

Manuel Guerrero Agüera

I.E.S. RAFAEL ALBERTI

PROYECTO FINAL CFGS DAM

ÍNDICE DE CONTENIDO

1. Introducción	Página 3
2. Descripción.....	Página 4
3. Instalación y preparación	Página 6
3.1. Requisitos previos	Página 6
3.2. Instalación y puesta en marcha.....	Página 6
3.3. Control de versiones (GitHub, GitLab, etc.).....	Página 6
4. Prototipado.....	Página 7
4.1. Diseño inicial y wireframes.....	Página 7
5. Diseño funcional.....	Página 8
5.1. Diagramas entidad-relación.....	Página 8
5.2. Casos de uso.....	Página 9
5.3. Diagramas de flujo.....	Página 17
6. Desarrollo.....	Página 18
6.1. Secuencia de desarrollo.....	Página 18
6.2. Dificultades encontradas.....	Página 22
6.3. Decisiones tomadas.....	Página 22
6.4. Herramientas de control de versiones y revisión de código.....	Página 22
7. Pruebas.....	Página 23
7.1. Descripción de las pruebas realizadas.....	Página 23
7.2. Automatización de pruebas unitarias.....	Página 23
8. Distribución.....	Página 24
8.1. Tecnologías utilizadas para la distribución.....	Página 24
8.2. Descripción del proceso de despliegue.....	Página 25
9. Manual.....	Página 26
9.1. Instalación de la aplicación.....	Página 26
9.2. Uso y funcionalidades principales.....	Página 27
10. Conclusiones	Página 41
10.1. Comparación del resultado con la idea inicial.....	Página 41
10.2. Posibles mejoras futuras.....	Página 42
11. Índice de tablas e imágenes.....	Página 43
12. Bibliografía y referencias.....	Página 44

1. INTRODUCCIÓN

SportyHub es una aplicación para móvil cuyo propósito principal es el de una plataforma de quedadas deportivas en la que los usuarios puedan conectar e interactuar entre ellos. Con la finalidad de que los usuarios tengan una herramienta con la que puedan organizar actividades para la práctica de diferentes deportes y también con la intención de asociar esta combinación de colores con algo más positivo para la salud de las personas.

El funcionamiento de esta aplicación consiste en que un usuario crea una actividad del deporte que desee entre las opciones disponibles en la aplicación, especificando detalles de la actividad como el lugar, la fecha y hora, duración, o el precio si tuviese.

Una vez el usuario cree la actividad, esta será visible para el resto de usuarios, de la cual podrán ver los detalles y demás usuarios que vayan a participar en la actividad. Los usuarios tendrán un listado de estas actividades que los usuarios han creado, de las que siempre podrán ver los detalles de la actividad. Los usuarios podrán decidir si unirse a la actividad si está coincide con sus intereses, y de si han pagado el precio para unirse, en caso de que la actividad fuese de pago.

En caso de que la actividad sea gratuita, el usuario podrá unirse siempre directamente sin restricción alguna, aunque siempre y cuando en la actividad a la cual se va a unir disponga de espacio para más participantes. En el caso contrario de que la actividad sea no gratuita, el usuario tendrá que pasar por una pasarela de pago y realizar un pago antes de poder unirse.

Se dispone de una serie de filtros de búsqueda en el listado de actividades con diferentes etiquetas para una mayor eficacia a la hora de buscar lo que el usuario realmente desea.

2. DESCRIPCIÓN

SportHub es una app para móviles creada para simplificar la planificación y el disfrute de actividades deportivas; además de favorecer la interacción entre individuos que comparten aficiones deportivas similares.

El principal propósito consiste en ofrecer un espacio donde los usuarios puedan organizar y participar en eventos deportivos de forma más fácil y así promover la práctica del deporte en la sociedad. La aplicación brinda a los usuarios la posibilidad de visualizar actividades cercanas a su ubicación mediante la integración de la API de Google Maps para facilitar la identificación geográfica de dichos eventos.

Al momento de registrarse en la plataforma, los usuarios tendrán acceso a un catálogo de actividades clasificadas según su ubicación geográfica, tipo de deporte, costos y otros criterios para una búsqueda más personalizada. Asimismo, podrán consultar información detallada como fecha, hora, duración y costos estimados para participar en las mismas, uniéndose si cumplen los requisitos establecidos.

Roles de usuario

Existen tres roles dentro de SportyHub:

- **Usuario normal:** Puede crear actividades, unirse a eventos y formar parte de equipos.
- **Líder de equipo:** Además de las funciones de un usuario normal, puede gestionar equipos y registrar a sus miembros en actividades.
- **Administrador:** Se encarga de la gestión de la plataforma, teniendo el control sobre todo lo que conlleva gestionar el sistema.

Gestión de equipos y participación colectiva

Los usuarios pueden formar equipos dentro de la aplicación, personalizar la información de este.

Método de inscripción y pagos

SportyHub permite la inscripción en eventos gratuitos sin restricciones, siempre que haya espacios para participantes disponibles. En el caso de actividades de pago, los usuarios deben de completar un proceso a través de la pasarela de pago integrada antes de confirmar su participación. Los pagos se gestionan mediante un sistema que retiene el importe hasta la finalización del evento, evitando fraudes y boicots en las actividades.

Infraestructura y tecnologías utilizadas

La aplicación está desarrollada con las siguientes tecnologías:

- **Android Studio (Java)** en el desarrollo de la aplicación móvil.
- **Spring Boot** para el desarrollo de la API que gestiona la lógica de entidades y la conexión con la base de datos.
- **PostgreSQL** como base de datos principal.
- **API de Google Maps y Lugares** para la localización, lugares y la visualización de actividades en mapas.
- **Cloudinary** para la gestión de imágenes subidas por el usuario.
- **Render** para el despliegue de la API en la nube.
- **Paypal** para la pasarela de pago.

SportyHub ofrece una plataforma intuitiva y eficiente para que los usuarios encuentren actividades deportivas que se ajusten a sus preferencias, promoviendo la interacción y el deporte en la comunidad.

3. INSTALACIÓN Y PREPARACIÓN

3.1. Requisitos previos

Para la instalación y uso de la aplicación no se requiere de gran cosa, para ello es necesario lo siguiente:

- Un dispositivo móvil con una versión **Android 13 o superior**, debido a las dependencias que usa esta aplicación y a sus respectivas versiones, es incompatible con versiones Android anteriores a esta.
- Conexión a internet en el dispositivo móvil para poder interactuar y realizar peticiones a la API, para poder cargar los datos en la aplicación.

Con solo esto ya estaremos listos para poder proceder a la instalación y uso de la app.

3.2. Instalación y puesta en marcha

No es necesaria ninguna configuración previa del entorno de uso para la utilización de la aplicación. Simplemente abriendo el archivo APK de la aplicación podremos instalar y empezar a usar la aplicación directamente sin pasos adicionales. En cuanto a la API a la que se conecta la aplicación, está ubicada y funcionando en la nube por lo que no es necesario nada más que instalar la aplicación para usarla.

3.2. Control de versiones

Aunque principalmente se ha llevado a cabo el desarrollo de la aplicación en el entorno local, he utilizado GitHub como repositorio final para guardar el código y poder garantizar su seguridad y disponibilidad. Durante el desarrollo, mayoritariamente no se ha implementado una estrategia de control de versiones, sin embargo, el código fuente se encuentra accesible en GitHub para futuras mejoras o mantenimiento.

4. PROTOTIPADO

4.1. Wireframe de la aplicación

Para el prototipado de la aplicación he realizado un diseño wireframe, para verlo en más detalle, lo tienes disponible en el repositorio junto a este documento o bien pulsando [aquí](#).



5. DISEÑO FUNCIONAL

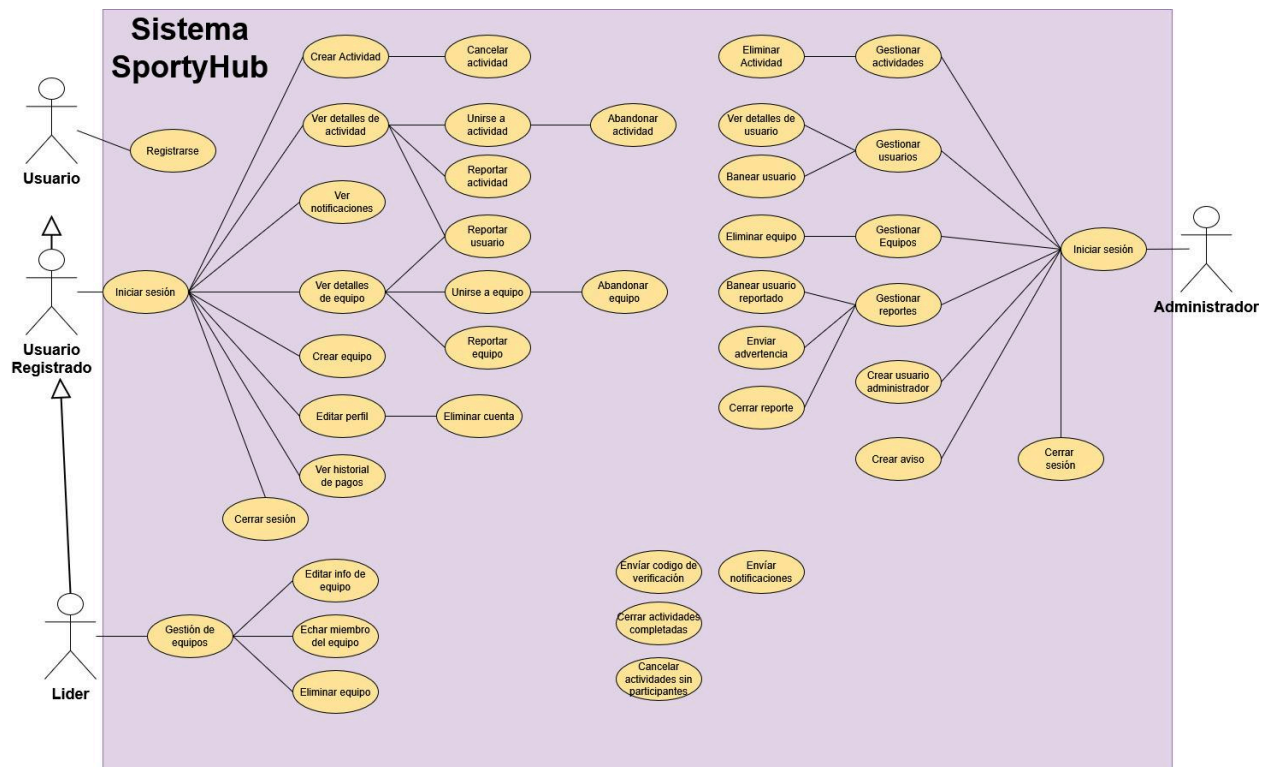
5.1. Diagrama entidad-relación

El diagrama entidad-relación de SportyHub, cuenta con múltiples entidades como Actividad, Equipo, Pago, Notificación, etc. Entre estas entidades se relacionan mayoritariamente con una relación de tipo Many2Many con lo cual, termina con la creación de otras entidades adicionales. El diagrama por aquí se ve algo pequeño pero puedes verlo completo y en detalle pulsando [aquí](#).



