



**ULPGC**  
**Universidad de**  
**Las Palmas de**  
**Gran Canaria**

**eii**

**ESCUELA DE**  
**INGENIERÍA INFORMÁTICA**

**Trabajo de Fin de Grado**

---

**Diseño y Desarrollo de una Aplicación Móvil para  
la Creación y Gestión de Eventos como Red  
Social**

**TITULACIÓN:** Grado en Ingeniería Informática

**AUTOR:** Jorge Álamo Santana

---

**TUTORIZADO POR:**

Dr. Luis Hernández Acosta

Las Palmas de Gran Canaria

Julio de 2025

**SOLICITUD DE DEFENSA DE TRABAJO DE FIN DE TÍTULO**

D/D<sup>a</sup> Jorge Álamo Santana, autor/a del Trabajo de Fin de Título Diseño y Desarrollo de una Aplicación Móvil para la Creación y Gestión de Eventos como Red Social, correspondiente a la titulación Grado de Ingeniería Informática, plan 41.

**S O L I C I T A**

que se inicie el procedimiento de defensa del mismo, para lo que se adjunta la documentación requerida, haciendo constar que

☒ [ X ] se autoriza / ☐ [ ] no se autoriza, la grabación en audio de la exposición y turno de preguntas.

Asimismo, con respecto al registro de la propiedad intelectual/industrial del TFT, declara que:

☐ [ ] Se ha iniciado o hay intención de iniciarlo (defensa no pública).  
☒ [ X ] No está previsto.

Y para que así conste firma la presente. (fecha en firma electrónica)

El/la estudiante

Fdo.: \_\_\_\_\_

A rellenar y firmar **obligatoriamente** por el/la/los/las tutor/a/es/as.

En relación a la presente solicitud, se informa: (firmar donde corresponda)

**Positivamente** (en caso de detección de copia, esta firma quedará invalidada)

**Negativamente** (la justificación en caso de informe negativo deberá incluirse en el TFT05)

Fdo.:

Fdo.: \_\_\_\_\_

**DIRECTOR DE LA ESCUELA DE INGENIERÍA INFORMÁTICA**

## Agradecimientos

Quiero agradecer en primer lugar, a mis padres, por haberme apoyado incondicionalmente en todo momento, tanto en los buenos como en los malos. Gracias por confiar siempre en mí, por su paciencia y por ser mi mayor ejemplo de esfuerzo y constancia.

A mi hermana, por estar siempre a mi lado, animándome cuando más lo necesitaba y dándome fuerzas para seguir adelante.

A Nerea, por estar ahí en los momentos difíciles, por escucharme, entenderme y ayudarme en todo lo que estaba en su mano, siendo un pilar fundamental durante todo este proceso.

Y por supuesto, a mis amigos, por haberme dado la moral y la energía para continuar hacia adelante, pasase lo que pasase. Gracias por los ánimos, las risas y el apoyo que han hecho que este camino sea mucho más llevadero.

Y, por último, a mi tutor Luis, por su ayuda, dedicación y toda su atención en todo momento, guiándome durante este proyecto y facilitando que pudiese desarrollarlo de la mejor manera posible.

A todos ustedes, gracias de corazón. Este trabajo también es un reflejo de su apoyo y cariño.

# Índice general

|                                                                        |           |
|------------------------------------------------------------------------|-----------|
| <b>Resumen .....</b>                                                   | <b>8</b>  |
| <b>Abstract .....</b>                                                  | <b>9</b>  |
| <b>Capítulo 1: Introducción .....</b>                                  | <b>10</b> |
| <b>1.1. Contexto .....</b>                                             | <b>10</b> |
| <b>1.2. Estado actual .....</b>                                        | <b>10</b> |
| <b>1.2.1. Eventbrite .....</b>                                         | <b>10</b> |
| <b>1.2.2. TopEvent .....</b>                                           | <b>12</b> |
| <b>1.2.3. Meetup .....</b>                                             | <b>13</b> |
| <b>1.3. Motivación .....</b>                                           | <b>14</b> |
| <b>1.4. Objetivos .....</b>                                            | <b>15</b> |
| <b>1.5. Estructura de la memoria .....</b>                             | <b>16</b> |
| <b>Capítulo 2: Competencias específicas cubiertas .....</b>            | <b>18</b> |
| <b>Capítulo 3: Aportaciones y alineamiento con objetivos ODS .....</b> | <b>21</b> |
| <b>3.1. Aportaciones .....</b>                                         | <b>21</b> |
| <b>3.2. Alineamiento con objetivos ODS .....</b>                       | <b>21</b> |
| <b>Capítulo 4: Análisis y diseño .....</b>                             | <b>22</b> |
| <b>4.1. Metodología .....</b>                                          | <b>22</b> |
| <b>4.2. Planificación del proyecto .....</b>                           | <b>22</b> |
| <b>4.3. Tecnologías utilizadas .....</b>                               | <b>25</b> |
| <b>4.4. Requisitos .....</b>                                           | <b>26</b> |
| <b>4.4.1. Requisitos de usuario .....</b>                              | <b>26</b> |
| <b>4.4.2. Requisitos del sistema .....</b>                             | <b>28</b> |
| <b>4.5. Diseño .....</b>                                               | <b>29</b> |
| <b>4.5.1. Diseño de la interfaz y mockups .....</b>                    | <b>29</b> |
| <b>4.5.2. Diseño de la base de datos .....</b>                         | <b>42</b> |
| <b>Capítulo 5: Implementación .....</b>                                | <b>46</b> |
| <b>5.1. Estructura general del código .....</b>                        | <b>46</b> |
| <b>5.2. Librerías y dependencias .....</b>                             | <b>47</b> |
| <b>5.3. Base de datos .....</b>                                        | <b>48</b> |
| <b>5.4. Componentes y clases .....</b>                                 | <b>51</b> |
| <b>5.4.1. Componentes .....</b>                                        | <b>52</b> |
| <b>5.4.2. Páginas .....</b>                                            | <b>54</b> |
| <b>5.4.3. Servicios .....</b>                                          | <b>61</b> |
| <b>5.4.4. Guards .....</b>                                             | <b>63</b> |

|                                                          |           |
|----------------------------------------------------------|-----------|
| <b>5.5. Pruebas y validación .....</b>                   | <b>64</b> |
| <b>Capítulo 6: Guía de usuario .....</b>                 | <b>65</b> |
| <b>Capítulo 7: Conclusiones y trabajos futuros .....</b> | <b>79</b> |
| <b>7.1. Conclusiones .....</b>                           | <b>79</b> |
| <b>7.2. Trabajos futuros .....</b>                       | <b>80</b> |
| <b>Bibliografía .....</b>                                | <b>81</b> |

## Índice de ilustraciones

|                                                                                                                         |           |
|-------------------------------------------------------------------------------------------------------------------------|-----------|
| <b>Ilustración 1: Pantallas de navegación de la aplicación Eventbrite. Fuente: ExcelentWebWorld .....</b>               | <b>11</b> |
| <b>Ilustración 2: Pantallas de navegación de la aplicación TopEvent. Fuente: Google Play - TopEvent.....</b>            | <b>12</b> |
| <b>Ilustración 3: Pantallas de navegación de la aplicación Meetup. Fuente: TreceBits .....</b>                          | <b>13</b> |
| <b>Ilustración 4: Paleta de colores .....</b>                                                                           | <b>30</b> |
| <b>Ilustración 5: Logo de Evnet .....</b>                                                                               | <b>30</b> |
| <b>Ilustración 6: Formularios de inicio de sesión y registro .....</b>                                                  | <b>31</b> |
| <b>Ilustración 7: Diseños descartados de mi perfil.....</b>                                                             | <b>32</b> |
| <b>Ilustración 8: Diseño final de mi perfil.....</b>                                                                    | <b>32</b> |
| <b>Ilustración 9: Diseño de página de editar perfil .....</b>                                                           | <b>33</b> |
| <b>Ilustración 10: Diseños de perfil público y listas de seguidores/seguídos .....</b>                                  | <b>34</b> |
| <b>Ilustración 11: Diseño de creación de un nuevo evento.....</b>                                                       | <b>35</b> |
| <b>Ilustración 12: Diseño de la página principal .....</b>                                                              | <b>36</b> |
| <b>Ilustración 13: Diseño de post del evento.....</b>                                                                   | <b>37</b> |
| <b>Ilustración 14: Diseño de evento .....</b>                                                                           | <b>38</b> |
| <b>Ilustración 15: Diseño de más información del evento descartado .....</b>                                            | <b>38</b> |
| <b>Ilustración 16: Diseño de más información del evento .....</b>                                                       | <b>39</b> |
| <b>Ilustración 17: Diseños de chat, album, ajustes y editar detalles de un evento .....</b>                             | <b>40</b> |
| <b>Ilustración 18: Diseño de mis eventos.....</b>                                                                       | <b>41</b> |
| <b>Ilustración 19: Diseño de buscar eventos.....</b>                                                                    | <b>42</b> |
| <b>Ilustración 20: Diagrama entidad-relación de la base de datos. Fuente: Mermaid.....</b>                              | <b>44</b> |
| <b>Ilustración 21: Página de inicio de sesión (login) .....</b>                                                         | <b>66</b> |
| <b>Ilustración 22: Página de registro (signup).....</b>                                                                 | <b>67</b> |
| <b>Ilustración 23: Página principal (home).....</b>                                                                     | <b>68</b> |
| <b>Ilustración 24: Página de notificaciones (notifications) .....</b>                                                   | <b>68</b> |
| <b>Ilustración 25: Página de evento (event).....</b>                                                                    | <b>69</b> |
| <b>Ilustración 26: Sección de lista de la compra de la página de más información del evento (event more info) .....</b> | <b>70</b> |
| <b>Ilustración 27: Sección de participantes de la página de más información del evento (event more info).....</b>       | <b>70</b> |
| <b>Ilustración 28: Página de añadir participantes (add participants) .....</b>                                          | <b>71</b> |
| <b>Ilustración 29: Página de mi perfil (my profile) .....</b>                                                           | <b>72</b> |
| <b>Ilustración 30: Sección de seguidores o seguidos.....</b>                                                            | <b>72</b> |
| <b>Ilustración 31: Página de editar perfil (edit profile) .....</b>                                                     | <b>73</b> |
| <b>Ilustración 32: Desplegable de cambiar contraseña .....</b>                                                          | <b>73</b> |
| <b>Ilustración 33: Perfil de usuario (profile) .....</b>                                                                | <b>74</b> |
| <b>Ilustración 34: Página de crear nuevo evento, primera etapa (create event).....</b>                                  | <b>75</b> |
| <b>Ilustración 35: Página de crear nuevo evento, segunda etapa (create event shopping list) ..</b>                      | <b>76</b> |
| <b>Ilustración 36: Página de crear nuevo evento, tercera etapa (create event participants) ..</b>                       | <b>77</b> |
| <b>Ilustración 37: Página del buscador de eventos (search) .....</b>                                                    | <b>78</b> |
| <b>Ilustración 38: Página de mis eventos (my events) .....</b>                                                          | <b>78</b> |

# Índice de tablas

**Tabla 1: Planificación inicial del proyecto ..... 24**

**Tabla 2: Tabla users de la base de datos ..... 49**

**Tabla 3: Tabla events de la base de datos ..... 50**

**Tabla 4: Tabla notifications de la base de datos ..... 51**

## Resumen

Evnet es una plataforma móvil para la creación y gestión de eventos, planteada como una red social. Permite a los usuarios publicar eventos públicos o privados, añadir detalles (descripción, fecha, lugar, temática, etc.) y crear listas de la compra con división automática de gastos entre los participantes. Se pueden invitar participantes mediante enlaces únicos o desde la propia app. Cada evento se clasifica por categoría, y los usuarios pueden ver los eventos públicos de quienes siguen. Desarrollada con Ionic, Angular y Supabase, la plataforma cuenta con una interfaz adaptable a dispositivos móviles. El proyecto se ha desarrollado siguiendo la metodología ágil SCRUM, con actividades de análisis, diseño, implementación, pruebas y documentación a lo largo de los distintos sprints.

## Abstract

Evnet is a mobile platform for creating and managing events, designed as a social network. It allows users to publish public or private events, add details (description, date, location, theme, etc.) and create shopping lists with automatic expense splitting among participants. They can be invited via unique links or directly from the app. Each event is categorized, and users can view public events from those they follow. Developed with Ionic, Angular, and Supabase, the platform features a responsive interface adaptable to mobile devices. The project was developed following the agile SCRUM methodology, with activities including analysis, design, implementation, testing, and documentation carried out across different sprints.

# Capítulo 1: Introducción

## 1.1. Contexto

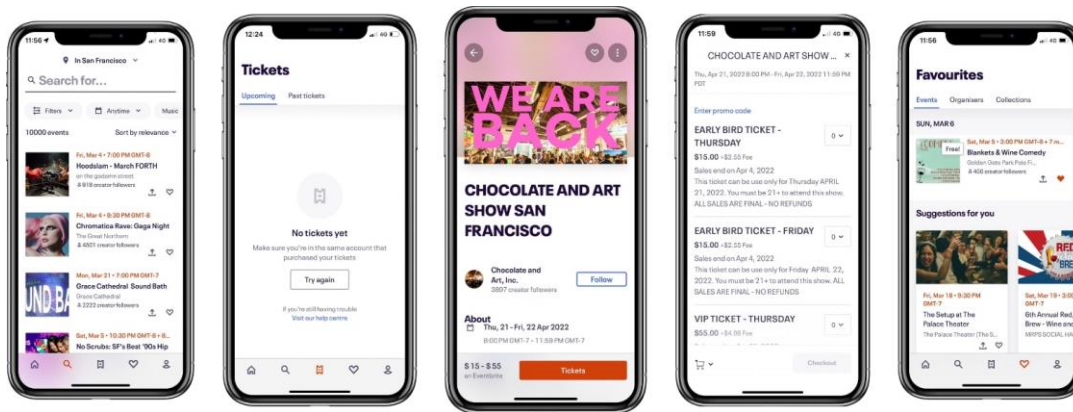
En las últimas décadas, la organización y participación en eventos (ya sean encuentros informales, reuniones de grupo o grandes espectáculos públicos) ha experimentado una transformación significativa gracias al auge de las tecnologías móviles y las redes sociales. Plataformas consolidadas han facilitado la creación, promoción y gestión de eventos, agilizando procesos de inscripción, venta de entradas y difusión. No obstante, estas herramientas suelen presentar limitaciones en cuanto a la interacción colaborativa entre participantes y la gestión integrada de aspectos logísticos y económicos. Además, ninguna de estas aplicaciones está concebida para funcionar como una red social fácil de usar: sus funcionalidades sociales suelen ser parciales, no intuitivas o secundarias, lo que dificulta la construcción de comunidades activas y el intercambio fluido de contenidos entre los usuarios.

## 1.2. Estado actual

Con el fin de contextualizar adecuadamente el desarrollo de *Evnet*, se ha llevado a cabo un análisis detallado de tres plataformas actuales que representan diferentes enfoques en la organización de eventos. Este análisis permite identificar sus funcionalidades, carencias y enfoques sociales, con el propósito de comprender el panorama actual y detectar oportunidades de mejora.

### 1.2.1. Eventbrite

*Eventbrite* es un servicio consolidado centrado en la venta de entradas y el registro de asistentes para eventos de diferentes temáticas, contando con una interfaz sencilla para crear y personalizar páginas de eventos. Sus pantallas de navegación se pueden observar en la Ilustración 1.



**Ilustración 1: Pantallas de navegación de la aplicación Eventbrite. Fuente: [ExcelentWebWorld](#)**

Ofrece las siguientes características:

- Personalización de páginas de eventos con imágenes, descripciones y categorías.
- Procesamiento de pagos integrado y control de entradas mediante códigos QR.
- Herramientas de marketing y analítica para envío de correos, promoción en redes sociales y generación de informes.
- Visibilidad pública y comunidad amplia, gracias a su gran base de usuarios.
- Funcionalidad social básica, como la opción de seguir a usuarios, aunque esta característica está poco integrada y resulta poco intuitiva.

La aplicación presenta las siguientes limitaciones:

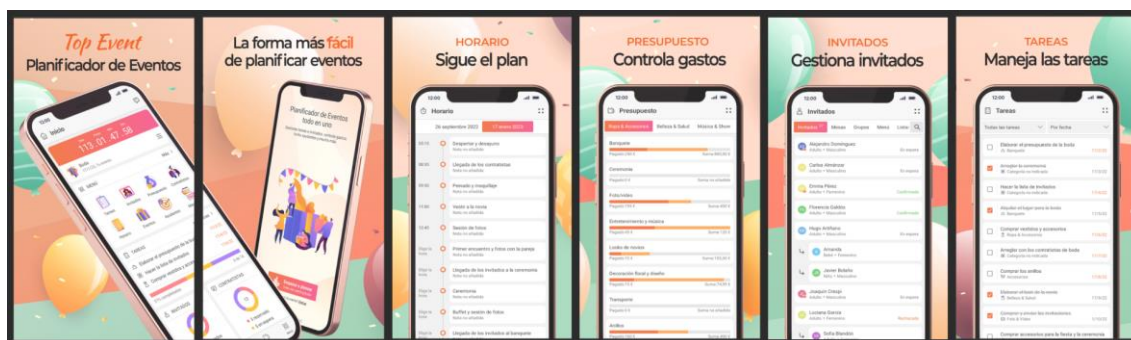
- Separación de funcionalidades: en dispositivos móviles, las funcionalidades de unirse a eventos y crear eventos se separan en dos aplicaciones diferentes.
- Poca interacción de los participantes: los eventos están planteados para únicamente informarte del mismo y adquirir las entradas, imposibilitando la interacción entre los asistentes con medios como chats de texto o álbumes de fotos.
- Costo: se aplican comisiones por venta de entradas, lo que puede impactar en el margen de beneficio en eventos pequeños.

Comparación con nuestra plataforma: nuestra plataforma se diferencia al incluir, además de las funciones básicas de registro y venta de entradas, herramientas internas para la gestión de gastos compartidos, listas de compra colaborativas y un *feed* personalizado donde los usuarios pueden ver los eventos publicados por las personas a las que siguen. Además, podrán explorar todos los eventos públicos creados por cualquier usuario directamente desde la aplicación, favoreciendo así la conexión y el descubrimiento dentro

de la red social interna. Esto la posiciona como una solución más completa y social, no solamente centrada en la venta de entradas.

### 1.2.2. TopEvent

*TopEvent* está enfocado en la planificación de eventos sociales como bodas, cumpleaños, aniversarios, fiestas y otros eventos similares, presentando una solución integral para gestionar todos los aspectos organizativos de este tipo de celebraciones. En la Ilustración 2 se muestran sus pantallas de navegación.



