



viu

**Universidad
Internacional
de Valencia**

HomeMate: Aplicación web para alquiler de habitaciones y búsqueda de compañeros de piso

|| Titulación:
Máster en desarrollo de
aplicaciones y servicios
web
Curso Académico
2024-2025

|| Alumno/a: Santos
Martínez, Natalia
DNI: 26577651R

Director/a del TFT:
Ruben Perez Ibáñez

|| Convocatoria:

PRIMERA

Índice general

Índice de figuras

Índice de tablas

| | |
|--|-----------|
| 1 Introducción | 2 |
| 1.1 Motivación del proyecto | 2 |
| 2 Objetivos | 4 |
| 2.1 Objetivos tecnológicos | 4 |
| 2.2 Requisitos funcionales del proyecto | 4 |
| 3 Marco tecnológico | 6 |
| 3.1 Estado del arte | 6 |
| 3.2 Tecnologías utilizadas | 7 |
| 3.2.1 React | 7 |
| 3.2.2 NodeJS | 8 |
| 3.2.3 Firebase | 8 |
| 3.2.4 Cloudinary | 8 |
| 3.2.5 GitHub | 9 |
| 3.2.6 Open Street Map y Leaflet | 9 |
| 3.2.7 Trello | 9 |
| 3.2.8 Figma | 10 |
| 3.2.9 Postman | 10 |
| 4 Metodología | 11 |
| 4.1 Gestión del proyecto | 11 |
| 4.1.1 Diagrama de casos de uso | 11 |
| 4.1.2 Diagrama de base de datos | 12 |
| 4.1.3 Historias de usuario | 13 |
| 4.1.4 Mapa de historias de usuario | 14 |
| 4.1.5 Sprints | 15 |
| 4.1.6 Diagrama de Gantt | 16 |
| 4.2 Gestión de tareas con Trello | 18 |
| 4.3 Modelos para la interfaz (MockUps) | 19 |
| 4.3.1 Menús | 19 |
| 4.3.2 Login y Registro usuarios | 20 |
| 4.3.3 Publicar habitación | 20 |
| 4.3.4 Perfil de usuario y opciones de configuración como compañero de piso | 21 |
| 5 Resultados obtenidos | 23 |
| 5.1 Vistas de la aplicación final | 23 |
| 5.1.1 Menú usuario invitado vs autenticado | 23 |
| 5.1.2 Inicio de sesión | 23 |
| 5.1.3 Registro usuario | 24 |
| 5.1.4 Crear habitación | 26 |
| 5.1.5 Consultar habitaciones | 28 |
| 5.1.6 Actualizar o eliminar habitación | 31 |
| 5.1.7 Consultar o actualizar perfil personal | 31 |

| | | |
|-----------------------------------|--|-----------|
| 5.1.8 | Consultar compañeros de piso | 32 |
| 5.1.9 | Chat | 34 |
| 5.1.10 | Programar visita | 35 |
| 5.2 | Backend - ApiRest | 36 |
| 5.2.1 | EndPoints públicos | 37 |
| 5.2.2 | EndPoints privados | 37 |
| 5.3 | Aspectos destacables de la aplicación | 38 |
| 5.3.1 | Autenticación con Google | 38 |
| 5.3.2 | Autenticación Nativa con Tokens Personalizados | 40 |
| 5.3.3 | Sincronización con Google Calendar | 42 |
| 5.3.4 | Mapas Interactivos con OpenStreetMap y Leaflet | 44 |
| 5.3.5 | Seguridad del sistema con reCAPTCHA | 45 |
| 6 | Conclusiones | 48 |
| 7 | Trabajo futuro | 49 |
| Referencias bibliográficas | | 50 |
| A | Anexo I | 52 |
| B | Anexo II | 57 |

Índice de figuras

| | | |
|------|---|----|
| 3.1 | Frameworks frontend más utilizados en 2024 | 7 |
| 4.1 | Diagrama de casos de uso (Fuente: Elaboración propia) | 12 |
| 4.2 | Diagrama UML (Fuente: Elaboración propia) | 13 |
| 4.3 | Mapa de historias de usuario | 15 |
| 4.4 | Diagrama de Gantt | 17 |
| 4.5 | Tablero en Trello | 18 |
| 4.6 | Items de la tarea | 19 |
| 4.7 | Menú mockup (Fuente: Elaboración propia) | 19 |
| 4.8 | Login y Registro mockup (Fuente: Elaboración propia) | 20 |
| 4.9 | Publicar habitación mockup (Fuente: Elaboración propia) | 21 |
| 4.10 | Perfil usuario mockup (Fuente: Elaboración propia) | 22 |
| 5.1 | Vista Menú usuario invitado (Fuente: Elaboración propia) | 23 |
| 5.2 | Vista Menú usuario autenticado (Fuente: Elaboración propia) | 23 |
| 5.3 | Vista Inicio sesión (Fuente: Elaboración propia) | 24 |
| 5.4 | Vista Mensajes error login (Fuente: Elaboración propia) | 24 |
| 5.5 | Vista Registro usuario (Fuente: Elaboración propia) | 25 |
| 5.6 | Vista Mensajes error registro usuario (Fuente: Elaboración propia) | 26 |
| 5.7 | Vista Crear habitación paso 1 (Fuente: Elaboración propia) | 27 |
| 5.8 | Vista Crear habitación paso 2 (Fuente: Elaboración propia) | 27 |
| 5.9 | Vista Crear habitación paso 3 (Fuente: Elaboración propia) | 28 |
| 5.10 | Vista Consultar habitaciones (Fuente: Elaboración propia) | 29 |
| 5.11 | Vista Consultar detalles de la habitación (Fuente: Elaboración propia) | 30 |
| 5.12 | Vista Actualizar o eliminar habitación (Fuente: Elaboración propia) | 31 |
| 5.13 | Vista Consultar o actualizar perfil personal (Fuente: Elaboración propia) | 32 |
| 5.14 | Vista Consultar compañeros de piso (Fuente: Elaboración propia) | 33 |
| 5.15 | Vista Consultar compañero de piso (Fuente: Elaboración propia) | 34 |
| 5.16 | Vista Chat (Fuente: Elaboración propia) | 35 |
| 5.17 | Vista Fecha y hora visita (Fuente: Elaboración propia) | 35 |
| 5.18 | Vista Invitación visita Gmail (Fuente: Elaboración propia) | 36 |
| 5.19 | Vista Visita en Google Calendar (Fuente: Elaboración propia) | 36 |

Índice de tablas

| | | |
|------|---|----|
| 3.1 | Comparativa entre HomeMate, Idealista y Fotocasa | 6 |
| 4.1 | Historia de usuario - Registro usuarios | 14 |
| 4.2 | Historia de usuario - Acceso usuarios | 14 |
| A.1 | Historia de usuario - Publicar habitación | 52 |
| A.2 | Historia de usuario - Publicar habitación | 52 |
| A.3 | Historia de usuario - Eliminar habitación | 53 |
| A.4 | Historia de usuario - Consultar habitación | 53 |
| A.5 | Historia de usuario - Filtrar en búsqueda de habitación | 53 |
| A.6 | Historia de usuario - Acceso perfil usuario | 54 |
| A.7 | Historia de usuario - Actualizar perfil usuario | 54 |
| A.8 | Historia de usuario - Publicar mi perfil de compañero de piso | 54 |
| A.9 | Historia de usuario - Eliminar mi perfil de compañero de piso | 55 |
| A.10 | Historia de usuario - Consultar compañero de piso | 55 |
| A.11 | Historia de usuario - Contacto por chat | 55 |
| A.12 | Historia de usuario - Cerrar sesión | 56 |
| A.13 | Historia de usuario - Gestión visitas | 56 |

Resumen

Este Trabajo de Fin de Máster tiene como objetivo principal el desarrollo de una plataforma web que facilite el proceso de encontrar alojamiento compartido y compañeros de piso en España. La creciente demanda de este tipo de soluciones, debido a los elevados precios del alquiler y la necesidad de compartir gastos, ha mostrado que los métodos tradicionales, como las redes sociales o los tablones de anuncios, no siempre son suficientes.

HomeMate se concibe como una alternativa innovadora, que centraliza las ofertas de habitaciones y los perfiles de posibles compañeros de piso en una única plataforma. La aplicación se ha desarrollado utilizando tecnologías modernas como React para el frontend, NodeJS para el backend y Firebase como base de datos. Además, integra herramientas como Cloudinary para la gestión de imágenes, OpenStreetMap y Leaflet para la visualización de mapas y la sincronización con Google Calendar para la gestión de visitas.

El desarrollo del proyecto se ha basado en metodologías ágiles, organizando el trabajo en sprints y utilizando Trello para la gestión de tareas. Los resultados obtenidos incluyen funcionalidades clave como la publicación y gestión de anuncios de habitaciones, la consulta de perfiles de compañeros de piso, un sistema de chat para la comunicación entre usuarios, y la programación de visitas con integración en Google.

Palabras clave: alquiler de habitaciones, compañeros de piso, chat, React, NodeJS, Firebase, Google Calendar, reCAPTCHA, OpenStreetMap.

Agradecimientos

Me gustaría agradecer a mi tutor del trabajo final de máster, Rubén Pérez Ibáñez, por su ayuda y guía durante todo el proceso de este proyecto.

También quiero dar las gracias a mi familia y amigos por su apoyo, ánimo a lo largo de todo el curso.

1. Introducción

En los últimos años, la búsqueda de alojamiento compartido ha dejado de ser una práctica exclusiva de los estudiantes para convertirse en una necesidad creciente entre diversos grupos de la población, debido al aumento de los precios del alquiler y las dificultades para acceder a la compra de vivienda. Compartir piso se ha establecido como una opción económica para muchas personas, permitiéndoles reducir los costos de la vivienda, pero también implica ciertos desafíos relacionados con la convivencia y la búsqueda de compañeros de piso confiables.

Actualmente, los métodos más comunes para encontrar compañeros de piso o habitaciones disponibles suelen incluir publicaciones en redes sociales como Instagram y el uso de plataformas de tablones de anuncios, como Idealista. Sin embargo, estos medios presentan varias limitaciones: la falta de organización, la ausencia de un sistema de verificación de identidad y la escasa confianza entre los usuarios. Además, compartir los datos personales y de contacto en estas plataformas puede generar inseguridad y vulnerabilidad.

Con el objetivo de mejorar este proceso y resolver estas limitaciones, este Trabajo de Fin de Máster (TFM) propone el desarrollo de una aplicación web que centralice las ofertas y demandas de alojamientos compartidos en España. El desarrollo incluirá un backend implementado en NodeJS, un frontend con React y una base de datos basada en Firebase.

1.1. Motivación del proyecto

Según un estudio realizado por María Hernández en el periódico *El Mundo* durante el año 2024, “*compartir piso y alquilar habitaciones para vivir ya no es sólo cosa de estudiantes. Tampoco es una moda, sino una necesidad ante los precios récord que batén mes tras mes los alquileres y el número cada vez menor de pisos completos disponibles. El mercado se ha dado la vuelta y en ciudades como Madrid, Barcelona o Valencia, la oferta de habitaciones y apartamentos compartidos ya supera a la oferta de pisos enteros*” (Mundo, 2024).

Otro estudio realizado en 2024 por Fotocasa Research. “*El comienzo del curso escolar marca el punto álgido anual para el alquiler de habitaciones en viviendas compartidas, una actividad en la que en 2024 participan un 3 % de españoles en España. De ese porcentaje, el 55 % de quienes han alquilado habitación en un piso compartido lo hace porque no pueden pagar por el arrendamiento de un inmueble completo. Es un porcentaje significativamente más alto que el 44 % registrado en 2023 con once puntos porcentuales de crecimiento en tan solo un año.*” (Fotocasa, 2024).

Por lo tanto, la creciente demanda de alojamiento compartido impulsada por los altos costos del alquiler, resalta la relevancia de este proyecto. La plataforma propuesta busca abordar estas necesidades mediante un entorno organizado y seguro, que facilite la búsqueda de habitaciones y compañeros de piso. Con herramientas como una calculadora de costos, un chat integrado y la posibilidad de programar visitas desde la aplicación, esta solución tecnológica pretende garantizar una experiencia más confiable y transparente para los usuarios,

mejorando significativamente el proceso actual.

2. Objetivos

Como ya se ha comentado en el apartado 1, el objetivo de este proyecto de fin de máster consiste en desarrollar una aplicación web para facilitar la búsqueda de habitaciones y compañeros de piso en España. La plataforma está diseñada tanto para quienes buscan alojamiento como para aquellos que ofrecen habitaciones o desean presentarse como posibles compañeros de piso.

En la sección 2.1 se detallarán las tecnologías empleadas en el desarrollo de la aplicación, mientras que en la sección 2.2 se describirán los requisitos esenciales que debe cumplir del proyecto.

2.1. Objetivos tecnológicos

La aplicación se desarrolla utilizando tecnologías avanzadas para garantizar una experiencia ágil, segura y escalable. El frontend está construido con React, lo que permite crear una interfaz interactiva y altamente reactiva, mejorando la navegación del usuario.

El backend se gestiona con Node.js y Express, proporcionando una API rápida y flexible que maneja eficientemente las solicitudes y la interacción con la base de datos. Firebase se encarga de la autenticación de usuarios y de almacenar los datos, asegurando tanto la protección de la información como la escalabilidad de la plataforma.