5.2. Diagrama de casos de uso

Para los casos de uso tenemos el siguiente diagrama con los correspondientes casos:



Como se puede ver, en este sistema tenemos cuatro actores diferentes con sus respectivos casos que pueden llevar a cabo.

- Descripción de casos de uso:

Puesto a que hay casos que son parecidos los unos de los otros voy a poner la descripción de algunos casos de uso de entre los más relevantes y diferentes entre sí. En este caso voy a mostrar la descripción de los siguientes casos de uso:

- **1.** Crear Actividad
- **2.** Unirse a Actividad
- **3.** Reportar Actividad
- **4.** Ver notificaciones
- **5.** Eliminar usuario
- **6.** Registrarse
- **7.** Banear usuario.

1. Crear Actividad

Nombre	Crear Actividad	
Descripción	El usuario crea una nueva actividad en la plataforma introduciendo los datos de esta.	
Actores	Usuario registrado	
Pre condiciones	El usuario debe estar registrado y autenticado previamente en la aplicación.	
Flujo Principal	1	El usuario pulsa el botón para crear la actividad.
	2	El usuario rellena el formulario con los detalles de la actividad (precio, fecha, etc).
	3	El usuario pulsa guardar para crear finalmente la actividad.
	4	El sistema comprueba y valida la información ingresada.
	5	El sistema guarda la información en la base de datos.
	6	El sistema muestra un mensaje de confirmación exitosa al usuario.
Flujo Alternativo	1	Si la información introducida por el usuario es inválida, el sistema mostrará un mensaje al usuario pidiendo que corrija los datos.
Post condiciones	La nueva actividad se ha creado y ahora está disponible para que los demás usuarios se unan.	

2. Unirse a actividad

Nombre	Unirse a actividad			
Descripción	El usuario quiere ser participante de una actividad y decide unirse			
Actores	Usuario registrado			
Pre condiciones	El usuario debe estar registrado y autenticado previamente en la aplicación.			
Flujo Principal	1	El usuario pulsa en la actividad que le interesa.		
	2	El sistema devuelve la información la información de la actividad y muestra los detalles de esta en una nueva pantalla.		
	3	El usuario pulsa sobre el botón de unirse.	3B	El usuario pulsa el botón de pagar.
	4	El sistema comprueba que haya sitio disponible en la actividad	4B	El sistema comprueba que haya plazas disponibles.
	5	El sistema registra la participación del usuario en la actividad.	5B	El sistema le muestra al usuario la plataforma de pago
	6	El sistema muestra un mensaje de confirmación al usuario.	6B	El usuario realiza el pago
			7B	El sistema registra el pago
			8B	El sistema muestra un mensaje de confirmación al usuario.
Flujo Alternativo	1	Si en la actividad seleccionada no quedan plazas disponibles, el sistema mostrará el error al usuario y este no podrá unirse.		
	2	Si el pago del usuario no es válido o se cancela, el sistema notificará al usuario de que no ha podido unirse a la actividad.		
Post condiciones	El usuario está registrado como participante de la actividad seleccionada.			

3. Reportar actividad

Nombre	Reportar actividad	
Descripción	El usuario quiere realizar un reporte sobre una actividad que considera poco apropiada	
Actores	Usuario registrado	
Pre condiciones	El usuario debe estar registrado y autenticado previamente en la aplicación.	
Flujo Principal	1	El usuario pulsa en la actividad la cual quiere reportar.
	2	El sistema devuelve la información la información de la actividad y muestra los detalles de esta en una nueva pantalla.
	3	El usuario pulsa sobre la opción de reportar actividad.
	4	El sistema le muestra una pantalla con un formulario para el reporte.
	5	El usuario rellena el formulario de reporte con el motivo y una descripción.
	6	El sistema guarda el reporte en la base de datos para que los administradores puedan revisarlo.
	7	El sistema devuelve al usuario a la pantalla de inicio y le muestra un mensaje de que el reporte se ha realizado correctamente.
Flujo Alternativo	1	Si los datos del formulario del reporte no son completados, el sistema mostrará un mensaje de error al usuario para que lo rellene.
Post condiciones	El usuario ha realizado la denuncia a la actividad seleccionada.	

4. Mostrar notificaciones

Nombre	Mostrar notificaciones	
Descripción	El usuario puede mirar las lista de notificaciones que tiene en su perfil.	
Actores	Usuario registrado	
Pre condiciones	El usuario debe estar registrado y autenticado previamente en la aplicación.	
Flujo Principal	1	El usuario toca su perfil
	2	El sistema le muestra un panel lateral de opciones.
	3	El usuario pulsa la opción de notificaciones.
	4	El sistema lleva al usuario a otra pantalla.
	5	El sistema recupera la información de las notificaciones del usuario y las envía a la aplicación.
	6	El sistema carga las notificaciones en la pantalla y muestra al usuario sus notificaciones.
Flujo Alternativo	1	Si el usuario no tiene notificaciones, el sistema le mostrara por pantalla de que no tiene ninguna.
Post condiciones	El usuario ha realizado la denuncia a la actividad seleccionada.	

5. Eliminar usuario

Nombre	Eliminar usuario	
Descripción	El usuario tiene la opción de decidir eliminar su cuenta	
Actores	Usuario registrado	
Pre condiciones	El usuario debe estar registrado y autenticado previamente en la aplicación.	
Flujo Principal	1	El usuario toca su perfil
	2	El sistema le muestra un panel lateral de opciones.
	3	El usuario pulsa la opción de ajustes.
	4	El sistema lleva al usuario a otra pantalla con una lista de ajustes.
	5	El usuario pulsa sobre la opción de perfil.
	6	El sistema lleva al usuario a una nueva pantalla para ajustes sobre su perfil, entre ellos eliminar la cuenta.
	7	El usuario pulsa sobre el botón de eliminar cuenta.
	8	El sistema pide confirmación al usuario
	9	El usuario confirma que desea eliminar su cuenta.
	10	El sistema sistema desloguea al usuario.
	11	El sistema cancela las actividades creadas por el usuario y sus participaciones pendientes.
	12	El sistema cambia de líder a los equipos creados por el usuario si estos tienen miembros, sino los elimina. Y elimina al usuario de los equipos donde este era miembro.
	13	El sistema notifica a los usuarios implicados, si hubiera, con las consecuencias de cancelar actividades/participaciones y de eliminar equipos/miembros.
	14	El sistema muestra al usuario un mensaje de que la eliminación de su cuenta ha sido exitosa.
Flujo Alternativo	1	Si el usuario no confirma la eliminación al sistema, esta acción se cancelará
Post condiciones	El usuario ha realizado la denuncia a la actividad seleccionada.	