*Ilustración 2: Pantallas de navegación de la aplicación TopEvent. Fuente: [Google Play - TopEvent](#)*

Ofrece las siguientes características:

- Gestión de tareas y presupuestos, con división de gastos y recordatorios.
- Organización de invitados por grupos y comunicación básica.
- Agenda y cronograma detallado para el día del evento.
- Control de proveedores y estadísticas generales.
- Sincronización en tiempo real entre usuarios y dispositivos.
- Cuenta regresiva y widgets personalizados.

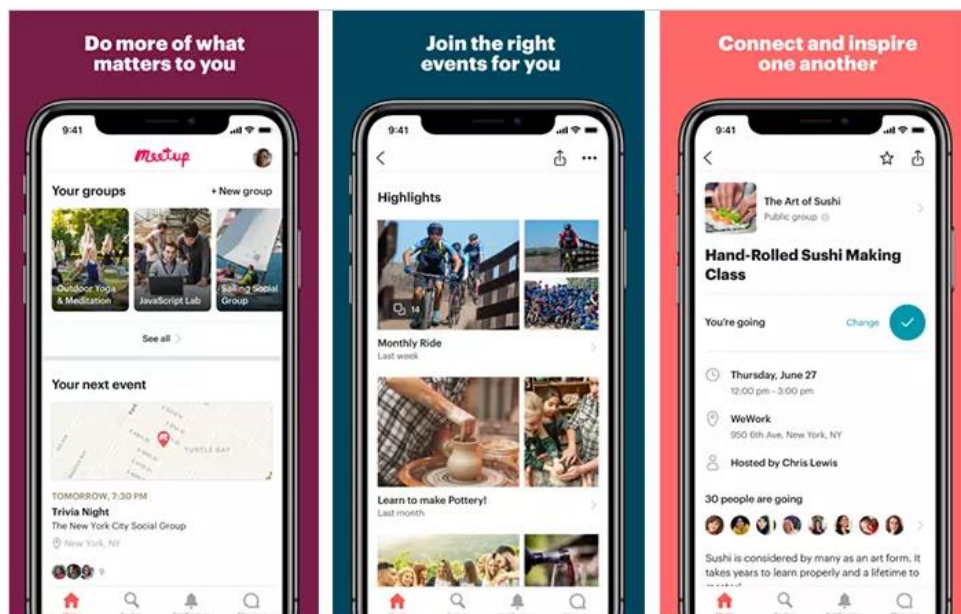
*TopEvent* presenta las siguientes limitaciones:

- Enfoque específico: Al estar principalmente orientada a eventos sociales y bodas, complica mucho su uso para otro tipo de eventos más sencillos y no tan elaborados.
- Complejidad: La aplicación es muy completa y te proporciona herramientas para controlar todos los elementos de un evento, pero eso hace que sea muy compleja y poco intuitiva para los usuarios.
- Poca interactividad social: No está enfocada en interactuar con otros usuarios, por lo que no tienes la posibilidad de buscar o seguir eventos.

Comparación con nuestra plataforma: mientras *TopEvent* se especializa en eventos personales complejos, nuestra plataforma busca combinar la planificación logística con funciones sociales y colaborativas. Así, podrá adaptarse tanto a reuniones informales como a grandes encuentros públicos, ofreciendo además exploración de eventos y comunidad integrada.

### 1.2.3. Meetup

*Meetup* se centra en la organización y promoción de reuniones y eventos dirigidos a comunidades con intereses comunes, facilitando la conexión entre personas a través de grupos temáticos. Podemos ver las pantallas de navegación en la Ilustración 3.



**Ilustración 3: Pantallas de navegación de la aplicación Meetup. Fuente: [TreceBits](#)**

Ofrece las siguientes características:

- Creación de grupos y eventos temáticos, permitiendo a los usuarios formar comunidades basadas en intereses específicos.
- Proporciona mecanismos para gestionar la asistencia, enviar notificaciones y mantener informados a los miembros sobre novedades o cambios en los eventos.
- Facilita que los usuarios descubran nuevas comunidades y eventos afines, ampliando sus redes de contacto mediante recomendaciones y sugerencias personalizadas.

- Muestra la ubicación geográfica de los eventos, ayudando a los usuarios a encontrar encuentros cercanos y planificar su asistencia con mayor facilidad.

*Meetup* presenta las siguientes limitaciones:

- Coste elevado para organizadores: En los últimos años, las tarifas para gestionar grupos y eventos han aumentado, lo que puede ser una barrera para pequeñas comunidades o eventos gratuitos.
- Personalización y gestión logística limitada: La plataforma se enfoca en la conexión social y comunitaria, pero carece de herramientas avanzadas para gestionar aspectos operativos complejos, como la división de gastos o la administración de listas colaborativas de compra.

Comparación con nuestra plataforma: nuestra plataforma aspira a combinar lo mejor de ambos mundos. Por un lado, incorpora funcionalidades sociales similares a *Meetup*, tales como la posibilidad de seguir a otros usuarios, explorar eventos públicos y descubrir nuevas comunidades. Por otro, añade herramientas específicas para la gestión detallada de eventos, como la división automática de gastos, listas de compra colaborativas y soporte para eventos tanto privados como de gran escala.

### **1.3. Motivación**

En el panorama actual, resulta difícil encontrar una aplicación que funcione verdaderamente como una red social centrada en eventos, donde los usuarios puedan crear, descubrir y unirse a actividades de manera cómoda, intuitiva y sin complicaciones. La mayoría de las plataformas existentes tienden a centrarse en aspectos concretos, como la venta de entradas o la planificación logística, dejando de lado la interacción social y la sencillez de uso que caracteriza a las redes sociales modernas.

Además, las funciones sociales que sí están presentes suelen estar mal integradas o relegadas a un segundo plano. No se promueve la interacción constante entre usuarios ni la construcción de una comunidad en torno a los eventos, lo que dificulta tanto la visibilidad de nuevas actividades como la participación espontánea. El proceso para descubrir eventos cercanos o relacionados con los intereses del usuario a menudo es poco fluido o requiere múltiples pasos.

A esto se suma que, a la hora de organizar un evento, los usuarios suelen recurrir a una combinación de herramientas externas: aplicaciones de mensajería para comunicarse con

los asistentes, hojas de cálculo para dividir gastos, listas compartidas para gestionar compras, plataformas específicas para enviar invitaciones o vender entradas, y redes sociales para promocionarlo. Esta fragmentación genera una experiencia poco integrada, que añade complejidad innecesaria a algo que debería ser simple y accesible.

*Evnet* surge como respuesta a esta carencia, con la intención de convertirse en una plataforma social centrada en los eventos, donde los usuarios puedan seguir a otros, ver de forma clara todos los eventos públicos que han creado y recibir recomendaciones basadas en su red de contactos. Esta propuesta busca simplificar el acceso a los eventos y fomentar la participación, combinando lo mejor de una red social con las herramientas necesarias para gestionar y disfrutar de cada encuentro, todo en un mismo lugar, sin perder de vista la facilidad de uso y la experiencia intuitiva que los usuarios actuales demandan.

## **1.4. Objetivos**

El objetivo principal de este Trabajo de Fin de Grado es desarrollar una plataforma digital orientada a la creación y gestión de eventos, que permita tanto organizar eventos privados como públicos, todo ello en un entorno social accesible e intuitivo. La aplicación buscará consolidarse como una red social especializada en eventos, facilitando que los usuarios puedan crear, descubrir y unirse a eventos desde una misma interfaz. Para lograrlo, se plantea el diseño de una arquitectura modular y escalable que garantice el correcto funcionamiento de la plataforma, así como su mantenimiento y evolución futura. La experiencia de usuario se optimizará mediante una interfaz accesible, adaptada a dispositivos móviles, y centrada en la simplicidad y la comodidad de uso.

La plataforma integrará herramientas colaborativas que simplifiquen la planificación del evento, como la creación de listas de la compra compartidas, la división automática de gastos y un sistema de chat integrado que fomente la comunicación en tiempo real. También se incluirán funcionalidades sociales que permitan seguir a otros usuarios, visualizar eventos públicos organizados por personas seguidas, y acceder a perfiles donde se recopilan todos los eventos creados por un usuario. De esta manera, la aplicación se diferenciará por combinar planificación logística, gestión económica y una red social interna, lo que permitirá cubrir todas las fases de un evento sin necesidad de utilizar múltiples herramientas externas.

## 1.5. Estructura de la memoria

Para facilitar la lectura y comprensión de este Trabajo de Fin de Grado, la memoria se ha organizado en los siguientes capítulos:

### Capítulo 1. Introducción

Se contextualiza la necesidad de *Evnet* a raíz de la evolución de las tecnologías móviles y las redes sociales en la organización de eventos. Se analiza el estado actual de plataformas consolidadas (*Eventbrite*, *TopEvent* y *Meetup*), se expone la motivación del proyecto y se definen los objetivos generales.

### Capítulo 2. Competencias específicas cubiertas

Se justifica cómo el desarrollo de *Evnet* permite ejercitar y demostrar las competencias del Grado en Ingeniería Informática, desde la planificación y diseño de sistemas hasta la implementación de interfaces accesibles y seguras.

### Capítulo 3. Aportaciones y alineamiento con objetivos ODS

Se describen las contribuciones socio-económicas, técnicas y científicas de la plataforma, así como su correspondencia con los Objetivos de Desarrollo Sostenible (ODS).

### Capítulo 4. Análisis y diseño

Aquí se detalla la metodología ágil SCRUM y la planificación del proyecto, se presentan las tecnologías y herramientas seleccionadas, se recopilan los requisitos de usuario y de sistema, y se muestran los diseños de interfaz (*mockups*) y el modelo de datos (diagrama entidad-relación).

### Capítulo 5. Implementación

Se explica la organización del código en Ionic y Angular, las principales librerías y dependencias utilizadas, la estructura de la base de datos en Supabase, la descripción de los componentes, páginas, servicios y *guards*, y la estrategia de pruebas y validación que garantiza la calidad del sistema.

### Capítulo 6. Guía de usuario

Se ofrece un recorrido detallado por la aplicación desde la perspectiva del usuario final, con capturas de pantalla e indicaciones paso a paso para el registro, navegación por el

*feed*, creación y gestión de eventos, exploración, perfil y demás funcionalidades disponibles.

## **Capítulo 7. Conclusiones y trabajos futuros**

Se recogen los principales logros y aprendizajes obtenidos, se evalúan los objetivos alcanzados y las limitaciones encontradas, y se plantean posibles líneas de mejora y ampliación funcional para *Evnet* en futuras iteraciones.

## **Bibliografía**

Se incluye la lista completa de referencias bibliográficas utilizadas para respaldar los conceptos, metodologías y tecnologías empleadas a lo largo del proyecto.

## Capítulo 2: Competencias específicas cubiertas

### CI1

*"Capacidad para diseñar, desarrollar, seleccionar y evaluar aplicaciones y sistemas informáticos, asegurando su fiabilidad, seguridad y calidad, conforme a principios éticos y a la legislación y normativa vigente."*

Esta competencia se refleja en el desarrollo de una aplicación fiable y segura, con autenticación, control de acceso y gestión adecuada de datos personales. Se aplican buenas prácticas de desarrollo para garantizar la calidad del software.

### CI2

*"Capacidad para planificar, concebir, desplegar y dirigir proyectos, servicios y sistemas informáticos en todos los ámbitos, liderando su puesta en marcha y su mejora continua y valorando su impacto económico y social."*

El TFG se planifica siguiendo la metodología SCRUM y contempla su impacto como herramienta social. Se diseña como una plataforma escalable y con potencial de uso real, liderando su desarrollo de principio a fin.

### CI3

*"Capacidad para comprender la importancia de la negociación, los hábitos de trabajo efectivos, el liderazgo y las habilidades de comunicación en todos los entornos de desarrollo de software."*

El proyecto requiere organización autónoma, toma de decisiones tecnológicas y diseño de flujos de interacción claros, demostrando habilidades de liderazgo y comunicación en el diseño y desarrollo de la solución.

### CI8

*"Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados."*

Se ha elegido un stack tecnológico eficiente (Ionic, Angular, Supabase), garantizando una arquitectura robusta y segura, con una implementación modular y mantenible.

## **CI12**

*"Conocimiento y aplicación de las características, funcionalidades y estructura de las bases de datos, que permitan su adecuado uso, y el diseño y el análisis e implementación de aplicaciones basadas en ellos."*

La aplicación usa una base de datos relacional en Supabase para gestionar usuarios, eventos y notificaciones. Se diseña su estructura con integridad, rendimiento y consistencia.

## **CI13**

*"Conocimiento y aplicación de las herramientas necesarias para el almacenamiento, procesamiento y acceso a los sistemas de información, incluidos los basados en web."*

Se aplican tecnologías modernas para el almacenamiento y acceso a datos en la nube. Se desarrollan servicios web eficientes para interacción en tiempo real, carga de imágenes y gestión de eventos.

## **CI16**

*"Conocimiento y aplicación de los principios, metodologías y ciclos de vida de la ingeniería del software."*

El desarrollo sigue el ciclo completo de la ingeniería del software, desde la especificación hasta la implementación y pruebas. Se emplea SCRUM como metodología de gestión ágil del proyecto.

## **CI17**

*"Capacidad para diseñar y evaluar interfaces persona computador que garanticen la accesibilidad y usabilidad a los sistemas, servicios y aplicaciones informáticas."*

Se diseña una interfaz móvil intuitiva, centrada en la experiencia del usuario, asegurando usabilidad y accesibilidad en la navegación, la creación de eventos y la interacción entre usuarios.

## **TI1**

*"Capacidad para comprender el entorno de una organización y sus necesidades en el ámbito de las tecnologías de la información y las comunicaciones."*

La propuesta responde a una necesidad social concreta: una plataforma unificada para organizar eventos sociales, abordando la falta de integración entre herramientas existentes y mejorando la experiencia de usuario.

## **TI6**

*"Capacidad de concebir sistemas, aplicaciones y servicios basados en tecnologías de red, incluyendo Internet, web, comercio electrónico, multimedia, servicios interactivos y computación móvil."*

La plataforma se desarrolla como una aplicación móvil basada en red, integrando servicios como chat, álbum de fotos, subida de contenidos, enlaces de invitación y visualización de eventos públicos.

## **Capítulo 3: Aportaciones y alineamiento con objetivos ODS**

### **3.1. Aportaciones**

Este Trabajo de Fin de Grado representa una aportación relevante tanto en el ámbito técnico como en el entorno social y económico. Desde el punto de vista tecnológico, el proyecto propone el desarrollo de una aplicación móvil que integra múltiples funcionalidades que actualmente se encuentran dispersas en distintas plataformas. Al ofrecer en una sola herramienta la creación de eventos, la gestión colaborativa de recursos y gastos, la comunicación entre participantes y una dimensión social con descubrimiento de eventos y usuarios, se presenta una solución completa y novedosa dentro del ecosistema de aplicaciones móviles.

En el plano socioeconómico, la plataforma promueve la participación social, el fortalecimiento de comunidades locales y la organización de actividades con mayor eficiencia y transparencia. Esto puede traducirse en una reducción de costes y tiempo en la planificación de encuentros, así como en la promoción de hábitos colaborativos y responsables entre los usuarios. La posibilidad de gestionar eventos públicos y privados con herramientas intuitivas contribuye a que más personas se animen a organizar actividades sociales o culturales, democratizando el acceso a la organización de eventos.

### **3.2. Alineamiento con objetivos ODS**

El trabajo se alinea especialmente con el ODS 11 (Ciudades y comunidades sostenibles), ya que fomenta la creación de comunidades activas y cohesionadas mediante la organización de eventos sociales en entornos digitales accesibles y seguros. Asimismo, guarda relación con el ODS 8 (Trabajo decente y crecimiento económico) y el ODS 9 (Industria, innovación e infraestructuras), al proponer una solución tecnológica que facilita la interacción entre personas y la gestión eficiente de recursos en contextos colaborativos.

## Capítulo 4: Análisis y diseño

### 4.1. Metodología

Para el desarrollo de este proyecto se ha aplicado la metodología ágil *SCRUM*, ampliamente utilizada en la industria del software por su enfoque iterativo, incremental y centrado en la entrega continua de valor. *SCRUM* permite adaptar el proceso de desarrollo a cambios en los requisitos, algo esencial en proyectos como este, donde la experiencia de usuario y las funcionalidades evolucionan a medida que se valida el diseño o surgen nuevas necesidades.

El trabajo se ha estructurado en *sprints* de dos semanas cada uno, con objetivos concretos y entregables funcionales al final de cada ciclo. El equipo de desarrollo ha estado formado únicamente por mí, mientras que Luis Miguel Hernández Acosta, tutor del TFG, ha asumido el rol de *Product Owner*. Cada dos *sprints* se realizó una reunión de seguimiento, en la que se revisaba el avance global, se extraían lecciones aprendidas y se ajustaban las metas y requisitos para los siguientes ciclos.

Durante cada *sprint* se han llevado a cabo actividades de análisis de requisitos, diseño de interfaces, implementación de funcionalidades, pruebas unitarias y de integración. Para gestionar el flujo de trabajo y el progreso se emplearon tableros *Kanban* en “GitHub Projects”, que permitieron visualizar el estado de cada tarea, registrar bloqueos y gestionar cambios de prioridad. Gracias a esta metodología, el desarrollo de *Evnet* ha sido ágil y ordenado, facilitando la incorporación de *feedback* y garantizando una rápida adaptación a los desafíos técnicos y de usabilidad encontrados en el camino.

