UNIVERSIDAD DE MÁLAGA

E.T.S. INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE MÁLAGA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
GRADUADO EN INGENIERÍA INFORMÁTICA

APLICACIÓN iOS PARA FOMENTAR GRUPOS COLABORATIVOS Y ACTIVIDADES

iOS APPLICATION TO PROMOTE COLLABORATIVE GROUPS AND ACTIVITIES

Realizado por
Matvey Dergunov Bushmanov

Tutorizado por
David Santo Orcero

Departamento
Lenguajes y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA
MÁLAGA, SEPTIEMBRE DE 2023

Fecha defensa: X de Septiembre de 2023

Resumen

En la actualidad, existe una gran cantidad de aplicaciones móviles diseñadas para facilitar las relaciones entre personas. Estas aplicaciones, conocidas como redes sociales, han adquirido una enorme popularidad entre adultos y jóvenes en la última década, transformando la forma en que las personas interactúan entre sí.

Por un lado, encontramos redes sociales centradas en publicaciones, como Facebook, Twitter e Instagram, donde los usuarios asumen roles de consumidores y productores. Por otro lado, existen redes sociales orientadas a la comunicación, como WhatsApp y Telegram, en las que podemos identificar distintos tipos de interacción: usuario a usuario, usuario a grupo y grupo a grupo.

No obstante, hay pocas aplicaciones dedicadas a fomentar la colaboración y actividades en grupo para alcanzar objetivos comunes entre las personas, donde los usuarios puedan encontrar a otros con intereses similares y realizar actividades conjuntas, como por ejemplo, formar una banda de música.

El propósito de este trabajo de fin de grado es desarrollar una aplicación móvil que permita a los usuarios encontrar personas con intereses afines y llevar a cabo actividades en grupo. Para ello, se ha creado una aplicación móvil para iOS utilizando desarrollo nativo y el lenguaje de programación Swift. Algunas de las características de la app incluyen: registro de usuarios, creación de grupos, búsqueda de grupos, chat de grupo, entre otras.

Palabras clave: Redes sociales, Aplicaciones móviles, Comunicación, Colaboración, Interacción, Desarrollo nativo, Swift, iOS, Actividades en grupo, Intereses comunes.

Abstract

Currently, there is a wide range of mobile applications designed to facilitate relationships between people. These applications, known as social networks, have gained immense popularity among adults and young people in the last decade, transforming the way people interact with each other.

On one hand, we find social networks focused on posts, such as Facebook, Twitter, and Instagram, where users assume roles as consumers and producers. On the other hand, there are social networks geared towards communication, like WhatsApp and Telegram, in which we can identify different types of interaction: user-to-user, user-to-group, and group-to-group.

However, there are few applications dedicated to promoting collaboration and group activities to achieve common goals among people, where users can find others with similar interests and engage in joint activities, such as forming a music band.

The purpose of this final degree project is to develop a mobile application that allows users to find people with similar interests and carry out group activities. To achieve this, a mobile application for iOS has been created using native development and the Swift programming language. Some of the features of the app include: user registration, group creation, group search, group chat, and more.

Keywords: Social Networks, Mobile Applications, Communication, Collaboration, Interaction, Native Development, Swift, iOS, Group Activities, Common Interests.

Índice general

Resumen I

Abstract III

1. Introducción 1
- 1.1. Motivación . . . 1
- 1.2. Objetivos . . . 1
- 1.3. Estructura del documento . . . 2

2. Estado del arte 3
- 2.1. Introducción . . . 3
- 2.2. Estado del arte . . . 3
  - 2.2.1. Meetup . . . 3
- 2.3. Conclusión . . . 5

3. Introducción 7
- 3.1. Swift . . . 7
- 3.2. Firebase . . . 8
- 3.3. XCode . . . 8
- 3.4. Visual Paradigm . . . 9

4. Análisis del sistema 11
- 4.1. Requisitos funcionales . . . 11
- 4.2. Requisitos no funcionales . . . 12
- 4.3. Casos de uso . . . 13

5. Diseño del sistema 23
  - 5.0.1. Modelo de datos . . . 23
  - 5.0.2. Navegación y vistas . . . 25
  - 5.0.3. Estructura del proyecto . . . 31
  - 5.0.4. Crear o unirse a evento . . . 32

6. Conclusiones y Líneas Futuras 35
- 6.1. Conclusiones . . . 35
- 6.2. Líneas futuras . . . 35

6.2.1. Mejoras en el chat 35
6.2.2. Automatización del sistema de reporte 36
6.2.3. Mejoras en la planificación de actividades 36
6.2.4. Aplicación multiplataforma 36
6.2.5. Internacionalización 36

7. Bibliografía **37**

Apéndice A. Manual de Usuario **39**
A.1. Registro 39
A.2. Iniciar sesión 40
A.3. Unirse a un grupo 41
A.4. Chatear 42
A.5. Abandonar grupo 43
A.6. Ajustes de usuario 43
A.7. Crear y modificar grupo 44
A.8. Crear y modificar eventos 45
A.9. Unirse y abandonar evento 45

1

Introducción

1.1 Motivación

Hoy en día, las aplicaciones móviles se han vuelto herramientas esenciales en nuestras vidas cotidianas, ya que facilitan la comunicación, la búsqueda de información, entre otras funciones. En este contexto, la colaboración y las actividades en grupo son aspectos cada vez más relevantes, y las personas buscan aplicaciones que les permitan conectarse con otros usuarios que compartan sus intereses y trabajar juntos en proyectos específicos.

Por otro lado, las redes sociales y las aplicaciones de comunicación han experimentado un crecimiento significativo en la última década, pero aún hay oportunidades para mejorar y ofrecer al usuario una experiencia en función de sus intereses y objetivos. Es aquí donde surge la necesidad de nuestra aplicación móvil, cuya finalidad es facilitar la conexión entre personas con intereses afines y promover la realización de actividades en grupo.

1.2 Objetivos

El objetivo de este Trabajo de Fin de Grado es desarrollar una aplicación móvil para iOS que permita a los usuarios encontrar personas con intereses afines y llevar a cabo actividades en grupo de manera efectiva y adaptada a sus necesidades. Algunas de las características de la app son: registro de usuarios, creación de grupos, búsqueda de grupos, chat de grupo, creación de eventos, entre otras.

1.3 Estructura del documento

- Capítulo 1: Introducción. En este primer capítulo se presenta la motivación, los objetivos y la estructura del documento.
- Capítulo 2: Estado del arte. En el segundo capítulo, se realiza un análisis de las aplicaciones móviles existentes en el ámbito de la colaboración y actividades en grupo.
- Capítulo 3: Tecnologías utilizadas. En el tercer capítulo, se describen las tecnologías empleadas en el desarrollo de la aplicación.
- Capítulo 4: Análisis del sistema. En el cuarto capítulo, se detallan las funcionalidades de la aplicación y cómo está diseñado el sistema.
- Capítulo 5: Diseño del sistema. En el quinto capítulo, se elabora sobre las decisiones que se han tomado a la hora de arquitectura y diseño de la aplicación.
- Capítulo 6: Conclusión. En el sexto capítulo, se realiza una valoración del proyecto y se comentan las posibles líneas futuras.
- Capítulo 7: Bibliografía. En el séptimo capítulo, se enumeran las fuentes consultadas para el desarrollo de la aplicación.

2

Estado del arte

2.1 Introducción

En este capítulo, se abordará el estado del arte en cuanto a aplicaciones móviles para la colaboración y actividades en grupo, destacando las características y funcionalidades más relevantes de las soluciones existentes. Además, se discutirán las áreas donde estas aplicaciones presentan oportunidades de mejora y cómo nuestra propuesta busca abordar dichas brechas.

2.2 Estado del arte

2.2.1 Meetup

Meetup es una plataforma en línea que se fundó en junio de 2002 por Scott Heiferman y cuatro co-fundadores. La idea de Meetup surgió de la experiencia de Heiferman de conocer a sus vecinos en la ciudad de Nueva York por primera vez después de los ataques del 11 de septiembre en las Torres Gemelas. La plataforma se creó con el objetivo de fomentar la comunidad y la interacción en persona.

Los usuarios de Meetup pueden buscar grupos en su localidad que se ajusten a sus intereses. Cada grupo está dirigido por un organizador, que puede programar eventos y administrar la página del grupo. Los miembros del grupo pueden ver los detalles del evento, incluyendo la ubicación y la hora, y RSVP para confirmar su asistencia. La aplicación también ofrece la posibilidad de interactuar con otros miembros a través de la sección de comentarios de cada evento.

- **Organización de grupos:** Los usuarios pueden formar y unirse a grupos basados en intereses comunes.
- **Planificación de eventos:** Los organizadores pueden planificar eventos dentro de su grupo, estableciendo la fecha, la hora, el lugar y los detalles del evento.
- **RSVP para eventos:** Los usuarios pueden confirmar su asistencia a los eventos y ver quién más planea asistir.
- **Comunicación:** Los organizadores y los miembros del grupo pueden comunicarse a través de mensajes dentro de la plataforma, permitiendo la coordinación y el intercambio de información.
- **Categorización de grupos:** Los grupos pueden ser categorizados en base a una variedad de intereses y temas, facilitando a los usuarios encontrar grupos que coincidan con sus intereses.
- **Calendario de eventos:** Los eventos futuros se muestran en un calendario, permitiendo a los usuarios ver los próximos eventos de un vistazo.
- **Eventos locales y virtuales:** Meetup permite la organización de eventos tanto en persona como virtuales, ofreciendo flexibilidad a los usuarios y organizadores.
- **Fotos y comentarios de eventos:** Después de un evento, los asistentes pueden publicar fotos y comentarios, proporcionando un registro de la actividad del grupo y permitiendo a los miembros interactuar después del evento.