6. Registrarse

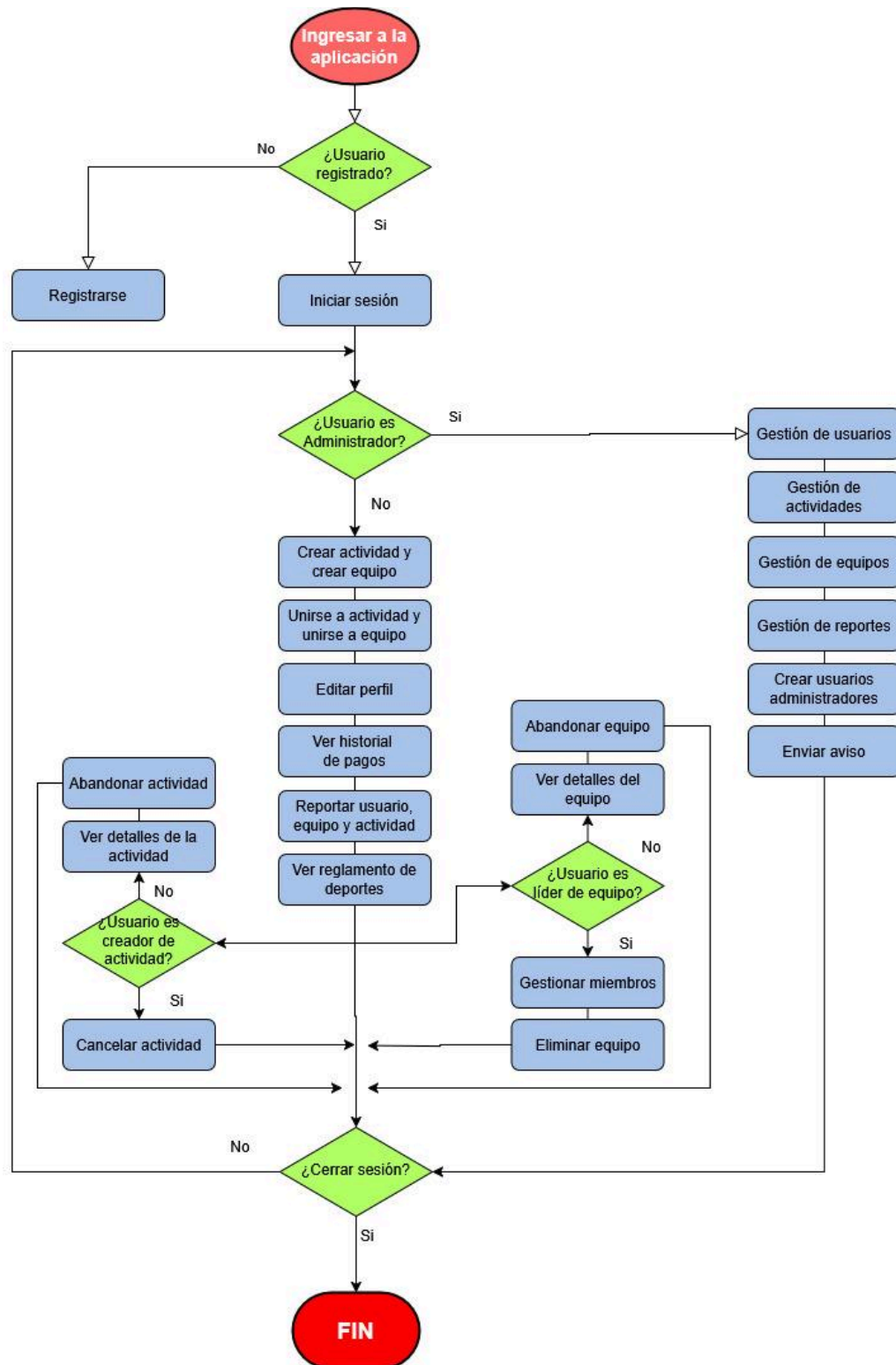
Nombre	Registrarse	
Descripción	El usuario se registra para poder usar las demás funciones de la app.	
Actores	Usuario	
Pre condiciones	No hay ninguna condición previa.	
Flujo Principal	1	El usuario toca el botón de registrarse.
	2	El sistema lo lleva a una pantalla con un formulario para que introduzca su correo y contraseña a usar.
	3	El usuario rellena el formulario y pulsa en continuar.
	4	El sistema lleva al usuario a otra pantalla con un campo donde introducir un código de confirmación.
	5	El sistema envía un correo con el código de confirmación a la dirección de correo que el usuario introdujo anteriormente.
	6	El usuario introduce el código de confirmación y pulsa en siguiente.
	7	El sistema lleva al usuario a una última pantalla con otro formulario con los datos (Nombre, nickname, pfp) para poder finalizar el registro.
	8	El usuario completa el formulario y pulsa en finalizar registro.
	9	El sistema envía y guarda la información del nuevo usuario en la base de datos.
	10	El sistema muestra un mensaje al usuario confirmando el registro.
	11	El sistema lleva al usuario a la pantalla de login y loguea al usuario automáticamente.
	12	El sistema lleva al usuario a la pantalla principal.
Excepciones	2a	Si las direcciones de correo o las contraseñas no coinciden, el sistema mostrará un error al usuario sobre ello.
	6a	Si el código de confirmación introducido por el usuario no coincide con el enviado, el sistema notificará al usuario de la discrepancia.
	8a	Si la información introducida no cumple con un formato adecuado o el nickname ya está siendo usado el sistema mostrará el error al usuario.
Post condiciones	El usuario ha finalizado el registro de su cuenta y ahora es un usuario registrado.	

7. Enviar advertencia

Nombre	Enviar advertencia	
Descripción	El administrador puede enviar advertencias al usuario que esté asociado a un reporte.	
Actores	Administrador	
Pre condiciones	El usuario debe ser administrador y estar logueado previamente.	
Flujo Principal	1	El administrador toca el botón el apartado de reportes.
	2	El sistema le muestra una pantalla con la lista de reportes pendientes.
	3	El administrador selecciona un reporte de la lista.
	4	El sistema lleva al administrador a otra pantalla con los detalles del reporte seleccionado.
	5	El administrador en el apartado de acciones, pulsa en el botón de enviar advertencia.
	6	El sistema carga una nueva pantalla con un formulario para la advertencia formada por un título y un mensaje.
	7	El administrador completa el formulario y pulsa el botón de envíar.
	8	El sistema envía y guarda la información de la advertencia.
	9	El sistema crea y envía una notificación al usuario asociado al reporte con la información de la advertencia.
	10	El sistema muestra por mensaje al administrador que la advertencia se ha creado y enviado correctamente.
	11	El sistema cambia ese reporte de pendiente a revisado y lleva al usuario de vuelta a la pantalla principal.
Excepciones	2a	Si no hay reportes pendientes, el sistema mostrará que no hay reportes que revisar.
	7a	Si los campos del formulario no han sido completados, el sistema mostrará al administrador el error.
Post condiciones	El administrador ha enviado una advertencia al usuario asociado a un reporte.	

5.3. Diagrama de flujo

Por aquí tenemos el diagrama de flujo básico de la aplicación:



6. DESARROLLO

6.1. Secuencia de desarrollo

La secuencia de desarrollo tanto de la aplicación como de la API se han dividido en las siguientes fases.

Planificación:

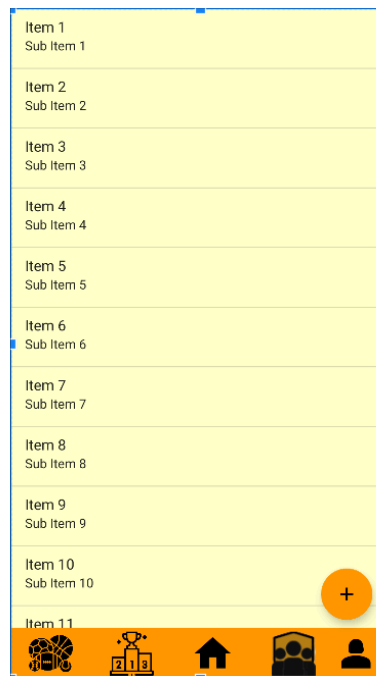
En esta fase inicial, se definieron los requisitos de la aplicación:

“Aplicación móvil cuyo propósito principal es el de que los usuarios puedan conectar entre ellos para la práctica de diferentes deportes. El funcionamiento consistiría en que un usuario crea una actividad del deporte que desee, especificando los detalles de la actividad como el lugar, la fecha y hora, número de personas máximo, o el precio si tuviese.

Una vez un usuario cree la actividad, será visible para el resto de usuarios que podrán unirse a esa actividad entre las diferentes que hayan publicadas por otros usuarios siempre y cuando estos cumplan con los requisitos para unirse a esta.”

De los cuales no todos pasarían a una fase superior de desarrollo debido a circunstancias como que no eran adecuadas o bien no daba tiempo para su desarrollo.

En esta fase también se realizó un diseño inicial de la aplicación:



Prototipado:

Para el prototipado de la aplicación he usado figma para realizar un diagrama wireframe de la aplicación. Este se usó como base para poder desarrollar las funciones básicas de la aplicación y de la API.

El diagrama wireframe de la aplicación está disponible en el repositorio donde se encuentra este documento y también pulsando [aquí](#).

Desarrollo Backend:

Para el desarrollo de la API he utilizado Spring Boot y el IDE IntelliJ Community. La API se ha desarrollado para que se conecte a una base de datos PostgreSQL remota alojada por la página [Neon](#) en su tier gratuito, y se ha usado el software de DBeaver para la gestión de la información de la base de datos, aunque realmente esto no es necesario, te permite el control de la base de datos de una manera más visual e intuitiva.

Se utiliza la librería JBCrypt para encriptar y desencriptar las contraseñas de los usuarios a la hora de crear sus cuentas y también para autenticar a los usuarios.

Para las peticiones a la API, al iniciar sesión en la aplicación, la API genera un token único para el usuario con la información para determinar y asegurar la confidencialidad de las peticiones, además de restringir el uso de estas a quienes no estén autorizados para ello.

Por último se ha subido esta API a la nube mediante [Render](#), para que esté disponible de manera remota.

Desarrollo Frontend:

Para el desarrollo de aplicación móvil se ha usado el IDE de Android Studio utilizando para todo el código de la aplicación el lenguaje Java.

Las peticiones a la API se realizan mediante la clase Retrofit, que es la que actúa como cliente HTTP y se encarga de transformar el código a peticiones HTTP para la API, y de transformar las respuestas de API en objetos para la aplicación.

Integraciones:

Para el funcionamiento de varias características de la aplicación se han usado varias integraciones. Cabe destacar que el uso de todas estas han sido gratuitas.

Google Maps Api: La API de google nos aporta los mapas detallados juntos con sus lugares y marcadores. También nos permite la búsqueda de lugares por nombre.

Cloudinary: Utilizado para subir las imágenes de los usuarios y de los equipos, y almacenarlas en un lugar en la nube.

Paypal: Se emplea principalmente para unirse a las actividades de pago. Esta otorga la simulación de los pagos en la aplicación.

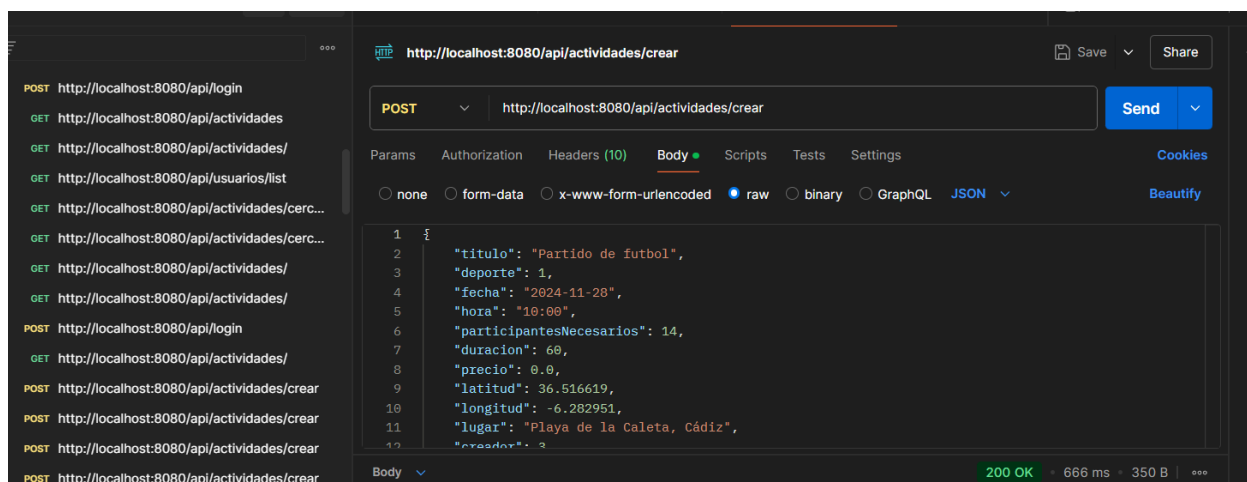


*Uso de la API de Google Maps en la Aplicación

Pruebas y ajustes:

Para las pruebas y para la depuración de errores se han ido usando Logs por todo el código para corregir los errores y bugs del código. Tanto en la aplicación como en el desarrollo de la API.