### 4.2. Planificación del proyecto

Antes de comenzar con el desarrollo de la aplicación, se elaboró una planificación inicial estructurada en *sprints* de dos semanas, siguiendo la metodología ágil *SCRUM* descrita en el apartado anterior. Esta planificación contemplaba la estimación de horas por fase y la definición detallada de tareas específicas para cada *sprint*. Podemos observar la planificación inicial propuesta en la Tabla 1.

| <b>Fases</b><br>(ver ANEXO)                        | <b>Duración<br/>Estimada<br/>(horas)</b> | <b>Tareas</b><br>(nombre y descripción, obligatorio al menos una por fase)         |
|----------------------------------------------------|------------------------------------------|------------------------------------------------------------------------------------|
| Sprint 0: Planificación y diseño                   | 30                                       | Tarea 0.1: Definir requisitos del sistema y preparar <i>product backlog</i> .      |
|                                                    |                                          | Tarea 0.2: Diseño inicial de la interfaz y experiencia de usuario.                 |
|                                                    |                                          | Tarea 0.3: Diseño de la arquitectura del sistema.                                  |
|                                                    |                                          | Tarea 0.4: Configuración del entorno de desarrollo.                                |
|                                                    |                                          | Tarea 0.5: Revisión y planificación para el Sprint 1.                              |
| Sprint 1: Desarrollo básico de la aplicación       | 60                                       | Tarea 1.1: Configuración del proyecto en Ionic con Angular y conexión a Supabase.  |
|                                                    |                                          | Tarea 1.2: Configuración de la base de datos y API REST.                           |
|                                                    |                                          | Tarea 1.3: Implementación de la navegación básica y estructura general.            |
|                                                    |                                          | Tarea 1.4: Desarrollo de la funcionalidad de creación y visualización de eventos.  |
|                                                    |                                          | Tarea 1.5: Implementación de autenticación de usuarios con Supabase Auth.          |
|                                                    |                                          | Tarea 1.6: Pruebas básicas y ajustes iniciales.                                    |
|                                                    |                                          | Tarea 1.7: Revisión y planificación para el Sprint 2.                              |
| Sprint 2: Gestión de eventos y funciones iniciales | 60                                       | Tarea 2.1: Implementación de lista de compras dentro de los eventos.               |
|                                                    |                                          | Tarea 2.2: Desarrollo de división automática de gastos entre participantes.        |
|                                                    |                                          | Tarea 2.3: Permitir que los participantes agreguen elementos a la lista de compra. |
|                                                    |                                          | Tarea 2.4: Ajuste de pagos en función de contribuciones individuales.              |
|                                                    |                                          | Tarea 2.5: Pruebas de las funcionalidades y resolución de errores.                 |
|                                                    |                                          | Tarea 2.6: Revisión y planificación para el Sprint 3.                              |
| Sprint 3: Comunicación y red social                | 60                                       | Tarea 3.1: Implementación de chat en los eventos.                                  |
|                                                    |                                          | Tarea 3.2: Creación del sistema de seguidores y seguidos.                          |

|                                                   |    |                                                                             |
|---------------------------------------------------|----|-----------------------------------------------------------------------------|
|                                                   |    | Tarea 3.3: Sección donde se muestren eventos públicos de personas seguidas. |
|                                                   |    | Tarea 3.4: Sistema de invitaciones a eventos con enlaces únicos.            |
|                                                   |    | Tarea 3.5: Filtros para explorar eventos públicos y pruebas.                |
|                                                   |    | Tarea 3.6: Revisión y planificación para el Sprint 4.                       |
| Sprint 4: Álbum de fotos y exploración de eventos | 50 | Tarea 4.1: Integración de subida de imágenes en Supabase Storage.           |
|                                                   |    | Tarea 4.2: Implementación de vistas ordenadas para eventos inscritos.       |
|                                                   |    | Tarea 4.3: Optimización del diseño y mejora de la interfaz.                 |
|                                                   |    | Tarea 4.4: Pruebas completas de sincronización y rendimiento.               |
|                                                   |    | Tarea 4.5: Feedback y ajustes según resultados de las pruebas.              |
| Documentación y presentación                      | 40 | Tarea 5.1: Optimización del código y mejora del rendimiento.                |
|                                                   |    | Tarea 5.2: Redacción de la memoria técnica del proyecto.                    |
|                                                   |    | Tarea 5.3: Preparación de la presentación y demos funcionales.              |
|                                                   |    | Tarea 5.4: Ensayos de presentación y feedback final.                        |

**Tabla 1: Planificación inicial del proyecto**

Durante el desarrollo del proyecto, se siguió en líneas generales la planificación establecida al inicio, aunque fue necesario realizar ciertos ajustes en el orden, prioridad y alcance de algunas tareas, en función de la evolución del proyecto, del tiempo disponible y de las decisiones tomadas durante los *sprints*.

La mayoría de las tareas planificadas fueron finalmente ejecutadas conforme al cronograma, incluyendo fases clave como la definición de requisitos, el diseño de la arquitectura, la configuración del entorno de desarrollo, y la implementación de funcionalidades fundamentales como la creación y visualización de eventos, la autenticación de usuarios, la página principal, el perfil de los usuarios, el sistema de seguidores y seguidos, la búsqueda de eventos públicos y el apartado de eventos inscritos.

No obstante, algunas tareas planificadas no llegaron a implementarse en esta versión del proyecto debido a limitaciones de tiempo y a un reajuste en las prioridades para centrarse en los elementos más críticos del *producto mínimo viable* (MVP). Entre estas tareas se encuentran la posibilidad de que los participantes agreguen elementos a la lista de la

compra, el ajuste de pagos según contribuciones individuales, el sistema de chat dentro de los eventos, los filtros de exploración avanzada de eventos y el álbum de fotos colaborativo.

Estas decisiones se tomaron durante las revisiones internas realizadas al finalizar ciertos *sprints*, donde también se analizaron las funcionalidades ya implementadas y se valoró su impacto en la experiencia de usuario. Así, el orden de algunas tareas se modificó y se optó por dejar en segundo plano aquellas que, si bien aportaban valor, no eran esenciales para validar la propuesta inicial del proyecto.

### 4.3. Tecnologías utilizadas

Para el desarrollo de este proyecto se han empleado diversas tecnologías y herramientas que permiten construir una aplicación móvil moderna, escalable y con una experiencia de usuario cuidada. A continuación, se describen las principales:

- **Ionic con Angular:** Framework principal para el desarrollo del *frontend*. Gracias a Ionic se ha podido crear una aplicación multiplataforma (Android, iOS y web) reutilizando la misma base de código. Angular, por su parte, facilita la estructuración de la lógica de negocio y la gestión de componentes de forma modular.
- **TypeScript:** Lenguaje de programación utilizado para la mayor parte del desarrollo. Al ser un superconjunto de JavaScript que añade tipado estático, ayuda a reducir errores y mejora el mantenimiento del código.
- **Supabase:** Plataforma que ofrece una solución *backend* completa, incluyendo autenticación de usuarios, base de datos en la nube y almacenamiento de archivos. Supabase facilita la integración rápida de estas funcionalidades mediante su API REST y SDKs.
- **PostgreSQL:** Motor de base de datos relacional utilizado a través de Supabase, encargado de almacenar datos relacionados con usuarios y eventos, entre otros.
- **Supabase Storage:** Servicio que permite guardar y gestionar imágenes subidas por los usuarios.

- **Figma:** Herramienta empleada para diseñar la interfaz de usuario (UI) y crear prototipos interactivos. Ha sido clave para definir la estructura visual de la aplicación y validar flujos de navegación antes de implementarlos.
- **GitHub:** Plataforma utilizada para el control de versiones del código, lo que ha permitido mantener un historial de cambios y coordinar el progreso del desarrollo.
- **GitHub Projects:** Sistema de tableros Kanban empleado para organizar las tareas, planificar los *sprints* y hacer seguimiento del estado de cada funcionalidad.

## 4.4. Requisitos

En este apartado se describen los requisitos identificados para el correcto desarrollo del sistema. Primero se presentan los requisitos de usuario, que recogen las necesidades y expectativas desde el punto de vista del usuario final. Posteriormente, se detallan los requisitos del sistema, que definen las funciones y características técnicas que debe implementar la aplicación para dar respuesta a esas necesidades.

### 4.4.1. Requisitos de usuario

- **RU1.** Como usuario, quiero poder registrarme e iniciar sesión con correo electrónico o redes sociales, para acceder a la aplicación fácilmente.
- **RU2.** Como usuario, quiero poder crear eventos públicos o privados, para organizar quedadas o actividades.
- **RU3.** Como usuario, quiero invitar a personas concretas a eventos privados mediante enlaces únicos, para mantener el control sobre quién puede acceder.
- **RU4.** Como usuario, quiero poder editar los detalles del evento (nombre, fecha, hora, lugar, descripción, imágenes, precio, número máximo de participantes) en cualquier momento antes de su celebración.
- **RU5.** Como usuario, quiero poder eliminar un evento que haya creado, para que no sea visible si finalmente no se va a realizar.
- **RU6.** Como usuario, quiero poder unirme a eventos públicos o aceptar invitaciones a eventos privados, para participar en ellos.

- **RU7.** Como usuario, quiero ver en un apartado específico todos los eventos a los que estoy inscrito, ordenados por fecha, para tener claro mi calendario de actividades.
- **RU8.** Como usuario, quiero poder salir de un evento al que me había inscrito, para liberar mi plaza si ya no puedo asistir.
- **RU9.** Como usuario, quiero poder gestionar una lista de la compra vinculada a un evento, para organizar mejor qué se necesita.
- **RU10.** Como usuario, quiero poder marcar qué ítems de la lista de la compra llevaré yo, para que quede claro entre los participantes.
- **RU11.** Como usuario, quiero que el gasto total del evento se divida automáticamente entre los participantes, para no tener que calcularlo manualmente.
- **RU12.** Como usuario, quiero poder ver en todo momento cuánto dinero debo aportar según los ítems que me corresponden.
- **RU13.** Como usuario, quiero comunicarme en tiempo real con otros participantes de un evento a través de un chat, para coordinar detalles.
- **RU14.** Como usuario, quiero poder recibir notificaciones cuando haya mensajes nuevos, cambios importantes en un evento o cuando me inviten a uno.
- **RU15.** Como usuario, quiero poder subir y visualizar fotos en un álbum de cada evento, para compartir recuerdos con los demás.
- **RU16.** Como usuario, quiero poder seguir a otros usuarios y que me sigan, para crear una red social dentro de la aplicación.
- **RU17.** Como usuario, quiero ver fácilmente los eventos públicos creados por las personas que sigo, para descubrir actividades interesantes.
- **RU18.** Como usuario, quiero poder explorar todos los eventos públicos de la plataforma usando filtros (categoría, ubicación, fecha), para encontrar eventos que me interesen.

- **RU19.** Como usuario, quiero poder gestionar mi perfil (foto, nombre, contraseña, etc.), para mantener mi identidad actualizada.
- **RU20.** Como usuario, quiero que la aplicación funcione correctamente en dispositivos móviles y tablets, para usarla desde cualquier lugar.

#### **4.4.2. Requisitos del sistema**

- **RS1.** El sistema debe permitir el registro y autenticación de usuarios mediante correo electrónico y redes sociales (Google/Facebook).
- **RS2.** El sistema debe permitir a los usuarios crear eventos, indicando si son públicos o privados.
- **RS3.** El sistema debe generar enlaces únicos para invitar a eventos privados y validar esos enlaces al acceder.
- **RS4.** El sistema debe permitir la edición de la información de un evento por parte de su creador antes de que ocurra.
- **RS5.** El sistema debe permitir eliminar eventos creados por el usuario, eliminándolos de la vista de otros usuarios.
- **RS6.** El sistema debe permitir que los usuarios se inscriban en eventos públicos o privados (si disponen del enlace).
- **RS7.** El sistema debe ofrecer una vista para que cada usuario consulte los eventos en los que está inscrito, ordenados por fecha.
- **RS8.** El sistema debe permitir a un usuario cancelar su inscripción en un evento.
- **RS9.** El sistema debe permitir crear y gestionar listas de la compra asociadas a un evento.
- **RS10.** El sistema debe permitir que los usuarios marquen los ítems de la lista de la compra que se comprometen a llevar.
- **RS11.** El sistema debe calcular y mostrar la división automática de gastos entre participantes según ítems asignados.
- **RS12.** El sistema debe mostrar a cada usuario su parte proporcional de gasto en el evento.

- **RS13.** El sistema debe incluir un chat en tiempo real para cada evento, accesible solo por los participantes inscritos.
- **RS14.** El sistema debe enviar notificaciones a los usuarios ante mensajes nuevos, cambios de estado o invitaciones.
- **RS15.** El sistema debe permitir subir, almacenar y mostrar imágenes en un álbum vinculado a cada evento.
- **RS16.** El sistema debe permitir que los usuarios sigan a otros usuarios.
- **RS17.** El sistema debe mostrar los eventos públicos creados por los usuarios a los que sigue un usuario concreto.
- **RS18.** El sistema debe permitir explorar todos los eventos públicos y filtrarlos por categoría, ubicación y fecha.
- **RS19.** El sistema debe permitir a los usuarios editar su perfil (foto, datos personales, contraseña).
- **RS20.** El sistema debe funcionar correctamente en dispositivos móviles y tablets.

## 4.5. Diseño

### 4.5.1. Diseño de la interfaz y mockups

Para garantizar una experiencia de usuario clara, intuitiva y coherente, se ha comenzado el proceso de diseño por la creación de prototipos mediante la herramienta “Figma”. Este paso previo permite visualizar la estructura de la aplicación antes de iniciar la implementación, facilitando la detección temprana de mejoras y la validación de requisitos funcionales y estéticos.

- **Paleta de colores y logo**

La paleta de colores elegida para el diseño de la aplicación es la mostrada en la Ilustración 4, pensada para transmitir una imagen moderna, accesible y cercana al usuario. Esta combinación de colores busca reflejar el carácter social y colaborativo del proyecto, ofreciendo una apariencia visual equilibrada que resulta atractiva y funcional.



*Ilustración 4: Paleta de colores*

El logo de la aplicación ha sido creado con el apoyo de herramientas de inteligencia artificial generativa, partiendo de conceptos clave como eventos, celebración y comunidad. Posteriormente, se realizaron ajustes manuales para integrarlo con la paleta de colores seleccionada, consiguiendo un diseño coherente y reconocible que refuerza la identidad visual de la aplicación y aporta un estilo moderno y distintivo, tal y como podemos observar en la Ilustración 5.



*Ilustración 5: Logo de Evnet*

- **Formularios de inicio de sesión y registro**

Como vemos en la Ilustración 6, los formularios de inicio de sesión y registro están diseñados para ofrecer una experiencia de usuario sencilla e intuitiva. Se ha

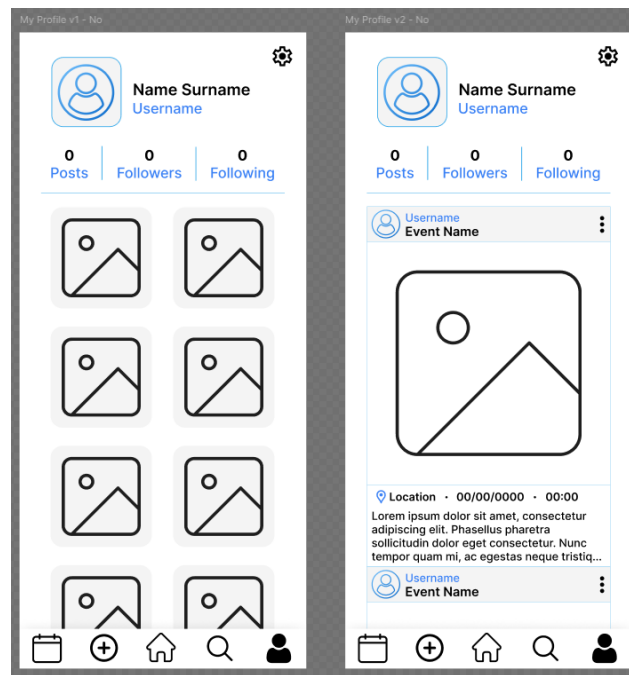
priorizado la claridad en los campos y la accesibilidad, facilitando el acceso rápido a la aplicación tanto para usuarios nuevos como para usuarios recurrentes.

Además, el diseño sigue la línea visual definida en la paleta de colores y mantiene la coherencia con el estilo general de la interfaz, garantizando una navegación fluida y agradable desde el primer contacto con la plataforma.

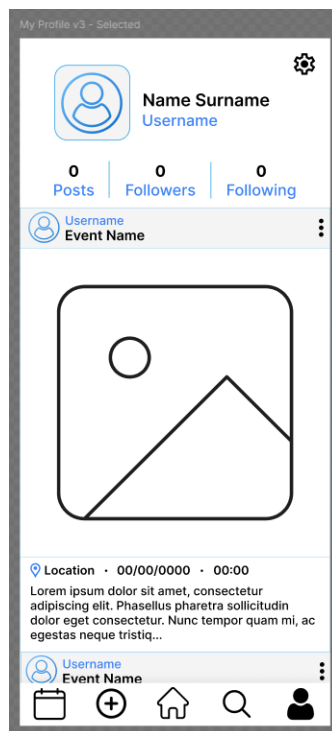
***Ilustración 6: Formularios de inicio de sesión y registro***

- **Perfil de usuario**

Durante el proceso de diseño de la sección "Mi perfil", se desarrollaron tres versiones diferentes para evaluar distintas opciones de presentación y funcionalidad. Tras analizar cada propuesta y considerar aspectos de usabilidad y estética, se seleccionó la versión final que mejor se adapta a las necesidades del usuario y a la coherencia visual del proyecto, mostrada en la Ilustración 8, pudiendo observar las opciones descartadas en la Ilustración 7. Este enfoque iterativo permitió mejorar la experiencia de usuario y optimizar la interfaz.

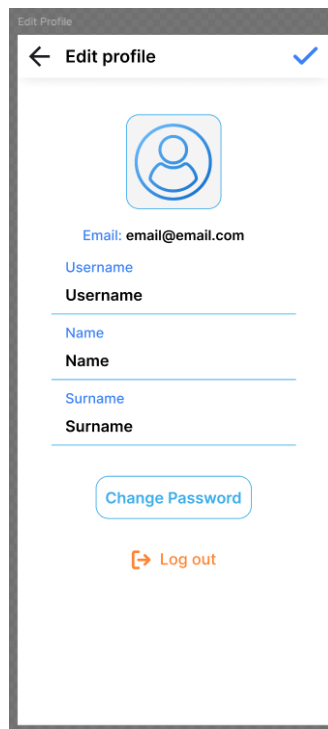


**Ilustración 7: Diseños descartados de mi perfil**



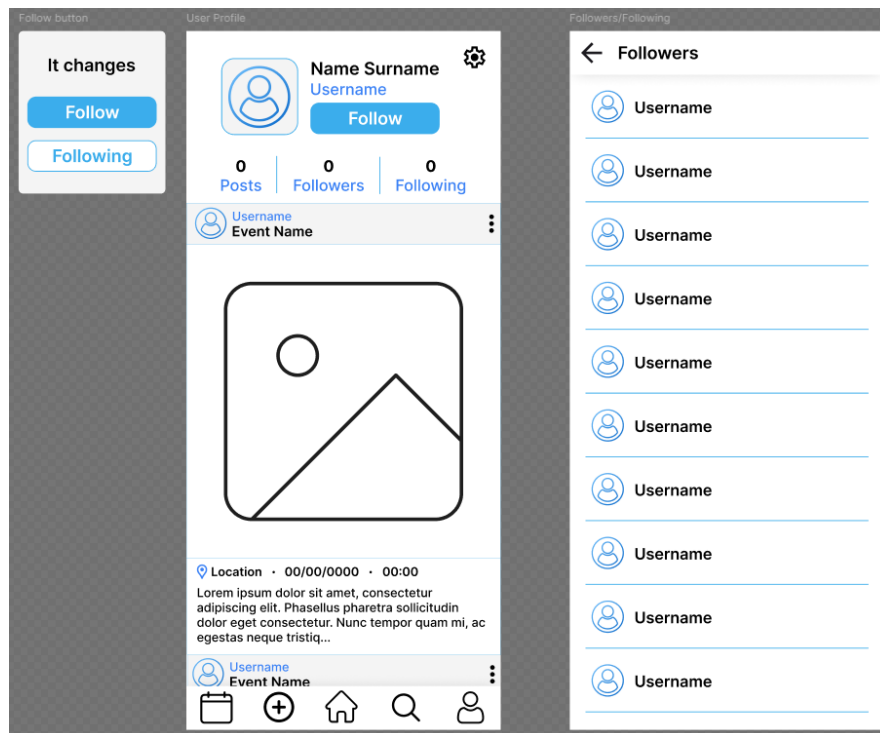
**Ilustración 8: Diseño final de mi perfil**

Al pulsar en el icono del engranaje, el usuario accede al apartado de “Editar perfil” mostrado en la Ilustración 9, donde puede modificar sus datos personales (foto, nombre de usuario, nombre, apellidos y contraseña) además de cerrar sesión, facilitando así la gestión y personalización de su cuenta.