Para el almacenamiento y manejo de contenido multimedia, como imágenes y vídeos, se utiliza Cloudinary, lo que optimiza el rendimiento y reduce los tiempos de carga. El proceso de desarrollo y colaboración se gestiona a través de GitHub, donde se controla el versionado del código, y Trello, que facilita la organización y seguimiento de tareas y plazos. Además, Figma se utiliza para diseñar la interfaz de usuario, permitiendo una visualización clara de la estructura de la aplicación antes de la implementación.

2.2. Requisitos funcionales del proyecto

En esta sección se describen los requisitos que la aplicación debe cumplir para garantizar el éxito del desarrollo del proyecto.

- **Acceso aplicación:** El sistema permitirá a los usuarios registrarse directamente desde la aplicación, utilizando un formulario nativo, o acceder mediante sus credenciales de Google. Además, como medida de seguridad adicional, el proceso de registro incluirá la integración de reCAPTCHA para prevenir accesos no autorizados y garantizar que los registros sean realizados por usuarios reales.
- **Gestión habitaciones:** Los usuarios invitados tendrán acceso a los detalles de las habitaciones disponibles, pero no podrán realizar cambios. Solo los usuarios autenticados podrán registrar, actualizar o eliminar habitaciones, siempre que sean aquellas creadas por ellos mismos.

- **Gestión perfil:** Una vez autenticados, los usuarios tendrán la posibilidad de visualizar y editar la información de su perfil personal.
- **Gestión comapañeros de piso:** Los usuarios autenticados podrán ofrecerse como compañeros de piso mediante la habilitación de esta opción en su perfil personal. Los invitados tendrán acceso a todos los perfiles de compañeros de piso que son públicos.
- **Gestión chat:** La aplicación permitirá a los usuarios autenticados iniciar conversaciones privadas con los propietarios de las habitaciones que les resulten de interés. Este sistema de mensajería facilitará la comunicación directa entre ambas partes.
- **Gestión de visitas:** Aquellos usuarios que hayan iniciado un chat con un propietario podrán solicitar visitas a las habitaciones. Estas visitas se sincronizarán automáticamente con el calendario de Google de ambos participantes en el chat.

Además, todos los formularios de la aplicación incluirán validaciones de datos para garantizar la correcta entrada de información. En caso de que algún campo sea incorrecto o esté incompleto, el sistema informará al usuario mediante mensajes claros y específicos, facilitando la corrección del error.

3. Marco tecnológico

En este capítulo se presenta el marco tecnológico del proyecto, abordando tanto el estado del arte como las tecnologías utilizadas. En primer lugar, en la sección [3.1](#) se analiza el estado del arte, revisando la evolución, tendencias y soluciones actuales relacionadas con el proyecto. Posteriormente, en la sección [3.2](#) se describen las tecnologías utilizadas, incluyendo las herramientas, lenguajes de programación y plataformas seleccionadas, justificando su elección en función de los objetivos de este proyecto.

3.1. Estado del arte

En la Tabla [3.1](#) se han analizado las principales funcionalidades de las plataformas más relevantes en el mercado para la búsqueda de compañeros de piso y habitaciones.

Idealista¹ es una plataforma líder en el sector inmobiliario en España que permite buscar viviendas en alquiler, compra o compartidas, y facilita la conexión entre propietarios e interesados. Fotocasa² es otra de las principales plataformas inmobiliarias, conocida por su amplio catálogo de viviendas y herramientas de búsqueda avanzadas.

Como se puede observar, todas las aplicaciones ofrecen filtros de búsqueda o chat integrado, pero HomeMate se diferencia por incluir características adicionales como el planificador de visitas integrado con Google Calendar, la calculadora de costos y la búsqueda por compañeros de piso. Estas diferencias justifican el desarrollo de una aplicación propia, centrada en optimizar la experiencia de quienes buscan compartir vivienda, ofreciendo herramientas que van más allá de lo que las plataformas inmobiliarias tradicionales proporcionan.

| Característica | HomeMate | Idealista | Fotocasa |
|---------------------------------------|----------|-----------|----------|
| Búsqueda de compañeros de piso | X | - | - |
| Búsqueda por habitaciones | X | X | - |
| Filtros de búsqueda | X | X | X |
| Gestor de visitas con Google Calendar | X | - | - |
| Calculadora de costos | X | - | - |
| Chat integrado | X | X | X |

Tabla 3.1: Comparativa entre HomeMate, Idealista y Fotocasa

¹<https://www.idealista.com/>

²<https://www.fotocasa.es/>

3.2. Tecnologías utilizadas

En este apartado se presentan las tecnologías utilizadas en el proyecto, como lenguajes de programación y frameworks, elegidas por su capacidad para cumplir con los requisitos del desarrollo.

React

“ReactJS es una de las librerías más populares de JavaScript para el desarrollo de aplicaciones móviles y web. Creada por Facebook, React contiene una colección de fragmentos de código JavaScript reutilizables utilizados para crear interfaces de usuario (UI) llamadas componentes.” (Hostinger, 2024b).

A través de una encuesta realizada por la plataforma de recursos humanos de Jobbloo: *“React sigue siendo el framework más popular, consolidándose como la opción preferida para construir interfaces de usuario dinámicas y escalables. Angular y Vue.js también mantienen una fuerte presencia.”* (Jobbloo, 2024).

A continuación, se puede ver gráficamente esta afirmación:

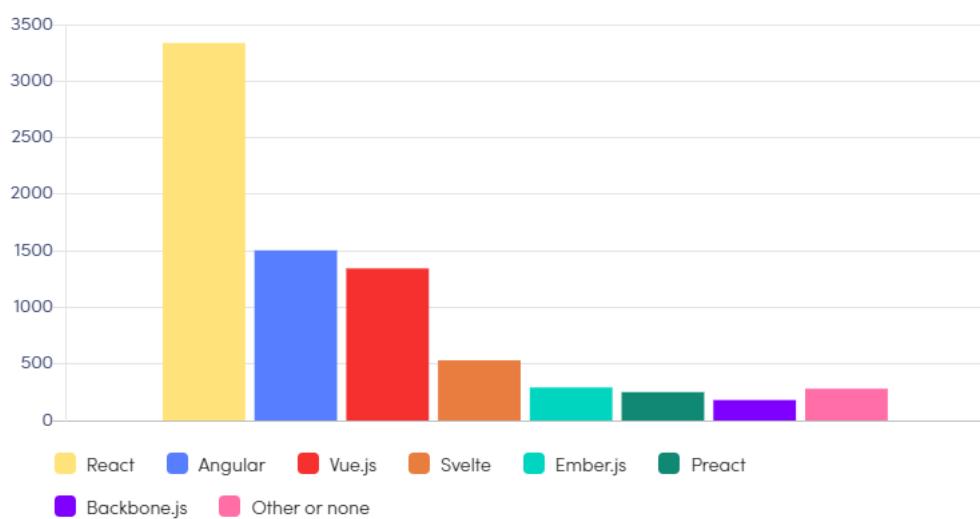


Figura 3.1: [Frameworks frontend más utilizados en 2024](#)
(Appscript, 2024).

Se ha escogido React para el desarrollo del proyecto, en lugar de Angular o Vue, debido a su gran popularidad y consolidación como el framework más utilizado en 2024, lo que asegura una amplia comunidad de desarrolladores y una abundante documentación. Además, React destaca por su enfoque basado en componentes reutilizables, lo que facilita el desarrollo y mantenimiento de interfaces de usuario. También cabe destacar su facilidad con la que se integra con herramientas como Firebase, Google Calendar y OpenStreetMap.

NodeJS

“Node.js es un entorno de tiempo de ejecución de JavaScript que se utiliza para crear aplicaciones escalables del lado del servidor y de red a través de servidores privados virtuales. Ofrece operaciones de entrada/salida (E/S) no bloqueantes y está construido según una arquitectura asincrónica basada en eventos para ayudar a los desarrolladores a crear diversos proyectos de forma eficiente y sencilla.” (Hostinger, 2024a).

Tras un amplio estudio de la competencia, se ha podido concluir que esta tecnología destaca frente a competidores como Python, PHP y Java gracias a su modelo asincrónico y no bloqueante, ideal para manejar múltiples solicitudes simultáneamente. Esta capacidad resulta beneficiosa en la aplicación desarrollada, ya que requiere una respuesta rápida en funcionalidades como el sistema de chat en tiempo real y la gestión de solicitudes de habitaciones.

Firebase

“Firebase de Google es una plataforma en la nube para el desarrollo de aplicaciones web y móvil. Está disponible para distintas plataformas (iOS, Android y web), con lo que es más rápido trabajar en el desarrollo. Aunque fue creada en 2011 pasó a ser parte de Google en 2014, comenzando como una base de datos en tiempo real.” (Mora, 2020).

Se ha escogido esta base de datos en el proyecto porque dispone de una funcionalidad clave denominada Firebase Authentication. *“Firebase Authentication proporciona servicios de backend, SDK fáciles de usar y bibliotecas de IU ya elaboradas para autenticar a los usuarios en la aplicación desarrollada. Admite la autenticación mediante contraseñas, números de teléfono, proveedores de identidad federada populares, como Google, Facebook y Twitter, etc.”* (Firebase, 2024).

Además, Firebase también se ha utilizado para implementar un sistema de chat entre usuarios dentro de la aplicación. Este sistema permite la creación de salas de chat personalizadas entre los usuarios, el almacenamiento de mensajes en tiempo real y su recuperación eficiente. Mediante Firebase Firestore y sus capacidades de sincronización en tiempo real, los usuarios pueden enviar y recibir mensajes instantáneamente, garantizando una comunicación fluida y fiable.

Cloudinary

“Cloudinary es una plataforma en la nube diseñada específicamente para la gestión de medios digitales. Esta herramienta facilita la carga, almacenamiento, manipulación y entrega de imágenes y videos, permitiendo a desarrolladores y empresas optimizar sus recursos visuales de manera eficiente.” (Rondón, 2024)

Para el almacenamiento de imágenes, se ha optado por utilizar Cloudinary en lugar de Firebase. Esto se debe a que Firebase requiere actualmente añadir una tarjeta de crédito para habilitar el almacenamiento de imágenes, lo cual puede suponer una limitación para algunos desarrolladores. Por otro lado, Cloudinary proporciona una solución gratuita para el almacenamiento de imágenes, ofreciendo la generación de URLs que pueden ser utilizadas en Firebase.

GitHub

“Github es un portal creado para alojar el código de las aplicaciones de cualquier desarrollador, y que fue comprada por Microsoft en junio del 2018. La plataforma está creada para que los desarrolladores suban el código de sus aplicaciones y herramientas, y que como usuario no sólo puedes descargarte la aplicación, sino también entrar a su perfil para leer sobre ella o colaborar con su desarrollo.” (Xataka, 2024)

En este proyecto, su uso ha sido de gran utilidad para llevar un control de versiones del código, lo que facilita el seguimiento de los cambios realizados y la posibilidad de revertir a versiones anteriores si es necesario. Además, GitHub actúa como una copia de seguridad en la nube, protegiendo el trabajo ante pérdidas accidentales.

Open Street Map y Leaflet

“Leaflet es la biblioteca líder de código abierto de JavaScript para mapas interactivos compatibles con dispositivos móviles. Leaflet está diseñado teniendo en cuenta la simplicidad, el rendimiento y la facilidad de uso. Funciona de manera eficiente en todas las principales plataformas de escritorio y móviles, se puede ampliar con muchos complementos.” (Leaflet, 2023)

“Openstreetmaps es un proyecto internacional colaborativo para crear mapas gratuitos de todo el mundo. Este proyecto existe desde el año 2004 y entre las personas que colaboran en él se encuentran cartógrafos, profesionales de GIS e ingenieros, y puede usarse libremente.” (Lumion, 2023)

Se han escogido Leaflet y OpenStreetMap debido a que ambas son herramientas gratuitas y de código abierto, lo que permite reducir costos y asegurar una alta flexibilidad en su implementación y personalización.

Trello

“Trello es un software de oficina diseñado para la gestión de actividades y la administración de proyectos de manera colectiva. Este programa informático está optimizado para organizar información en formatos visuales simples que facilitan la realización de tareas y el cumplimiento de objetivos. Esta plataforma está optimizada para hacer más sencillas las tareas de los equipos de trabajo. Mediante una interfaz interactiva, los usuarios pueden acceder a información ordenada sobre un proyecto, un plan de trabajo o una meta a cumplir.” (Terreros, 2024)

Se ha utilizado Trello en el proyecto por su capacidad para organizar y gestionar tareas de manera visual y eficiente. Facilita la gestión por sprints, permitiendo dividir el trabajo en tareas claras, asignables y priorizables. Su interfaz intuitiva ayuda a controlar el progreso y optimizar los recursos del equipo.

Figma

"Figma es una plataforma de diseño basada en la nube que permite a los usuarios crear interfaces de usuario (UI), prototipos interactivos, y colaborar en proyectos de diseño de forma simultánea desde cualquier lugar. Lanzado en 2016, Figma ha ganado rápidamente popularidad entre diseñadores de todo el mundo gracias a su enfoque en la colaboración en tiempo real y su accesibilidad a través de cualquier navegador web. " (CEI, 2024)

Se ha escogido Figma por ser una herramienta gratuita que permite crear interfaces y prototipos de manera eficiente, además de facilitar la colaboración en tiempo real sin necesidad de instalaciones previas.