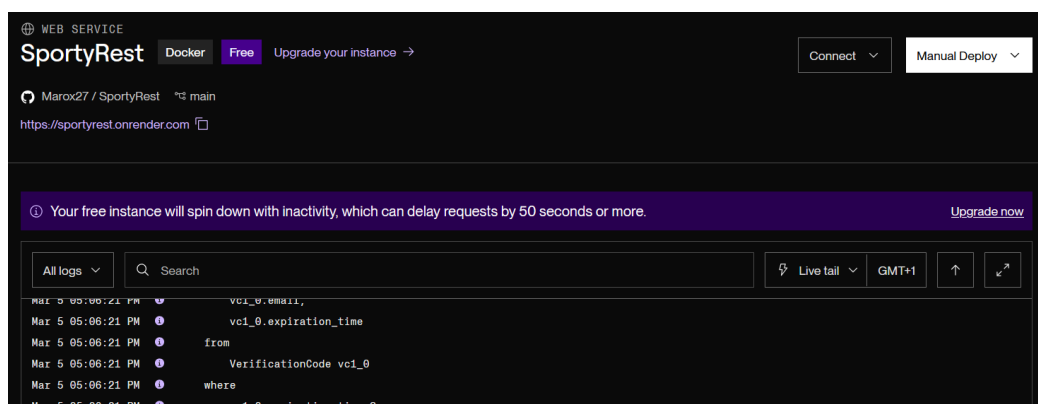
Adicionalmente para las pruebas de peticiones a la API se ha usado el software de postman.



Despliegue:

Finalmente la API se ha publicado en Render y configurado todo lo necesario como las variables de entorno, mediante un docker y se han realizado las pruebas pertinentes en dispositivos reales.

Uno de los problemas a raíz de esto ha sido las limitaciones del tier gratuito de Render, ya que estas hacen que las peticiones requieran de un mayor tiempo de respuesta, además de que esta se desactiva pasado un poco de tiempo de inactividad.



6.2. Dificultades encontradas

Han sido múltiples los obstáculos encontrados que impedían el correcto funcionamiento de la aplicación, principalmente entre ellos:

- Problemas varios con el emulador de android a la hora de realizar pruebas, como que los mapas no cargaran, se quedase colgado, etc.
- El recorte de imágenes y su posterior subida a Cloudinary.
- La integración de la API de Google Maps y Google Places en el proyecto.
- La captura de órdenes de la API de PayPal.
- La eliminación de un usuario con actividades y equipos activos, con todo lo que conllevaba su eliminación.

6.3. Decisiones tomadas

En ciertos puntos del desarrollo he tomado decisiones clave para el funcionamiento de ciertas funciones en la aplicación, que bajo mi punto de vista, eran necesarias y mejoraban de alguna manera su comportamiento.

- Retener los pagos de las actividades hasta que estas se completaran con éxito, con la finalidad de evitar estafas.
- Dividir el registro en pasos para mejorar la experiencia de registro del usuario.
- Incorporar un código de verificación en el registro para evitar registro de personas con direcciones de correo que no le pertenecen.
- Quitar el derecho a un reembolso a la participación de una actividad cuando queda menos de 1 hora para que comience, esto para evitar que los usuarios no se responsabilicen de su acción y para evitar perjudicar al resto de participantes.

6.4. Herramientas de control de versiones y revisión de código

Para el control de versiones de la aplicación y de la API se ha usado gitHub, aunque mayoritariamente este control de versiones se realizó en local, hasta que se ha subido la API a Render y que he almacenado el código de manera remota. En estas revisiones entre un cambio y otro se han realizado pruebas prácticas con un dispositivo real para la aplicación y el aspecto de esta, además de comprobar funciones que incluían a su vez el uso de la API. Y también de Postman para las peticiones y la funcionalidad de la API.

7. PRUEBAS

7.1. Descripción de las pruebas realizadas

Para el correcto funcionamiento tanto de la app como de la API se han realizado múltiples pruebas entre las que podemos destacar:

- **Pruebas manuales en Postman:** Usado para verificar los endpoints de la API y comprobar si la respuesta es la esperada. Se ha usado esta prueba para la creación de usuarios, inicio de sesión, inscripción en actividades, creación de equipos, obtención de provincias, y otras funcionalidades clave.
- **Pruebas en dispositivos físicos y emuladores:** Debido a las ciertas limitaciones del emulador de Android Studio, se han realizado pruebas de funcionamiento tanto en emulador, como en dispositivos reales. Uno de los mayores errores detectados gracias a estas pruebas, fue el de descubrir que los mapas de la API de Google Maps no cargan correctamente en el emulador, mientras que en el dispositivo real sí.
- **Pruebas de integración:** Estas pruebas se han centrado en el testeo de funcionamiento de todas las integraciones (Maps, Cloudinary y PayPal) con la aplicación, además de evidentemente comprobar que nuestra API propia comunicaba correctamente con la base de datos PostgreSQL y con servicios externos como Cloudinary y PayPal.
- **Pruebas funcionales:** Además de las pruebas propias del funcionamiento en la app, se ha evaluado la experiencia del usuario en la app para asegurar que los flujos como registro de cuenta, inicio de sesión, unirse en actividades, etc, eran intuitivos y sin errores.

7.2. Automatización de pruebas unitarias

En este proyecto no se han aplicado pruebas automáticas debido a la naturaleza del proyecto y el tiempo disponible. En su lugar las pruebas han sido de carácter manual, y aunque más usuarios además de mí han participado en el testeo funcional de la aplicación final, sería bastante recomendable incluir test automatizados en un futuro para poder mejorar la estabilidad de la app.

8. DISTRIBUCIÓN

8.1. Tecnologías utilizadas para la distribución

Debido a que tenemos tanto una aplicación móvil como una API, las tecnologías de distribución se dividen en:

- **Backend (API):**
 - **Render:** Para el despliegue de la API en la nube mediante docker.
 - **PostgreSQL en Neon:** Para la base de datos de PostgreSQL.
 - **Cloudinary:** Para el almacenamiento de imágenes que suban a la app.
- **Frontend (App móvil):**
 - **Generación de APK en Android Studio:** Para distribuir la aplicación en dispositivos Android.

Debido a que la aplicación no está pensada actualmente para una descarga al público, utilizamos simplemente la generación del APK que nos incorpora Android Studio. Si en algún momento se planease subir la app a Google Play, sería necesario de un pequeño proceso extra que explicaré brevemente el siguiente punto.

8.2. Descripción del proceso de despliegue

Como hemos mencionado en el punto anterior nuestro proyecto tenemos por una parte la API y por otro lado la aplicación móvil, por lo que el despliegue se separa en:

- **Despliegue de la API en Render:**
 1. Subida del código de la API al repositorio de GitHub.
 2. Configuración en Render para desplegar un docker automáticamente con el contenido desde GitHub.
 3. Configuración de variables de entorno necesarias para la API (credenciales para la base de datos, Cloudinary, PayPal, Maps, etc).
 4. Finalmente probar con postman o con la aplicación para verificar el correcto funcionamiento de los endpoints.

- **Distribución de la App móvil:**
 1. Compilar y generar el archivo APK en Android Studio.
 2. Instalar y realizar pruebas en dispositivos móviles reales.
 3. Repartir el APK para pruebas con usuarios.

Si en el futuro se planea subir la aplicación a Google Play, es necesario crearse una cuenta de desarrollador de Google Play y crear una clave de carga para poder firmar el apk con una clave de desarrollador y luego terminar de completar la verificación en Google Play Console.

9.MANUAL

9.1. Instalación de la aplicación

Los pasos a seguir para instalar el archivo APK son los siguientes:

1. **Descargar el archivo APK** desde la fuente proporcionada.
2. **Permitir la instalación desde orígenes desconocidos**, aunque esto puede no ser necesario si ya tenías la opción habilitada:
 - Ir a **Ajustes > Seguridad**.
 - Y activar la opción "**Permitir instalación desde fuentes desconocidas**".
3. **Abrir el archivo APK** y seguir las instrucciones en pantalla para completar la instalación.
4. **Abrir la aplicación** y conceder los permisos requeridos (ubicación y notificaciones, almacenamiento, si aplica).

*** Recordatorio importante:** Tal y como expliqué al principio de los requisitos previos, es necesario tener una versión de Android 13 o superior para poder instalar y usar la aplicación.