***Ilustración 9: Diseño de página de editar perfil***

Por último, en el apartado de perfil de usuario se incluyen también los diseños correspondientes al perfil público de otros usuarios y la pantalla para visualizar la lista de usuarios a los que el usuario sigue, así como aquellos que le siguen, tal y como podemos ver en la Ilustración 10.



*Ilustración 10: Diseños de perfil público y listas de seguidores/seguidos*

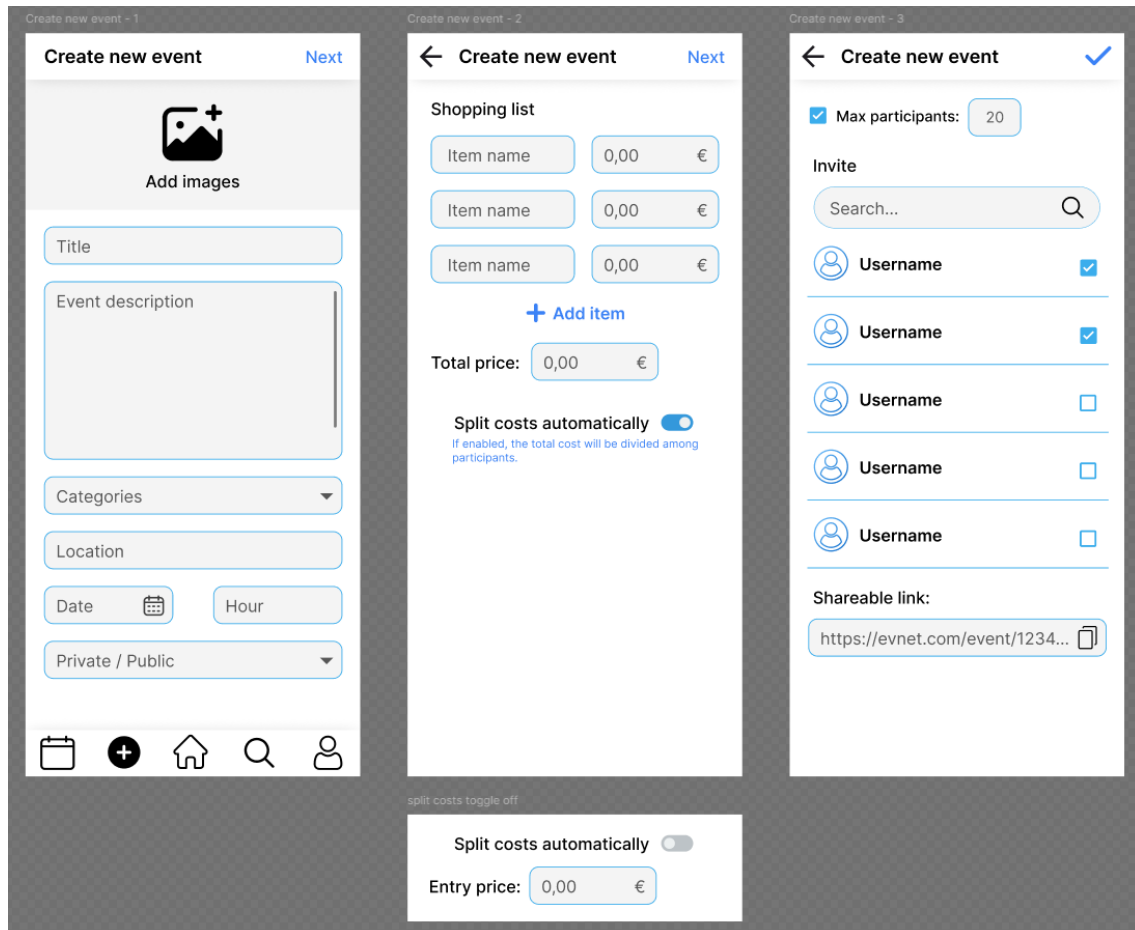
- **Creación de un nuevo evento**

Como vemos en la Ilustración 11, el proceso de creación de un nuevo evento se ha dividido en tres diseños principales, cada uno con su propia funcionalidad y campos específicos para facilitar la configuración completa del evento.

En el primer diseño, el usuario puede añadir las imágenes representativas del evento, así como definir información básica como el título, la descripción, las categorías, la ubicación, la fecha y la hora. Además, se establece la privacidad del evento, permitiendo escoger si será público o privado.

El segundo diseño se centra en la gestión económica del evento. Aquí se incluye la lista de la compra, compuesta por distintos ítems con nombre y precio. El precio total se calcula automáticamente a partir de los precios individuales de los ítems, pero también es posible modificarlo manualmente si se desea un valor diferente. Además, existe un interruptor que, si está activo, indica que el coste total de la lista se dividirá entre los asistentes. En caso contrario, aparecerá un campo para introducir el precio de la entrada al evento, que será obligatorio para los participantes.

Finalmente, el tercer diseño incluye la opción de establecer un número máximo de participantes, que puede ser desactivado si no se desea limitar el aforo. Además, permite buscar y seleccionar usuarios para invitarlos directamente al evento. En la parte inferior, se muestra un enlace para copiar, que facilita la difusión del evento y el acceso mediante ese link.

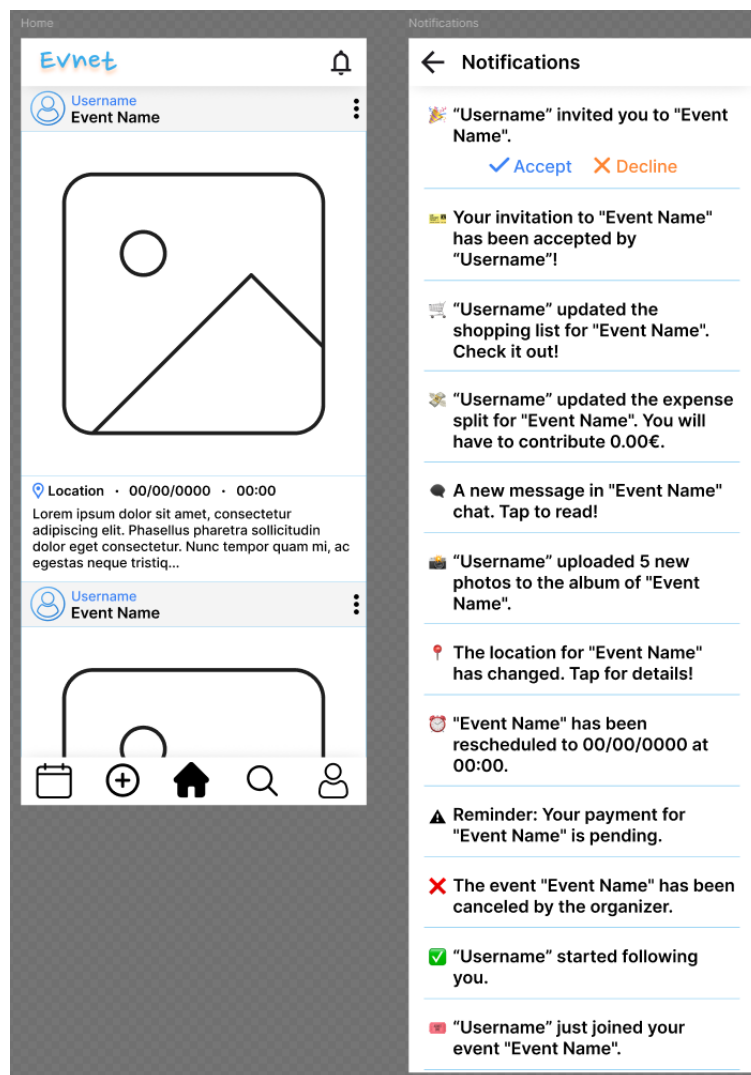


*Ilustración 11: Diseño de creación de un nuevo evento*

- **Página principal**

La página principal de la aplicación está diseñada como el punto de encuentro donde el usuario puede ver los eventos creados por las personas a las que sigue, fomentando así la interacción y la participación en su comunidad. En este apartado se muestra una lista actualizada y ordenada de los eventos públicos más relevantes, permitiendo al usuario explorar fácilmente las actividades próximas.

Además, desde la página principal se puede acceder al apartado de notificaciones, donde el usuario recibe alertas sobre novedades importantes, como invitaciones a eventos, mensajes nuevos o actualizaciones en eventos a los que está inscrito. Esto garantiza que el usuario esté siempre informado y pueda responder rápidamente a cualquier interacción dentro de la plataforma. Podemos observar estos diseños en la Ilustración 12.

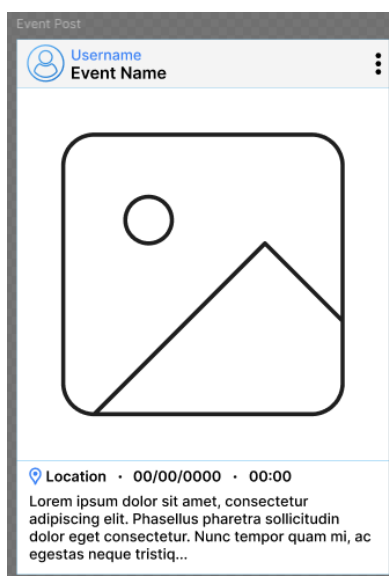


*Ilustración 12: Diseño de la página principal*

- **Diseño del evento**

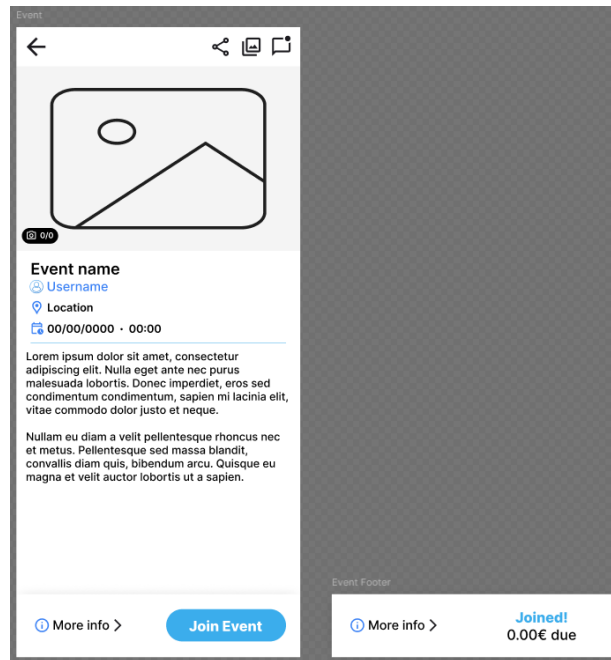
En la aplicación, los eventos se presentan inicialmente a través de un componente tipo "post", mostrado en la Ilustración 13, pensado para ofrecer de un vistazo la información más relevante y motivar la participación del usuario. Este

componente incluye una cabecera donde se muestra la foto de perfil y el nombre de usuario del creador, junto con el título del evento. Justo debajo, se incorpora un carrusel de imágenes destacadas del evento, que permite al usuario deslizar y visualizar varias fotos. En la parte inferior del post, se muestra de forma resumida la ubicación, la fecha y hora del evento, junto con un fragmento de la descripción, invitando así a ampliar la información.



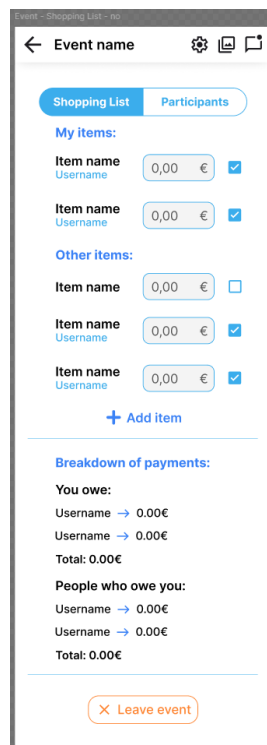
**Ilustración 13: Diseño de post del evento**

Al pulsar sobre este componente, el usuario accede a la página completa del evento, que podemos ver en la Ilustración 14, diseñada para ofrecer todos los detalles necesarios. En esta vista, se muestra nuevamente el carrusel de imágenes del evento, el título completo, el nombre de usuario y foto de perfil del creador, así como la ubicación, fecha y hora exactas, y la descripción completa del evento. En la parte inferior de esta página se encuentra un *footer* que ofrece dos opciones principales: la primera, “*More info*”, permite acceder a una vista más detallada con datos concretos sobre el evento, como la lista de la compra o la lista de participantes; la segunda opción, “*Join event*”, es un botón que, al pulsarlo, confirma automáticamente la inscripción del usuario, oculta el botón original y muestra en su lugar el estado de inscrito junto con la cantidad económica que debe aportar para participar.



**Ilustración 14: Diseño de evento**

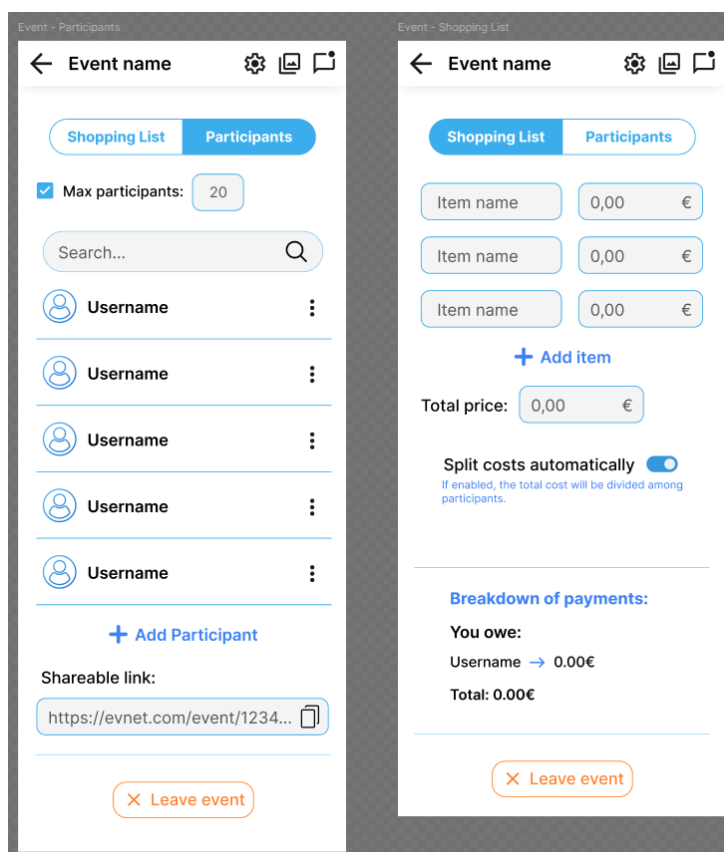
En la vista detallada del evento se encuentra el apartado “Más información”, diseñado inicialmente con una estructura que contemplaba funcionalidades adicionales, como la gestión de contribuciones individuales a la lista de la compra mostrada en la Ilustración 15.



**Ilustración 15: Diseño de más información del evento descartado**

Sin embargo, finalmente se optó por un diseño más simplificado al no incluir esta funcionalidad, priorizando la claridad y la facilidad de uso.

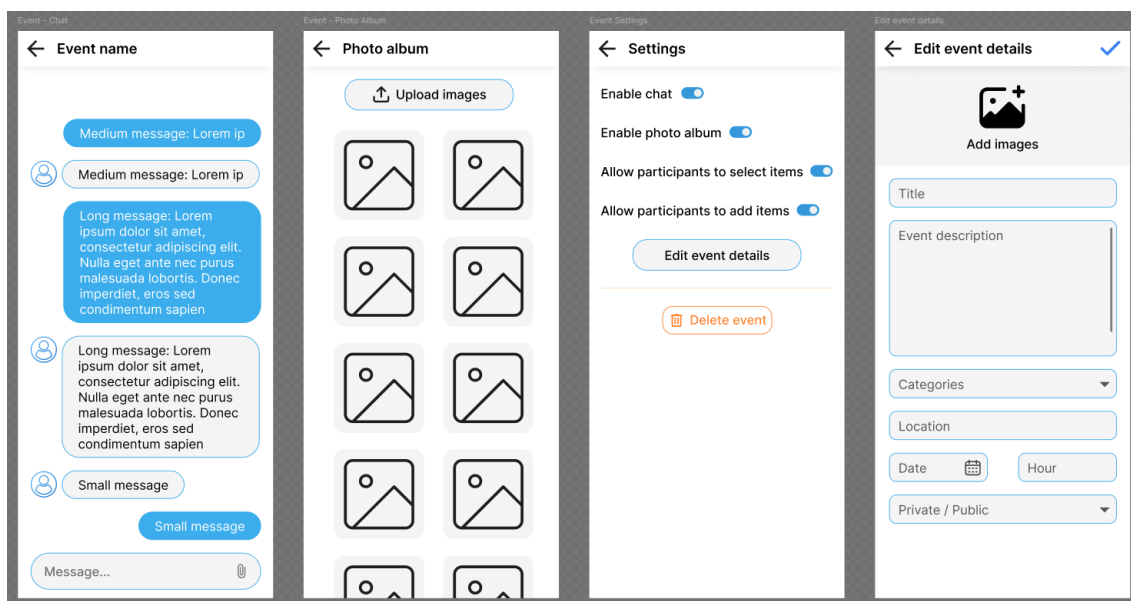
Como vemos en la Ilustración 16, este apartado se divide principalmente en dos secciones: la sección de lista de la compra, donde se muestran todos los ítems añadidos por el organizador o participantes, junto con su precio individual. Se incluye también el precio total del evento calculado a partir de la suma de estos ítems. Además, se indica la cantidad que debe aportar el usuario inscrito y, en su caso, a quién debe pagarla, permitiendo así una gestión más clara y directa de los gastos compartidos. Y la sección de participantes, la cual muestra un listado de los usuarios que ya se han unido al evento (mostrando su foto de perfil y nombre de usuario), donde se podrá invitar a nuevos usuarios para ampliar la lista de participantes.