Postman

"Postman es una herramienta de colaboración y desarrollo que permite a los desarrolladores interactuar y probar el funcionamiento de servicios web y aplicaciones. Proporcionando una interfaz gráfica intuitiva y fácil de usar para enviar solicitudes a servidores web y recibir las respuestas correspondientes. " (FormadoresIT, 2023)

Se ha elegido Postman para realizar las pruebas de la API diseñada en Node.js debido a su facilidad de uso, su interfaz intuitiva y sus amplias funcionalidades para la ejecución de solicitudes HTTP, lo que ha permitido verificar el correcto funcionamiento de los distintos endpoints de la aplicación.

4. Metodología

En esta sección se describen los enfoques y técnicas utilizadas para gestionar y desarrollar el proyecto de manera efectiva. Se ha adoptado una metodología ágil, centrada en la organización del trabajo en **historias de usuario** que detallan las necesidades del usuario final. Además, se ha utilizado un **mapa de historias de usuario** para visualizarlas y priorizarlas adecuadamente. El desarrollo se ha llevado a cabo en ciclos de **sprints**, lo que ha permitido entregar incrementos del proyecto de manera continua. También se ha empleado el **diagrama de casos de uso** para representar las interacciones del usuario con el sistema y el **diagrama de Gantt** para planificar y gestionar los plazos del proyecto de forma clara.

4.1. Gestión del proyecto

La gestión del proyecto es clave para asegurar un desarrollo organizado y eficiente de la aplicación web. En esta sección se describen las metodologías y herramientas utilizadas para coordinar las fases del proyecto, garantizar su avance y cumplir con los objetivos, asegurando que las necesidades de los usuarios sean cubiertas adecuadamente.

Diagrama de casos de uso

El diagrama de casos de uso es una herramienta gráfica que se utiliza en la ingeniería de software para representar las interacciones entre los usuarios y el sistema software.

El diagrama de casos de uso que se presenta en la Figura 4.1, incluye cuatro actores principales: dos APIs externas (Leaflet, utilizada para la visualización de mapas, y la API de Ciudades, que proporciona el listado completo de ciudades) y dos tipos de usuarios internos. El invitado tiene acceso limitado a funcionalidades, tan solo puede consultar habitaciones disponibles y buscar compañeros de piso, mientras que el usuario autenticado puede gestionar sus propias habitaciones, además de acceder a un conjunto más amplio de funcionalidades.

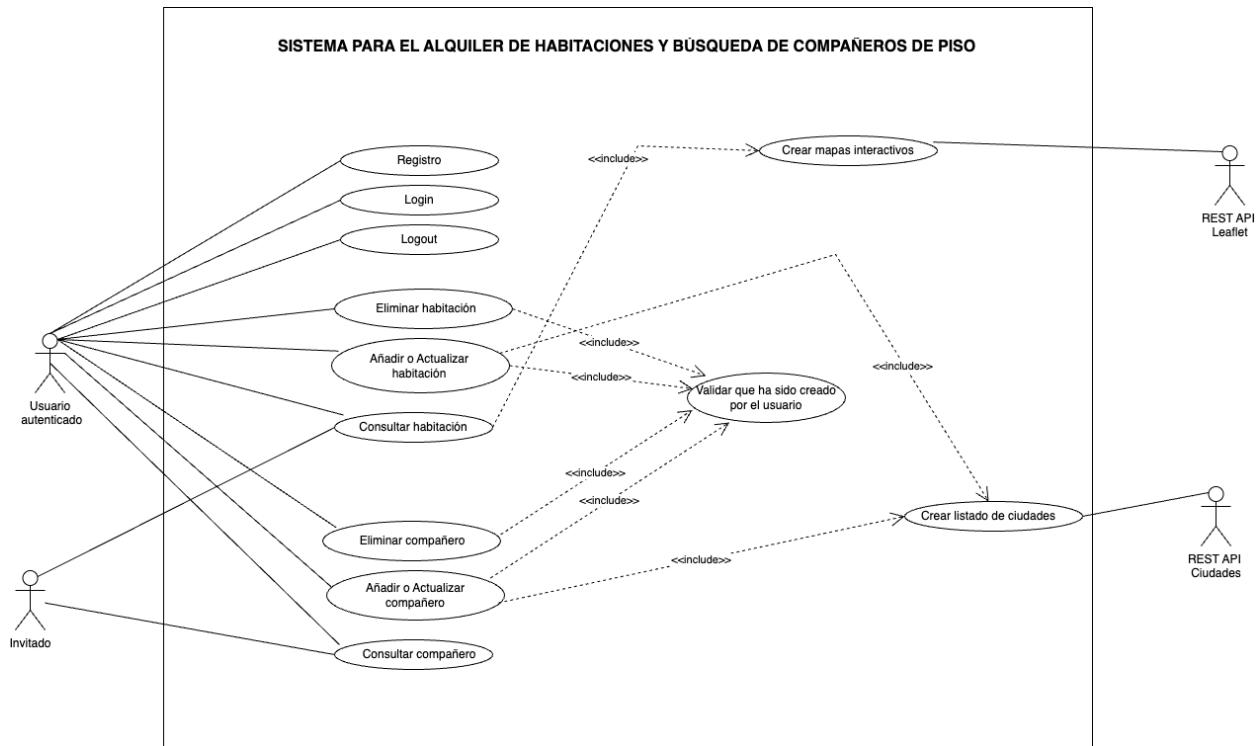


Figura 4.1: [Diagrama de casos de uso](#) (Fuente: Elaboración propia))

Diagrama de base de datos

En este proyecto se utiliza Firebase Firestore como sistema de base de datos no relacional basado en documentos. Dado que Firestore organiza la información mediante colecciones y documentos en lugar de tablas, el diagrama UML presentado a continuación sirve como referencia conceptual para la estructura de datos. La implementación final emplea documentos que pueden incluir información anidada en forma de mapas o establecer relaciones mediante referencias a otros documentos.

En la base de datos mostrada en la Figura 4.2, se emplean dos tipos de relaciones: las representadas por mapas y las de referencias a documentos.

- Los **mapas** permiten almacenar información jerárquica dentro de un solo campo, como el campo `address` en el documento `room`, que agrupa subcampos como `street`, `city`, `zipCode`, `country`. Esto permite representar datos complejos sin necesidad de crear múltiples documentos.
- Las **referencias a documentos** se usan cuando un campo almacena el ID de otro documento en una colección diferente, lo que permite establecer relaciones entre documentos sin duplicar datos.

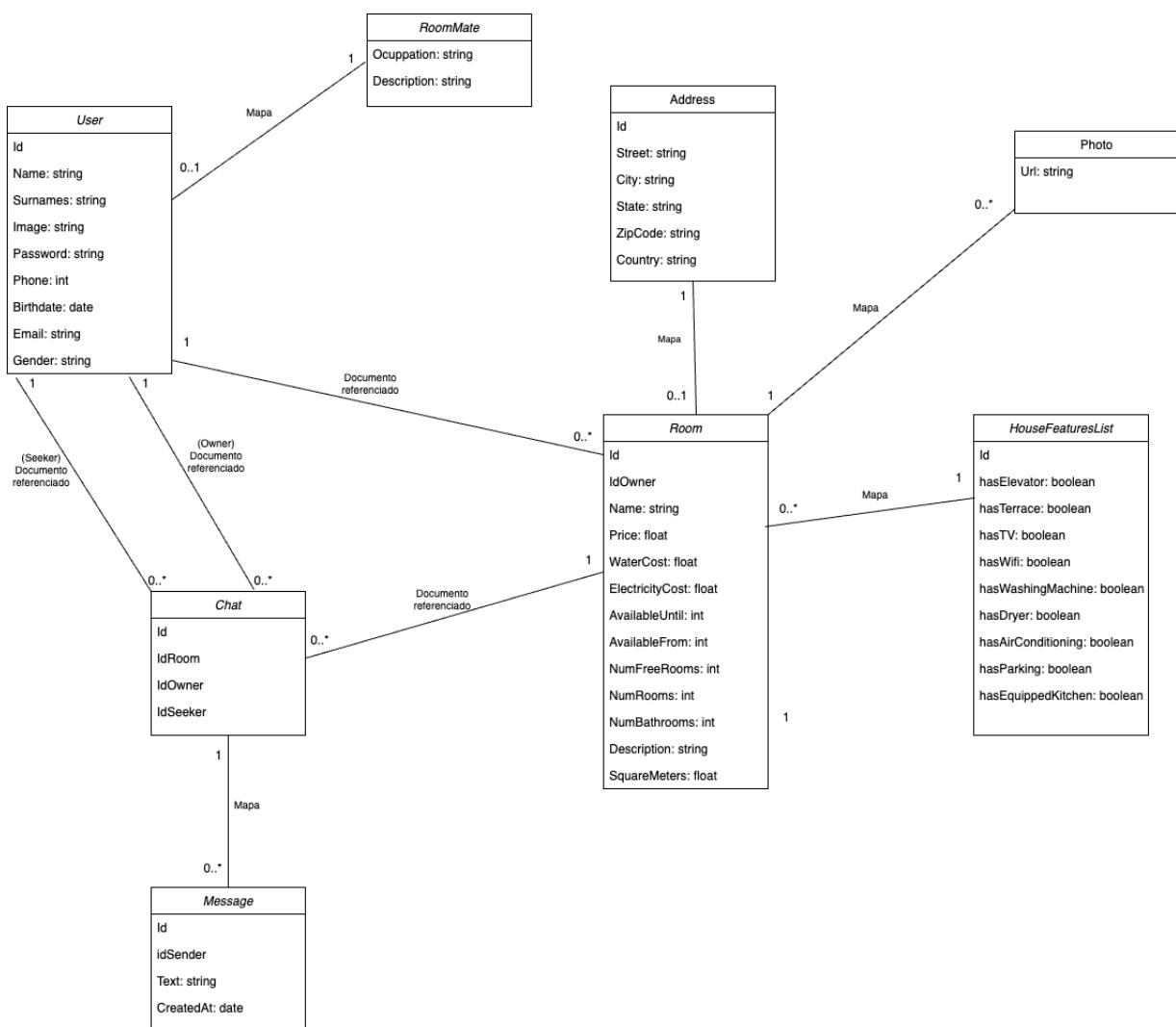


Figura 4.2: Diagrama UML (Fuente: Elaboración propia)

Historias de usuario

Las historias de usuario son fundamentales en el desarrollo ágil, ya que definen las funcionalidades de la aplicación desde la perspectiva del usuario final. En lugar de detallar especificaciones técnicas, se enfocan en los objetivos y expectativas de los usuarios que interactuarán con la aplicación.

En esta subsección se presentan las principales historias de usuario que guiarán el diseño y la implementación, enfocándose en las necesidades de los Buscadores y Anunciantes de habitaciones y compañeros de piso. A continuación, se describirán únicamente las historias de usuario relacionadas con el registro y acceso del usuario, mientras que el resto de historias de usuario se encuentran detalladas en el [Anexo A](#).

| Registro de usuarios | | | |
|--|------------------|-------------------|--------------------|
| Descripción: Como usuario quiero poder registrar mis datos para poder acceder a la aplicación. | | | |
| Validación: | | | |
| <ul style="list-style-type: none"> ■ Los datos introducidos tienen un formato válido ■ Los datos obligatorios (nombre, apellidos y email) han sido completados ■ El email del usuario no ha sido registrado previamente | | | |
| Id | Prioridad | Estimación | Dependencia |
| 1 | 1 | 25h | Ninguna |

Tabla 4.1: Historia de usuario - Registro usuarios

| Acceso usuarios | | | |
|--|------------------|-------------------|--------------------|
| Descripción: Como usuario quiero acceder a la aplicación a través de una interfaz en la que tenga que introducir el usuario y la contraseña. | | | |
| Validación: | | | |
| <ul style="list-style-type: none"> ■ Se permite el acceso a los usuarios que hayan sido previamente registrados en la aplicación ■ Se permite el acceso a los usuarios que hayan sido validados por Google ■ Denegar acceso con credenciales erróneas | | | |
| Id | Prioridad | Estimación | Dependencia |
| 2 | 1 | 17h | 1 |

Tabla 4.2: Historia de usuario - Acceso usuarios

Mapa de historias de usuario

El mapa de historias de usuario es una herramienta visual que organiza las funcionalidades de la aplicación en tres categorías: temas, epics y tareas. Esta estructura facilita la planificación y visualización de las tareas necesarias para el desarrollo de la aplicación.

- Los **temas** representan las características principales de la aplicación y agrupan las historias de usuario relacionadas. En la Figura 4.3, se identifican en color lila.
- Los **epics** son funcionalidades complejas que generalmente no pueden completarse en un solo sprint. A menudo, se dividen en varias historias de usuario y agrupan múltiples tareas. Están representadas en color verde.

- Las **tareas** son las unidades de trabajo más específicas. Se han definido para ser lo más breves y autónomas posible, permitiendo su implementación de manera aislada. Se muestran en color amarillo.

A continuación en la Figura 4.3, se presenta el mapa diseñado para el proyecto que está dividido en tres sprints.