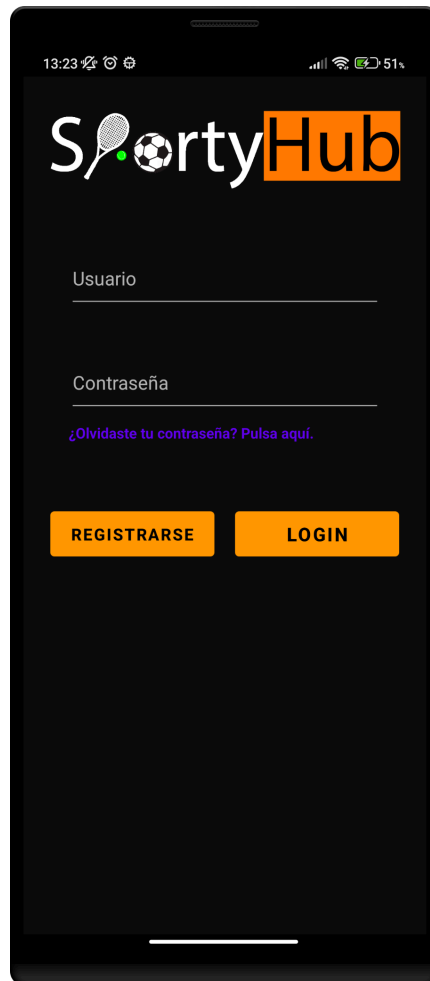
9.2. Uso y funcionalidades principales

En este apartado voy a mostrar cómo empezar a usar la aplicación y a cómo se realizan las funciones principales de la app. Las funciones que explicaré son:

- **A) Crear cuenta y registrarse**
- **B) Crear una actividad**
- **C) Unirse a una actividad**

Empezar a usar la aplicación:

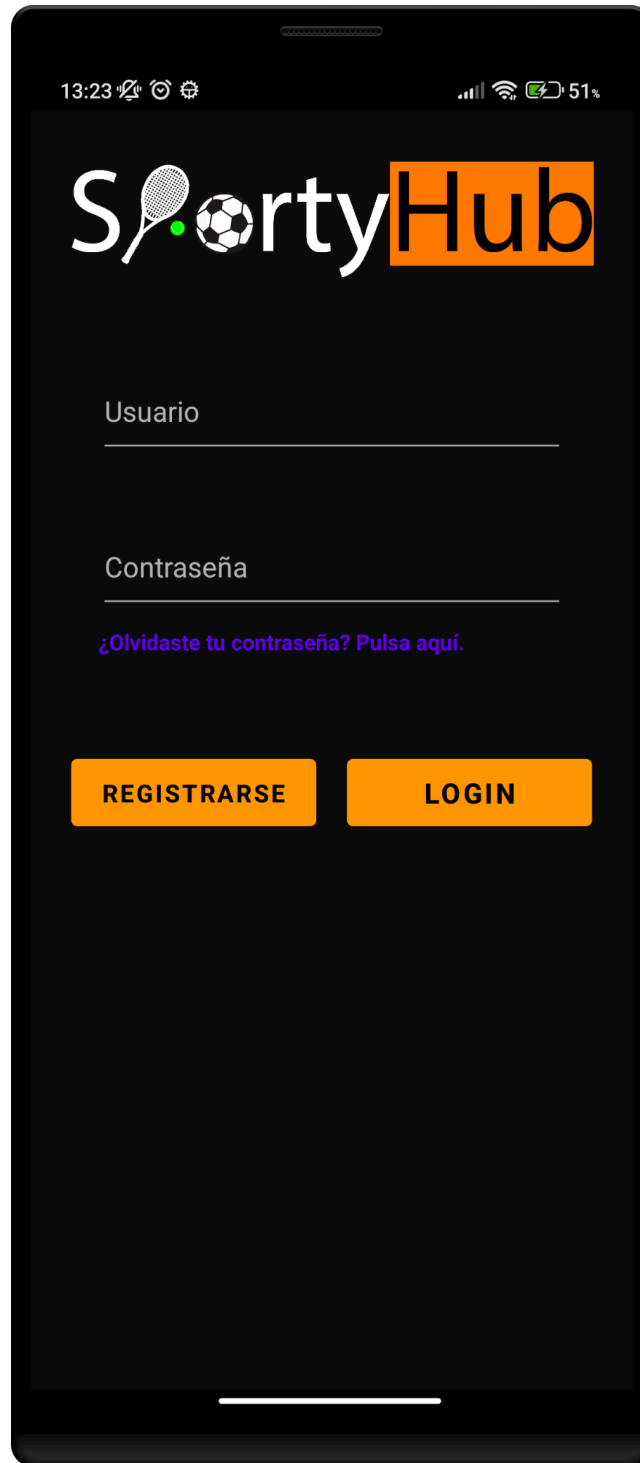
Una vez instalada la aplicación, lo primero que veremos al entrar en la app será la pantalla de inicio de sesión.



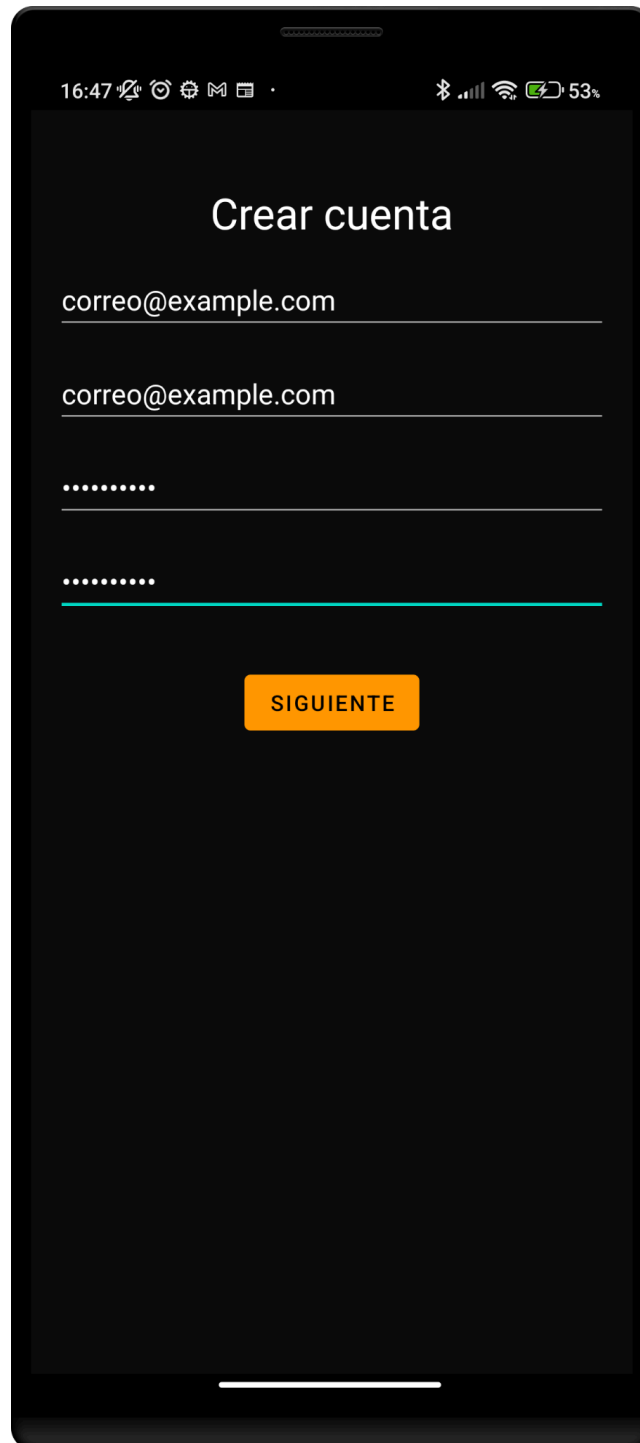
Una vez aquí, para poder empezar a usar la aplicación, lo primero que debemos de hacer es crearnos una cuenta nueva y registrarnos.

A) Crear una cuenta y registrarse:

Para poder registrarnos en la aplicación en la pantalla inicial debemos de pulsar el botón que pone **"REGISTRARSE"**.

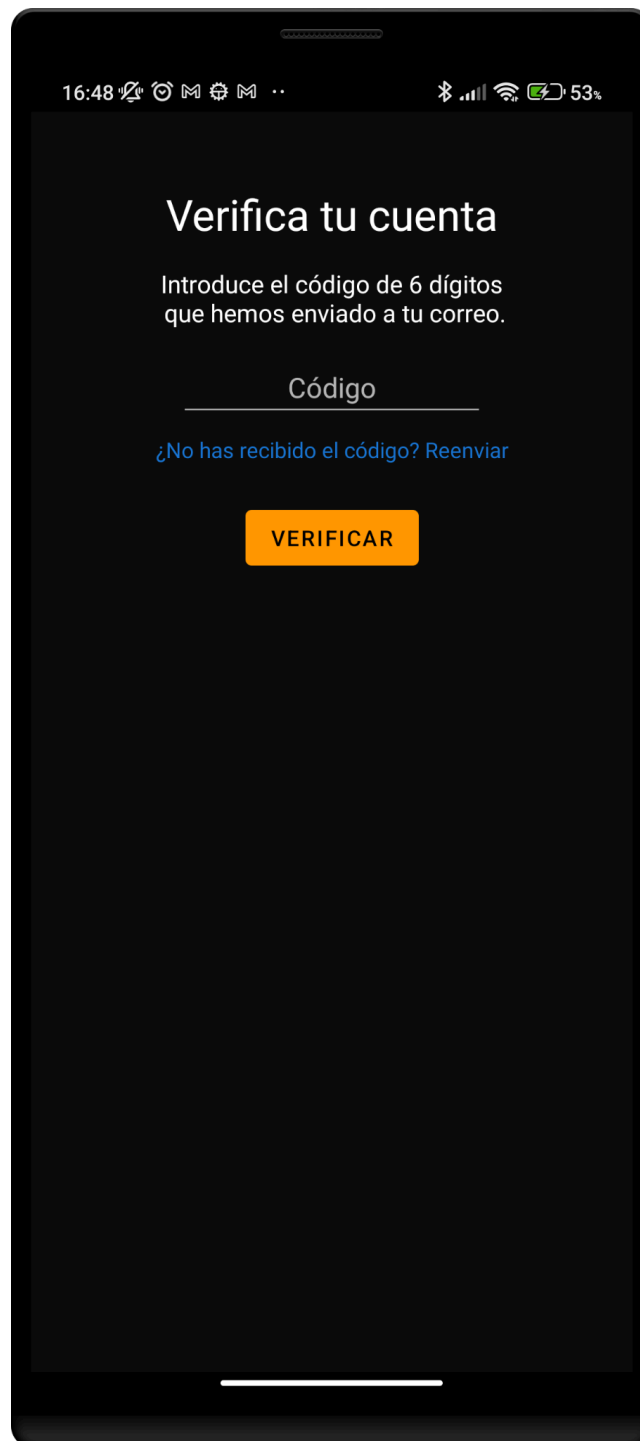


Pasaremos a la primera pantalla de registro donde deberemos de rellenar los campos con el correo y contraseña que vayamos a usar en la aplicación. Una vez introducidos los datos y si estos coinciden, pulsaremos el botón **"SIGUIENTE"**.