**Ilustración 16: Diseño de más información del evento**

En la cabecera del apartado “Más información” del evento se encuentran accesos directos a tres funcionalidades principales: el chat del evento, el álbum de fotos y los ajustes del evento.

Desde el chat, los participantes pueden comunicarse en tiempo real para coordinar detalles o compartir información relevante. El álbum de fotos permite a los usuarios subir, visualizar y gestionar las imágenes asociadas al evento. Por último, en el apartado de ajustes, el organizador puede modificar distintos aspectos del evento, como activar o desactivar el chat y el álbum de fotos, así como editar detalles del propio evento para mantenerlo actualizado. Estos cuatro últimos diseños se pueden observar en la Ilustración 17.

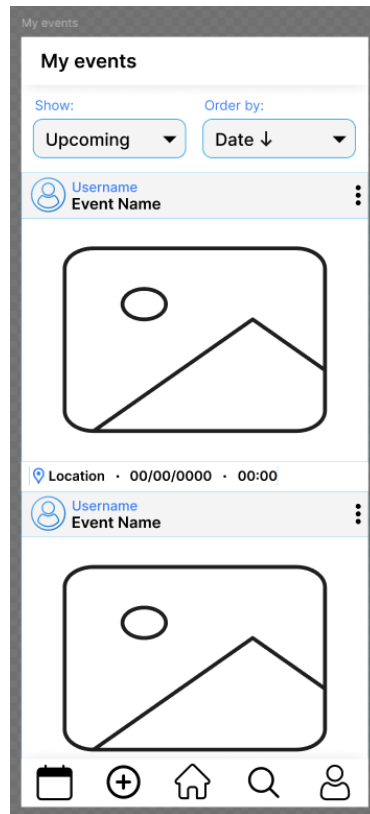


*Ilustración 17: Diseños de chat, álbum, ajustes y editar detalles de un evento*

- **Mis eventos**

Es el apartado de la aplicación donde el usuario puede consultar de forma clara y organizada todos los eventos en los que está inscrito, tal y como se muestra en la Ilustración 18. Para facilitar la navegación y la búsqueda, se ofrecen diferentes opciones de filtrado y ordenación: el usuario puede visualizar únicamente los eventos futuros, los eventos pasados o bien todos los eventos juntos.

Además, se incluye la posibilidad de ordenar la lista según distintos criterios, como por ejemplo la fecha de realización en orden ascendente o descendente, permitiendo priorizar los eventos más próximos o repasar eventos ya celebrados de manera sencilla y rápida.

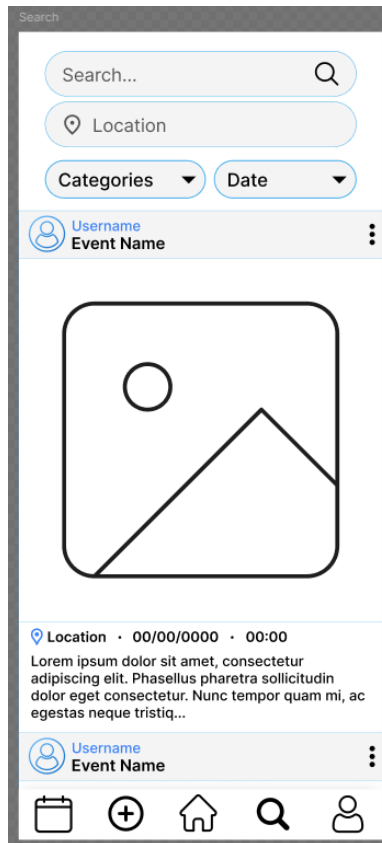


*Ilustración 18: Diseño de mis eventos*

- **Buscar eventos**

Esta es la sección de la aplicación que permite descubrir todos los eventos públicos creados por otros usuarios. Desde este apartado, los usuarios pueden realizar búsquedas por título, lo que facilita encontrar eventos concretos o relacionados con sus intereses.

También, como vemos en la Ilustración 19, se incluyen los filtros por ubicación, categorías y fecha, permitiendo así que cada usuario encuentre de forma rápida y sencilla los eventos que mejor se adapten a sus preferencias o disponibilidad.



*Ilustración 19: Diseño de buscar eventos*

#### 4.5.2. Diseño de la base de datos

La base de datos se ha modelado utilizando un enfoque relacional, complementado con algunos campos de tipo *array* y *jsonb* para dotarla de mayor flexibilidad y adaptarse mejor a las necesidades del proyecto.

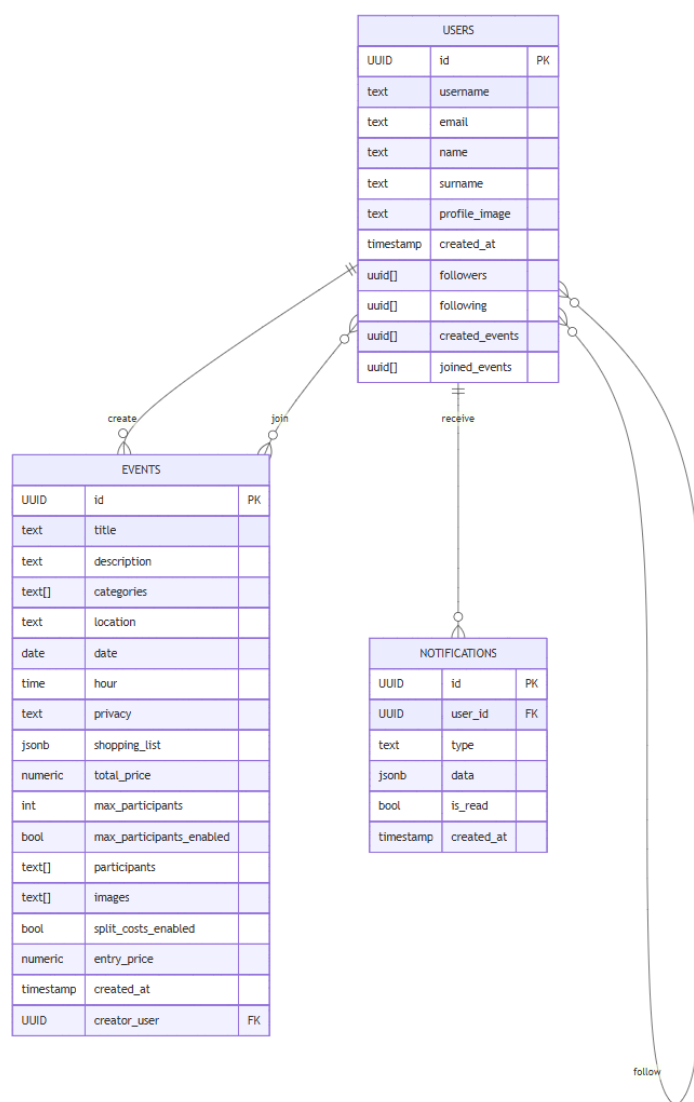
La base de datos se compone de tres tablas principales:

- **users:** contiene la información de cada usuario registrado en la plataforma, identificados de forma única mediante el campo “id” (clave primaria). Además de datos básicos como el “username”, “email”, “name” y “surname”, esta tabla incluye otros campos como la “profile\_image”, la fecha de creación (“created\_at”) y varios *arrays* de *uuid* (“followers”, “following”, “created\_events” y “joined\_events”) que permiten registrar qué usuarios siguen a otros, qué eventos ha creado un usuario y en cuáles está inscrito.
- **events:** almacena la información relativa a los eventos creados por los usuarios. Cada evento cuenta con su propio identificador único (“id”) y campos descriptivos

como “title”, “description”, “categories”, “location”, así como datos adicionales como “date”, “hour”, “privacy”, “shopping\_list” (en formato *jsonb* para permitir mayor flexibilidad), “total\_price”, “max\_participants” o “entry\_price”. A través del campo “creator\_user” (clave foránea), cada evento se vincula con el usuario que lo ha creado, estableciendo una relación de uno a muchos (un usuario puede crear varios eventos).

- **notifications:** recoge las notificaciones generadas por el sistema y dirigidas a los usuarios. Cada notificación tiene su identificador único (“id”), el identificador del usuario destinatario (“user\_id”, clave foránea que enlaza con “users”), el tipo de notificación (“type”), un campo “data” en formato *jsonb* para guardar información adicional (como el mensaje de la notificación en sí u otros datos), el estado de lectura (“is\_read”) y la fecha de creación (“created\_at”).

En la Ilustración 20 se muestra el diagrama entidad-relación que representa gráficamente la estructura y las relaciones existentes entre las tablas.



**Ilustración 20: Diagrama entidad-relación de la base de datos. Fuente: [Mermaid](#)**

En el diagrama se pueden observar las siguientes relaciones:

- Un usuario puede crear uno o varios eventos (relación 1:N), a través del campo “creator\_user” en la tabla “events”.
- Un usuario puede recibir varias notificaciones (relación 1:N), gracias al campo “user\_id” en la tabla “notifications”.
- Existe una relación de participación de muchos a muchos (N:N) entre “users” y “events”, reflejada mediante el array “participants” en la tabla “events” y el array “joined\_events” en “users”.
- Existe una relación de seguimiento entre usuarios (N:N), implementada mediante los arrays “followers” y “following” en la tabla “users”.

La notación empleada en el diagrama es la habitual en *diagramas entidad-relación*:

- PK indica clave primaria.
- FK indica clave foránea.
- $\| - o \{$  representa una relación de uno a muchos (1:N).
- $\} o - o \{$  representa una relación de muchos a muchos (N:N).

## Capítulo 5: Implementación

### 5.1. Estructura general del código

La estructura de carpetas del proyecto se encuentra dentro del directorio *src*, que es el directorio raíz donde se desarrolla la aplicación en Ionic con Angular. Esta organización sigue las convenciones y buenas prácticas recomendadas por ambos frameworks para mantener un código modular, escalable y fácil de mantener.

Dentro de *src* se encuentran las siguientes carpetas principales:

- **app:** Contiene el núcleo de la aplicación, donde se encuentran los componentes, servicios, módulos y demás elementos que conforman la lógica y funcionalidad principal.
- **assets:** Carpeta destinada a almacenar recursos estáticos como imágenes e iconos.
- **environments:** Incluye los archivos de configuración para diferentes entornos (desarrollo y producción), permitiendo ajustar parámetros específicos según el contexto en el que se despliegue la aplicación.
- **theme:** Contiene los estilos globales y variables de diseño utilizados en la aplicación, aprovechando el sistema de *theming* de Ionic para personalizar la apariencia y mantener la coherencia visual en toda la interfaz.

Dentro de la carpeta *app*, que como se mencionó anteriormente constituye el núcleo funcional de la aplicación, se organizan las siguientes subcarpetas principales para mantener el código modular y estructurado:

- **components:** Contiene los componentes reutilizables de la aplicación. Estos son bloques de interfaz independientes y autocontenidos que pueden ser utilizados en distintas páginas o partes de la app para facilitar la reutilización y mantener la consistencia visual y funcional.
- **guards:** Incluye los guardias de rutas (*route guards*) que controlan el acceso a ciertas páginas o funcionalidades, implementando lógica para proteger rutas según condiciones específicas, como la autenticación del usuario.

- **pages:** Alberga las páginas o vistas principales de la aplicación. Cada página representa una pantalla o sección específica, organizada en componentes Angular que gestionan tanto la presentación como la lógica asociada a esa parte del sistema.
- **services:** Contiene los servicios que proporcionan funcionalidades y lógica compartida, tales como gestión de datos, comunicación con APIs, manejo de estado o cualquier otra lógica que deba estar centralizada y accesible desde diferentes partes de la aplicación.

## 5.2. Librerías y dependencias

A continuación se describen las principales librerías y dependencias utilizadas en el proyecto. Todas ellas forman parte del ecosistema de Ionic y Angular, y se han elegido para cubrir diferentes necesidades como acceso a la base de datos, diseño de interfaces móviles, funcionalidades nativas y gestión de datos.

- **Supabase**

Plataforma *backend* que proporciona servicios como base de datos (“PostgreSQL”), autenticación de usuarios, almacenamiento de archivos y generación automática de *API REST* y *WebSocket*. Mediante su SDK, la aplicación puede realizar operaciones como registro, login, consulta de datos y subida de archivos de forma sencilla y segura.

- **Capacitor**

Librería desarrollada por el propio equipo de Ionic que actúa como puente para acceder a funcionalidades nativas de los dispositivos (como cámara, vibración, almacenamiento local, etc.). Gracias a Capacitor, la aplicación puede compilarse como app móvil nativa y aprovechar características del hardware del dispositivo.

- **Swiper**

Librería para crear carruseles y sliders. Utilizada en la aplicación para mostrar de forma atractiva las imágenes de los eventos.

- **RxJS**

Librería que permite trabajar con datos asíncronos y eventos mediante el uso de *observables*, facilitando la composición, transformación y manejo de flujos de datos de forma reactiva, algo especialmente útil en Angular para gestionar peticiones HTTP, formularios y actualizaciones en tiempo real.

- **UUID**

Librería para la generación de identificadores únicos universales (UUID). Se utiliza para crear de forma segura IDs de usuarios, eventos o notificaciones, asegurando que no haya colisiones en la base de datos.

### 5.3. Base de datos

Para el desarrollo de la aplicación se ha utilizado “Supabase” como *Backend-as-a-Service*, que proporciona una base de datos relacional gestionada sobre “PostgreSQL”, junto con servicios de autenticación, almacenamiento y *API REST* en tiempo real.

La base de datos se ha estructurado teniendo en cuenta los requisitos funcionales del sistema y la necesidad de mantener una arquitectura escalable, segura y fácil de consultar. A continuación, se describen las principales tablas y su función:

- **Tabla users**

La tabla *users*, mostrada en la Tabla 2, es la encargada de almacenar toda la información principal relacionada con los usuarios registrados en la aplicación.

Incluye campos básicos como identificador único, nombre de usuario y correo electrónico, así como datos opcionales como nombre o apellidos.

Además, contiene *arrays* para gestionar las relaciones sociales (seguidores y seguidos) y las referencias a los eventos creados y a los que el usuario se ha inscrito, lo que facilita la funcionalidad de red social y gestión de eventos dentro de la plataforma.

| <b>Campo</b>   | <b>Tipo</b> | <b>Descripción</b>                                                       |
|----------------|-------------|--------------------------------------------------------------------------|
| id             | Uuid        | Identificador único del usuario (clave primaria).                        |
| username       | Text        | Nombre de usuario único.                                                 |
| email          | Text        | Correo electrónico único.                                                |
| name           | Text        | Nombre real del usuario.                                                 |
| surname        | Text        | Apellidos del usuario.                                                   |
| profile_image  | Text        | URL o referencia a la imagen de perfil.                                  |
| created_at     | timestamp   | Fecha y hora de creación del registro (valor por defecto: fecha actual). |
| followers      | uuid[]      | Array con los IDs de usuarios que siguen a este usuario.                 |
| following      | uuid[]      | Array con los IDs de usuarios a los que sigue este usuario.              |
| created_events | uuid[]      | Array con los IDs de los eventos creados por el usuario.                 |
| joined_events  | uuid[]      | Array con los IDs de los eventos en los que el usuario está inscrito.    |

**Tabla 2: Tabla users de la base de datos**

- **Tabla events**

La tabla *events*, que podemos ver en la Tabla 3, es el núcleo para la gestión de los eventos dentro de la aplicación.

En ella se almacena toda la información relevante de cada evento creado por los usuarios, incluyendo título, descripción, localización, fecha y hora, además de otros campos pensados para enriquecer la experiencia social como las imágenes, lista de la compra o participantes.

También permite controlar la privacidad del evento, limitar el número de asistentes y registrar si el coste se reparte entre los participantes o se gestiona mediante precio de entrada.

| <b>Campo</b>                    | <b>Tipo</b> | <b>Descripción</b>                                                               |
|---------------------------------|-------------|----------------------------------------------------------------------------------|
| <b>id</b>                       | uuid        | Identificador único del evento.                                                  |
| <b>title</b>                    | text        | Título del evento.                                                               |
| <b>description</b>              | text        | Descripción detallada del evento.                                                |
| <b>categories</b>               | text[]      | Categorías o etiquetas asociadas al evento.                                      |
| <b>location</b>                 | text        | Ubicación donde se celebrará el evento.                                          |
| <b>date</b>                     | date        | Fecha del evento.                                                                |
| <b>hour</b>                     | time        | Hora de inicio del evento.                                                       |
| <b>privacy</b>                  | text        | Privacidad del evento (público o privado).                                       |
| <b>shopping_list</b>            | jsonb       | Lista de la compra asociada al evento.                                           |
| <b>total_price</b>              | numeric     | Coste total del evento, calculado automáticamente o editado manualmente.         |
| <b>max_participants</b>         | integer     | Número máximo de participantes permitidos en el evento.                          |
| <b>max_participants_enabled</b> | boolean     | Indica si está activo el límite de participantes.                                |
| <b>participants</b>             | text[]      | Array con los identificadores de los usuarios inscritos en el evento.            |
| <b>images</b>                   | text[]      | Array de imágenes asociadas al evento.                                           |
| <b>split_costs_enabled</b>      | boolean     | Indica si el coste del evento se divide automáticamente entre los participantes. |
| <b>entry_price</b>              | numeric     | Precio de entrada al evento, si no se reparte el coste de la lista de la compra. |
| <b>created_at</b>               | timestamp   | Fecha y hora de creación del evento.                                             |
| <b>creator_user</b>             | uuid        | Identificador del usuario creador del evento.                                    |

*Tabla 3: Tabla events de la base de datos*

- **Tabla notifications**

La tabla *notifications*, que se puede observar en la Tabla 4, se encarga de gestionar todas las notificaciones que recibe un usuario dentro de la aplicación.

Estas notificaciones permiten informar de acciones importantes, como la creación de nuevos eventos por usuarios seguidos, invitaciones a eventos privados o cualquier otro cambio relevante.

Además, cuenta con un campo destinado a almacenar información extra que varía según el tipo de notificación, y un indicador que permite saber si la notificación ya ha sido leída.

| Campo      | Tipo      | Descripción                                                                        |
|------------|-----------|------------------------------------------------------------------------------------|
| id         | uuid      | Identificador único de la notificación.                                            |
| user_id    | uuid      | Identificador del usuario que recibe la notificación.                              |
| type       | text      | Tipo de notificación (por ejemplo: invitación, nuevo evento, actualización, etc.). |
| data       | jsonb     | Información adicional específica de la notificación, almacenada en formato JSON.   |
| is_read    | boolean   | Indica si el usuario ya ha leído la notificación.                                  |
| created_at | timestamp | Fecha y hora en la que se creó la notificación.                                    |

*Tabla 4: Tabla notifications de la base de datos*

## 5.4. Componentes y clases

La aplicación está formada por diferentes elementos organizados en componentes, páginas, servicios y *guards*, cada uno con un papel específico dentro de la estructura del proyecto. Estos elementos trabajan conjuntamente para proporcionar la funcionalidad requerida y mantener un diseño modular y escalable. A continuación se describen las principales clases y componentes que conforman la aplicación, agrupados según su función dentro del sistema.