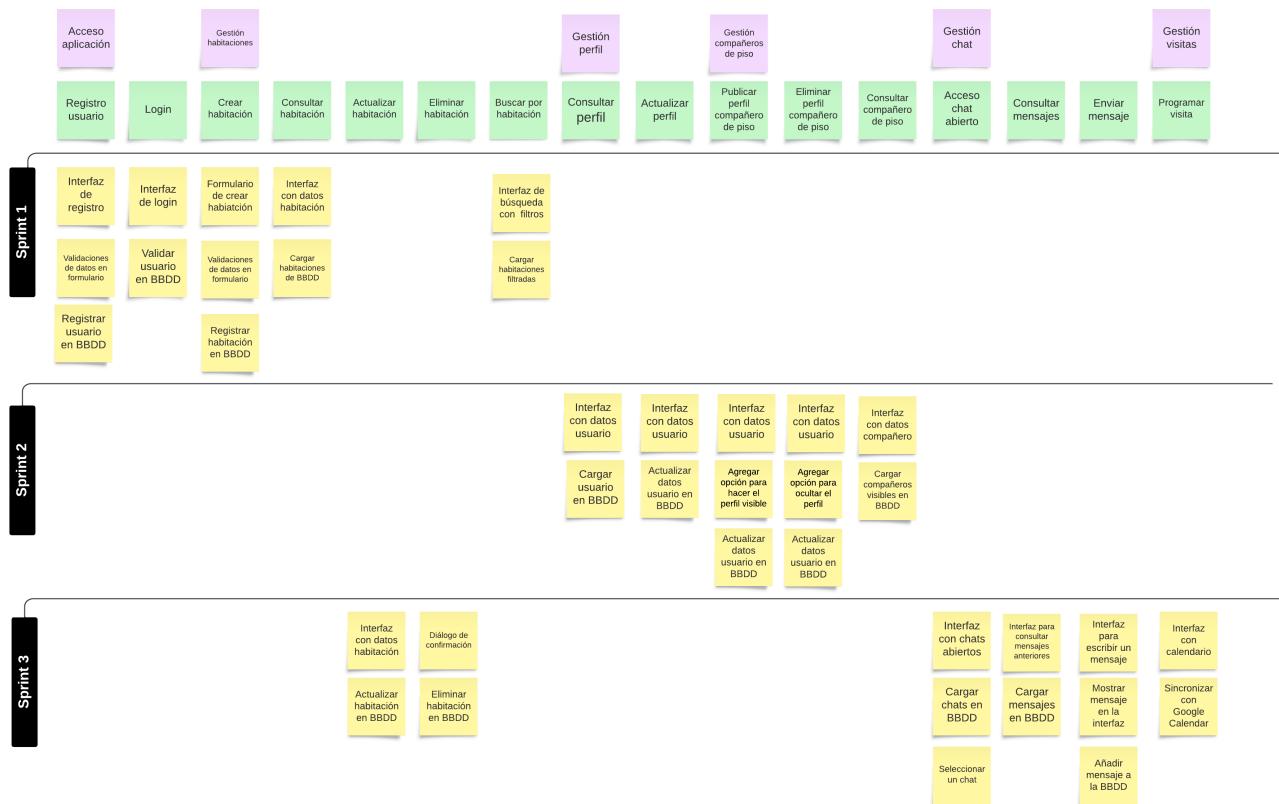


Figura 4.3: Mapa de historias de usuario
(Fuente: Elaboración propia)

Sprints

Los sprints son períodos de tiempo en los que se lleva a cabo un conjunto específico de tareas dentro del marco de trabajo ágil, especialmente en metodologías como Scrum. En este proyecto, se ha determinado que cada sprint tiene una duración de 3 semanas, y su objetivo es completar el conjunto de funcionalidades definidas previamente.

En el desarrollo de este proyecto, se han generado tres sprints para organizar y estructurar las tareas de manera eficiente.

- En este **primer sprint**, se implementarán las funcionalidades básicas esenciales para que la aplicación sea operativa. Esto incluirá el registro de usuarios, la autenticación (login), la creación y consulta de habitaciones, así como la capacidad de filtrar habitaciones según las preferencias del usuario.

- En el **segundo sprint**, el enfoque estará en mejorar la interacción del usuario con la plataforma. Se añadirá la funcionalidad para consultar el perfil del usuario, actualizar el perfil y activar los perfiles de compañeros de piso, permitiendo a los usuarios ofrecerse como compañeros en la plataforma.
- El **último sprint** se centrará en la implementación de la funcionalidad de chat para facilitar la comunicación entre usuarios, además de la capacidad de actualizar y eliminar habitaciones. También se permitirá a los usuarios consultar los compañeros de piso disponibles, facilitando así la búsqueda de compañeros con intereses similares. Adicionalmente, se incluirá la funcionalidad de programar visitas directamente desde la plataforma, las cuales se sincronizarán automáticamente con Google Calendar.

Diagrama de Gantt

Se ha elaborado un diagrama de Gantt presentado en la Figura 4.4 como planificación inicial del proyecto, estableciendo una organización semanal que abarca el período comprendido entre noviembre y febrero.

En el desarrollo backend se han establecido tres sprints, cada uno con una duración de tres semanas, reservando la última semana para actividades de detección y corrección de posibles errores. Cabe destacar que el diseño de mockups está directamente vinculado a cada sprint: antes de iniciar un nuevo sprint, es imprescindible haber diseñado los mockups correspondientes a las tareas que se abordarán en dicho sprint. Esta planificación asegura una dependencia entre las fases de diseño y desarrollo.



Figura 4.4: **Diagrama de Gantt**
(Fuente: Elaboración propia)

4.2. Gestión de tareas con Trello

A continuación, se puede observar el tablero que se ha diseñado en Trello para la aplicación de HomeMate. Se han definido tres tipos de tarjetas:

1. Pendiente: Representan tareas que aún no han comenzado.
2. En desarrollo: Indican tareas que están actualmente en proceso.
3. Finalizada: Señalan las tareas que ya han sido completadas.

En el tablero de Trello mostrado en la Figura 4.5, se ha implementado un sistema de organización que clasifica las tarjetas, tanto por etiquetas como por prioridad. Las etiquetas identifican el sprint al que pertenece cada tarjeta: **verde** para el Sprint 1, **naranja** para el Sprint 2 y **morado** para el Sprint 3. Además, las tarjetas están ordenadas dentro de cada columna según su nivel de prioridad, lo que permite centrar la atención en las tareas más importantes en cada sprint.

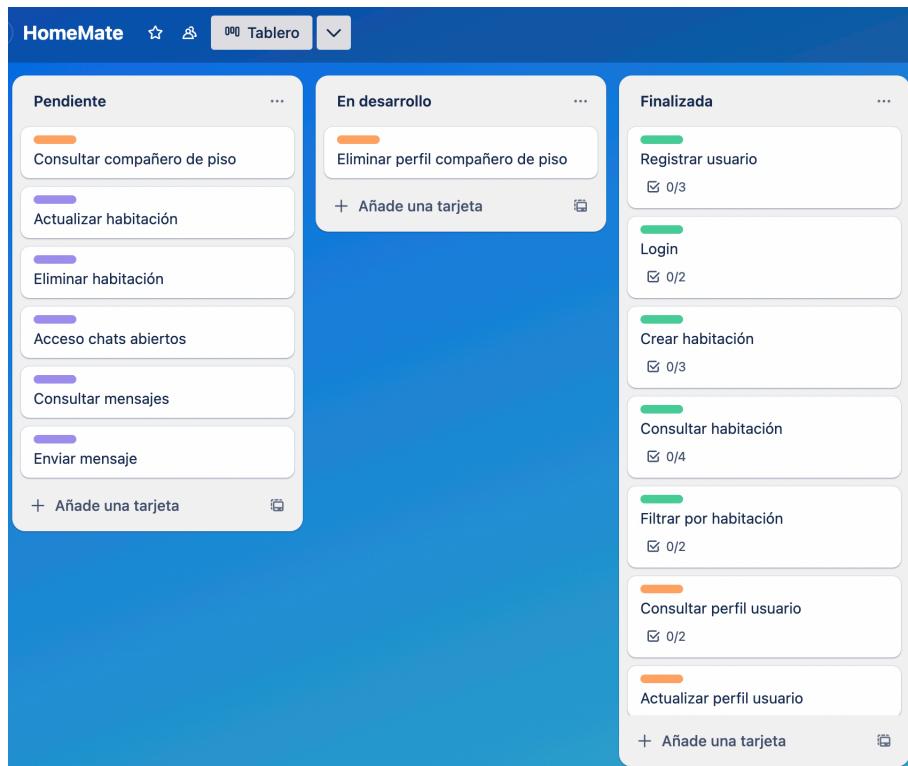


Figura 4.5: Tablero en Trello
(Fuente: Elaboración propia)

Además, cada tarea puede incluir un conjunto de ítems detallados. Estos han sido especificados para asegurar que en el momento de desarrollar se tenga claro qué se debe lograr, permitiendo así considerar la tarea como completada, tal como se muestra en la Figura 4.6.

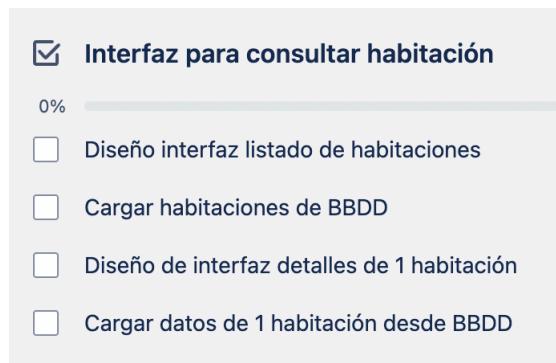


Figura 4.6: Items de la tarea
(Fuente: Elaboración propia)

4.3. Modelos para la interfaz (MockUps)

Los mockups son representaciones gráficas de la interfaz de usuario de la aplicación web, diseñadas para proporcionar una visión clara y detallada de cómo se verá y funcionará el producto final.

Los mockups han sido diseñados con la herramienta de Figma en la fase de diseño del proyecto. A continuación, se presentan algunos de los diseños clave que servirán como base para la implementación de la aplicación.

Menús

Como se muestra en la Figura 4.7, se ha diseñado un boceto inicial del menú de navegación de la aplicación, adaptable según el estado del usuario (autenticado o no). Para usuarios no autenticados, el menú ofrece opciones básicas, mientras que para los usuarios autenticados incluye servicios como acceso al chat, publicación de anuncios, gestión de propiedades y visualización del perfil personal.

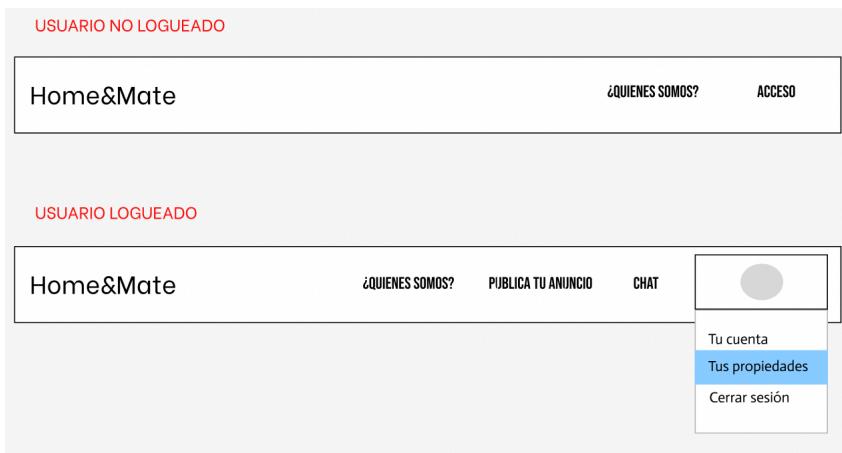


Figura 4.7: Menú mockup (Fuente: Elaboración propia))

Login y Registro usuarios

La aplicación cuenta con un formulario de login y registro que permite a los usuarios autenticarse o crear una cuenta de manera sencilla, como se muestra en la Figura 4.8.

The mockup displays two side-by-side forms: 'Registro usuario' (Registration) on the left and 'Log in' on the right.

Registro usuario:

- REGISTRO USUARIO**
- Icono de perfil.
- *Nombre: [Input field]
- *Apellidos: [Input field]
- Género: [Input field] con un ícono de flecha abajo.
- *Email: [Input field]
- Fecha de nacimiento: [Input field]
- Teléfono: [Input field]
- *Contraseña: [Input field]
- *Repetir contraseña: [Input field]
- Registrarme** [Botón]

Log in:

- INICIO SESIÓN**
- Email: [Input field]
- Contraseña: [Input field]
- ¿Eres un nuevo usuario? [Regístrate aquí](#)
- También puedes
- Sign in with Google [Botón]
- Iniciar sesión** [Botón]

Figura 4.8: Login y Registro mockup (Fuente: Elaboración propia))

Publicar habitación

En la Figura 4.9 se ha diseñado un mockup para la funcionalidad de Publicar Habitación, que permite a los usuarios registrados ofrecer sus habitaciones disponibles.

The figure displays two sequential screens for publishing a room listing. The first screen, titled 'Publicar anuncio (Ubicación)', asks for location details: Calle, Número, Ciudad, Código postal, and Provincia. It includes a back button ('Volver atrás') and a next button ('Siguiente'). The second screen, titled 'Publicar anuncio (Características piso)', asks for room characteristics: Nombre del anuncio, Precio (€), m2, Disponible desde, Disponible hasta, Gasto agua (€), Gasto luz (€), Número habitaciones libres, Número baños, and a list of other features with checkboxes (Terraza, Calefacción, Cocina equipada, Wifi, Parking, Amueblado, TV, Lavadora, Gimnasio, Aire acondicionado, Secadora, Piscina). It also includes a text area for 'Descripción' and a final 'Siguiente' button.