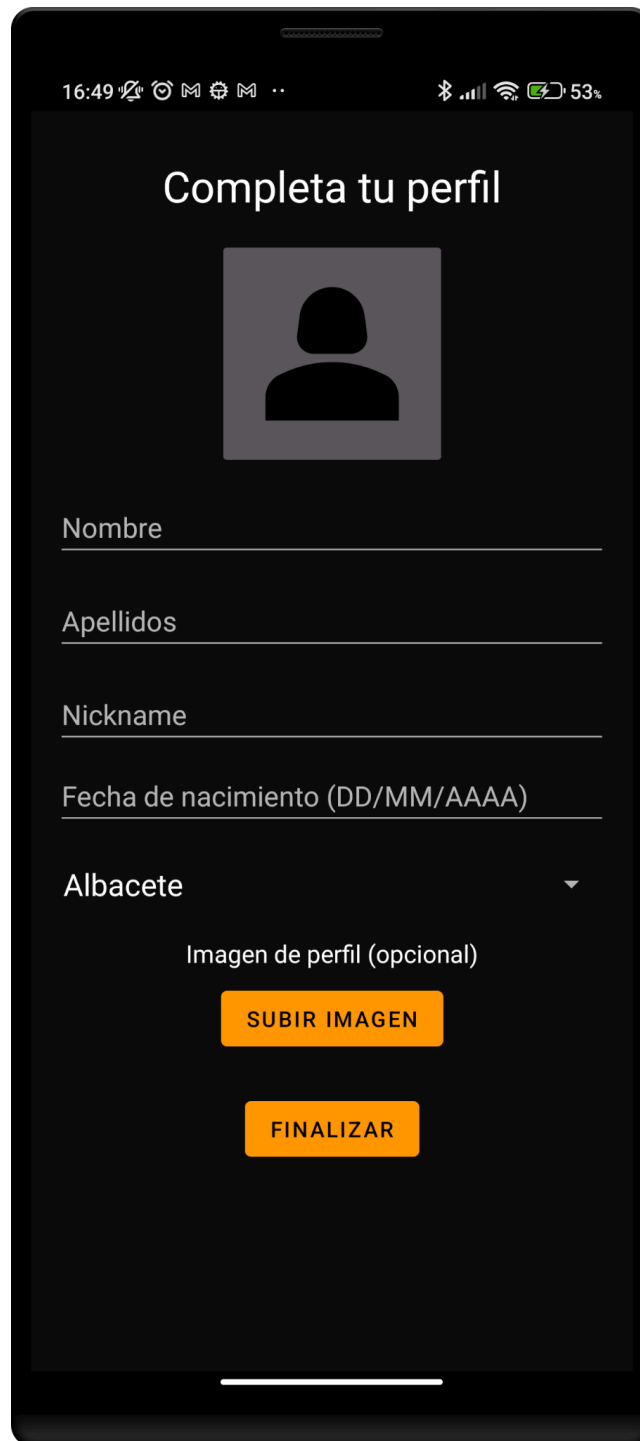


The screenshot shows a mobile application interface for creating an account. The screen has a dark background. At the top, there is a status bar with the time 16:47, various notification icons, and a battery level of 53%. The main heading is "Crear cuenta" in white text. Below the heading, there are four input fields. The first two fields are for email, both containing the text "correo@example.com". The next two fields are for a password, both containing a series of dots. A red underline is visible under the second password field. At the bottom of the form, there is a red button with the text "SIGUIENTE" in white capital letters.

Al hacer esto el sistema nos enviará un código de confirmación al correo introducido para comprobar que somos nosotros. Para continuar introducimos el código en el campo y le damos al botón de **"VERIFICAR"**.




Lo siguiente que nos saldrá, es un formulario para completar los datos de nuestro usuario. Una vez que estemos satisfechos con la información de nuestro perfil, pulsaremos sobre el botón **"FINALIZAR"** para terminar de crear nuestra cuenta.

A smartphone screen displaying a profile completion form. The status bar at the top shows the time 16:49, various notification icons, and a battery level of 53%. The form has a dark background with white text. At the top, it says "Completa tu perfil" above a placeholder for a profile picture. Below this are input fields for "Nombre", "Apellidos", "Nickname", and "Fecha de nacimiento (DD/MM/AAAA)". A dropdown menu is set to "Albacete". At the bottom, there is a label "Imagen de perfil (opcional)" followed by two orange buttons: "SUBIR IMAGEN" and "FINALIZAR".

16:49 53%

Completa tu perfil



Nombre

Apellidos

Nickname

Fecha de nacimiento (DD/MM/AAAA)

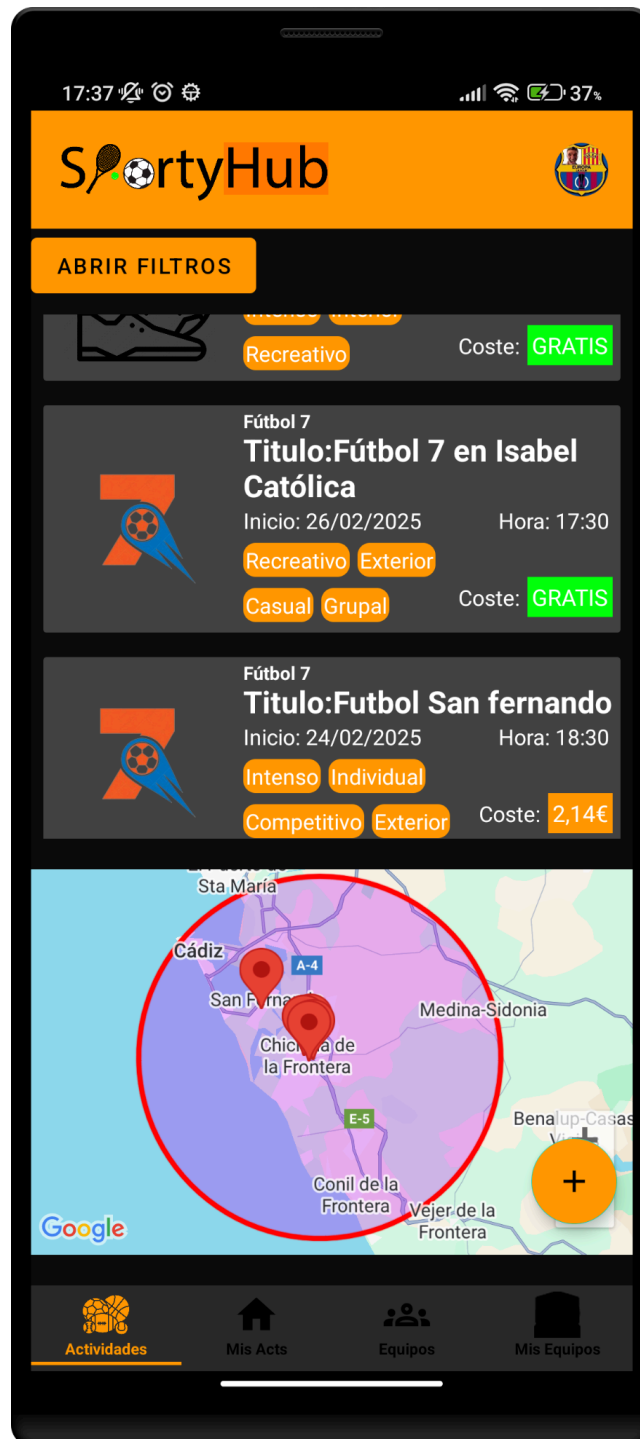
Albacete ▼

Imagen de perfil (opcional)

SUBIR IMAGEN

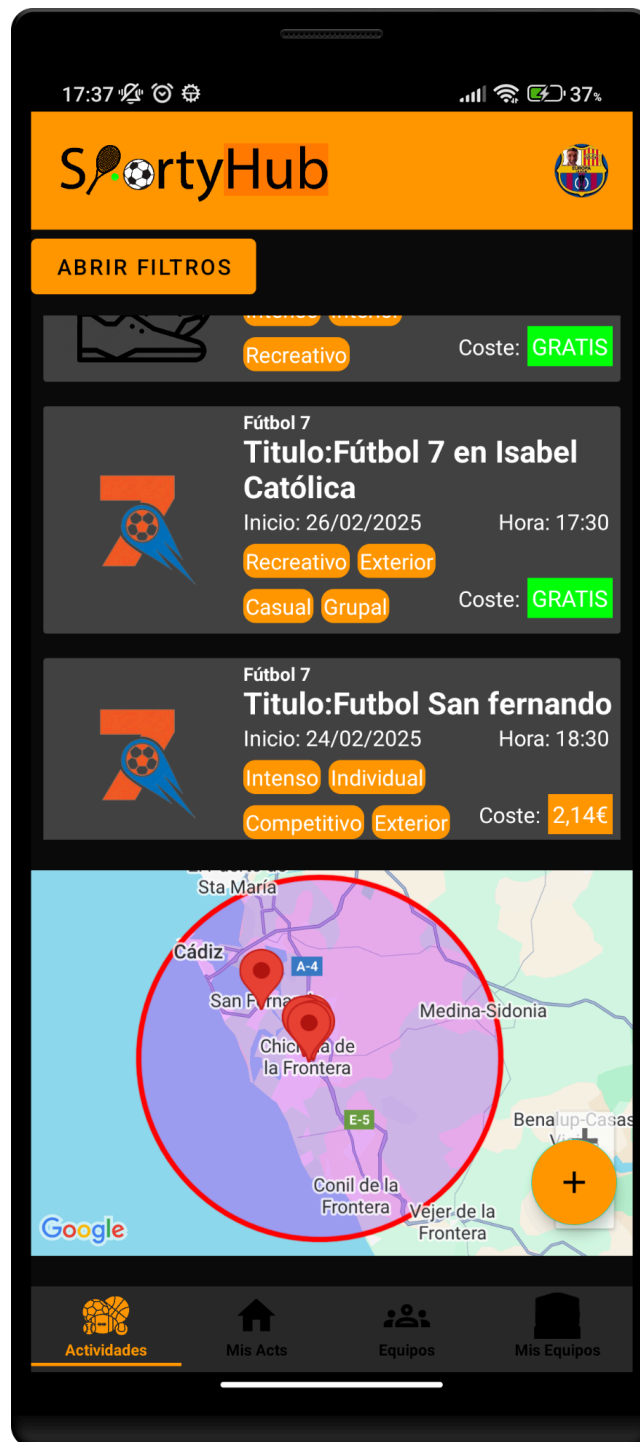
FINALIZAR

Al terminar de crear nuestra cuenta, el sistema iniciará sesión por nosotros de manera automática y nos llevará a la pantalla principal. A partir de aquí ya estás listo para poder usar todas las funciones de la aplicación.