### 5.4.1. Componentes

Los componentes son unidades reutilizables que encapsulan la lógica y la presentación de pequeñas partes de la interfaz de usuario. Facilitan la construcción modular de la aplicación y permiten mantener un código más limpio y organizado. En este apartado se describen los componentes más relevantes que conforman la aplicación y sus funcionalidades principales.

- **EventPost**

Componente que muestra una vista resumida de un evento con información clave como el creador, título, ubicación, fecha, hora, descripción e imágenes asociadas. Permite navegar al perfil del creador y a la página detallada del evento al hacer clic en sus elementos, además de cargar dinámicamente las imágenes mediante un servicio externo.

- **EventImageCarousel**

Componente encargado de mostrar un carrusel de imágenes utilizando la librería “Swiper”, que permite la navegación y paginación visual entre las imágenes proporcionadas. Optimiza la carga mediante *lazy loading* para mejorar el rendimiento y la experiencia del usuario.

- **EventImagesInput**

Componente que permite al usuario seleccionar, previsualizar y gestionar múltiples imágenes para un evento. Muestra una vista previa de las imágenes cargadas y ofrece la opción de eliminar cualquiera de ellas. Maneja la selección de archivos desde el sistema, filtrando solo archivos de imagen válidos y renombrándolos para evitar conflictos. Actualiza y sincroniza las imágenes seleccionadas a través de un servicio compartido que mantiene el estado del formulario del evento.

- **ShareableLink**

Componente que muestra un campo de texto con un enlace compartible de un evento específico, generado dinámicamente a partir del identificador único del evento. Permite al usuario copiar fácilmente el enlace al portapapeles mediante un botón integrado, facilitando la difusión del evento.

- **EventHeader**

Componente que representa la cabecera de una pantalla o sección del evento, mostrando un título dinámico y botones de navegación y acción. Incluye opciones para volver atrás, compartir, acceder a configuraciones, galería y chat.

- **EventFooter**

Componente que muestra información y acciones relacionadas con la participación en un evento, incluyendo un botón para unirse si no se ha unido aún o un resumen del estado si ya participa, como el importe pendiente a pagar. También ofrece un acceso rápido a más información del evento, facilitando la interacción y el seguimiento desde la parte inferior de la pantalla.

- **SearchBar**

Componente que proporciona un campo de entrada para realizar búsquedas, permitiendo al usuario introducir texto y activar la búsqueda mediante la tecla *Enter* o pulsando el icono correspondiente. Emite el término de búsqueda para ser gestionado por componentes padres.

- **InvitationNotification**

Componente que muestra una notificación interactiva relacionada con invitaciones a eventos, permitiendo al usuario aceptar o rechazar la invitación mediante botones visibles solo cuando corresponde, y mostrando un estado confirmado cuando la invitación ha sido aceptada. También facilita la navegación directa al evento asociado al hacer clic sobre la notificación.

- **ProfileImage**

Componente que muestra la imagen de perfil de un usuario, con opción a ser editable para permitir la selección y actualización de una nueva imagen mediante un diálogo de archivos. Cambia visualmente para indicar cuándo la imagen es editable y emite el archivo seleccionado para que el componente padre pueda gestionar la actualización.

- **ProfileFollowButton**

Componente que permite a un usuario seguir o dejar de seguir a otro, mostrando el estado actual y actualizando dinámicamente la interfaz. Gestiona la interacción con el servicio de backend para modificar el estado de seguimiento y emite eventos para informar al componente padre sobre los cambios en el estado de seguimiento.

- **ProfileCounters**

Componente que muestra contadores de la cantidad de publicaciones, seguidores y seguidos de un usuario. Permite la navegación a las listas de seguidores y seguidos al hacer clic en sus respectivos contadores.

- **UserFollowerFollowing**

Componente que muestra información básica de un usuario, incluyendo su imagen de perfil y nombre de usuario, y permite navegar a su perfil al hacer clic.

- **Header**

Componente que representa la cabecera de una página con opciones configurables para mostrar botones de retroceso, confirmación y avance. Permite mostrar un título dinámico y emitir eventos cuando se activan los botones de “Next” o “Check”, facilitando la navegación y acciones del usuario dentro de la aplicación.

- **Footer**

Componente que implementa una barra de navegación inferior con pestañas, permitiendo al usuario acceder rápidamente a las secciones principales de la aplicación. Cada pestaña muestra un icono que cambia según si está seleccionada o no, y la navegación se actualiza dinámicamente según la ruta actual.

### 5.4.2. Páginas

Las páginas conforman la estructura principal de la aplicación y representan las diferentes vistas a las que puede acceder el usuario para interactuar con las funcionalidades disponibles. Cada página está diseñada para cumplir un propósito específico dentro del flujo general, facilitando tareas como la autenticación, la gestión del perfil, la creación y

visualización de eventos, así como la interacción social con otros usuarios. A continuación se describen las páginas más relevantes que componen la aplicación.

- **Home**

La página *Home* actúa como el centro del *feed* social de la aplicación, donde el usuario puede descubrir y explorar los eventos organizados por las personas que sigue. Al entrar, se muestra un encabezado personalizado (*HomeHeader*) que también indica si el usuario tiene notificaciones no leídas.

El contenido principal se compone de una lista dinámica de eventos (*EventPost*) cargados desde la base de datos de Supabase. Estos eventos se obtienen en bloques gracias a un sistema de paginación implementado con *infinite scroll*: al desplazarse hacia abajo, se cargan automáticamente más eventos, mejorando la experiencia de exploración sin necesidad de recargar la página manualmente.

Si el usuario no sigue a nadie o todavía no existen eventos disponibles, se muestra un mensaje que lo indica de forma clara.

- **MyProfile**

La página *MyProfile* permite al usuario ver y gestionar su propio perfil dentro de la aplicación. Al entrar, se muestra la información básica del usuario, como nombre completo, nombre de usuario y foto de perfil (*ProfileImage*). También incluye un acceso rápido a la página de editar perfil mediante un icono de configuración en la parte superior.

Justo debajo, se visualizan los contadores de actividad (*ProfileCounters*), que indican el número de eventos creados, seguidores y seguidos. Finalmente, se listan de forma ordenada todos los eventos creados por el usuario (*EventPost*), permitiendo revisar rápidamente su historial de publicaciones y actividades en la plataforma.

- **EditProfile**

La página *EditProfile* ofrece al usuario un espacio para actualizar y gestionar los datos de su perfil de forma sencilla e intuitiva. Al acceder, se muestra un encabezado (*Header*) con opciones para volver atrás o guardar los cambios realizados.

En el centro de la página, el usuario puede modificar su foto de perfil (*ProfileImage*), que es editable y permite previsualizar la nueva imagen seleccionada antes de guardarla. También se visualiza la dirección de correo electrónico asociada a la cuenta, que se muestra como dato informativo, ya que no es editable directamente desde esta vista.

Para editar los datos personales, el usuario dispone de varios campos (*EditProfileInput*) para actualizar el nombre, apellido y nombre de usuario. Además, la página ofrece accesos directos para cambiar la contraseña (*ChangePasswordButton*) o cerrar sesión (*LogoutButton*).

- **Profile**

La página *Profile* permite visualizar el perfil público de cualquier usuario de la aplicación, distinto al usuario autenticado. Muestra información básica como la foto de perfil, nombre completo y nombre de usuario, acompañados de un botón para seguir o dejar de seguir a ese usuario (*ProfileFollowButton*), facilitando la interacción social.

Al igual que en *MyProfile*, se incluyen los contadores de publicaciones, seguidores y seguidos para ofrecer una visión rápida de la actividad y la comunidad del usuario. Bajo esta sección, se despliega la lista de eventos creados por el usuario, permitiendo explorar su actividad reciente dentro de la plataforma.

La página se carga dinámicamente según el identificador del usuario seleccionado en la URL. Si el usuario intenta acceder a su propio perfil mediante esta ruta, es redirigido automáticamente a *MyProfile* para evitar duplicidad y mantener una experiencia coherente.

- **CreateEvent**

La página *CreateEvent* permite al usuario crear un nuevo evento de forma intuitiva y estructurada mediante un formulario reactivo. En la parte superior, un encabezado personalizado con el título "Create new event" incluye un botón para avanzar a la siguiente etapa del proceso, facilitando la navegación progresiva.

El formulario principal está compuesto por múltiples campos controlados que validan la entrada del usuario en tiempo real para garantizar que todos los datos

necesarios se completen correctamente antes de continuar. Estos campos incluyen la carga de imágenes del evento (*EventImagesInput*), título, descripción, categorías seleccionables, ubicación, fecha, hora y nivel de privacidad. Cada *input* posee validaciones que indican al usuario si algún campo está incompleto o contiene errores, mejorando la usabilidad y la calidad de los datos ingresados.

Al presionar el botón "Next", los datos del formulario se almacenan temporalmente en un servicio compartido (*EventFormDataService*), junto con un UUID generado para identificar el evento, y se incluye automáticamente el identificador del usuario creador obtenido desde el sistema de autenticación (Supabase). Esto permite una navegación fluida hacia la siguiente pantalla donde se puede completar información adicional del evento.

- **CreateEventShoppingList**

Esta página es el segundo paso en el flujo de creación de eventos y está dedicada a definir la lista de compras necesaria para el evento. El encabezado incluye un botón de retroceso para volver a la pantalla anterior y un botón "Next" para avanzar al siguiente paso.

El núcleo de la interfaz es una lista dinámica donde el usuario puede agregar, editar y eliminar ítems de compra. Cada ítem incluye un campo para el nombre y otro para el precio, ambos vinculados bidireccionalmente para mantener los datos actualizados en tiempo real. Los precios se suman automáticamente para mostrar un total actualizado de la lista de compras, brindando al usuario una visión clara del costo total estimado.

Además, incluye un componente para dividir automáticamente los costos, que permite configurar si los gastos se repartirán entre participantes y establecer un precio de entrada si corresponde.

Al avanzar, todos los datos recopilados (lista de compras, total, y configuración de división de costos) se almacenan en un servicio compartido (*EventFormDataService*) para mantener el estado del evento en creación. Si el usuario intenta acceder a esta pantalla sin haber iniciado el proceso correctamente, es redirigido a la página inicial de creación para asegurar la secuencia lógica.

- **CreateEventParticipants**

Esta página es el paso final en el flujo de creación de eventos y está dedicada a configurar los participantes del evento. El encabezado incluye un botón de retroceso para volver a la pantalla anterior y un botón de confirmación (*check*) para finalizar la creación del evento.

La interfaz permite activar o desactivar un límite máximo de participantes, y si está activado, se puede definir el número máximo permitido. Además, ofrece un buscador dinámico para encontrar usuarios por su nombre de usuario, mostrando resultados que excluyen a los ya seleccionados.

Los usuarios seleccionados para invitar se muestran en una lista separada y pueden ser eliminados en cualquier momento, actualizando automáticamente las listas disponibles. También se incluye un componente que muestra un enlace compartible del evento para facilitar su difusión.

Al finalizar, todos los datos recopilados (configuración de participantes, lista de invitados y detalles del evento) se almacenan en un servicio compartido (*EventFormDataService*). Se crea el evento en la base de datos mediante *SupabaseService*, se envían invitaciones a los usuarios seleccionados y el creador se une automáticamente al evento. Si el usuario intenta acceder a esta pantalla sin iniciar correctamente el proceso, es redirigido a la página inicial de creación para mantener la secuencia lógica.

- **Event**

Esta página muestra los detalles completos de un evento específico seleccionado por el usuario. En primer lugar, se presenta un carrusel de imágenes (*EventImageCarousel*) que permite visualizar las fotos relacionadas con el evento.

El cuerpo principal contiene la información clave del evento: el título, el nombre y foto del creador (que al pulsar lleva al perfil del usuario creador), la ubicación del evento con un ícono representativo, y la fecha junto con la hora formateada para fácil lectura. También se muestra una descripción detallada que ofrece más contexto sobre el evento.

Toda la información se carga dinámicamente al entrar en la vista, recuperando los datos desde el *backend* a través del servicio *SupabaseService*. Esto incluye las imágenes, el perfil y nombre del creador, y los datos principales del evento.

Finalmente, si el evento está correctamente cargado, se muestra un pie de página específico para el evento (*EventFooter*).

- **EventMoreInfo**

Esta página ofrece una vista detallada y gestionable de la información adicional de un evento, dividiéndose en dos modos principales: la lista de compras y la lista de participantes, entre los cuales el usuario puede alternar mediante un switch (*ShoppingListParticipantsSwitch*).

En el modo “Lista de la compra”, el usuario puede agregar, editar y eliminar ítems con nombre y precio. Los precios se suman automáticamente para mostrar un total actualizado, y un componente permite configurar la división automática de costos, incluyendo un precio de entrada si corresponde (*SplitCostsAutomatically*). Además, se presenta un desglose claro de cuánto debe pagar el usuario, calculado según si es el creador, si la división está habilitada, o si solo aplica el precio de entrada. Los cambios realizados se guardan automáticamente tras breves pausas en la edición, asegurando que estos no se pierdan.

En el modo “Participantes”, se puede gestionar el número máximo de participantes, buscar usuarios y visualizar los seleccionados con su imagen y nombre. También está disponible un botón para añadir nuevos participantes, que redirige a la pantalla correspondiente (*AddParticipants*). También, se muestra un enlace para compartir el evento y un botón para abandonar el evento si el usuario lo desea.

El encabezado muestra el título dinámico del evento y un botón de configuración para ajustes adicionales. Al cargar la página, se recuperan todos los datos relevantes desde el *backend* mediante el servicio *SupabaseService*: detalles del evento, lista de compras, participantes, configuración de costos y más.

Se emplean mecanismos para detectar cambios en la lista de compras, el precio de entrada y la configuración de división de costos, con un temporizador que

activa la función de auto-guardado para sincronizar los datos con el *backend* de manera eficiente y transparente para el usuario.

- **MyEvents**

Esta página presenta la lista de eventos en los que el usuario ha participado o está inscrito.

Dentro del contenido principal, se despliega una lista de publicaciones (*EventPost*) que representan cada evento, cargadas dinámicamente desde el *backend* mediante el servicio *SupabaseService*. Si el usuario no ha participado en ningún evento, se muestra un mensaje informativo indicando que aún no ha unido a eventos.

El listado utiliza un scroll infinito que permite cargar más eventos automáticamente conforme el usuario se acerca al final de la lista, mejorando la experiencia al evitar largas cargas iniciales y mostrando progresivamente más contenido. Cuando no quedan más eventos por cargar, el scroll infinito se deshabilita para evitar nuevas solicitudes.

Al entrar en la vista, la página obtiene el identificador del usuario actual y carga los primeros eventos en los que participa, manteniendo la paginación mediante límites y offsets para controlar la cantidad de datos descargados.

- **Search**

Esta página permite al usuario buscar eventos mediante un campo de búsqueda situado en la parte superior (*SearchBar*), facilitando la entrada de texto para filtrar los eventos disponibles.

Al ingresar texto en la barra de búsqueda, la página actualiza dinámicamente los resultados mostrados, que se despliegan en una lista de publicaciones (*EventPost*) correspondientes a los eventos encontrados que coinciden con la consulta. Si no hay resultados para la búsqueda realizada, se muestra un mensaje indicando que no se encontraron eventos.

Para mejorar la experiencia, la lista utiliza scroll infinito, permitiendo cargar progresivamente más resultados conforme el usuario se desplaza hacia abajo,

optimizando el rendimiento y evitando sobrecargar la pantalla con demasiados datos al inicio.

La búsqueda y la paginación se gestionan mediante llamadas al servicio *SupabaseService*, que realiza consultas paginadas y entrega los resultados correspondientes según el texto ingresado. Al entrar en la vista, la página inicializa su estado, limpiando resultados previos y preparando el scroll infinito para nuevas búsquedas.

### 5.4.3. Servicios

En esta sección se presentan los servicios clave que soporta la aplicación, abarcando desde la gestión de la comunicación con la base de datos hasta el manejo centralizado del estado y la lógica relacionada con la creación y edición de eventos. Además, se incluyen utilidades que facilitan operaciones comunes y mejoran la experiencia de usuario, como la generación de identificadores únicos o la gestión de alertas.

- **SupabaseService**

En el núcleo de la aplicación se encuentra *SupabaseService*, un servicio único y cohesionado que actúa como capa de abstracción sobre las distintas funcionalidades que ofrece Supabase. Al instanciarse, este servicio configura el cliente de Supabase con la URL y la clave aportadas por el entorno, lo que permite a los distintos métodos comunicarse de forma segura y uniforme con la plataforma. De este modo, todas las operaciones de lectura y escritura, desde la consulta de tablas hasta la gestión de almacenamiento de imágenes, se ejecutan a través de una única instancia compartida, garantizando consistencia y facilitando la inyección de dependencias en los componentes de la aplicación.

La autenticación de usuarios se gestiona mediante métodos como “signUp”, “signIn” y “signOut”, que permiten crear nuevas cuentas, iniciar sesión con credenciales y cerrar la sesión limpiamente, incluidos los canales de *realtime* que se usan para notificaciones y actualizaciones en vivo. En el proceso de registro se comprueban las contraseñas, se crea el usuario en la base de datos de autenticación y se enlaza posteriormente con un perfil en la tabla *users*, donde se almacenan campos adicionales como el nombre de usuario y la imagen de perfil.

A continuación, *SupabaseService* ofrece una serie de métodos para la gestión de perfiles de usuario: con “*getUserId*” se obtiene el identificador del usuario autenticado, mientras que “*getUserProfileData*” y “*getEditProfileData*” permiten recuperar respectivamente toda la información de perfil y los campos editables. Para modificar estos datos se utiliza “*updateProfile*”, que aparte de actualizar los valores en la tabla *users* gestiona la subida de la nueva imagen al *bucket* de *Storage*, y para la seguridad de las credenciales se incorporan “*changePassword*” y “*verifyCurrentPassword*”, que facilitan el cambio y la validación de la contraseña actual.

En lo relativo a los eventos, el servicio abstrae tanto la subida de imágenes (a través de “*uploadEventImages*”) como la creación y actualización de registros en la tabla *events* mediante “*createEvent*”, generando UUIDs de forma automática cuando es necesario. Para la consulta de datos existen métodos como “*getEventById*”, que recupera un evento completo, “*getEventImageUrl*”, que convierte un *path* de almacenamiento en una URL pública, y “*getEventMoreInfo*”, que retorna la información detallada necesaria en la vista de más información del evento.