Figura 2.1: Logo Swift

2.3 Conclusión

Meetup se centra en fomentar las interacciones en el mundo real a través de experiencias compartidas. Sin embargo, su enfoque es más general y no está diseñado específicamente para la colaboración en proyectos concretos o para la formación de grupos con el objetivo de realizar actividades específicas a largo plazo. Además, a pesar de su popularidad, Meetup no está exento de críticas, particularmente en lo que respecta a la experiencia del usuario y a las tarifas de suscripción para los organizadores de grupos.

Nuestra propuesta busca abordar estas brechas al proporcionar una plataforma más personalizada y simple, orientada a la colaboración en proyectos y actividades específicas. A diferencia de Meetup, nuestra aplicación se enfocará en la formación de grupos con objetivos a largo plazo y proporcionará herramientas robustas para la gestión de proyectos y la comunicación en grupo.

3

Introducción

3.1 Swift

Swift[1] es un lenguaje de multiparadigma de código abierto lanzado en 2014 por Apple, orientado al desarrollo de aplicaciones para iOS y macOS. Swift esta fuertemente enfocado en facilitar la experiencia al desarrollador a través de seguridad, eficiencia y expresividad del lenguaje. Por ejemplo, la memoria se gestiona automáticamente mediante un recuento de referencias preciso y determinista, manteniendo al mínimo el uso de la memoria sin la sobrecarga de la recolección de basura. Ofrece la posibilidad de escribir código concurrente con simples palabras clave incorporadas que definen el comportamiento asíncrono, haciendo que el código sea más legible y menos propenso a errores.

Figura 3.1: Logo Swift

Swift se propone como un reemplazo para los lenguajes basados en C (C, C++ y Objective-C). Como tal, Swift es comparable a esos lenguajes en rendimiento para la mayoría de las tareas. Aún así, ofrece interoperabilidad con Objective-C

3.2 Firebase

Firebase es una plataforma de desarrollo de aplicaciones creada por Google, diseñada para ayudar a los desarrolladores a crear, mejorar y hacer crecer sus aplicaciones de manera más eficiente. Proporciona una variedad de herramientas y servicios que cubren varias áreas del desarrollo de aplicaciones, incluyendo alojamiento en la nube, bases de datos en tiempo real, autenticación de usuarios, análisis de aplicaciones, pruebas A/B, etc.

Figura 3.2: Logo Visual Paradigm

Una de las características más notables de Firebase es su base de datos en tiempo real, que permite a los desarrolladores almacenar y sincronizar datos entre usuarios en tiempo real. Esto es especialmente útil para aplicaciones que requieren actualizaciones en tiempo real, como juegos en línea, aplicaciones de chat, y aplicaciones colaborativas.

Además, Firebase también incluye una potente autenticación de usuarios, permitiendo a los desarrolladores implementar sistemas de inicio de sesión seguros y personalizados sin tener que construirlos desde cero. Firebase admite varios métodos de autenticación, incluyendo correo electrónico/contraseña, autenticación telefónica, Google, Facebook, Twitter, y GitHub.

Otra de las características a mencionar es Firebase Analytics, que ofrece a los desarrolladores una visión detallada del uso y el comportamiento de sus aplicaciones. Esta herramienta puede ayudar a los desarrolladores a entender cómo los usuarios interactúan con sus aplicaciones, lo que puede ser útil para la toma de decisiones y la optimización del rendimiento de la aplicación.

3.3 XCode

Xcode es un entorno de desarrollo integrado (IDE) desarrollado por Apple para el desarrollo de software en Mac. Es la principal herramienta utilizada para desarrollar software para iOS, macOS, watchOS y tvOS. Xcode proporciona una amplia variedad de herramientas para el desarrollo de software, incluyendo compiladores para el lenguaje de programación Swift y Objective-C, herramientas para la construcción de interfaces de usuario, y utilidades para la gestión de dispositivos, el despliegue de aplicaciones y la gestión de la App Store.

Además de sus capacidades de programación, Xcode también ofrece una variedad de características para facilitar el desarrollo de software. Por ejemplo, el depurador de Xcode permite a los desarrolladores examinar el estado de un programa en cualquier punto de su ejecución, mientras que su perfilador ayuda a identificar cuellos de botella en el rendimiento del software. Xcode también incluye un editor de interfaces de usuario gráficas, que permite a los desarrolladores diseñar las interfaces de sus aplicaciones de manera visual.

Xcode también se integra con otras herramientas y servicios de Apple. Por ejemplo, se puede utilizar junto con el sistema de gestión de versiones de Git para el control de versiones, y con el servicio de integración continua de Apple, Xcode Server, para la automatización de las pruebas y la construcción de aplicaciones.

En resumen, Xcode es una herramienta integral para el desarrollo de software en las plataformas de Apple, proporcionando todo lo necesario para diseñar, desarrollar, probar y desplegar aplicaciones.

3.4 Visual Paradigm

Visual Paradigm es una aplicación de software diseñada para equipos de desarrollo de software para modelar sistemas de información de negocios y gestionar procesos de desarrollo. Además del soporte para modelado, esta tecnología proporciona generación de informes y capacidades de ingeniería de código, incluyendo generación de código. Esta tecnología puede revertir diagramas de ingeniería a partir del código y proporcionar ingeniería de ida y vuelta para varios lenguajes de programación. Visual Paradigm cuenta con diagramas de Lenguaje Unificado de Modelado (UML), Diagrama de Relaciones de Entidades (ERD), y diagramas de Mapeo de Objetos Relacionales (ORMD) esenciales en el diseño de sistemas y bases de datos.

Figura 3.3: Logo Visual Paradigm

4

Análisis del sistema

En esta sección, se llevará a cabo una revisión detallada y especificación de los requisitos de la aplicación. Se comienza el análisis con los requisitos funcionales, los cuales detallan las funciones y características que el sistema brinda al usuario. También se definen los requisitos no funcionales, que describen cómo el sistema debe operar y cuál debe ser su comportamiento. Para concluir esta sección, se describirán los casos de uso necesarios para satisfacer los requisitos funcionales. Este análisis es crucial para obtener información de alta calidad y minimizar el potencial de errores en el diseño o la implementación de la aplicación.

4.1 Requisitos funcionales

- **RF0. Registro**: El usuario se dará de alta en la aplicación aportando los datos necesarios como son el correo electrónico, contraseña, nombre de usuario y localización preferida.
- **RF1. Inicio de sesión**: El usuario, con el correo electrónico y contraseña creada durante el registro, podrá iniciar sesión en la aplicación.
- **RF2. Restablecer contraseña**: El usuario, proporcionando su correo electrónico, podrá solicitar restablecer su contraseña, a través de un correo que recibe en su correo electrónico.
- **RF3. Cerrar sesión**: El usuario podrá cerrar sesión en la aplicación.

- **RF4. Editar perfil**: El usuario tendrá la posibilidad de modificar datos como el nombre de usuario y la localización preferida.
- **RF5. Crear grupo**: El usuario podrá crear un nuevo grupo de actividad proporcionando datos como lugar, tipo de grupo, límite de integrantes y descripción del grupo.
- **RF6. Explorar grupos**: El usuario podrá a través de un mapa, explorar los grupos disponibles cercanos, pudiendo ver sobre estos la información básica como es el lugar, número actual de integrante y descripción del grupo.
- **RF7. Unirse a un grupo**: El usuario podrá unirse a un grupo siempre y cuando haya espacio disponible.
- **RF8. Salir del grupo**: El usuario podrá salir del grupo en el momento que sea, siempre y cuando no sea el último miembro.
- **RF9. Visualizar listado de grupos**: El usuario podrá visualizar de un vistazo rápido los grupos a los que pertenece y ver el número de mensajes no leídos correspondiente al chat del grupo.
- **RF10. Chat grupal**: El usuario, podrá usar el correspondiente chat de grupo para mandar mensajes en tiempo real en el chat grupal asociado al grupo.
- **RF11. Crear evento**: El usuario propietario del grupo, podrá crear y planificar una actividad a la que los demás miembros del grupo podrán unirse.
- **RF12. Unirse a evento**: El usuario podrá unirse a un evento creado por el propietario del grupo.

4.2 Requisitos no funcionales

- **RNF0. Confirmación de correo electrónico**: Como medida de seguridad, se requerirá que los usuarios confirmen su dirección de correo electrónico antes de poder acceder a la aplicación. Esto ayudará a prevenir el uso indebido y garantizará que los correos enviados por la aplicación lleguen a los usuarios correctos.
- **RNF1. Límite de creación de grupos**: Para evitar la sobrecarga del sistema, cada usuario solo podrá crear un máximo de dos grupos.
- **RNF2. Límite de grupos**: Un usuario solo puede ser miembro de 4 grupos simultáneamente.

- **RNF3. Rendimiento**: La aplicación debe ser capaz de manejar un gran número de usuarios simultáneamente sin degradar la calidad del servicio.
- **RNF4. Seguridad**: Los datos del usuario, incluyendo información personal y contraseñas, deben ser almacenados de manera segura para prevenir el acceso no autorizado.
- **RNF5. Internacionalización**: La aplicación deberá estar disponible en inglés y español para facilitar su uso a un público más amplio.
- **RNF6. Modo claro/oscuro**: La aplicación deberá permitir a los usuarios cambiar entre un modo claro y oscuro, mejorando así la experiencia del usuario y permitiéndole adaptar la interfaz a sus preferencias personales.
- **RNF7. Eliminación de cuenta**: El usuario podrá eliminar su cuenta en cualquier momento, eliminando así todos los datos asociados a la misma.