B) Crear actividad:

Para crear una actividad, vamos a partir de la pantalla principal.



En esta pantalla deberemos de pulsar el botón naranja con el símbolo “+” que se encuentra en la parte inferior derecha de la pantalla.

Cuando pulsemos el botón se nos abrirá esta nueva pantalla con un formulario que rellenaremos con la información que queremos que tenga nuestra actividad. Cabe destacar que todos los datos son obligatorios de rellenar, excepto la descripción y la selección de etiquetas.

18:48 100% 45%

Crear Nueva Actividad

Titulo de la actividad

Titulo

Deporte

Fútbol 7

Selecciona etiquetas

Casual ★ Intenso 🔥

Recreativo 😊 Competitivo 🏆

Individual 👤 Grupal 👥

Interior 🏠 Exterior 🌳

Número de Participantes (Min 14, Máx 24)

14

Fecha Hora

Duración 30 Precio

Una vez que estemos satisfechos con la información puesta en el formulario, le daremos al botón **“CREAR”**.

The screenshot shows a mobile application interface for creating an event. At the top, there are two tabs: "Recreativo" (selected) and "Competitivo". Below these are four buttons: "Individual" (selected), "Grupal", "Interior" (selected), and "Exterior". A text field for "Número de Participantes (Min 14, Máx 24)" contains the value "14". Below this are two input fields for "Fecha" and "Hora". Further down are a "Duración" field set to "30" and a "Precio" field. A slider control is positioned below the duration field. The "Ubicación" section includes a search bar with the placeholder "Buscar un sitio" and a map view showing a location marked with a red pin. The map includes labels for "na de ntera", "EDRERA", and "A-9034". At the bottom of the form is a large orange button labeled "CREAR".

18:48 45%

Recreativo Competitivo

Individual Grupal

Interior Exterior

Número de Participantes (Min 14, Máx 24)

14

Fecha Hora

Duración 30 Precio

Ubicación

Buscar un sitio

na de ntera

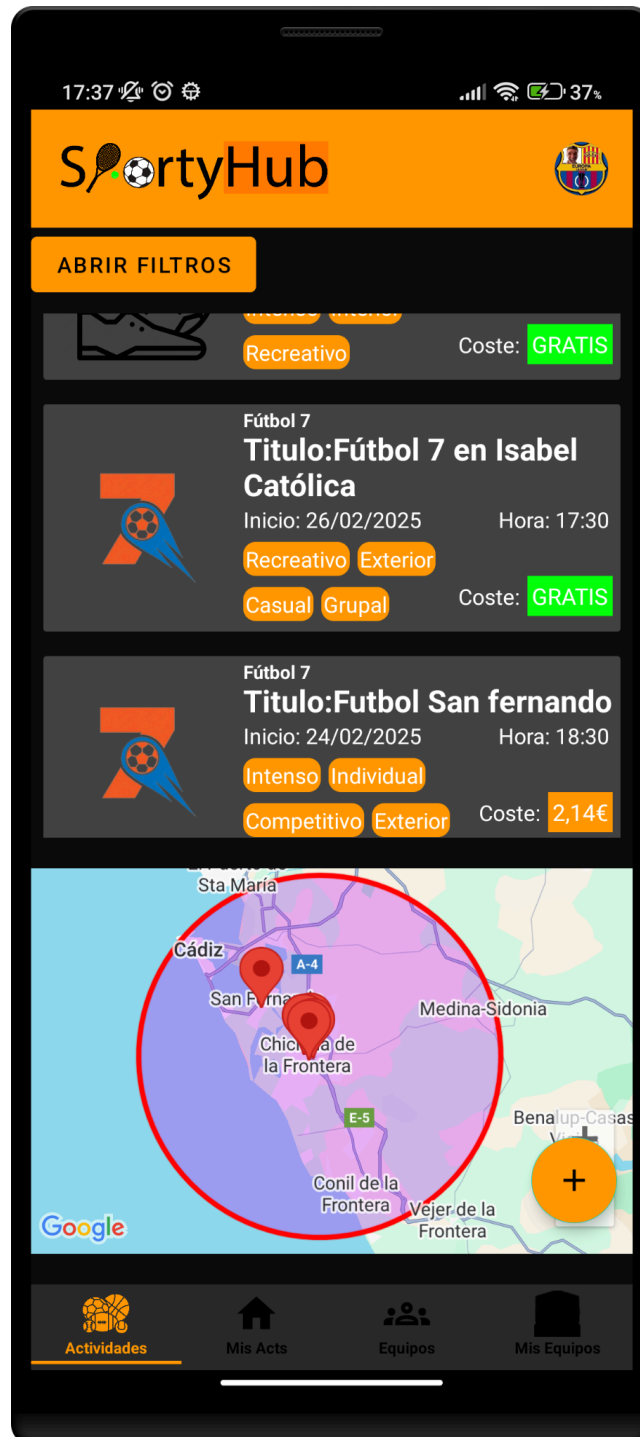
EDRERA

A-9034

Google

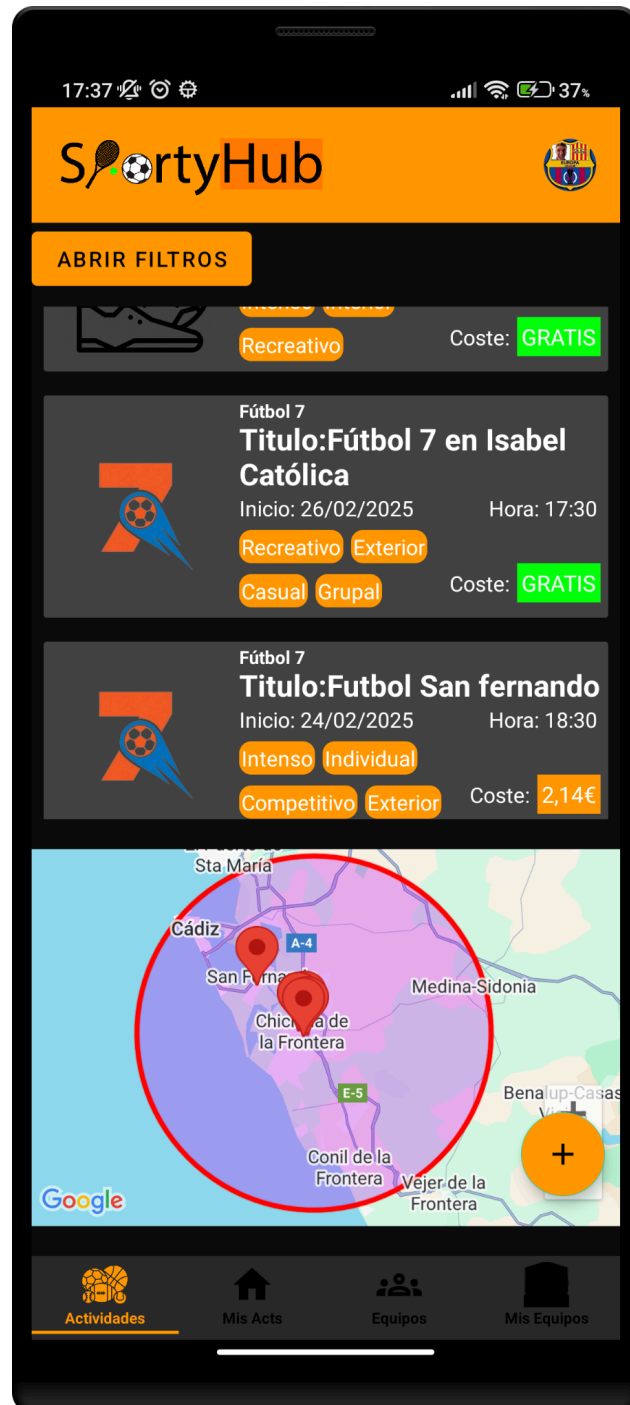
CREAR

Si todos los datos están correctos y no hay ningún problema, el sistema nos devolverá a la pantalla principal, donde ahora nuestra actividad creada estará disponible para que el resto de usuarios pueda verla y unirse. Las actividades creadas también aparecerán en el apartado con el icono de la casa, llamado mis actividades (“**Mis Acts**”).



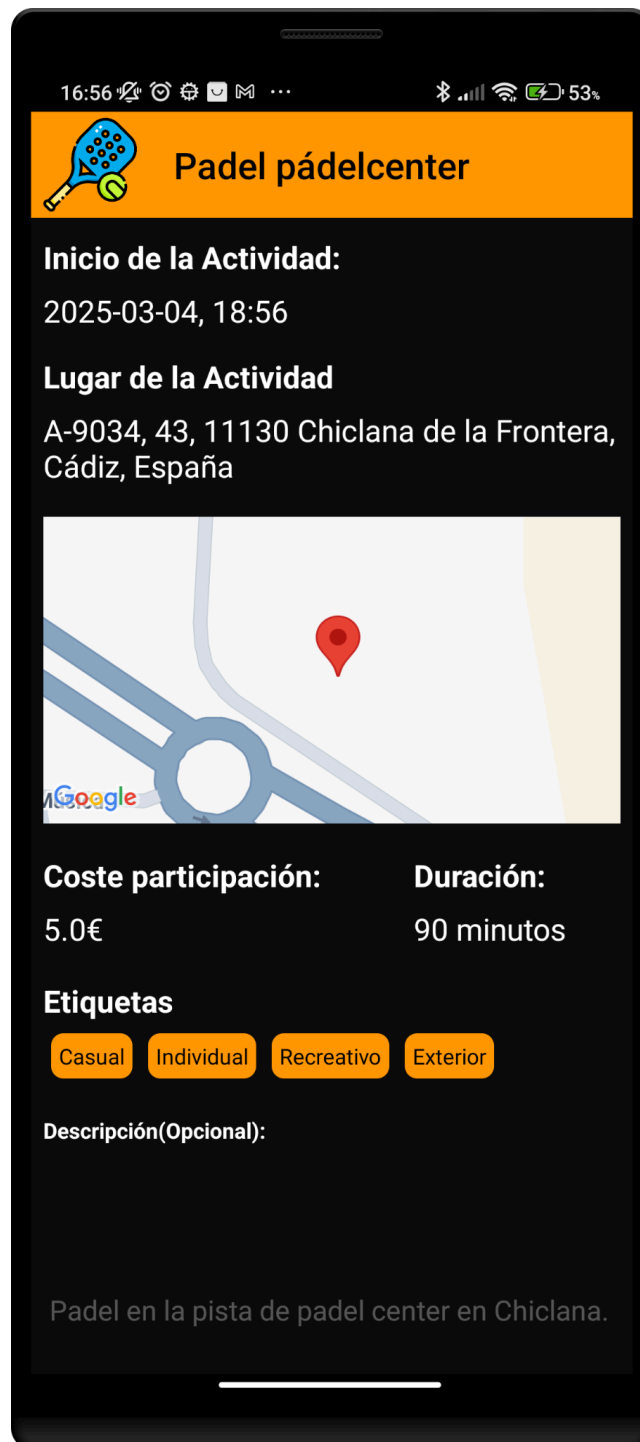
C) Unirse a una actividad:

Para poder unirnos a la actividad que otro usuario haya creado, primero en la pantalla principal, miraremos entre las diferentes actividades que tenemos en la lista o bien en el mapa de abajo. Esta lista puede reducirse aplicando filtros a la búsqueda pulsando sobre la el botón **“ABRIR FILTROS”** y aplicando las especificaciones deseadas.

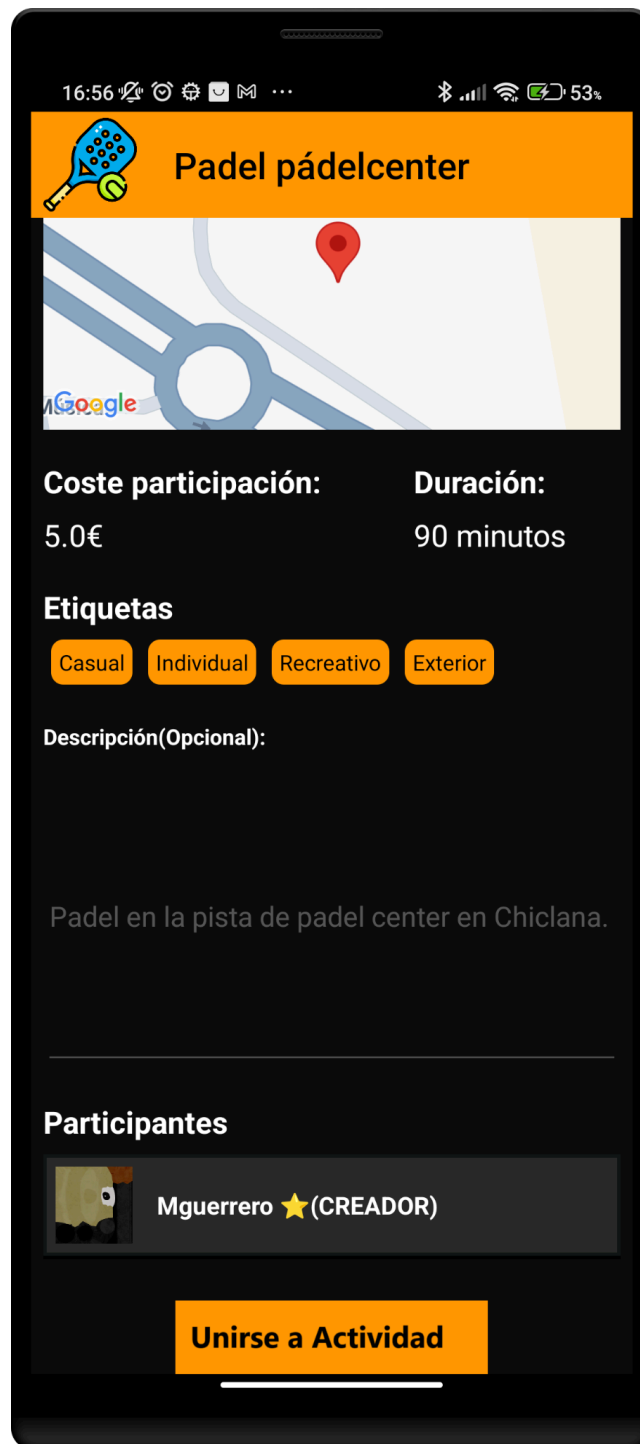


Una vez decidida, pulsamos sobre la actividad o sobre su marcador en el mapa.

Una vez seleccionamos una actividad, el sistema nos llevará a otra pantalla donde nos mostrará la información en detalle de la actividad que hemos seleccionado.



Si la actividad nos conviene para unirnos, simplemente tendremos que darle al botón que pone **unirse** o **pagar**, este botón varía dependiendo de si la actividad tiene un coste por participar o no. En este caso la actividad es de pago por lo que nos pondrá “**Pagar con PayPal**”



En caso de tener que pagar, al pulsar el botón el sistema nos pedirá nuestra cuenta de PayPal para poder realizar el pago. Una vez pagado tendrá su resumen de pago.

Hecho esto ya estaremos unidos a la actividad, y solo quedaría asistir a la actividad en cuestión. En caso de que la actividad se cancelase, el sistema notificará al usuario y realizará la devolución del coste de la participación si lo hubiese. Además ahora esta actividad saldrá en el apartado de mis actividades, junto al resto de actividades que haya creado o estemos participando.



Y con esto habremos cubierto como utilizar las funciones principales de la app.

10.CONCLUSIONES

10.1. Comparación del resultado con la idea inicial

La aplicación resultante es muy diferente en cuanto a la idea inicial, para contrastar vamos a listar las similitudes y las diferencias con la idea inicial.

- **Similitudes con la idea inicial:**

- Se ha logrado desarrollar una aplicación funcional para organizar quedadas deportivas.
- Se han implementado las funciones principales como la creación de actividades, la gestión de equipos y la inscripción a las actividades con pago.
- Se ha logrado que los usuarios y los equipos tengan personalización gracias a poder subir y bajar las imágenes de la nube con Cloudinary.
- Se ha creado un mapa con marcadores que ponen a las actividades sus ubicaciones correspondientes, esto junto al apartado de filtros hace que la experiencia de búsqueda del usuario sea mucho mejor,

- **Diferencias en el camino:**

- **Registro por pasos:** Al principio no se había pensado el hecho de realizar el registro por pasos. Finalmente se decidió implementar para mejorar la seguridad y experiencia del usuario.
- **Sistema de pagos:** El enfoque inicial no retenía los pagos de los participantes hacía los creadores de la actividades. Se decidió implantar para evitar los casos de estafa.
- **Limitaciones:** Algunas funcionalidades como el chat de equipo y que los líderes de los equipos pudiesen unir a los miembros no se implementaron finalmente en esta versión, pero podrían añadirse más adelante.

10.2. Posibles mejoras futuras

Debido a las limitaciones algunas ideas iniciales y otras que han surgido durante el desarrollo no han podido llevarse a cambio. Entre estas ideas tenemos:

- **Implementación de chat en equipos y actividades** para mejorar la comunicación entre los miembros del equipo y los participantes de las actividades pudiendo así mejorar la organización.
- **Diversidad en el sistema de pagos**, considerando otras pasarelas además de PayPal, como Google Pay o Bizum por ejemplo.
- **Actividades recurrentes**, para los usuarios que quieren realizar una actividad con un día y una fija, semana tras semana.
- **Sección de torneos**, para que los usuarios pudiesen crear y participar con sus equipo, para hacer la app mas interesante y satisfacer a los usuarios más competitivos
- **Pruebas automatizadas** para mejorar la estabilidad y detectar errores antes del lanzamiento.

11. ÍNDICE DE TABLAS E IMÁGENES

Tablas:

1. Tabla 1: Crear Actividad - Caso de Uso	Página 10
2. Tabla 2: Unirse a Actividad - Caso de Uso	Página 11
3. Tabla 3: Reportar Actividad - Caso de Uso	Página 12
4. Tabla 4: Mostrar Notificaciones - Caso de Uso	Página 13
5. Tabla 5: Eliminar Usuario - Caso de Uso	Página 14
6. Tabla 6: Registrarse - Caso de Uso	Página 15
7. Tabla 7: Enviar Advertencia - Caso de Uso	Página 16

Imágenes:

1. Imagen 1: Wireframe de la Aplicación	Página 7
2. Imagen 2: Diagrama Entidad-Relación	Página 8
3. Imagen 3: Diagrama de Casos de Uso	Página 9
4. Imagen 4: Diagrama de Flujo Básico	Página 17
5. Imagen 5: Diseño inicial de la aplicación.....	Página 18
6. Imagen 6: Demostración de uso API Google Maps.....	Página 20
7. Imagen 7: Uso de Postman.....	Página 21
8. Imagen 8: Demostración de uso de Render.....	Página 21
9. Imagen 9: Demostración de uso API Google Maps.....	Página 20
10. Imagen 10: Demostración de uso API Google Maps.....	Página 20
11. Imagen 11: Pantalla inicial de la aplicación.....	Página 27
12. Imágenes 12-17: Manual como crear usuario.....	Págs 28-32
13. Imágenes 18-22: Manual como crear actividad.....	Págs 33-36
14. Imágenes 23-27: Manual como crear usuario.....	Págs 37-40

12. BIBLIOGRAFÍAS Y REFERENCIAS

1. **Google Maps API Documentation. Google Developers.**
<https://developers.google.com/maps/documentation>
2. **PayPal API Documentation. PayPal Developers.**
<https://developer.paypal.com/docs/api/overview/>
3. **Cloudinary API Documentation. Cloudinary Developers.**
<https://cloudinary.com/documentation>
4. **Spring Boot Reference Documentation.**
<https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/>
5. **PostgreSQL Documentation. PostgreSQL Global Development**
<https://www.postgresql.org/docs/>
6. **Render Documentation. Render Developers.**
<https://render.com/docs>