Además, *SupabaseService* incorpora funcionalidades de búsqueda y listados paginados: “*searchEventsPaginated*” permite filtrar eventos públicos por título, “*getRecommendedEvents*” ofrece los últimos eventos creados, “*getMyCreatedEventsByUserId*” y “*getCreatedEventsByUserId*” obtienen los eventos propios (públicos o privados), “*getJoinedEvents*” filtra las participaciones por estado y orden, y “*getEventsFromFollowedUsers*” muestra los eventos de los usuarios a los que se sigue.

La lógica de relaciones sociales está implementada con “*followUser*”, “*unfollowUser*” e “*isFollowing*” para gestionar los *arrays* de seguidores y seguidos, complementado por “*getFollowersByUserId*” y “*getFollowingByUserId*” para recuperar listados de perfiles. Por último, el sistema de notificaciones se sustenta en “*sendEventInvitations*” para generar invitaciones, “*getNotificationsForUser*” para obtener todas las alertas, “*markNotificationAsRead*” y “*updateNotificationType*” para actualizar su estado, “*hasUnreadNotifications*” para comprobar si quedan mensajes pendientes y

“getEventTitleById” para referenciar claramente los eventos implicados en cada notificación.

- **EventFormDataService**

El servicio *EventFormDataService* funciona como un almacén temporal en memoria para todos los datos necesarios durante el proceso de creación y edición de un evento en la aplicación. Su principal objetivo es mantener el estado del formulario mientras el usuario navega entre distintas pantallas o pasos, evitando así la pérdida de información antes de enviarla a la base de datos. Este servicio permite establecer y obtener datos del evento de forma individual o global, así como gestionar dinámicamente listas como las imágenes o la lista de la compra. Además, dispone de métodos para reiniciar el estado cuando sea necesario. Al implementarse como *singleton* (*providedIn: 'root'*), garantiza que los datos se mantengan consistentes y accesibles desde cualquier componente de la aplicación durante todo el flujo de creación o edición de un evento.

- **Utils**

El archivo *utils.ts* reúne funciones de utilidad generales que se utilizan en distintos puntos de la aplicación para simplificar tareas comunes. Incluye una función “showAlert” que muestra mensajes emergentes al usuario de forma sencilla usando el *AlertController* de Ionic, y una función “generateUUID” que permite generar identificadores únicos, útiles por ejemplo para asignar IDs a nuevos eventos o recursos temporales antes de almacenarlos en la base de datos. Estas utilidades ayudan a mantener el código más limpio, reutilizable y organizado.

#### 5.4.4. Guards

En esta aplicación se ha implementado un único *guard* llamado “authGuard”, cuya función principal es proteger las rutas que requieren que el usuario esté autenticado. Este *guard* comprueba de forma asíncrona, a través del servicio *SupabaseService*, si existe un usuario activo (es decir, si el usuario ha iniciado sesión). Si el usuario está autenticado, se permite el acceso a la ruta; en caso contrario, se redirige automáticamente a la página de *login*. De esta forma se garantiza que ciertas partes de la aplicación solo sean accesibles para usuarios registrados, mejorando la seguridad y la experiencia de usuario.

## 5.5. Pruebas y validación

Para asegurar la calidad, la fiabilidad y el correcto funcionamiento de la aplicación, se ha diseñado y ejecutado un conjunto exhaustivo de 27 *test suites*, que abarcan 161 pruebas unitarias en total. Estas tests incluyen desde la verificación de la inicialización de componentes básicos (*login*, *registro*, *perfiles*) hasta la comprobación de la lógica de negocio en servicios y *guards*. Cada prueba se ha centrado en validar los métodos y flujos más relevantes, garantizando que tanto los formularios como las rutas protegidas reaccionan de forma adecuada ante entradas válidas e inválidas, y que los mensajes de alerta y las redirecciones se activan según lo esperado.

En el ámbito de la gestión de eventos se han cubierto ampliamente tanto la página principal (*Home*) como las pantallas de creación de eventos, que incluyen el formulario inicial, la lista de la compra y la selección de participantes, así como la vista de detalle de evento y la sección de más información. Se ha comprobado que el componente de presentación de eventos (*EventPost*), el carrusel de imágenes, los controles de fecha y hora, la generación y copiado de enlaces compartibles, y la división automática de costes funcionan correctamente en escenarios diversos. Además, se han validado los flujos de unión y abandono de eventos, el envío de invitaciones y la actualización de la lista de participantes.

Para mejorar la experiencia de búsqueda y exploración, también se ha implementado pruebas sobre la funcionalidad de búsqueda con scroll infinito, así como sobre las páginas de “Mis eventos” y perfil público de otros usuarios, garantizando la correcta paginación, recarga y navegación entre los resultados.

De este modo, todas las partes críticas de la aplicación han sido validadas mediante pruebas automatizadas, aportando un alto grado de robustez y confianza antes de su despliegue.

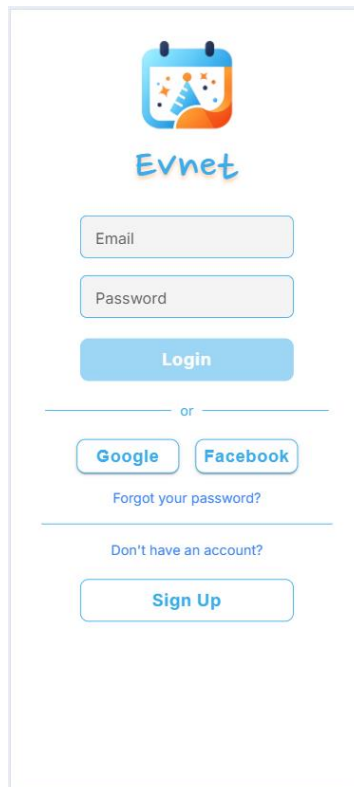
## Capítulo 6: Guía de usuario

En este capítulo se describe el funcionamiento general de la aplicación desde el punto de vista del usuario final. A lo largo de esta guía se recorren las principales pantallas y apartados de la aplicación: desde el proceso de registro e inicio de sesión, pasando por la creación y gestión de eventos, hasta el uso de las funciones sociales como el buscador, los perfiles de otros usuarios y la gestión de participantes y amigos. Además, se incluyen indicaciones sobre el manejo de la lista de la compra colaborativa, el control de gastos y otras utilidades pensadas para facilitar la organización de eventos de forma sencilla y eficiente.

En primer lugar, al abrir la aplicación, se muestra la página de inicio de sesión (*login*), tal como se puede ver en la Ilustración 21. En esta pantalla, el usuario que ya disponga de una cuenta puede acceder a la aplicación introduciendo su dirección de correo electrónico y su contraseña.

Cabe destacar que, por el momento, el inicio de sesión a través de servicios externos como Google o Facebook no está habilitado, por lo que el acceso debe realizarse obligatoriamente mediante las credenciales propias registradas en la aplicación.

En el caso de que el usuario todavía no disponga de una cuenta, podrá crearla fácilmente accediendo a la página de registro (*signup*). Para ello, únicamente debe pulsar el botón situado en la parte inferior de la pantalla, identificado con el texto “Sign Up”, que le llevará al formulario de creación de cuenta.

The image shows a login and sign-up form for 'Evnet'. At the top is a logo consisting of a blue and orange square with a white circle and dots inside, followed by the text 'Evnet' in a blue, rounded font. Below the logo are two input fields: 'Email' and 'Password', both with light gray borders. Under these is a blue 'Login' button. A horizontal line with the word 'or' in the center separates the login section from the sign-up section. Below the line are two buttons: 'Google' and 'Facebook', both with light blue borders. Under these is a link 'Forgot your password?' in blue text. Another horizontal line is below this link, followed by a link 'Don't have an account?' in blue text. At the bottom is a blue 'Sign Up' button.

***Ilustración 21: Página de inicio de sesión (login)***

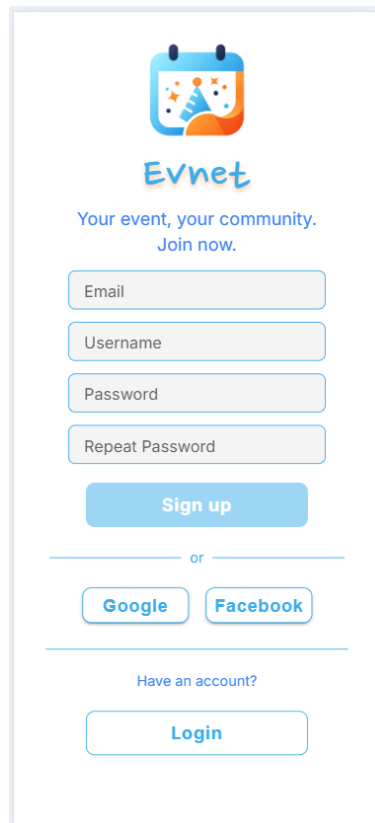
A continuación, al acceder a la página de registro (*signup*), el usuario tiene la posibilidad de crear una nueva cuenta completando los campos que se muestran en el formulario, tal como se aprecia en la Ilustración 22.

Concretamente, el usuario deberá introducir su correo electrónico (*email*), un nombre de usuario (*username*), una contraseña (*password*) y, por último, repetir la contraseña en el campo correspondiente (*repeat password*) para verificar que ambas coinciden.

Una vez rellenados todos los campos, bastará con pulsar el botón “Sign Up” para completar el proceso de registro y crear la cuenta.

Además, en la parte inferior de la pantalla se encuentra disponible un botón identificado como “Login”, que permite volver a la página de inicio de sesión, en caso de que el usuario lo desee.

Al igual que en la pantalla de *login*, las opciones de registro mediante cuentas externas como Google o Facebook permanecen deshabilitadas por el momento, siendo necesario completar el registro de forma manual.



**Evnet**

Your event, your community.  
Join now.

Email

Username

Password

Repeat Password

Sign up

or

Google Facebook

Have an account?

Login

***Ilustración 22: Página de registro (signup)***

Una vez iniciada la sesión correctamente, el usuario accederá a la página principal de la aplicación (*home*). Tal y como se muestra en la Ilustración 23, en esta pantalla se muestran los eventos creados por los usuarios a los que seguimos, ordenados cronológicamente según su fecha de creación, de forma que sea sencillo mantenerse al día de las nuevas actividades publicadas por los usuarios a los que seguimos.

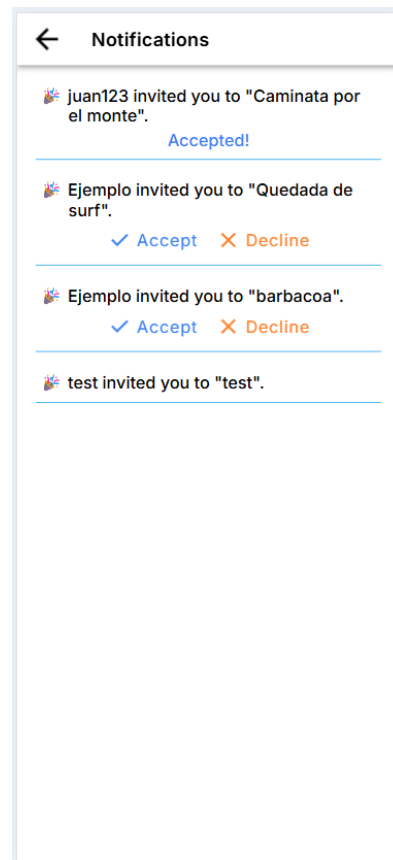
En la parte superior derecha de la pantalla principal se encuentra un icono de notificaciones, que permite acceder al apartado de notificaciones (Ilustración 24). Desde esta sección, el usuario puede consultar de manera centralizada las invitaciones a eventos que ha recibido, con la opción de aceptarlas o rechazarlas directamente, facilitando así la gestión de la participación en dichos eventos.

Finalmente, en la parte inferior de la pantalla se encuentra el *footer* o menú de navegación inferior, que permite al usuario desplazarse de forma rápida y cómoda entre las principales secciones de la aplicación. Concretamente, desde este menú se puede acceder (de izquierda a derecha) a la página de mis eventos, a la página para crear un nuevo evento,

a la página principal que estamos describiendo ahora mismo, al buscador de eventos y, por último, a la página de mi perfil.



*Ilustración 23: Página principal (home)*



*Ilustración 24: Página de notificaciones (notifications)*

Al pulsar sobre una publicación de un evento, la aplicación redirigirá al usuario a la página del propio evento, donde se muestra de forma detallada toda la información relacionada con el mismo: el título del evento, el nombre de usuario del creador junto a su foto de perfil, así como la ubicación, fecha y hora y una descripción completa del evento, tal y como se puede apreciar en la Ilustración 25.

En la cabecera de esta página, el usuario dispone de un icono en forma de flecha situado en la parte izquierda, que permite retroceder fácilmente a la vista anterior. En la parte derecha se encuentran tres iconos: uno para compartir el evento, función representada en la Ilustración 28, y otros dos destinados a acceder al álbum de fotos del evento y al chat del evento. Cabe destacar que estas dos últimas funciones (álbum y chat) aún no se encuentran implementadas, por lo que actualmente están desactivadas.

Por último, en la parte inferior de la página se presentan dos elementos importantes: un enlace para ver más información detallada del evento (*More info*) y un botón para unirse al evento. Al pulsar este botón, el usuario se unirá automáticamente al evento y, además, se le mostrará el importe económico que debe aportar para participar en él.



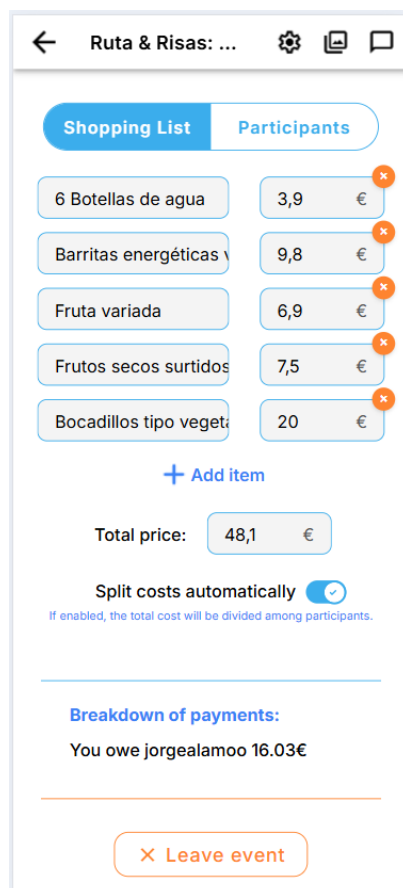
*Ilustración 25: Página de evento (event)*

Al pulsar sobre el enlace de más información (*More info*), se accede a una página adicional de detalles del evento. Esta página está organizada en dos apartados principales que permiten al usuario gestionar y consultar toda la información extra relacionada con el evento.

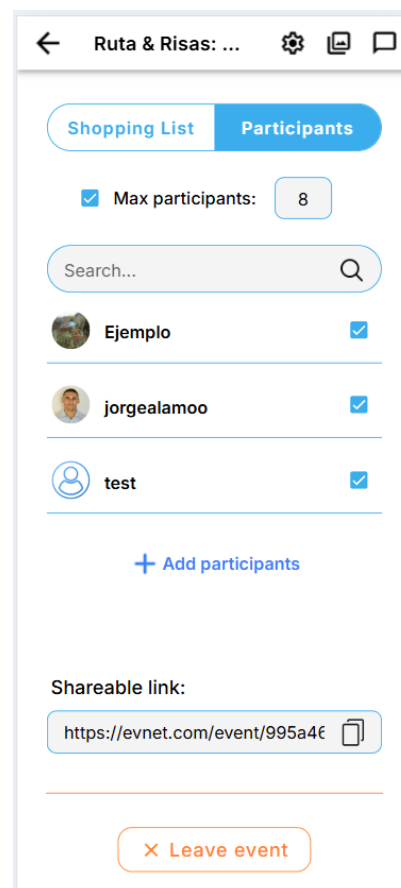
Por un lado, se encuentra la sección de la lista de la compra (*Shopping list*), representada en la ilustración 26. En esta sección se muestra de forma estructurada toda la lista de productos o ítems añadidos al evento, junto con sus respectivos precios, además del precio total de la compra. También se incluye una opción para dividir automáticamente los costes entre todos los participantes. Si esta opción se desactiva, se habilita un campo adicional que permite introducir manualmente el precio de entrada al evento. Además, en esta misma pantalla el usuario puede consultar la cantidad económica que debe aportar al

creador del evento y dispone de un botón en la parte inferior para salir del evento si así lo desea.

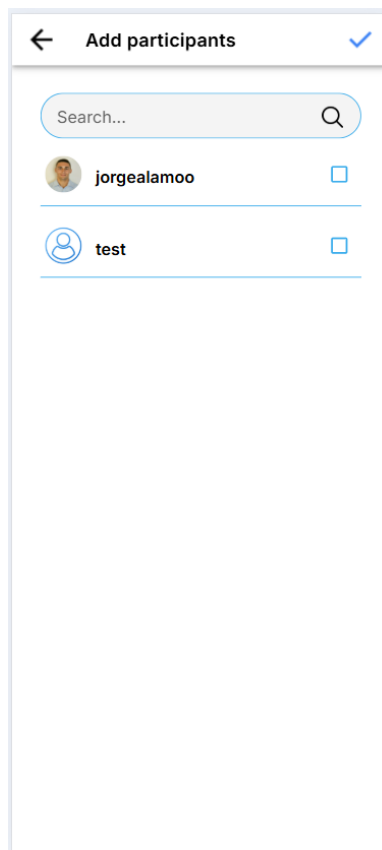
Por otro lado, la segunda parte corresponde al apartado de participantes (*Participants*), como se muestra en la ilustración 27. Aquí se listan los usuarios que ya forman parte del evento, junto con un campo editable que permite definir el máximo número de participantes admitidos. Este límite es opcional, por lo que puede desactivarse si no se quiere restringir el aforo del evento. Además, se incluye un enlace para compartir el evento con otros usuarios y una opción para invitar a nuevos participantes (*Add participants*). Al pulsar esta opción, se despliega un buscador que facilita encontrar usuarios e invitarlos al evento, tal como se muestra en la ilustración 28. Finalmente, en la parte inferior de esta sección también aparece un botón para que el usuario pueda salir del evento en cualquier momento.