Figura 4.9: Publicar habitación mockup (Fuente: Elaboración propia))

Perfil de usuario y opciones de configuración como compañero de piso

En la Figura 4.10 se presenta el mockup del perfil de usuario, que incluye una sección dedicada a la opción de ofrecerse como compañero de piso. Esta funcionalidad permite al usuario mostrar su perfil visible para todos los usuarios de la aplicación.

Perfil usuario autenticado

Home&Mate

PERFIL DEL USUARIO


CAMBiar FOTO
NOMBRE
GMAIL

INFORMACIÓN PERSONAL:

| | |
|----------------------|----------------------|
| *Nombre: | *Apellidos: |
| <input type="text"/> | <input type="text"/> |
| *Email: | Teléfono: |
| <input type="text"/> | <input type="text"/> |
| Género: | Fecha de nacimiento: |
| <input type="text"/> | <input type="text"/> |

COMPAÑERO DE PISO

Al seleccionar esta opción, aceptas ofrecerte como compañero de piso y compartir tu información de contacto con otros usuarios interesados.

Ofrecerme como compañero de piso

*Descripción personal:

*Ocupación: Edad:

Actualizar

Figura 4.10: Perfil usuario mockup (Fuente: Elaboración propia))

5. Resultados obtenidos

En esta sección se presentan los resultados obtenidos tras el desarrollo del proyecto, abarcando tanto la interfaz de usuario como el funcionamiento del backend. Se analizan las vistas finales de la aplicación, la estructura y operación de la API Rest, y las funcionalidades avanzadas incorporadas, como la sincronización con Google Calendar y la integración de mapas interactivos.

5.1. Vistas de la aplicación final

En esta sección se presentan las vistas finales de la aplicación HomeMate, desarrolladas a partir de los mockups creados en la fase de diseño, comentados en el apartado 4.3. Aunque estas vistas siguen la estructura y funcionalidad planificadas, se han realizado algunos ajustes durante el desarrollo para mejorar la usabilidad y adaptarse a las necesidades técnicas. A continuación, se detallan las distintas pantallas y su papel dentro de la aplicación.

Menú usuario invitado vs autenticado

La aplicación cuenta con dos menús adaptados según el tipo de usuario: invitado o autenticado. En la Figura 5.1 se muestra el menú adaptado para el usuario invitado, el cual permite explorar la sección **Quiénes somos**, que contiene listados con las habitaciones y compañeros de piso que se han subido en la plataforma. Además, se ofrece la opción de **Acceso** para iniciar sesión o registrarse y desbloquear funcionalidades adicionales.



Figura 5.1: [Vista Menú usuario invitado](#) (Fuente: Elaboración propia))

En la Figura 5.2 se presenta el menú adaptado para el usuario autenticado, el cual incluye acceso a la opción de **publicar anuncio**, un módulo de **chats** para comunicarse con otros usuarios, y secciones como **gestión de cuenta** y **mis propiedades** para administrar su contenido.



Figura 5.2: [Vista Menú usuario autenticado](#) (Fuente: Elaboración propia))

Inicio de sesión

En la Figura 5.3 se muestra la vista de inicio de sesión, que permite a los usuarios acceder a la aplicación mediante autenticación nativa o a través de Google. En la autenticación

nativa, el usuario introduce su correo electrónico y contraseña en el formulario. En la autenticación con Google, se facilita el acceso mediante una cuenta de Google. Para los nuevos usuarios, la vista incorpora un enlace a la página de Registro, permitiendo crear una nueva cuenta.

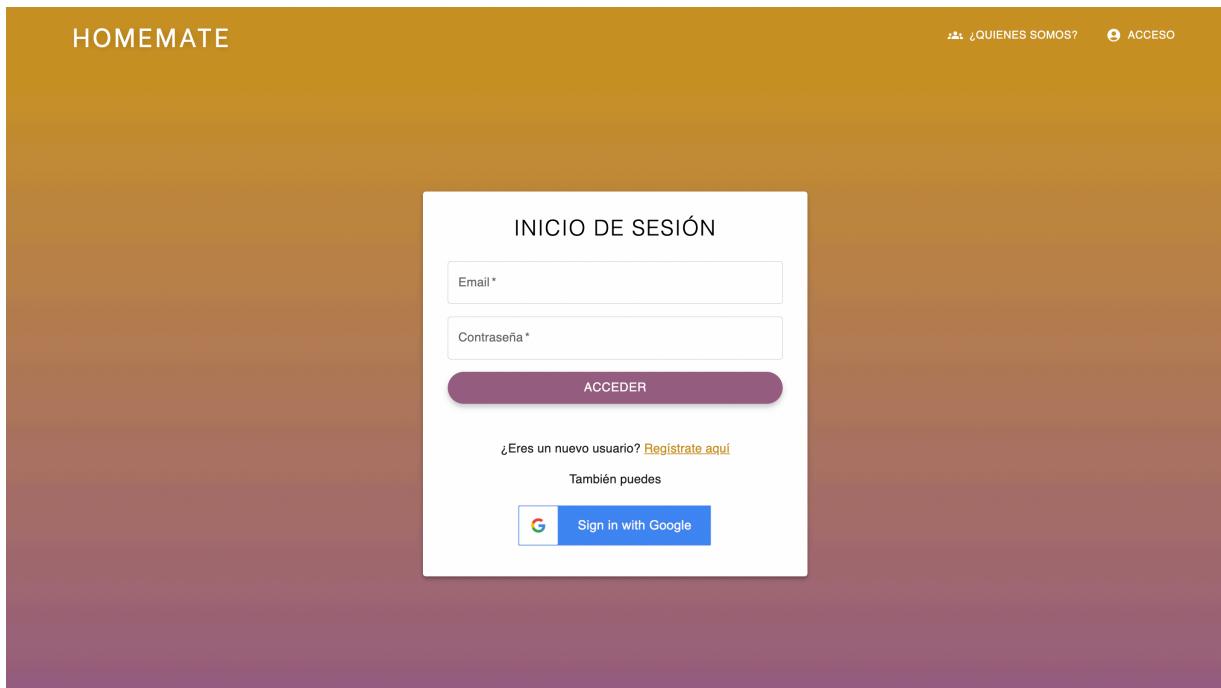


Figura 5.3: [Vista Inicio sesión](#) (Fuente: Elaboración propia))

En la Figura 5.4 se muestra la vista de inicio de sesión, en la cual se gestionan tres posibles errores para mejorar la experiencia del usuario:

- **Campos incompletos:** Si el usuario intenta iniciar sesión sin completar el correo o la contraseña.
- **Cuenta inexistente:** Si el correo ingresado no está registrado en la plataforma.
- **Credenciales inválidas:** Si el correo existe pero la contraseña no coincide.



Figura 5.4: [Vista Mensajes error login](#) (Fuente: Elaboración propia))

Registro usuario

En la vista de registro, presentada en la Figura 5.5, los nuevos usuarios pueden crear una cuenta completando campos como nombre, apellidos, fecha de nacimiento, foto de perfil, entre otros.

The screenshot shows the registration form for the HomeMate application. At the top left is the 'HOMEMATE' logo. On the right are links for 'QUIENES SOMOS?' and 'ACESO'. The main title 'REGISTRO' is centered above a placeholder icon of a person. Below the title are five input fields arranged in two rows: 'Nombre *' and 'Apellidos' in the first row, and 'Email *' and 'Teléfono' in the second row. A dropdown menu for 'Género' and a date picker for 'Fecha de nacimiento*' are also present. A password field for 'Contraseña *' follows. Below these fields is a note: 'Por favor, verifica que no eres un robot para completar el registro.' A reCAPTCHA checkbox labeled 'No soy un robot' is shown next to the reCAPTCHA logo. At the bottom is a large purple 'REGÍSTRAME' button. Below it, a link says '¿Ya estás registrado? [Inicia sesión](#)'.

Figura 5.5: Vista Registro usuario (Fuente: Elaboración propia))

Para asegurar la seguridad, cada campo del formulario cuenta con validaciones y se requiere la verificación mediante reCAPTCHA para prevenir registros automatizados. A continuación, en la Figura 5.6 se detallan algunos errores que pueden surgir durante el proceso de registro.

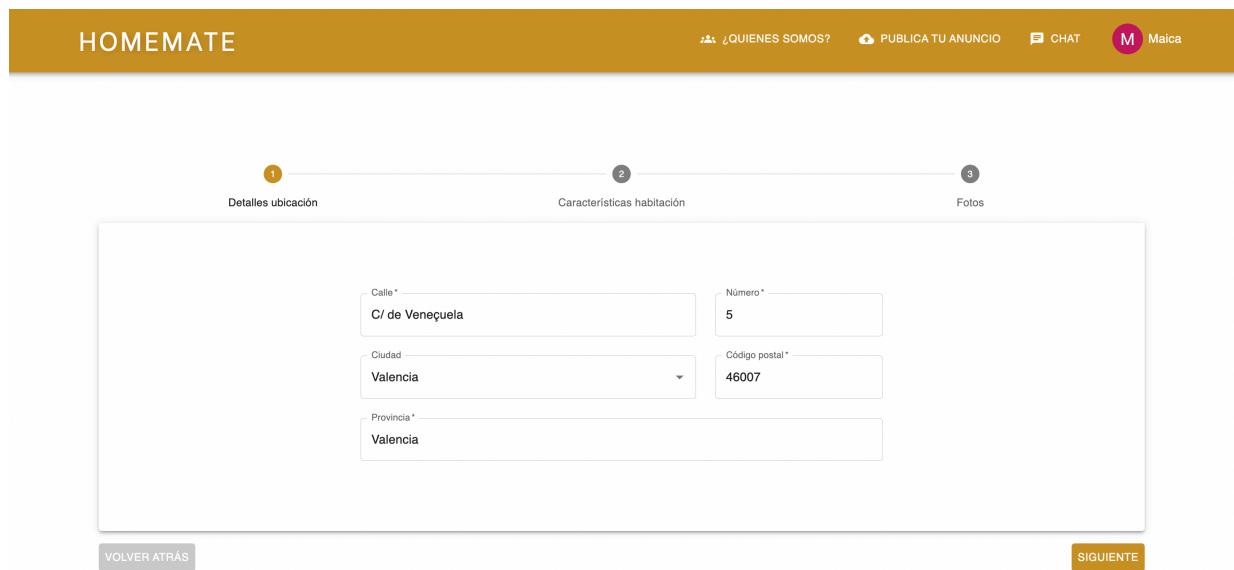
The screenshot shows the 'REGISTRO' (Registration) form on the HomeMate website. At the top, there is a red error message box stating: 'Por favor, corrige los errores antes de registrarte.' (Please correct the errors before registering). The registration form includes fields for Name*, Surname, Email*, Phone, Gender, Birthdate, and Password*. Below the form is a CAPTCHA section with a checkbox for 'No soy un robot'. A purple 'REGÍSTRATE' button is at the bottom, and a link '¿Ya estás registrado? Inicia sesión' is at the very bottom.

Figura 5.6: Vista Mensajes error registro usuario (Fuente: Elaboración propia)

Crear habitación

El proceso para crear una habitación se divide en tres pasos secuenciales: detalles de ubicación, características de la habitación y fotos. El usuario debe completar cada paso antes de poder avanzar al siguiente, asegurando que toda la información necesaria se ingrese de forma ordenada. Además, se ofrece la opción de volver atrás en cualquier momento para revisar o modificar los datos introducidos.

En el primer paso, como se muestra en la Figura 5.7 el usuario debe ingresar la dirección, ciudad y otros datos relacionados con la ubicación de la habitación.



1 Detalles ubicación 2 Características habitación 3 Fotos

Calle *
C/ de Venezuela

Número *
5

Ciudad
Valencia

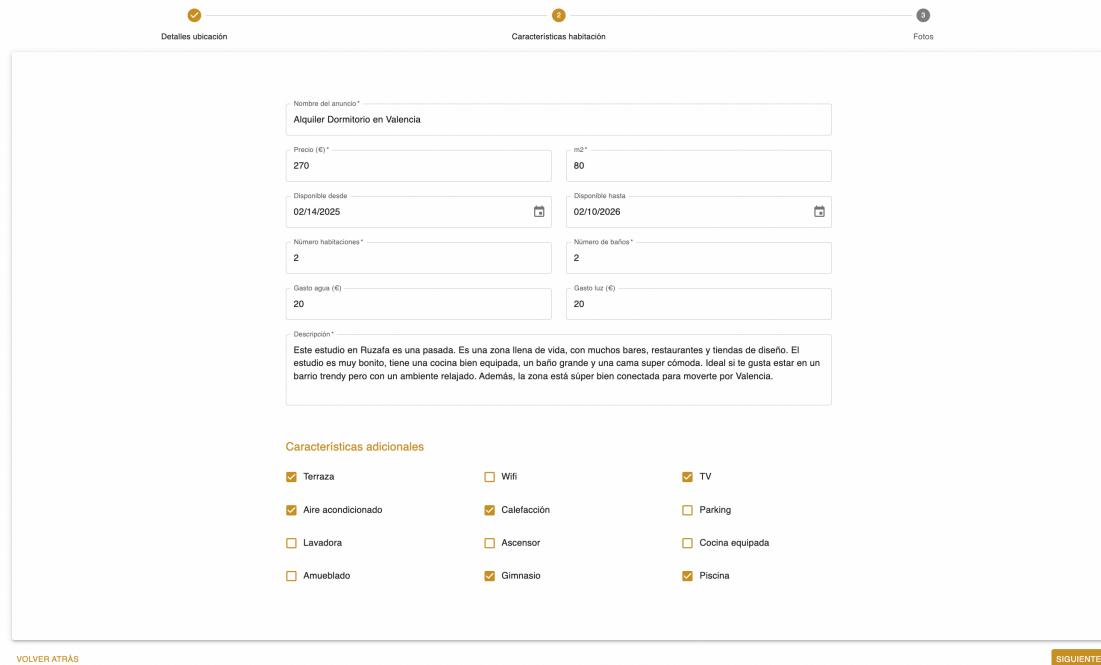
Código postal *
46007

Provincia *
Valencia

VOLVER ATRÁS SIGUIENTE

Figura 5.7: Vista Crear habitación paso 1 (Fuente: Elaboración propia))

La Figura 5.8 presenta el segundo paso del proceso, donde el usuario debe asignar un nombre al anuncio, indicar el precio por habitación, la disponibilidad de entrada y otros detalles importantes. Además, deberá seleccionar las características de la habitación, como terraza, wifi, lavadora, entre otras.