4.3 Casos de uso

Los casos de uso representan interacciones significativas entre los usuarios (conocidos como «actores») y el sistema que estamos diseñando. Estos casos nos permiten entender, documentar y comunicar cómo se espera que funcione el sistema en diversas situaciones, siempre desde la perspectiva del usuario.

Cada caso de uso describe una secuencia de eventos que ocurren durante una interacción específica del usuario con el sistema, mostrando cómo el sistema responde a las acciones del usuario. Los casos de uso son útiles para capturar y especificar los requisitos funcionales del sistema.

A continuación, se presentan los casos de uso más relevantes para nuestra aplicación.

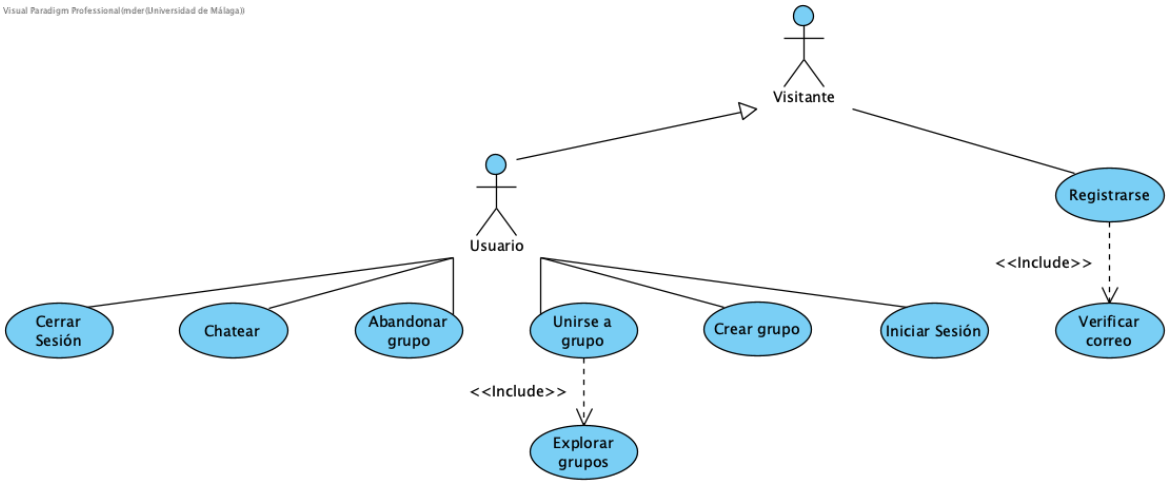

Figura 4.1: Diagrama casos de uso

Caso de uso	Registrarse (CU01)
Actor	Visitante
Descripción	El visitante se dará de alta en la aplicación aportando los datos necesarios como son el correo electrónico, contraseña, nombre de usuario y localización preferida.
Precondiciones	El usuario debe tener la aplicación instalada
Postcondiciones	El usuario queda registrado en el sistema
Flujo normal	1. El visitante abre la aplicación 2. Selecciona «Darse de alta» 3. Rellena el formulario con los datos necesarios 4. Acepta las condiciones de uso 5. Pulsa «Confirmar» 6. El sistema muestra mensaje de espera de confirmación de correo 7. El usuario accede a su buzón y confirma el correo 8. El usuario vuelve a la aplicación y pulsa «Correo confirmado» 9. El sistema comprueba el estado de confirmación 10. El usuario es redirigido a la pantalla principal
Flujo alternativo	1.1 El correo ya está registrado 1.1.1 El sistema muestra un mensaje de error 1.1.2 El visitante introduce un correo diferente 1.2 Los datos introducidos son inválidos 1.2.1 El sistema muestra un mensaje de error 1.2.2 El visitante corrige los datos

Cuadro 4.1: Especificaciones del caso de uso: Registrarse (CU01)

Caso de uso	Iniciar Sesión (CU02)
Actor	Usuario
Descripción	El usuario, con el correo electrónico y contraseña creada durante el registro, podrá iniciar sesión en la aplicación.
Precondiciones	El usuario tiene que tener la aplicación abierta y cuenta registrada
Postcondiciones	El usuario tiene la sesión iniciada y se encuentra en la pantalla principal
Flujo normal	1. El usuario introduce su correo y contraseña 2. El usuario realiza click en «Iniciar sesión» 3. El sistema comprueba los datos y realiza el inicio de sesión 4. El usuario es redirigido a la pantalla principal
Flujo alternativo	3.1 Correo electrónico no confirmado 3.1.1 El sistema comprueba que el correo no está confirmado, muestra una pantalla 3.1.2 El usuario da click en «Reenviar confirmación» 3.1.3 El sistema envía un correo de confirmación 3.1.4 El usuario confirma el correo electrónico 3.1.5 El usuario vuelve a iniciar sesión

Cuadro 4.2: Especificaciones del caso de uso: Iniciar Sesión (CU02)

Caso de uso	Cerrar Sesión (CU3)
Actor	Usuario
Descripción	El usuario podrá cerrar su sesión en la aplicación desde la configuración de perfil.
Precondiciones	El usuario tiene que tener la sesión iniciada y se encuentra en la configuración de perfil
Postcondiciones	El usuario tiene la sesión cerrada
Flujo normal	1. El usuario se encuentra en la configuración de perfil 2. El usuario selecciona «Cerrar Sesión» 3. El sistema muestra una pantalla de confirmación 4. El usuario selecciona «Confirmar» 5. El sistema cierra la sesión del usuario
Flujo alternativo	3.1 El usuario selecciona «Cancelar» en la pantalla de confirmación 3.1.1 El sistema cancela el cierre de sesión 3.1.2 El usuario permanece en la configuración de perfil

Cuadro 4.3: Especificaciones del caso de uso: Cerrar Sesión (CU3)

Caso de uso	Unirse a grupo (CU04)
Actor	Usuario
Descripción	El usuario podrá explorar los grupos disponibles cercanos a través de un mapa, pudiendo ver la información básica de estos como el lugar, número actual de integrantes y descripción del grupo. El usuario tiene la posibilidad de unirse al grupo.
Precondiciones	El usuario se encuentra en la vista de mapa
Postcondiciones	Queda registrada la membresía del usuario al grupo en el sistema
Flujo normal	1. El usuario explora el mapa buscando grupos cercanos 2. El usuario selecciona un grupo de interés 3. El sistema muestra una pantalla de detalles sobre el grupo, con un botón de Unirse 4. El usuario selecciona el botón de «Unirse» 5. El sistema registra en la base de datos la nueva membresía 6. El usuario es redirigido al chat del grupo
Flujo alternativo	3.1 El sistema comprueba que el usuario ya forma parte del grupo 3.1.1 El sistema ofrece la opción de «Salir del grupo» 3.1.2 El usuario cierra la pantalla de detalles del grupo 3.1.3 El usuario sigue explorando grupos en la vista de mapa

Cuadro 4.4: Especificaciones del caso de uso: Unirse a un grupo (CU04)

Caso de uso	Abandonar grupo (CU05)
Actor	Usuario
Descripción	El usuario podrá salir del grupo en cualquier momento, en caso de que sea el último miembro al momento de salir, tendrá la opción de eliminar el grupo
Pre-condiciones	El usuario es miembro del grupo
Post-condiciones	El usuario deja de ser miembro del grupo
Flujo normal	1. El usuario selecciona el listado de grupos 2. Selecciona el grupo que quiere abandonar 3. En el chat, pulsa sobre el nombre del grupo 4. El sistema muestra los detalles del grupo 5. El usuario pulsa en «Abandonar» 6. El sistema muestra una alerta de confirmación 7. El usuario pulsa «Confirmar» 8. El usuario es redirigido al listado de grupos
Flujo alternativo	6.1 Es el último miembro del grupo 6.1.1 El sistema muestra un mensaje de eliminar grupo 6.1.2 El usuario confirma la acción 6.1.3 El sistema elimina el grupo 6.1.4 El usuario es redirigido al listado de grupos

Cuadro 4.5: Especificaciones del caso de uso: Abandonar grupo (CU05)

Caso de uso	Crear grupo (CU06)
Actor	Usuario
Descripción	El usuario podrá crear un nuevo grupo de actividad proporcionando datos como lugar, tipo de grupo, límite de integrantes y descripción del grupo.
Pre-condiciones	El usuario tiene la sesión iniciada
Post-condiciones	Un nuevo grupo queda registrado en la base de datos
Flujo normal	1. El usuario selecciona la vista de listado de grupos 2. El usuario pulsa en el botón «Crear grupo» 3. El sistema presenta un formulario 4. El usuario rellena el formulario con los datos del nuevo grupo 5. El usuario pulsa en el botón «Crear» 6. El sistema comprueba los campos 7. El sistema registra el grupo en la base de datos 8. El usuario es redirigido al listado de grupos
Flujo alternativo	4.1 Los datos introducidos son inválidos 4.1.1 El sistema muestra un mensaje de error 4.1.2 El usuario corrige los datos

Cuadro 4.6: Especificaciones del caso de uso: Crear grupo (CU06)