**Ilustración 26:** Sección de lista de la compra de la página de más información del evento (event more info)



**Ilustración 27:** Sección de participantes de la página de más información del evento (event more info)



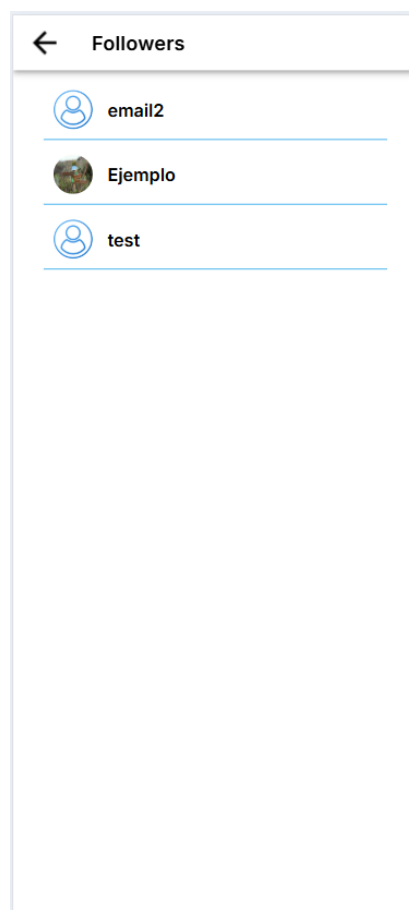
***Ilustración 28: Página de añadir participantes (add participants)***

A continuación, accederemos a nuestro perfil a través del menú de navegación inferior descrito anteriormente. Tal y como se puede observar en la Ilustración 29, en la página de mi perfil se muestra, en la parte superior, nuestra foto de perfil junto con nuestro nombre y apellidos y el nombre de usuario. Justo debajo encontramos los contadores de actividad, que indican el número de eventos creados, el número de seguidores y el número de seguidos. Si pulsamos sobre los contadores de seguidores o seguidos, se desplegará la lista correspondiente de usuarios para consultarla de forma detallada, como se ve en la Ilustración 30.

Más abajo, la página muestra de manera ordenada las publicaciones de los eventos creados por el usuario autenticado, organizadas por fecha de creación, permitiendo así tener una visión rápida de la actividad reciente.



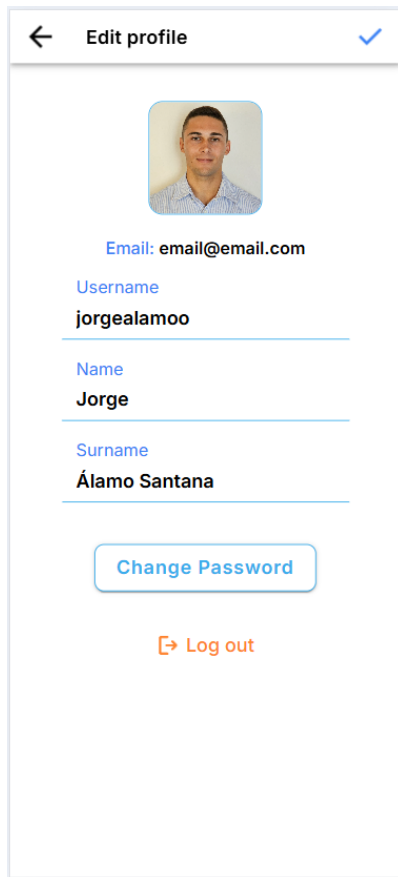
*Ilustración 29: Página de mi perfil (my profile)*



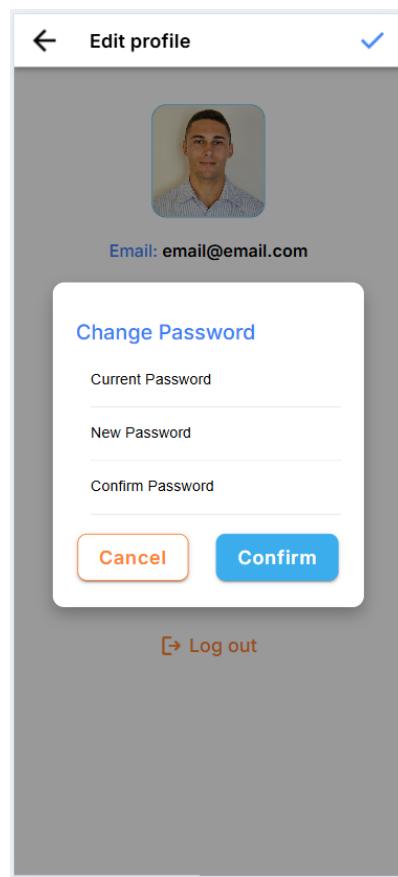
*Ilustración 30: Sección de seguidores o seguidos*

En la parte superior derecha de la pantalla, aparece un icono con forma de engranaje que, al pulsarlo, nos redirige a la página de edición del perfil (ilustración 31). En esta página, el usuario puede modificar su foto de perfil simplemente pulsando sobre ella, lo que abrirá el selector de medios para escoger una nueva imagen. Debajo de la foto se muestra, a modo informativo, el correo electrónico asociado a la cuenta, y justo después aparecen los campos editables para actualizar el nombre de usuario (*username*), el nombre (*name*) y los apellidos (*surname*).

Finalmente, en la parte inferior de la página de edición del perfil se encuentran dos botones: el primero permite cambiar la contraseña (*Change Password*), y al pulsarlo se abrirá una pantalla donde será necesario introducir la contraseña actual y después escribir la nueva contraseña y repetirla para confirmarla (Ilustración 32); el segundo botón permite cerrar sesión, finalizando así la sesión del usuario en la aplicación.



*Ilustración 31: Página de editar perfil (edit profile)*



*Ilustración 32: Desplegable de cambiar contraseña*

A continuación, tal y como se muestra en la ilustración 33, el perfil de usuario es visualmente muy similar a nuestro perfil personal, pero presenta algunas diferencias importantes: no aparece el icono del engranaje en la esquina superior derecha y, además, se incluye un botón de seguir (*Follow*) o siguiendo (*Following*) que permite al usuario pulsarlo para empezar a seguir o dejar de seguir a esa persona.



***Ilustración 33: Perfil de usuario (profile)***

Para crear un evento nuevo, podremos acceder directamente desde el menú de navegación inferior, pulsando en el icono del círculo con un más (+) en el centro. El proceso de creación del evento se divide en tres etapas, siendo la primera la que se muestra en la ilustración 34. En esta primera etapa, tendremos la opción de añadir imágenes pulsando sobre la sección superior que muestra el texto “Add Images”. Al hacerlo, se abrirá el selector de medios de nuestro dispositivo y, tras seleccionar las imágenes deseadas, estas se mostrarán en el orden escogido, con la posibilidad de eliminarlas o añadir nuevas en cualquier momento.

Justo debajo encontraremos varios campos para completar la información del evento: el título (Title), una descripción más detallada (*Event description*), las categorías a las que pertenece (*Categories*), la ubicación (*Location*), la fecha y hora del evento (*Event Date*) y, por último, el ajuste de privacidad del evento (*Event privacy*), donde podremos escoger si queremos que el evento sea público o privado. Una vez rellenados todos estos campos, podremos continuar con la creación del evento pulsando en el botón “Next”, situado en la parte derecha de la cabecera.

Create new event Next

  
Add Images

Title

Event description

Categories ▾

Location

Event Date:

dd/mm/yyyy  --:-- 

Select event privacy ▾

*Ilustración 34: Página de crear nuevo evento, primera etapa (create event)*

En la siguiente etapa del proceso de creación de un evento, se podrá definir y configurar la lista de la compra del evento. Como podemos ver en la Ilustración 35, esta lista está compuesta por distintos items, y cada uno de ellos cuenta con dos campos principales: el nombre del producto (*Item name*) y su precio, que por defecto aparece establecido en 0,00 €. Para añadir nuevos productos a la lista, basta con pulsar el botón “Add item”, mientras que, si queremos eliminar alguno, podremos hacerlo fácilmente pulsando en el icono de la equis (x) de color naranja que aparece junto al precio de cada item.

Cada vez que asignemos un precio a un producto, la aplicación recalculará de forma automática el precio total (*Total price*), que corresponde a la suma de todos los precios de los productos añadidos. Este campo también es editable, permitiendo que el usuario pueda fijar directamente el precio total sin necesidad de introducir precios individualmente para cada item, en caso de que así lo prefiera.

Además, se incluye la opción de activar o desactivar el reparto de costes automáticamente (*Split costs automatically*). Si se mantiene activada, el coste total se dividirá de manera equitativa entre todos los participantes apuntados al evento. En caso de desactivarla,

aparecerá un nuevo campo denominado precio de entrada (*Entry price*), donde se podrá establecer un importe fijo que cada participante deberá abonar. Una vez completada esta segunda etapa, podremos continuar con la creación del evento pulsando nuevamente el botón “Next” situado en la parte superior derecha de la pantalla.

← Create new event Next

Shopping list

|           |        |   |
|-----------|--------|---|
| Item name | 0,00 € | ✕ |
| Item name | 0,00 € | ✕ |
| Item name | 0,00 € | ✕ |

+ Add item

Total price: 0,00 €

Split costs automatically ☐

Entry price: 0,00 €

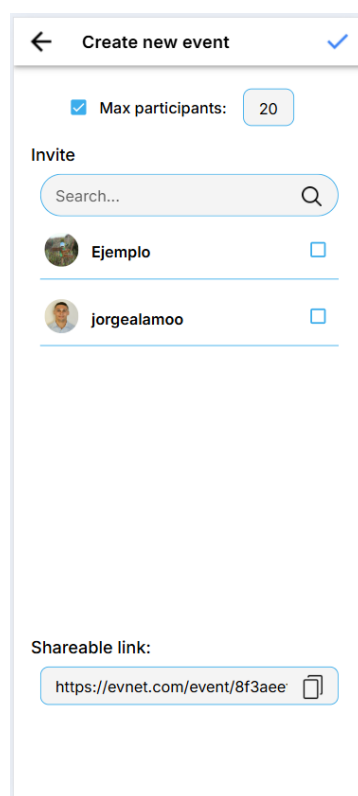
**Ilustración 35:** *Página de crear nuevo evento, segunda etapa (create event shopping list)*

En esta tercera etapa de creación del evento, mostrada en la Ilustración 36, podremos definir el número máximo de participantes (*Max participants*), que por defecto está establecido en 20. Si se prefiere organizar un evento sin límite de aforo, existe también la opción de desactivar esta restricción para que pueda apuntarse cualquier número de personas.

A continuación, la aplicación muestra una lista de usuarios a los que el usuario autenticado sigue, permitiendo seleccionar fácilmente a quiénes se desea invitar al evento. Además, se incluye un buscador para poder encontrar y añadir a otros usuarios por su nombre de usuario, ampliando así las posibilidades de personalizar la lista de invitados.

Por último, en la parte inferior de la pantalla, se encuentra disponible un enlace para compartir el evento (*Shareable link*), que podremos copiar pulsando el icono situado a la

derecha de dicho campo, facilitando así el envío del enlace por otras vías externas a la aplicación. Una vez finalizado todo el proceso de configuración, tan solo quedará pulsar el icono del tic (✓) que aparece en la parte derecha de la cabecera para crear definitivamente el evento.



***Ilustración 36: Página de crear nuevo evento, tercera etapa (create event participants)***

Las dos últimas pantallas de la aplicación son el buscador y mis eventos, a las que podemos acceder cómodamente desde el menú de navegación inferior, pulsando sobre los iconos de la lupa y del calendario, respectivamente.

Tal y como se aprecia en la Ilustración 37, en la página del buscador se muestran por defecto los últimos eventos creados por la comunidad. Si el usuario desea buscar un evento concreto, podrá hacerlo utilizando la barra de búsqueda situada en la parte superior de la pantalla, introduciendo el título del evento como criterio. Para mostrar los resultados de la búsqueda, bastará con pulsar el icono de la lupa de la barra o simplemente presionar la tecla “Enter”.

Por otro lado, al acceder a la página de mis eventos, se mostrarán todos los eventos en los que el usuario está actualmente inscrito, como podemos ver en la Ilustración 38. En caso

de que el usuario no esté apuntado a ningún evento, aparecerá en pantalla un mensaje informativo indicándolo de manera clara.



*Ilustración 37: Página del buscador de eventos (search)*



*Ilustración 38: Página de mis eventos (my events)*

## Capítulo 7: Conclusiones y trabajos futuros

### 7.1. Conclusiones

El Proyecto de Fin de Grado presentado ha consistido en el diseño, desarrollo e implementación de *Evnet*, una aplicación móvil que integra funciones de red social y gestión colaborativa de eventos. Durante el proyecto se ha logrado implementar un sistema completo para la creación y gestión de eventos, tanto públicos como privados, que incluye invitaciones mediante enlaces únicos, control de aforo y administración de asistentes, cubriendo de forma satisfactoria los requisitos de usuario y de sistema definidos en la fase inicial.

La dimensión social de *Evnet* se ha visto reforzada con la incorporación de un sistema de seguidores y seguidos, un *feed* principal que muestra los eventos de nuestro círculo y un módulo de exploración de eventos públicos. Estos elementos han potenciado la interacción comunitaria y la capacidad de descubrimiento de actividades, demostrando la viabilidad de un modelo que combina red social y gestión de eventos. Asimismo, la adopción de una arquitectura modular y escalable basada en Ionic, Angular y Supabase, junto a un *backend* robusto en PostgreSQL, asegura la compatibilidad multiplataforma, la mantenibilidad y la posibilidad de futuras extensiones sin comprometer el rendimiento.

A nivel técnico y personal, este proyecto ha supuesto una oportunidad de aprendizaje muy enriquecedora. Gracias a su desarrollo he podido profundizar y descubrir nuevas funcionalidades avanzadas de Ionic que desconocía previamente, lo que me ha permitido mejorar la calidad y usabilidad de la aplicación. Además, ha sido mi primera experiencia trabajando con Supabase como sistema de autenticación y base de datos, aprendiendo a integrarlo con el resto de la arquitectura de forma eficiente, algo que me ha resultado especialmente interesante por su potencia y sencillez de uso.

Si bien no todas las funcionalidades pensadas en un inicio pudieron ser implementadas (principalmente por limitaciones de tiempo) se optó por priorizar aquellas que garantizaran el correcto funcionamiento básico de la aplicación y sentasen las bases para seguir ampliándola en el futuro. Gracias a estas decisiones de diseño y priorización, *Evnet* se entrega como una aplicación plenamente funcional, abierta a mejoras y evoluciones que puedan enriquecer la experiencia del usuario.

En definitiva, este proyecto no solo me ha permitido aplicar y consolidar los conocimientos adquiridos durante la carrera, sino también afrontar nuevos retos técnicos, mejorar mis habilidades de diseño, desarrollo y organización, y demostrar la capacidad de crear una aplicación completa, funcional y preparada para seguir creciendo.

## **7.2. Trabajos futuros**

La aplicación Evnet presenta un gran potencial para ser ampliada con nuevas funcionalidades y mejoras que enriquecerían aún más la experiencia de uso y harían la herramienta más completa y versátil.

Entre las ideas más destacadas para trabajos futuros se encuentra la incorporación de un sistema de permisos y roles dentro de los eventos. Este permitiría, por ejemplo, designar administradores o coadministradores del evento que puedan gestionar tareas específicas, como editar la lista de la compra, invitar o eliminar participantes, o modificar detalles del evento, mientras que otros usuarios mantendrían permisos más limitados. Este sistema ofrecería un mayor control sobre la organización colaborativa y facilitaría la distribución de responsabilidades entre varios usuarios.

Otra mejora relevante sería permitir que los usuarios puedan hacer aportaciones a la lista de la compra, recalculando automáticamente el precio total del evento en función del valor de lo aportado por cada participante.

En cuanto a la dimensión social de la aplicación, un importante trabajo futuro sería el desarrollo del álbum de fotos compartido, donde los asistentes pudieran subir y compartir sus imágenes del evento, generando así un recuerdo colaborativo. Del mismo modo, la implementación de un chat específico para cada evento permitiría a los usuarios comunicarse de forma más directa y sencilla, facilitando la organización previa y el intercambio de impresiones durante y después del evento.

Por último, otra idea interesante sería integrar funcionalidades basadas en inteligencia artificial, como por ejemplo un asistente capaz de generar automáticamente una propuesta de lista de la compra a partir de un tema o tipo de evento y un presupuesto aproximado, simplificando así la planificación.

Estas, entre muchas otras posibles extensiones, reflejan el amplio margen de crecimiento que ofrece Evnet y su potencial para evolucionar hacia una herramienta aún más potente y flexible.

## Bibliografía

- [1] **Eventbrite**. (s.f.). Eventbrite. Recuperado el 17 de junio de 2025, de <https://www.eventbrite.es/>
- [2] **Planeador de Eventos y Fiestas – Aplicaciones en Google Play**. (s.f.). TopEvent - Apps en Google Play. Recuperado el 17 de junio de 2025, de <https://play.google.com/store/apps/details?id=app.topevent.android&hl=es>
- [3] **Meetup**. (s.f.). Meetup. Recuperado el 17 de junio de 2025, de <https://www.meetup.com/es-ES/>
- [4] **Excellent Web World**. (s.f.). Eventbrite App: Best App of the Week. Recuperado el 17 de junio de 2025, de <https://www.excellentwebworld.com/best-app-of-the-week/eventbrite-app/>
- [5] **TreceBits**. (2020, junio 11). Qué es Meetup y cómo funciona. Recuperado el 18 de junio de 2025, de <https://www.trecebits.com/que-es-meetup-como-funciona/>
- [6] **Wikipedia**. (2023, julio 5). Scrum (desarrollo de software). Recuperado el 22 de junio de 2025, de [https://es.wikipedia.org/wiki/Scrum\\_\(desarrollo\\_de\\_software\)](https://es.wikipedia.org/wiki/Scrum_(desarrollo_de_software))
- [7] **ReactiveX**. (s.f.). RxJS. Recuperado el 27 de junio de 2025, de <https://rxjs.dev/>
- [8] **Mermaid**. (s.f.). Mermaid Live Editor. Recuperado el 3 de julio de 2025, de <https://mermaid.live/edit>
- [9] **GitHub**. (s.f.). GitHub. <https://github.com/>
- [10] **Supabase**. (s.f.). Supabase. <https://supabase.com/>
- [11] **Ionic Framework**. (s.f.). Ionic Framework. <https://ionicframework.com/>
- [12] **Angular**. (s.f.). Angular. <https://angular.dev/>
- [13] **OpenAI**. (s.f.). ChatGPT. <https://chatgpt.com/>
- [14] **Uiverse**. (s.f.). Silent Otter 72 by alexruix. Recuperado el 16 de abril de 2025, de <https://uiverse.io/alexruix/silent-otter-72>
- [15] **Organización de las Naciones Unidas**. (s. f.). Objetivos de desarrollo sostenible. Naciones Unidas. Recuperado el 17 de junio de 2025, de <https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/>