1 Detalles ubicación 2 Características habitación 3 Fotos

Nombre del anuncio *
Alquiler Dormitorio en Valencia

Precio (€) *
270

m² *
80

Disponible desde
02/10/2025

Disponible hasta
02/10/2026

Número habitaciones *
2

Número de baños *
2

Gasto agua (€)
20

Gasto luz (€)
20

Descripción *
Este estudio en Ruzafa es una pasada. Es una zona llena de vida, con muchos bares, restaurantes y tiendas de diseño. El estudio es muy bonito, tiene una cocina bien equipada, un baño grande y una cama super cómoda. Ideal si te gusta estar en un barrio trendy pero con un ambiente relajado. Además, la zona está super bien conectada para moverte por Valencia.

Características adicionales

| | | |
|--|---|---|
| <input checked="" type="checkbox"/> Terraza | <input type="checkbox"/> Wifi | <input checked="" type="checkbox"/> TV |
| <input checked="" type="checkbox"/> Aire acondicionado | <input checked="" type="checkbox"/> Calefacción | <input type="checkbox"/> Parking |
| <input type="checkbox"/> Lavadora | <input type="checkbox"/> Ascensor | <input type="checkbox"/> Cocina equipada |
| <input type="checkbox"/> Arriendado | <input checked="" type="checkbox"/> Gimnasio | <input checked="" type="checkbox"/> Piscina |

VOLVER ATRÁS SIGUIENTE

Figura 5.8: Vista Crear habitación paso 2 (Fuente: Elaboración propia))

En la Figura 5.9 se muestra el último paso del proceso, donde el usuario sube imágenes de la habitación para proporcionar una representación visual de su estado.

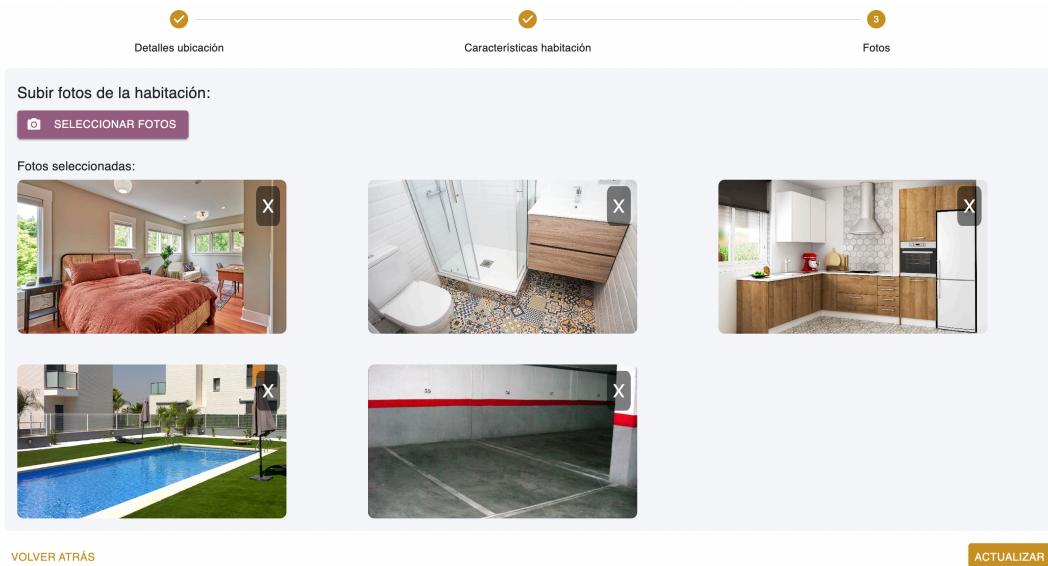


Figura 5.9: Vista Crear habitación paso 3 (Fuente: Elaboración propia))

Consultar habitaciones

La Figura 5.10 presenta la vista de listado de habitaciones y permite al usuario filtrar los resultados según varios criterios, como ciudad, metros cuadrados, número mínimo de habitaciones, entre otros. Esto facilita la búsqueda y permite encontrar opciones que se ajusten mejor a las necesidades del usuario.

HOMEMATE

[QUIENES SOMOS](#) [ACCESO](#)

FILTROS

Ordenar por
Más recientes

RESTABLECER

Total de 8 habitaciones



Alquiler Dormitorio en Valencia
C/ de Venezuela, Valencia
Disponible desde: 14/02/2025
2 habs - 80 m² - 2 baños



Habitación en centro de Madrid
C. de Calatrava, Madrid
Disponible desde: 28/02/2025
4 habs - 91 m² - 2 baños



Dormitorio en Valencia
C/ de Venezuela, Valencia
Disponible desde: 09/02/2025
2 habs - 60 m² - 2 baños



Alquiler de dormitorio en Valencia
C/ de Conca, Valencia
Disponible desde: 10/02/2025
4 habs - 98 m² - 2 baños

Figura 5.10: [Vista Consultar habitaciones](#) (Fuente: Elaboración propia)

Si el usuario selecciona alguna de las habitaciones del listado, el sistema muestra los detalles completos de la habitación como se muestra en la Figura 5.11. Se incluyen un visor con las imágenes, sus características, una calculadora automática para estimar los gastos totales y un mapa con la ubicación exacta.

The screenshot shows a detailed listing for a room for rent. At the top is a large image of a bedroom with a double bed covered in red bedding, a desk, and several windows. Below the image are five smaller thumbnail images of other parts of the property. To the left of the main image is the price '270 €/mes'. To the right is a purple button labeled 'CONTACTAR POR CHAT'. Below the main image are three status indicators: 'C/ de Venezuela' with a location pin, 'Disponible el 14/05/2025' with a calendar icon, and '2 habitaciones libres' with a bed icon.

Alquiler Dormitorio en Valencia

ESPECIFICACIONES DEL ALOJAMIENTO:

- Habitación en piso de 80 m²
- 2 baños
- Con terraza
- Sin ascensor

DESCRIPCIÓN:
Este estudio en Ruzafa es una pasada. Es una zona llena de vida, con muchos bares, restaurantes y tiendas de diseño. El estudio es muy bonito, tiene una cocina bien equipada, un baño grande y una cama super cómoda. Ideal si te gusta estar en un barrio trendy pero con un ambiente relajado. Además, la zona está súper bien conectada para moverte por Valencia.

COMODIDADES DEL ALOJAMIENTO:

Terraza TV Aire acondicionado Calefacción Gimnasio Piscina

CALCULADORA DE GASTOS:

| | |
|---------------------------------------|-----|
| Costo de alquiler (€) | 270 |
| Costo de agua (€) | 20 |
| Costo de electricidad (€) | 20 |
| Duración (meses) | 1 |
| Costo total estimado por 1 mes: 310 € | |

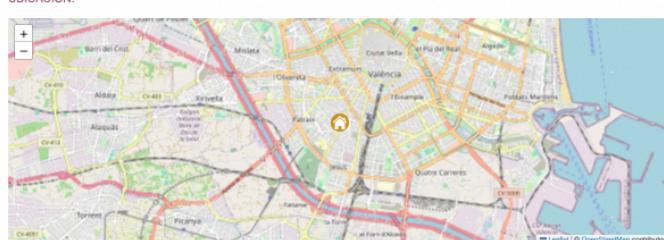
UBICACIÓN:

Figura 5.11: [Vista Consultar detalles de la habitación](#) (Fuente: Elaboración propia)

Actualizar o eliminar habitación

La Figura 5.12 presenta la vista de "Mis propiedades", donde el usuario puede ver un listado con todas las habitaciones que ha creado. Desde esta sección, el usuario tiene la opción de actualizar o eliminar cualquiera de las habitaciones según lo deseé.

The screenshot shows the 'MIS HABITACIONES' (My Apartments) section of the HomeMate application. At the top, there is a banner with the text 'HOMEMATE' and navigation links for '¿QUIENES SOMOS?', 'PUBLICA TU ANUNCIO', 'CHAT', and a user profile for 'Vienna'. Below the banner, it says 'Total de 1 habitaciones'. A single listing is shown: 'Alquiler de dormitorio en Barcelona' located at 'Calle Arrabal, Barcelona', available from '12/02/2025', with '2 habs - 100 m² - 2 baños'. The price is listed as '€390'. At the bottom of the listing are two buttons: 'ELIMINAR' (Delete) and 'ACTUALIZAR' (Update).

Figura 5.12: [Vista Actualizar o eliminar habitación](#) (Fuente: Elaboración propia))

Consultar o actualizar perfil personal

En la Figura 5.13 se puede observar la vista de perfil personal, donde el usuario puede presionar para actualizar su información. Si no lo hace, los campos permanecen deshabilitados. Al seleccionar la opción **Ofrecerme como compañero de piso**, el sistema habilita un conjunto de campos obligatorios adicionales que incluyen una descripción personal, el tipo de empleo y la ciudad de residencia, los cuales deben ser completados por el usuario.



Vienna Herraiz Hortelano
vienna@gmail.com

Información Personal

| | |
|----------------------------|------------------------------------|
| Nombre* Vienna | Apellidos Herraiz Hortelano |
| Email* vienna@gmail.com | Teléfono 676543456 |
| Género Mujer | Fecha de nacimiento* 02/10/2001 |

Compañero de piso

Al seleccionar esta opción, aceptas ofrecerte como compañero de piso y compartir tu información de contacto con otros usuarios interesados.

Ofrecerme como compañero de piso

Descripción
Soy Vienna, una persona bastante tranquila y soñadora. Me encanta leer, especialmente novelas y poesía, y siempre estoy buscando maneras de encontrar equilibrio en mi vida. Uno de mis pasatiempos favoritos es cuidar mi jardín, me da mucha paz. Aunque disfruto de la calma, también me encanta viajar y conocer nuevas culturas. Si alguna vez quieres hablar de libros, bienestar o compartir historias de viajes, ¡soy toda oídos!

Tipo de empleo
Estudiante y trabajando

Ciudad
Madrid

ACTUALIZAR INFORMACIÓN

Figura 5.13: Vista Consultar o actualizar perfil personal (Fuente: Elaboración propia)

Consultar compañeros de piso

En la sección de consultar compañeros de piso mostrada en la Figura 5.14, aparecerán los usuarios que han activado la opción 'Ofrecerme como compañero de piso' en su perfil. La búsqueda puede ser filtrada por ciudad, género, tipo de empleo y edad, permitiendo encontrar compañeros de piso que coincidan con las preferencias del usuario.

The screenshot shows the HomeMate platform's search interface. At the top, there is a navigation bar with links for 'QUIENES SOMOS?', 'PÚBLICA TU ANUNCIO', 'CHAT', and a user profile for 'Vienna'. Below the navigation bar is a search form titled 'FILTROS' with dropdown menus for 'Ciudad', 'Género', 'Tipo de empleo', 'Edad mínima', and 'Edad máxima', followed by a 'RESTABLECER' button. A message below the search bar states 'Total de 5 compañeros de piso'. Three user profiles are displayed in cards:

- Mario Gutiérrez Castro**
Edad: 27 años
Tipo de trabajo: Estudiante en prácticas
Bio: ¡Soy Mario! Me apasiona todo lo relacionado con la tecnología y me gusta mantenerme al día con las últimas tendencias. Siempre estoy buscando nuevas herramientas y proyectos para explorar. En mis tiempos libres me gusta desconectar con una caminata en la naturaleza, pero también soy un amante de la calma y la reflexión. Si tienes alguna
- Valeria Ruiz Hortelano**
Edad: 32 años
Tipo de trabajo: Trabajando a tiempo parcial
Bio: Soy Valeria, una persona extrovertida y curiosa. Me encanta conocer gente nueva y aprender de sus experiencias. Mi pasión por la música me ha llevado a tocar varios instrumentos, y siempre estoy buscando nuevas canciones o géneros para escuchar. Soy también una amante de los libros, especialmente los de ficción y desarrollo personal. En mi tiempo
- Pedro Herraiz Santos**
Edad: 23 años
Tipo de trabajo: En búsqueda de empleo
Bio: ¡Hola! Soy Pedro, un amante de la fotografía y los viajes. Siempre estoy buscando el lugar perfecto para capturar un paisaje único. Me encanta experimentar con diferentes estilos de fotografía y editar mis fotos para darles un toque especial. Aparte de la fotografía, me apasiona la tecnología y siempre estoy probando nuevos gadgets. En mis ratos

Figura 5.14: Vista Consultar compañeros de piso (Fuente: Elaboración propia)

Al seleccionar a un compañero de piso del listado, el sistema mostrará los detalles del usuario como en la Figura 5.15, con su nombre, descripción, teléfono de contacto, email, entre otras. Además se ofrecerá la posibilidad de contactar por chat directamente desde la plataforma.