Caso de uso	Chatear (CU07)
Actor	Usuario
Descripción	El usuario puede interactuar con otros miembros del grupo mediante el envío y recepción de mensajes en tiempo real.
Precondiciones	El usuario debe estar registrado, tener la sesión iniciada y ser miembro del grupo en cuestión.
Postcondiciones	El mensaje enviado por el usuario es visible para todos los miembros del grupo.
Flujo normal	1. El usuario selecciona el grupo 2. El sistema presenta la interfaz de chat del grupo 3. El usuario escribe un mensaje en el cuadro de texto 4. El usuario pulsa en el botón «Enviar» 5. El sistema envía el mensaje a todos los miembros del grupo
Flujo alternativo	4.1 El cuadro de texto está vacío 4.1.1 El sistema no permite el envío del mensaje 4.1.2 El usuario escribe un mensaje en el cuadro de texto

Cuadro 4.7: Especificaciones del caso de uso: Chatear (CU07)

Caso de uso	Crear evento (CU08)
Actor	Usuario
Descripción	El usuario podrá crear un nuevo evento dentro de un grupo proporcionando datos como lugar, fecha, hora y descripción del evento.
Pre-condiciones	El usuario tiene la sesión iniciada y es propietario del grupo en el que quiere crear el evento
Post-condiciones	Un nuevo evento queda registrado en la base de datos
Flujo normal	1. El usuario selecciona el grupo en el que quiere crear el evento desde el listado de grupos 2. El usuario pulsa en el botón «Crear evento» 3. El sistema presenta un formulario 4. El usuario rellena el formulario con los datos del nuevo evento 5. El usuario pulsa en el botón «Crear» 6. El sistema comprueba los campos 7. El sistema registra el evento en la base de datos 8. El usuario es redirigido al chat del grupo
Flujo alternativo	4.1 Los datos introducidos son inválidos 4.1.1 El sistema muestra un mensaje de error 4.1.2 El usuario corrige los datos

Cuadro 4.8: Especificaciones del caso de uso: Crear evento (CU08)

Caso de uso	Unirse a un evento (CU09)
Actor	Usuario
Descripción	El usuario podrá unirse a un evento existente dentro de un grupo.
Pre-condiciones	El usuario tiene la sesión iniciada y es miembro del grupo en el que se encuentra el evento
Post-condiciones	El usuario queda registrado como participante del evento en la base de datos
Flujo normal	1. El usuario selecciona el grupo en el que se encuentra el evento desde el listado de grupos 2. El usuario pulsa en el botón «Unirse a evento» 3. El sistema registra al usuario como participante del evento 4. El usuario es redirigido al chat del grupo

Cuadro 4.9: Especificaciones del caso de uso: Unirse a un evento (CU09)

5

Diseño del sistema

En esta sección se va a mostrar el diseño de la aplicación. Se detallará sobre el modelo de datos, aspectos de seguridad y diseño visual de la aplicación.

5.0.1 Modelo de datos

Para poder empezar con la implementación es de vital importancia definir cuáles son los datos que vamos a manipular en la aplicación, que entidades representan y como se relacionan estas entidades.

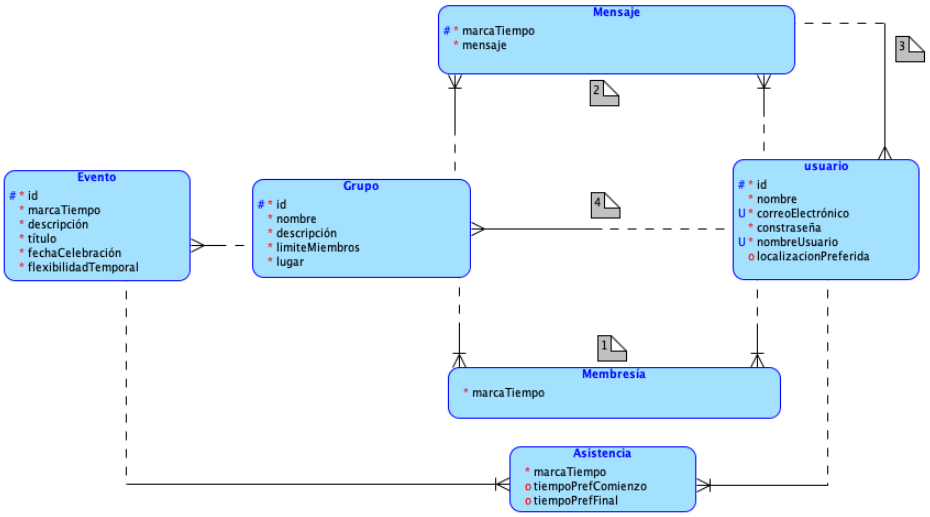

Figura 5.1: Modelo lógico de datos

Usuario

Un usuario representa a una persona que utiliza la aplicación. Cada usuario tiene un identificador único, un nombre, una dirección de correo electrónico, una contraseña, un nombre de usuario, una marca de tiempo de cuando se unió a la aplicación, y una ubicación preferida.

Grupo

Un grupo es una entidad que representa un conjunto de usuarios que se han unido para comunicarse y colaborar. Cada grupo tiene un identificador único, un nombre, una descripción, un límite de miembros, lugar y tipo de grupo. También el grupo mantiene la referencia al usuario dueño del grupo.

Mensaje

Un mensaje es una entidad que representa la comunicación entre usuarios dentro de un grupo.Cada mensaje mantiene una referencia al grupo y al usuario que lo envió, el contenido del mensaje, una marca de tiempo de cuando se envió. Cada mensaje también mantiene una lista con las referencias de los usuarios que han leído dicho mensaje.

Membresía

La membresía es una entidad intermedia que representa la relación entre los usuarios y los grupos. Es decir, denota que un usuario se ha unido a un grupo. Cada membresía tiene un identificador único, el identificador del usuario, el identificador del grupo, y una marca de tiempo de cuando el usuario se unió al grupo.

Evento

El entidad evento viene siempre asociada a un grupo, este describe el encuentro de grupo. Mantiene campos como título, descripción, marca de tiempo, fecha de celebración y un campo booleano que identifica la flexibilidad horaria.

Asistencia

La asistencia como su nombre indica, identifica la asistencia de un usuario a un evento. Tiene un identificador compuesto por dos claves foráneas, en este caso los identificadores de evento y usuario. Así como dos campos opcionales de tipo marca de tiempo que identifican la preferencia del tiempo de comienzo y final del evento en el caso en el que este último permita la flexibilidad horaria.

5.0.2 Navegación y vistas

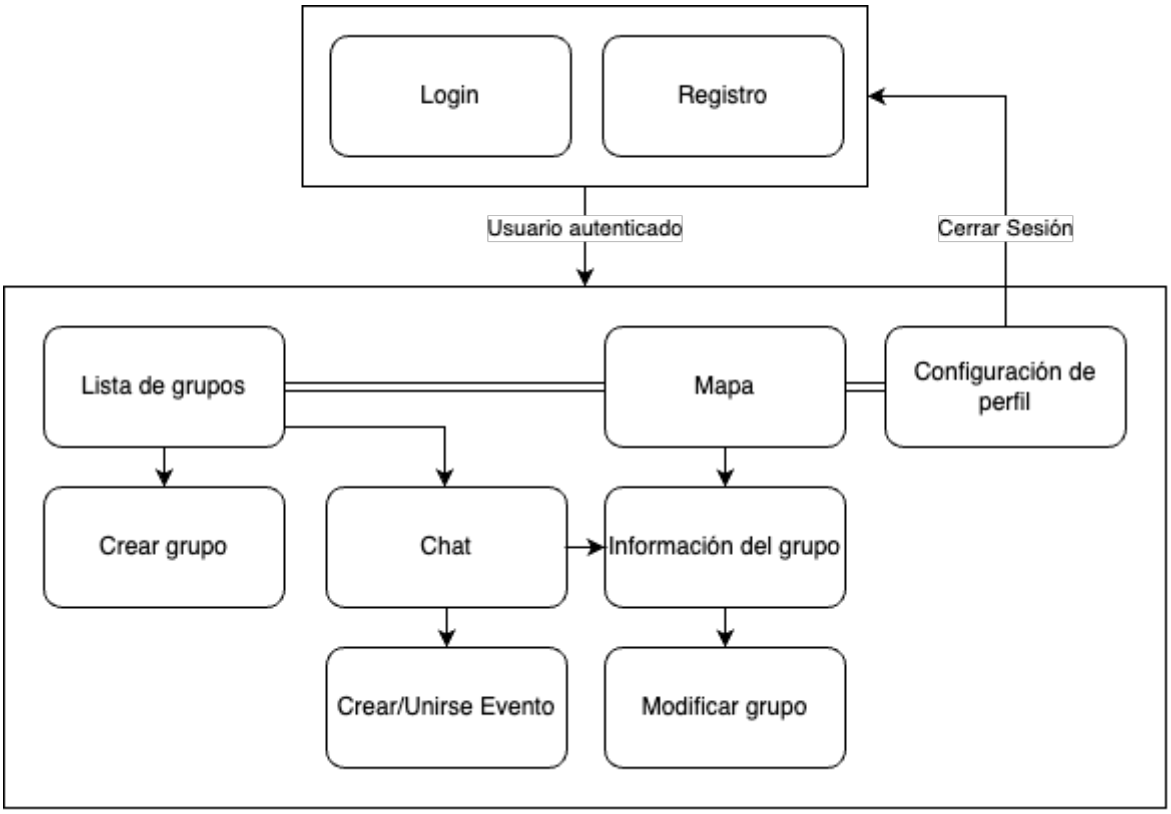

Figura 5.2: Diagrama de navegación

La vistas de la aplicación se han encapsulado en distintos grupos:

Login

La pantalla *Login* es el primer punto de entrada de nuestra aplicación, dónde el usuario puede tomar varias acciones: autenticarse, proceder al registro o restablecer contraseña.

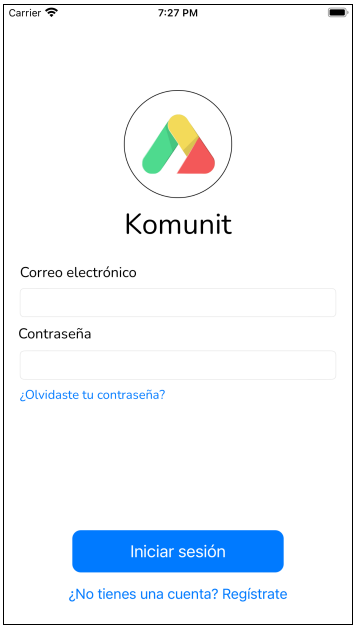

Figura 5.3: Inicio de sesión

Registro

La vista de registro en donde el visitante procederá a rellenar sus datos tales como nombre, nombre de usuario, contraseña, entre otros.

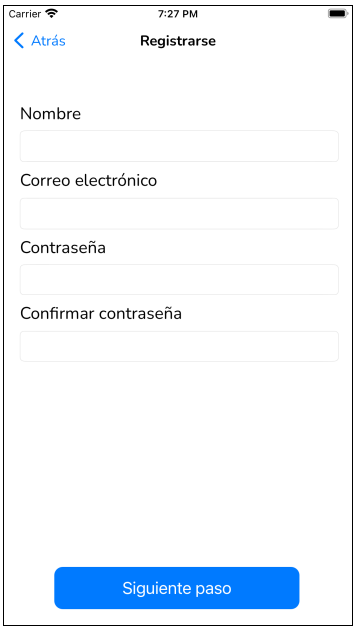

Figura 5.4: Registro de usuario

Confirmación de correo electrónico

El usuario para dar por finalizado el registro deberá confirmar su correo electrónico.