The screenshot shows a user profile page for 'Daniel'. At the top, there's a navigation bar with links for 'QUIENES SOMOS?', 'PUBLICA TU ANUNCIO', 'CHAT', and a user icon labeled 'Vienna'. Below the navigation is a large circular profile picture of a smiling young man with long hair, wearing a grey hoodie, standing outdoors with water droplets around him. To the right of the profile picture, the name 'Daniel' is displayed in bold. Underneath his name is a bio: '¡Hola, soy Daniel! La música y el arte son mis pasiones. Siempre tengo una playlist nueva que compartir y disfruto mucho descubriendo nuevos géneros. Me encanta asistir a conciertos y exposiciones de arte, siempre en busca de inspiración. Soy una persona sociable y me gusta estar rodeado de buenos amigos con quienes compartir risas y momentos especiales. Si quieras hablar de música, arte o simplemente necesitas una charla amena, ¡aquí estoy!'. Below the bio are several contact details: 'Edad: 25 años', 'Género: Hombre', 'Ocupación: Estudiante en prácticas', 'Email de contacto: daniel@gmail.com', and 'Teléfono de contacto: 654567898'. At the bottom right of the profile area is a yellow button with the text 'CONTACTAR POR CHAT'.

Figura 5.15: Vista Consultar compañero de piso (Fuente: Elaboración propia))

Chat

Desde los detalles de cualquier habitación, se puede iniciar un chat como aparece en la Figura 5.16, lo que crea automáticamente una nueva conversación entre el propietario y el interesado. En la pestaña de chats de cada usuario, aparecerán todas las conversaciones abiertas, con la posibilidad de enviar mensajes en cualquier momento.

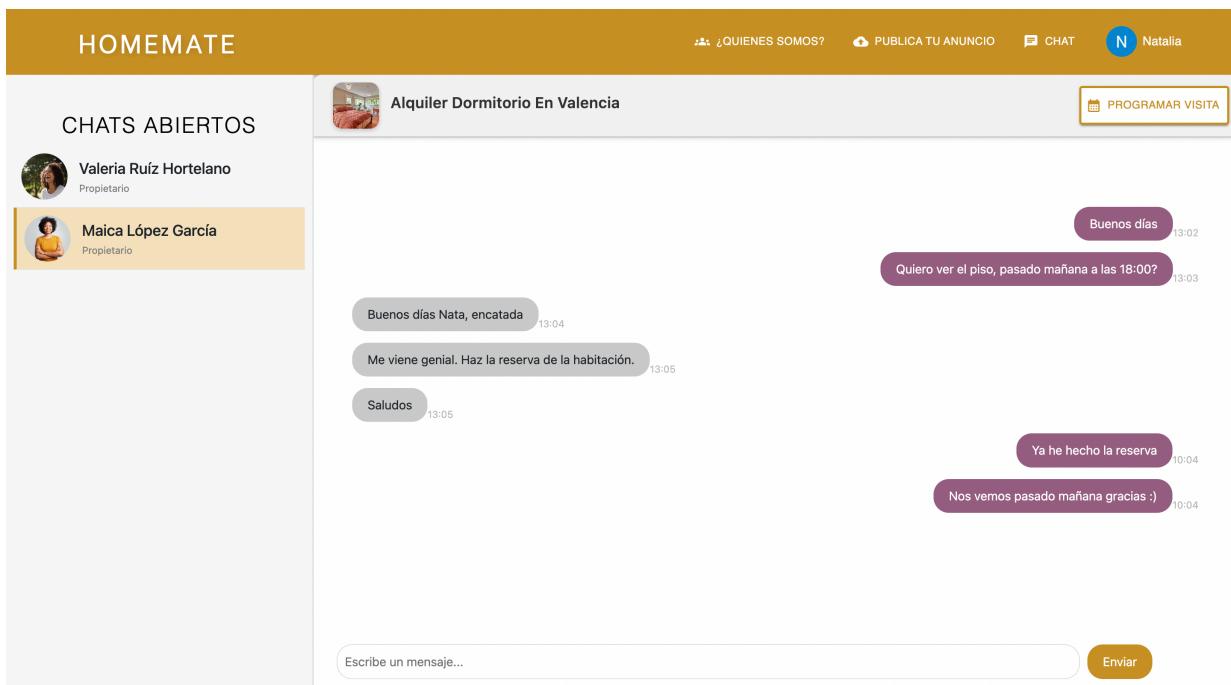


Figura 5.16: Vista Chat (Fuente: Elaboración propia))

Programar visita

Al hacer clic en Programar Visita, se abre una ventana como en la Figura 5.17 en la que el interesado puede elegir el día y la hora para agendarla.

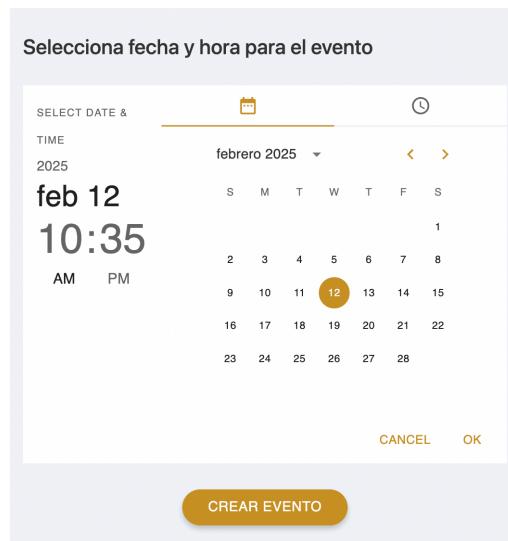


Figura 5.17: Vista Fecha y hora visita (Fuente: Elaboración propia))

Una vez que el interesado ha agendado la visita desde la aplicación de HomeMate, se

enviará un correo automático al propietario para que acepte o rechace la cita, como aparece en la Figura 5.18

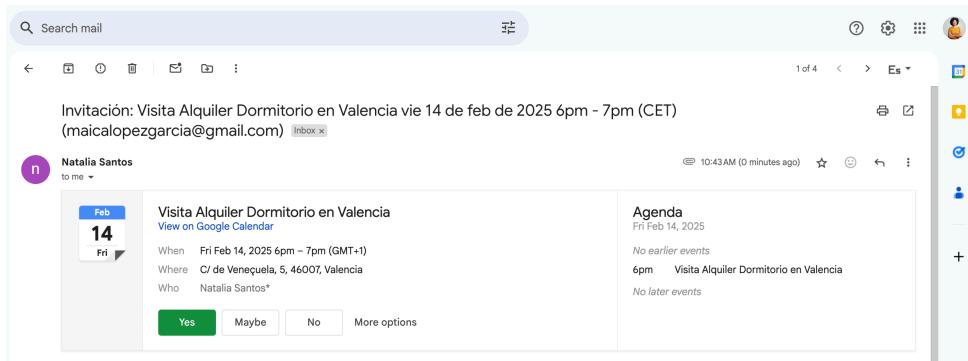


Figura 5.18: Vista Invitación visita Gmail (Fuente: Elaboración propia)

Cualquiera de las respuestas del propietario será notificada por correo al interesado, quien, al acceder a su Google Calendar, podrá ver el evento confirmado o cancelado según la decisión del propietario, como se muestra en la Figura 5.19.

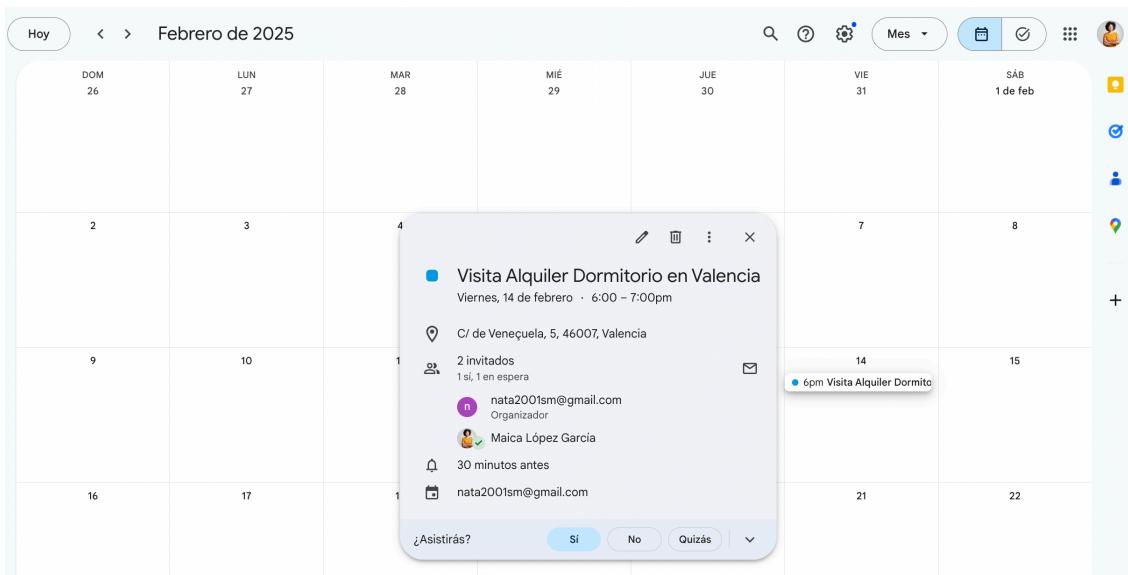


Figura 5.19: Vista Visita en Google Calendar (Fuente: Elaboración propia)

5.2. Backend - ApiRest

En esta sección se detallarán los endpoints generados usando Node.js y Express para gestionar las operaciones de la aplicación. Express, como framework, facilita la creación de rutas y el manejo de solicitudes HTTP, permitiendo dividir la lógica en rutas públicas y privadas. Las rutas públicas son accesibles sin autenticación, mientras que las rutas privadas

requieren un token de autenticación JWT para garantizar que solo los usuarios verificados puedan acceder a ellas.

EndPoints públicos

Los endpoints que se describen a continuación son accesibles sin necesidad de autenticación.

| User | |
|---------------------------------|---|
| /api/users/public/register | Permite a los nuevos usuarios registrarse en la aplicación. |
| /api/users/public/login | El sistema verifica si las credenciales son correctas y, si es así, les da acceso a la aplicación generando un token de autenticación para acciones futuras. |
| /api/users/public/listRoommates | Obtiene una lista de todos los compañeros de piso disponibles. El sistema consulta su base de datos y devuelve los usuarios registrados como posibles compañeros de cuarto. |
| /api/users/public/roommate/:id | Devuelve información detallada sobre un compañero de piso específico, identificado por su id. |
| Room | |
| /api/rooms/public/list | Obtiene el listado completo de todas las habitaciones que han sido registradas en la aplicación. |
| /api/rooms/public/:id | Permite obtener información detallada sobre una habitación específica, identificada por su id. |

EndPoints privados

Los endpoints detallados a continuación son accesibles solo para los usuarios autenticados, ya que requieren que la solicitud incluya un JWT válido en la cabecera para poder ser ejecutados. La validación de este JWT se lleva a cabo mediante un middleware antes de que se ejecute cualquier función asociada al endpoint solicitado. En caso de que el token no sea válido, el acceso será denegado.

| User | |
|--|---|
| /api/users/private/profile/:id | Permite que los usuarios autenticados consulten su perfil. Se muestra los datos del usuario al que corresponde el id proporcionado. |
| /api/users/private/update/:id | Permite actualizar la información de perfil del usuario al que corresponde el id proporcionado |
| Room | |
| /api/rooms/private/register | Permite a los usuarios registrar una nueva habitación en la plataforma. |
| /api/rooms/private/update/:id | Permite a los usuarios actualizar la información de una habitación existente, identificada por su id. |
| /api/rooms/private/delete/:id | Permite a los usuarios eliminar una habitación específica, identificada por su id. |
| /api/rooms/private/list/:id | Permite a los usuarios obtener todas las habitaciones registradas por ellos mismos. |
| Chat | |
| /api/chats/private/create | Permite iniciar un nuevo chat entre un propietario y un buscador. |
| /api/chats/private/messages/:id | Permite a los usuarios obtener todos los mensajes de un chat específico, identificado por su id. |
| /api/chats/private/send | Permite a los usuarios enviar un mensaje dentro de un chat |
| /api/chats/private/list/:id | Permite a los usuarios obtener todos los chats asociados a su cuenta, identificados por su id. |

5.3. Aspectos destacables de la aplicación

En este apartado se destacan las funcionalidades clave de HomeMate que mejoran la experiencia del usuario. Entre ellas, la autenticación con Google, la sincronización con Google Calendar y la integración de mapas interactivos con OpenStreetMap. Además, se presentan las medidas de seguridad implementadas, como el uso de reCAPTCHA.