Figura 5.5: Confirmación de correo

Mapa

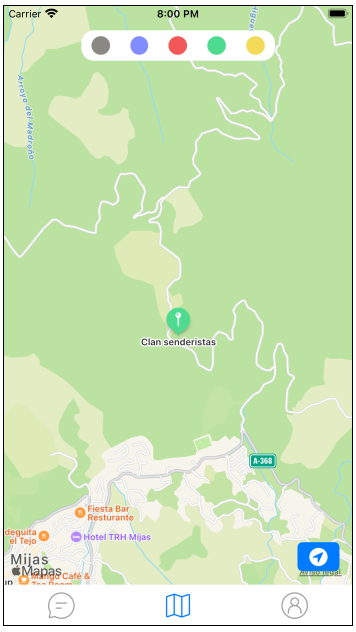

Figura 5.6: Vista mapa

Lista de grupos

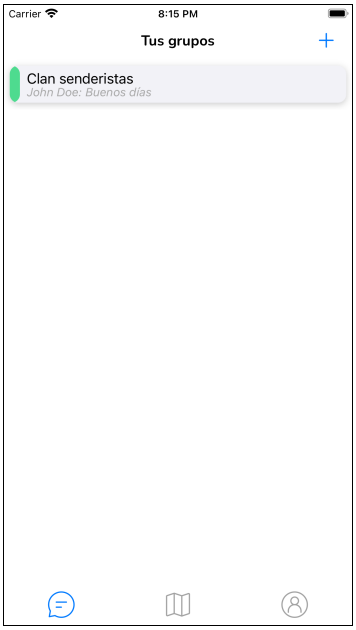

Figura 5.7: Lista de grupos

Chat

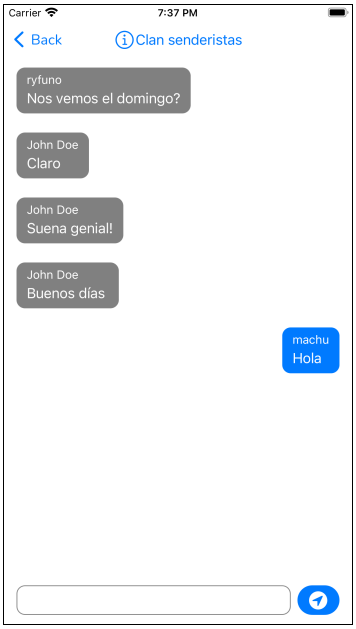

Figura 5.8: Vista de chat de grupo

Detalle del grupo

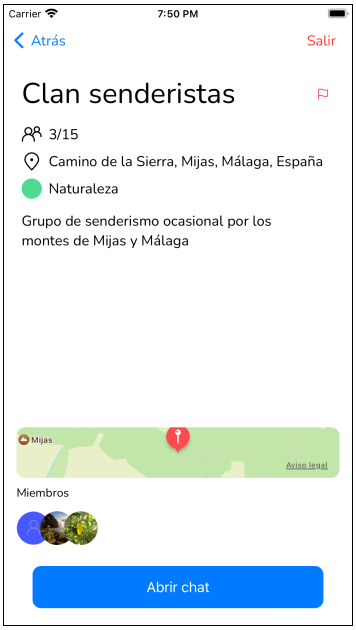

Figura 5.9: Vista detalle del grupo

Crear grupo

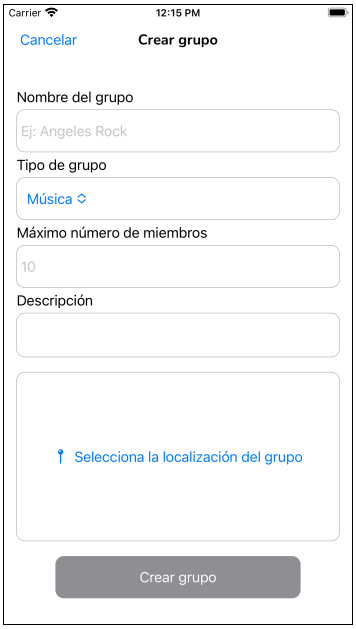

Figura 5.10: Vista Crear Grupo

Perfil

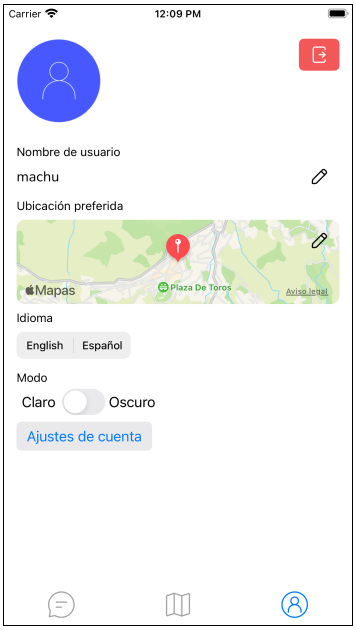

Figura 5.11: Vista Perfil

5.0.3 Estructura del proyecto

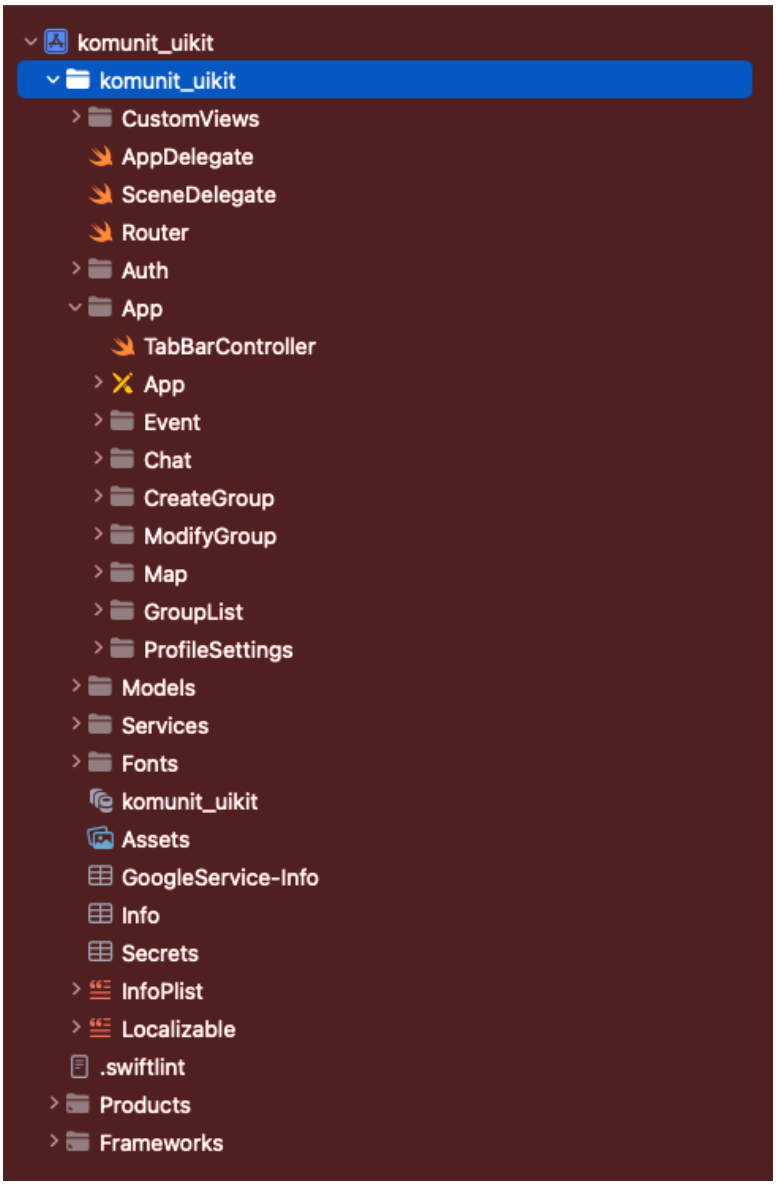

Figura 5.12: Estructura del proyecto

5.0.4 Crear o unirse a evento

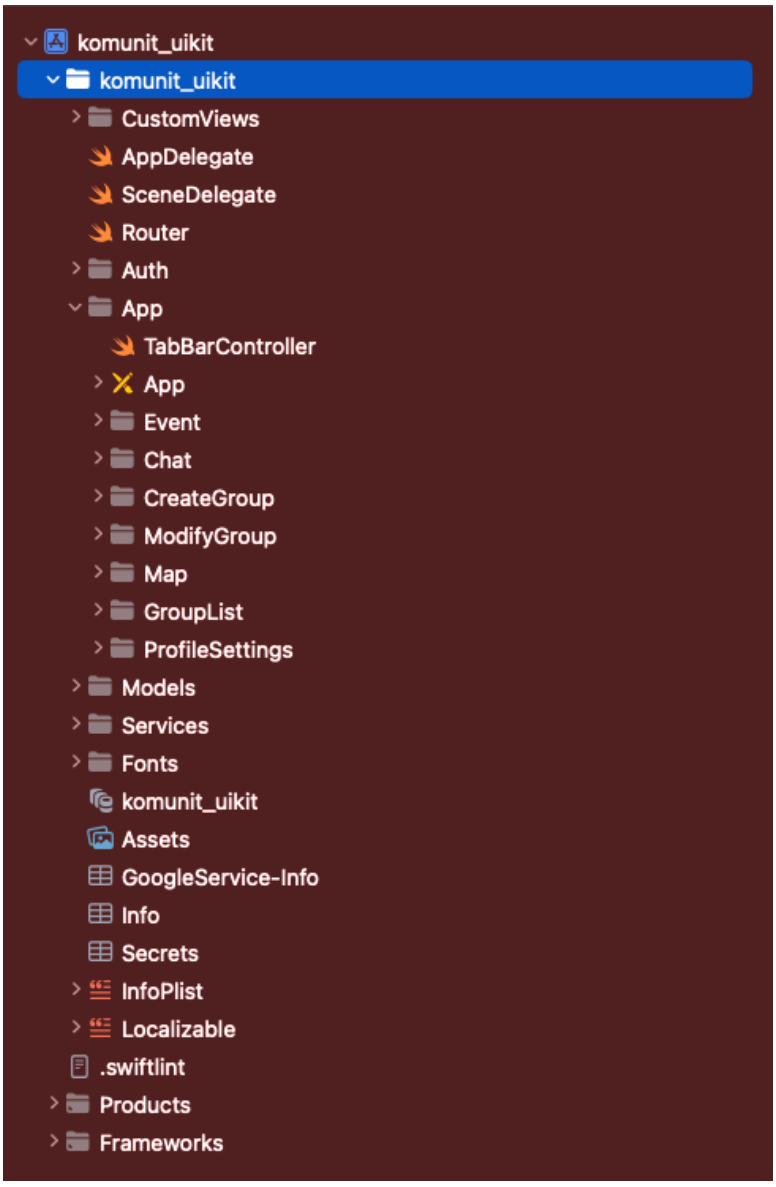

Figura 5.13: Estructura del proyecto

En la figura 5.13 se puede observar la jerarquía del proyecto. A continuación, se detallan los directorios y archivos más relevantes que componen la estructura del proyecto:

1. **AppDelegate.swift**: Es el archivo encargado de ser el punto de entrada de la aplicación. Configura Firebase, establece la apariencia de la aplicación y administra los eventos del ciclo de vida.

2. **SceneDelegate.swift**: Esta clase es encargada de la gestión de las escenas de la aplicación, incluyendo la configuración del controlador de vista inicial conforme

al estado de autenticación del usuario y la gestión de la alternancia entre los modos de visualización.

3. **Router.swift**: La responsabilidad de esta clase radica en la administración de la navegación de la aplicación, determinando qué controlador de vista mostrar en función del estado de autenticación del usuario.

4. **Auth**: Directorio que incluye las clases y recursos asociados con el proceso de autenticación de la aplicación, comprendiendo controladores de vista para el registro, la confirmación por correo electrónico, entre otros.

5. **App**: Directorio que alberga las clases y recursos correspondientes a la parte principal de la aplicación, siendo el punto de interacción para los usuarios una vez iniciada la sesión.

6. **Services**: Directorio que agrupa los servicios empleados en la aplicación, es decir, las clases que se encargan de la comunicación con la base de datos.

7. **Models**: Directorio que contiene los modelos de datos empleados en la aplicación.

8. **CustomViews**: Directorio en el cual se encuentran componentes de interfaz de usuario personalizados, utilizados en distintas partes de la aplicación.

9. **Animations**: Directorio que alberga archivos de animación empleados en la aplicación.

10. **Assets.xcassets**: Directorio que incluye los activos de imagen y color que se utilizan en la aplicación.

11. **Fonts**: Directorio que contiene las fuentes tipográficas personalizadas que se emplean en la aplicación.

12. **Localization**: La aplicación admite la localización, reflejada en la presencia de archivos de storyboard y archivos de cadena localizados, como es.lproj/Auth.strings y en.lproj/Localizable.strings.

13. **komunit_uikit.xcodeproj**: Archivo de proyecto de Xcode que reúne todas las configuraciones relacionadas con el proyecto.

6

Conclusiones y Líneas Futuras

6.1 Conclusiones

Conforme a la planificación estipulada al comienzo del proyecto, durante los últimos meses hemos logrado desarrollar la versión inicial de nuestra aplicación móvil. Esta primera versión, aunque con una interfaz sencilla y funcionalidades básicas, podría clasificarse como el Producto Mínimo Viable[2] de nuestra aplicación. Consideramos este punto de partida como un logro significativo, ya que la aplicación cumple con criterios esenciales de usabilidad, fiabilidad, funcionalidad y diseño, elementos suficientes para introducirla en el mercado.

6.2 Líneas futuras

En esta sección se presentarán las posibles mejoras y extensiones, que podemos implementar en nuestra aplicación, con el objetivo de mejorar la experiencia de usuario y aumentar el alcance de nuestra plataforma.

6.2.1 Mejoras en el chat

Vemos un amplio margen de mejora en la funcionalidad del chat de nuestra aplicación. Entre las potenciales mejoras que estamos considerando, destaca la posibilidad

de compartir imágenes y vídeos. Esta característica, común en muchas plataformas de chat, enriquecería la comunicación entre los usuarios, permitiendo un intercambio de información más completo y dinámico.

6.2.2 Automatización del sistema de reporte

Actualmente, nuestro sistema soporta el reporte de contenido inapropiado, pero este reporte en el actua estado debe ser revisado por un humano quién tomará la decisión de eliminar el contenido o no. En un futuro, se podría implementar un sistema de inteligencia artificial que analice el contenido reportado y decida si es apropiado o no. Esto permitiría automatizar el proceso de moderación, reduciendo el tiempo de respuesta y mejorando la experiencia del usuario.

6.2.3 Mejoras en la planificación de actividades

Desarrollar más a fondo este aspecto podría añadir un gran valor a nuestra aplicación. Mejorar esta herramienta permitiría a los grupos organizar sus actividades de manera más eficiente, por ejemplo añadir la posibilidad de además de poder sugerir la ventana temporal de disponibilidad, poder sugerir días. Asimismo poder asignar a un mismo día varias ventanas temporales. Siguiendo un enfoque similar al de la aplicación *LettuceMeet*[3], que permite a los usuarios agregar su disponibilidad en una cuadrícula para identificar el intervalo de tiempo que mejor se superpone con la disponibilidad de todos los miembros.

6.2.4 Aplicación multiplataforma

Para maximizar su accesibilidad y alcance, nuestra aplicación debe ser multiplataforma, es decir, compatible con diversos sistemas operativos y dispositivos. En este sentido, es esencial considerar desarrollar la aplicación para Android, dada su predominancia en el mercado global de dispositivos móviles.

6.2.5 Internacionalización

Si bien la aplicación actualmente soporta español e inglés, es fundamental para futura expansión global considerar la incorporación de otros idiomas. Aunque estos dos idiomas cubren una gran parte de la población mundial, una verdadera internacionalización implica la inclusión de una amplia gama de idiomas y regionalismos. Por ejemplo, podríamos considerar la adición de francés, alemán, italiano, portugués, chino, japonés, árabe, entre otros.

7

Bibliografía

[1] Erin Schaffer *What is Swift? Features, advantages, and syntax basics* https://www.educative.io/blog/swift-programming

[2] Stefano Gasbarrino *MVP: qué es el producto mínimo viable, cómo hacerlo y ejemplos* https://blog.hubspot.es/sales/producto-minimo-viable#que-es

[3] *LettuceMeet* https://lettucemeet.com/

Apéndice A

Manual de Usuario

En este anexo se facilitará un guía para el usuario. El propósito de esta guía es explicar al usuario cuáles son las capacidades de la aplicación y su modo de uso.

A.1 Registro

Para poder utilizar la aplicación el usuario deberá realizar primero el registro en el sistema. Este se realizará pulsando el botón de "¿No tienes una cuenta? Regístrate".

Posteriormente el usuario procederá a rellenar la información que ahí se le pide como nombre, correo electrónico y contraseña que mínimo deberá ser de 6 caracteres.

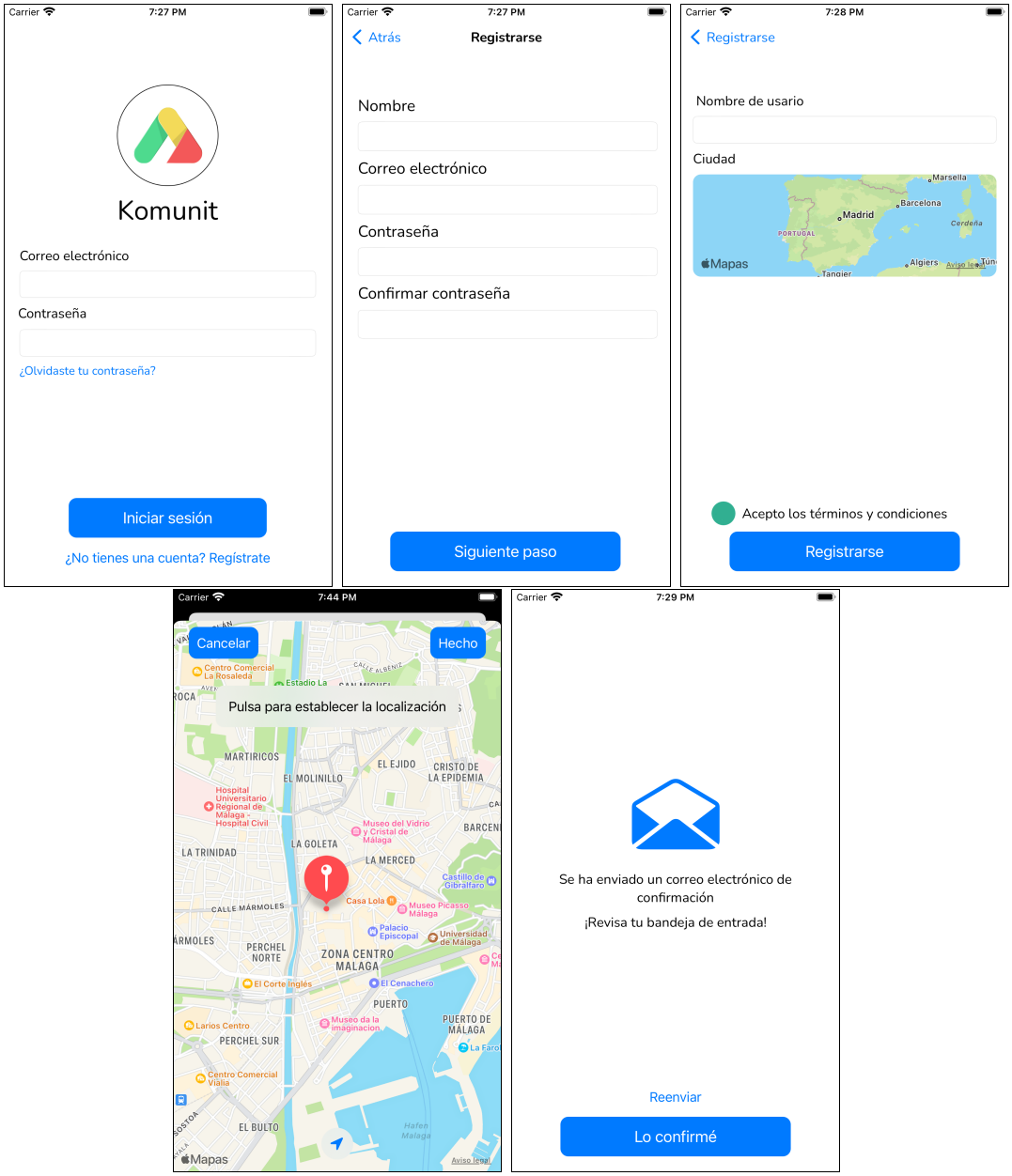

Figura A.1: Login y registro

En el siguiente paso deberá rellenar datos como nombre de usuario y localización preferida a través de un mapa interactivo, asimismo como aceptar los términos y condiciones.

Finalmente, para terminar el proceso de registro el usuario deberá confirmar su correo electrónico a través del enlace que le será enviado al buzón del correo electrónico especificado durante el registro.

A.2 Iniciar sesión

Para iniciar sesión simplemente el usuario debe proporcionar las credenciales que uso durante el proceso de registro.

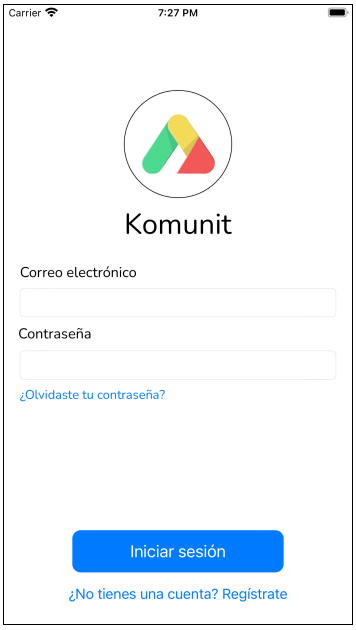

Figura A.2: Iniciar sesión

Después de rellenar correo y contraseña el usuario deberá pulsar el botón "Iniciar sesión", si las credenciales son correctas la aplicación redirigirá al usuario a la pantalla principal.

A.3 Unirse a un grupo

Estando en la pantalla principal, el usuario puede explorar grupos alrededor de su localización a través de un mapa interactivo.

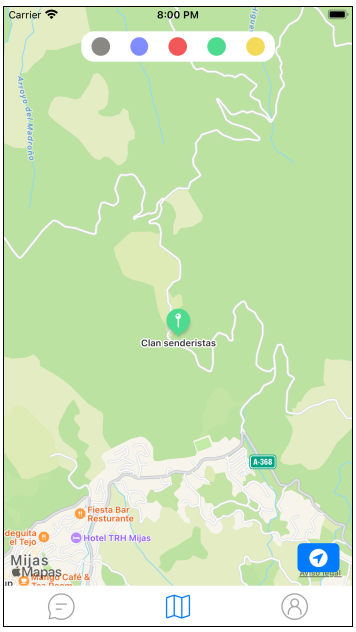

Figura A.3: Iniciar sesión

Pinchando sobre el grupo el usuario podrá visualizar una vista detallada de la información del grupo como título, descripción e integrantes. Para unirse a un grupo habrá que pulsar sobre el botón “Unirse”.

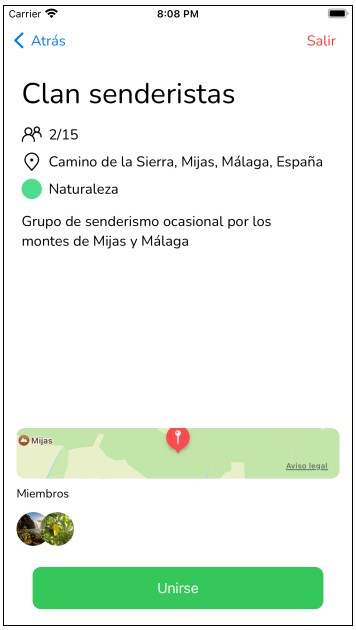

Figura A.4: Iniciar sesión

A.4 Chatear

Para acceder al chat de un grupo el usuario deberá dirigirse al listado de grupos pulsando en el primer icono del menú inferior de navegación. De esta forma podrá visualizar los grupos a los que pertenece ordenados por fecha del último mensaje.

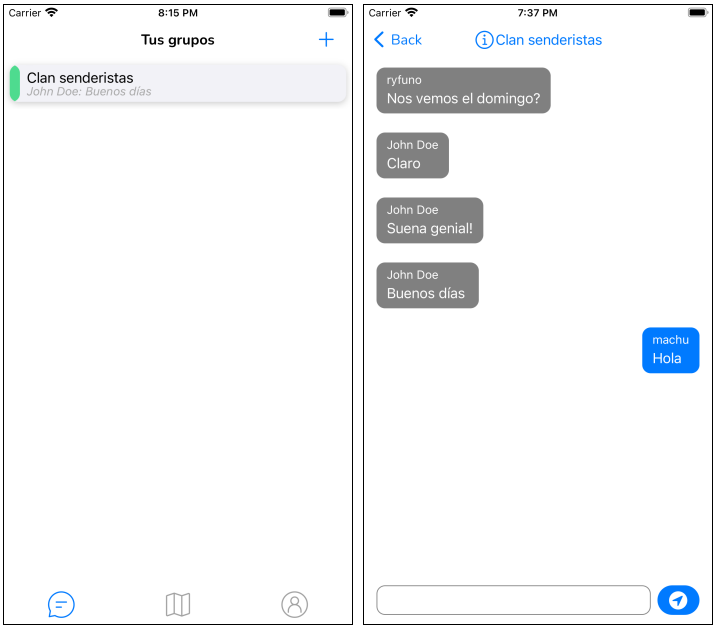

Figura A.5: Chat

Pulsando sobre el chat correspondiente, el usuario podrá acceder a los mensajes del grupo y mandar mensajes por su parte.

A.5 Abandonar grupo

Para abandonar grupo el usuario deberá en la pantalla de detalles del grupo. Pulsar sobre el botón "Salir" en la esquina superior derecha. Posteriormente deberá confirmar la acción a través de una ventana emergente.

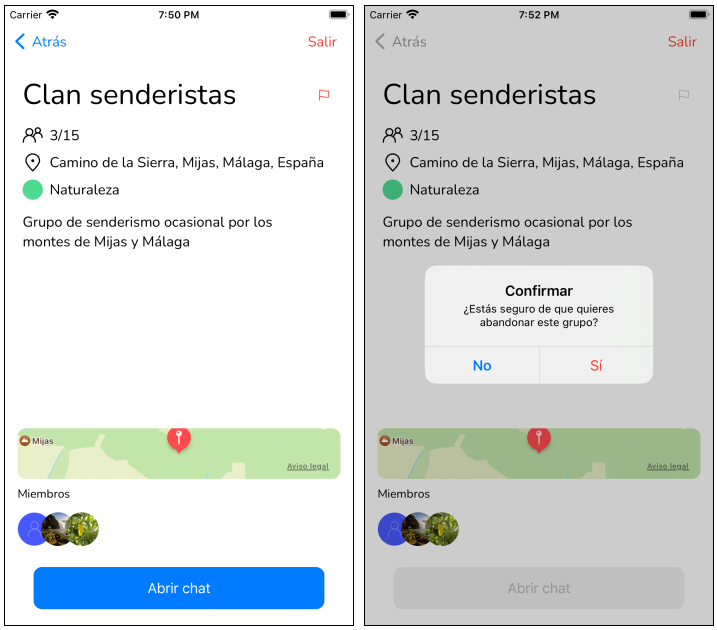

Figura A.6: Abandonar grupo

A.6 Ajustes de usuario

Para configurar distintos aspectos del perfil y de la aplicación el usuario debe pulsar en el tercer icono del menú de navegación. En esta pantalla podrá realizar distintas acciones como: actualizar nombre de usuario, localización preferida, idioma de la aplicación, cambiar de modo visual, actualizar foto de perfil y eliminar cuenta.

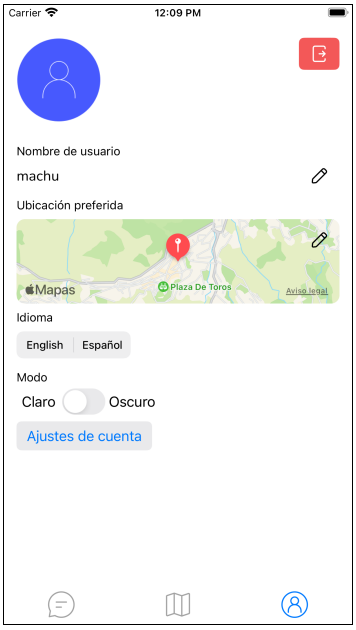

Figura A.7: Configuración de perfil

A.7 Crear y modificar grupo

Para crear un grupo el usuario deberá pulsar el icono de "+" en la parte superior derecha del listado de grupos.

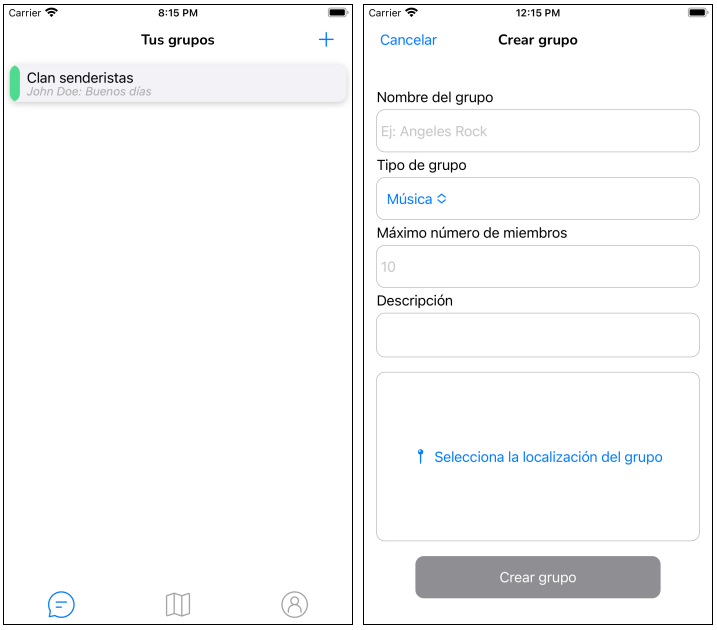

Figura A.8: Abandonar grupo

Deberá rellenar los datos necesarios como título, descripción, tipo de grupo, límite de integrantes y lugar de celebración. Para modificar un grupo simplemente el usuario creador deberá pulsar sobre la opción "Editar" en la pantalla de información del grupo.

A.8 Crear y modificar eventos

Para crear un evento, el usuario deberá pulsar sobre el botón "Crear evento" situado en la pantalla de chat.

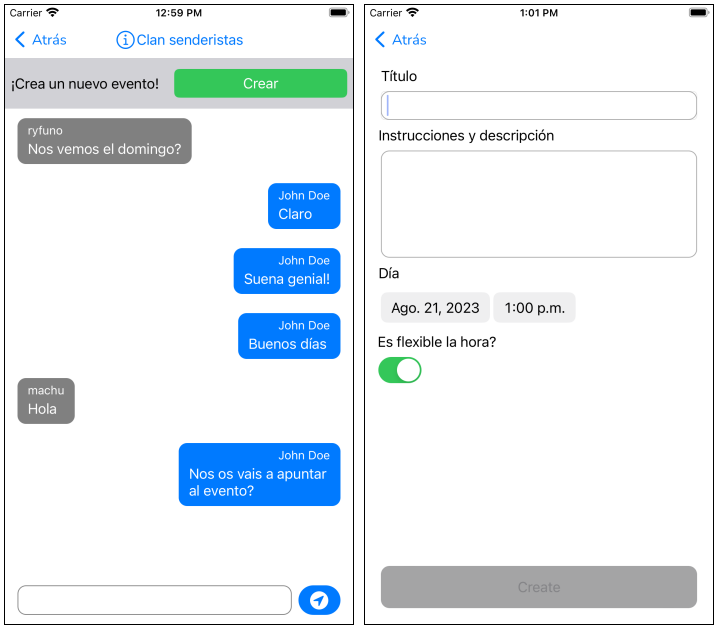

Figura A.9: Crear evento

Deberá rellenar la información requerida así como marca la opción de flexibilidad horaria, que permite especificar a la hora de unirse los integrantes del grupo su disponibilidad horaria para el día del evento. Las instrucciones para modificar un grupo son análogas.

A.9 Unirse y abandonar evento

El usuario podrá unirse a un evento pulsando el botón de "Unirse a evento" en caso de la existencia de este.

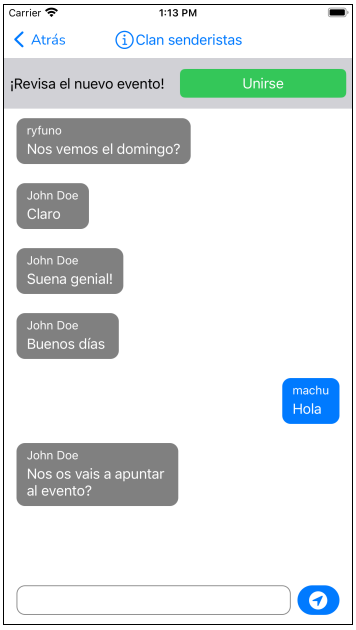

Figura A.10: Botón unirse a evento

Será mostrado una pantalla con los detalles del evento y si se da el caso podrá especificar su disponibilidad horaria a través de un selector de rango.

Figura A.11: Unirse a evento

Para abandonar el evento tiene que realizar las mismas acciones y en vez de “Unirse” se mostrará la opción “Desapuntarse”.

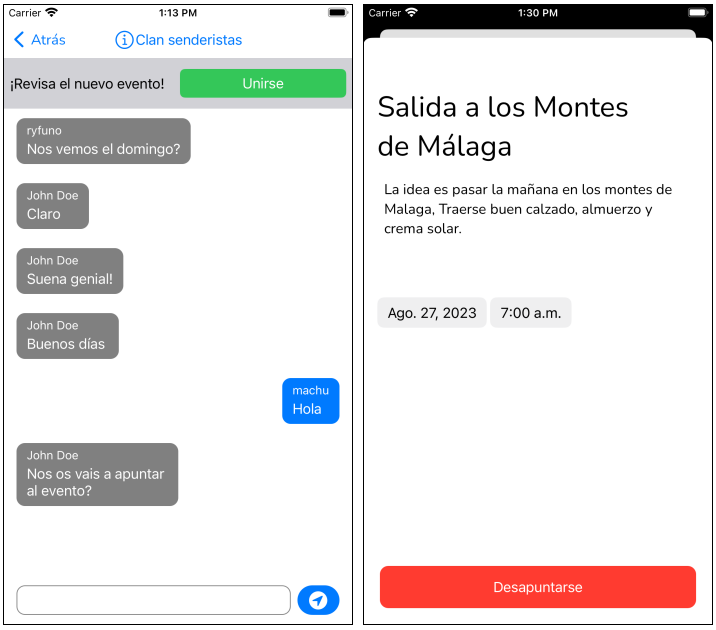

Figura A.12: Despuntarse del evento

UNIVERSIDAD DE MÁLAGA | uma.es

E.T.S DE INGENIERÍA INFORMÁTICA

E.T.S de Ingeniería Informática
Bulevar Louis Pasteur, 35
Campus de Teatinos
29071 Málaga