Autenticación con Google

Para integrar el inicio de sesión con Google en la aplicación, fue necesario crear una aplicación con Firebase Console y obtener las credenciales correspondientes para habilitar su funcionamiento.

```
import { initializeApp } from "firebase/app";
```

```
import {getAuth} from "firebase/auth";
import {getFirestore} from "firebase/firestore";

const firebaseConfig = {
    apiKey: "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX",
    authDomain: "homemate-4a772.firebaseio.com",
    projectId: "homemate-4a772",
    storageBucket: "homemate-4a772.firebaseio.storage.app",
    messagingSenderId: "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX",
    appId: "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
};

const app = initializeApp(firebaseConfig);

export const auth=getAuth();
export const db=getFirestore(app);
export default app;
```

Listing 5.1: Credenciales de Google

Una vez que las credenciales están almacenadas en la aplicación, se ejecuta el método handleLogin, que gestiona el inicio de sesión de un usuario mediante Google utilizando Firebase Authentication.

En primer lugar, se crea una instancia del proveedor de Google (GoogleAuthProvider) y luego se abre una ventana emergente para que el usuario ingrese sus credenciales a través de signInWithPopup. Si la autenticación es exitosa, obtiene la información del usuario y guarda los datos del usuario en Firestore. Finalmente, actualiza el estado de autenticación y redirige al usuario a la página de inicio de la aplicación. En caso de error, muestra un mensaje de advertencia.

```
import { GoogleAuthProvider, signInWithPopup } from "firebase/auth";
import { auth, db } from "../firebaseConfig/firebase";

const handleLogin = async () => {
    const provider = new GoogleAuthProvider();

    try {
        const result = await signInWithPopup(auth, provider);
        const user = result.user;

        if (user) {
            const { firstName, lastName } = getNameParts(user.displayName);
        }
    }
}
```

```
const userData = {
    id: user.uid,
    email: user.email,
    name: firstName,
    surnames: lastName,
    phone: user.phoneNumber,
    image: user.photoURL,
};

await saveUserToFirestore(userData);
saveUserToLocalStorage(userData, user.stsTokenManager.accessToken);

setAuth(userData);
navigate("/home");
}

} catch (error) {
toast.error("Error al iniciar sesión con Google. Inténtalo de nuevo.");
console.error("Google Login Error:", error);
}
};
```

Listing 5.2: Inicio de sesión con Google

Autenticación Nativa con Tokens Personalizados

La autenticación nativa se refiere al proceso de validación de usuarios sin necesidad de depender de servicios de terceros como Google, lo que implica el uso de credenciales propias (correo electrónico y contraseña). En este contexto, la aplicación permite que los usuarios inicien sesión utilizando sus datos de acceso personales. Para mejorar la seguridad y la experiencia del usuario, se implementa el uso de tokens personalizados que facilitan la autenticación en Firebase.

En el lado del cliente, el proceso comienza cuando el usuario ingresa su correo electrónico y contraseña en el formulario de inicio de sesión.

```
const credentials = { email, password };

const response = await fetch('http://localhost:5000/users/public/login', {
    method: "POST",
    body: JSON.stringify(credentials),
    headers: { "Content-Type": "application/json" },
});
```

Listing 5.3: Login lado del cliente

El servidor recibe estas credenciales y las valida. Si los datos son correctos, el servidor genera un token personalizado para autenticar al usuario en Firebase. Este token personalizado es devuelto al cliente como parte de la respuesta del servidor.

```
const login = async (req, res) => {
  const { email, password } = req.body;

  try {
    const isPasswordValid = await bcrypt.compare(password,
      userData.password);
    if (!isPasswordValid) {
      return res.status(401).json({ status: "error",
        message: "Credenciales inválidas." });
    }
    const customToken = await admin.auth().createCustomToken
      (userData.id);

    return res.status(200).json({
      status: "success",
      message: "Se ha iniciado sesión con éxito.",
      user: {
        id: userData.id,
        name: userData.name,
        surnames: userData.surnames || '',
        email: userData.email,
        image: userData.image || 'default.png',
      },
      token: customToken,
    });
  } catch (error) {
    return res.status(500).json({ status: "error", message:
      "Error al iniciar sesión", error: error.message });
  }
};
```

Listing 5.4: Login lado del cliente

En la siguiente etapa, el cliente obtiene el ID token del usuario, el cual es utilizado para identificar de manera segura al usuario durante las interacciones con el backend y en futuras solicitudes. Este token es almacenado en localStorage para que la aplicación pueda mantener al usuario autenticado incluso después de recargar la página.

```
const data = await request.json();

if(data.status === "success"){
    const userCredential = await signInWithCustomToken(auth,
        data.token);
    const idToken = await userCredential.user.getIdToken();

    localStorage.setItem("token", idToken);
    localStorage.setItem("user", JSON.stringify(data.user));
    setAuth(data.user);
```

Listing 5.5: Login lado del cliente

Sincronización con Google Calendar

Para integrar Google Calendar, se requiere configurar un proyecto en Google Cloud Console¹. Se deberán generar credenciales OAuth 2.0, que incluyen un Client ID y una API Key. Estos elementos permiten autenticar a los usuarios y conectar la aplicación con su calendario.

Posteriormente, será necesario utilizar el SDK de Firebase para permitir que los usuarios se autentiquen con Google. Esta funcionalidad ha sido explicada anteriormente en el apartado de 'Autenticación con Google'.

Se inicializa la API de Google Calendar y se configura con los parámetros obtenidos en el proyecto de Google Cloud Console. Además, se establece el estado isSignedIn según si el usuario ya está autenticado con Google.

```
const initClient = () => {
    gapi.client.init({
        apiKey: "XXXXXXXXXXXXXXXXXXXXXXXXXXXX",
        clientId: "XXXXXXXXXXXXXXXXXXXXXXXXXXXX",
        scope: "https://www.googleapis.com/auth/calendar.events"
            ,
        discoveryDocs: ["https://www.googleapis.com/discovery/v1
            /apis/calendar/v3/rest"],
    }).then(() => {
        const authInstance = gapi.auth2.getAuthInstance();

        const isAuthenticated = authInstance.isSignedIn.get();
        setIsSignedIn(isAuthenticated);
    });
};
```

¹<https://console.cloud.google.com/>

Listing 5.6: Inicialización de la API de Google Calendar

A continuación, se construye un objeto que contiene la información del evento, como el título, la ubicación, las fechas de inicio y fin, los asistentes, etc.

```
const eventStart = selectedDateTime.tz('Europe/Madrid', true).  
    toISOString();  
const eventEnd = selectedDateTime.add(1, 'hour').tz('Europe/  
    Madrid', true).toISOString();  
  
const event = {  
    summary: "Visita alojamiento: " + roomDetails.features.name,  
    location: `${roomDetails.ubicationDetails.street}, ${  
        roomDetails.ubicationDetails.number}, ${roomDetails.  
        ubicationDetails.zipCode}, ${roomDetails.ubicationDetails  
        .city}`,  
    start: {  
        dateTime: eventStart,  
        timeZone: "Europe/Madrid",  
    },  
    end: {  
        dateTime: eventEnd,  
        timeZone: "Europe/Madrid",  
    },  
    attendees: [  
        { email: roomDetails.owner.email },  
        { email: auth.email },  
    ],  
};
```

Listing 5.7: Creación de un Evento en Google Calendar

Finalmente, se utiliza la API de Google Calendar para insertar el evento en el calendario principal del usuario.

```
const request = gapi.client.calendar.events.insert({  
    calendarId: "primary",  
    resource: event,  
});  
  
request.execute((event) => {
```

```
    if (event.htmlLink) { alert("Evento creado"); }
    else { console.log("Error al crear el evento:", event); }
};
```

Listing 5.8: Envío de petición a API de Google Calendar

Mapas Interactivos con OpenStreetMap y Leaflet

En aplicaciones web que requieren mostrar mapas interactivos o servicios basados en ubicación, es esencial obtener las coordenadas geográficas (latitud y longitud) a partir de una dirección o lugar. Para ello, se puede utilizar la API de OpenStreetMap, un servicio de geocodificación que permite convertir una dirección en coordenadas geográficas.

```
export const getCoordinates = async (address) => {
  const url = 'https://nominatim.openstreetmap.org/search?q=${encodeURIComponent(address)}&format=json';

  try {
    const response = await axios.get(url);
    if (response.data.length > 0) {
      const location = response.data[0];
      const latitude = location.lat;
      const longitude = location.lon;
      return { latitude, longitude };
    } else {
      console.log("Dirección no encontrada.");
      return null;
    }
  } catch (error) {
    console.error("Error al obtener coordenadas:", error);
    return null;
  }
};
```

Listing 5.9: Inicio de sesión con Google

Para mostrar la ubicación geográfica de un lugar en un mapa interactivo, se puede diseñar un componente en React que integre la librería Leaflet. Esta librería facilita la creación de mapas interactivos, y en este caso, se emplea para consumir la función encargada de obtener las coordenadas geográficas de una dirección. Una vez obtenidas las coordenadas, se visualiza el mapa con un marcador personalizado en la ubicación correspondiente.

```
import { MapContainer, TileLayer, Marker, Popup } from 'react-leaflet';
```

```
import L from 'leaflet';
import 'leaflet/dist/leaflet.css';

const Map = ({ latitude, longitude }) => {
  const icon = new L.Icon({ iconUrl: require('../assets/logo.png'),
    iconSize: [32, 32], iconAnchor: [16, 32], popupAnchor: [0, -32],
  });

  return (
    <MapContainer center={[latitude, longitude]} zoom={13} >
      <TileLayer
        url="https://[s].tile.openstreetmap.org/{z}/{x}/{y}.png"
        attribution='&copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors'
      />
      <Marker position={[latitude, longitude]} icon={icon}>
        <Popup>
          Ubicación de la habitación
        </Popup>
      </Marker>
    </MapContainer>
  );
};

export default Map;
```

Listing 5.10: Inicio de sesión con Google

Seguridad del sistema con reCAPTCHA

En la aplicación, se ha implementado Google reCAPTCHA como una medida de seguridad para garantizar que el usuario que se está registrando no sea un bot.

Para integrar reCAPTCHA en la aplicación, se ha utilizado la librería react-google-recaptcha, que facilita la implementación del servicio en aplicaciones React. La clave del sitio (sitekey) es necesaria para interactuar con la API de Google y debe ser obtenida a través del registro del sitio en la consola de Google reCAPTCHA.

```
import React, { useState } from 'react';
import ReCAPTCHA from 'react-google-recaptcha';

const Recaptcha = ({ onVerify }) => {
  const [verified, setVerified] = useState(false);
  const handleRecaptcha = (token) => {
```

```
if (token) {
    setVerified(true);
    onVerify(token);
} else {
    setVerified(false);
}
};

return (
<div>
    <ReCAPTCHA
        sitekey="XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
        onChange={handleRecaptcha}
    />
    {verified ? <p>Verificación completada</p> : <p>Por favor,
        verifica que no eres un robot.</p>}
</div>
);
};

export default Recaptcha;
```

Listing 5.11: Inicio de sesión con Google

El token generado por reCAPTCHA se obtiene cuando el usuario completa la verificación. Este token es utilizado en el servidor para verificar que el desafío ha sido completado correctamente.

```
const [recaptchaToken, setRecaptchaToken] = useState(null);

const handleRecaptcha = (token) => {
    setRecaptchaToken(token);
};
```

Listing 5.12: Inicio de sesión con Google

Para mejorar la experiencia de usuario, se utilizan mensajes de validación para informar al usuario sobre el estado del formulario. Si el reCAPTCHA no es completado, se muestra un mensaje pidiendo al usuario que lo complete.

```
if (!recaptchaToken) {
    toast.error("Por favor, verifica que no eres un robot.");
    return;
}
```

Listing 5.13: Inicio de sesión con Google

Una vez que el formulario es validado (con todos los campos llenos y el reCAPTCHA completado), los datos del usuario junto con el token de reCAPTCHA se envían al servidor para completar el proceso de registro.

6. Conclusiones

El desarrollo de la aplicación HomeMate ha permitido abordar la problemática de la búsqueda de habitaciones y compañeros de piso en España, proporcionando una solución digital eficiente y accesible para todos los usuarios. A lo largo del proyecto, se ha logrado cumplir con los objetivos planteados, integrando diversas tecnologías que han permitido la creación de una plataforma completamente funcional y escalable.

Uno de los principales logros de este trabajo ha sido la implementación de un sistema de autenticación dual, que permite a los usuarios la opción de registrarse y acceder a la plataforma mediante autenticación nativa o a través de Google. Además, se ha conseguido la integración de herramientas clave como Firebase, que ha facilitado la gestión de la base de datos y la comunicación en tiempo real a través del chat.

En cuanto a la usabilidad, la interfaz desarrollada con React se ha diseñado con el objetivo de ser intuitiva y fácil de navegar, lo que podría contribuir a una mejor experiencia para los usuarios. Además, la integración de OpenStreetMap y Leaflet ha permitido la visualización detallada de la ubicación de las habitaciones.

En cuanto a la gestión del proyecto, el uso de Trello ha sido de gran ayuda para organizar las tareas y hacer un seguimiento del trabajo realizado. Además, el Diagrama de Gantt ha permitido distribuir las actividades de manera clara y asegurar que se avanzara según lo previsto. Como resultado, se han cumplido los plazos establecidos, e incluso el proyecto se finalizó con un pequeño margen de días extra respecto a la planificación inicial.

A nivel de impacto, esta aplicación responde a una necesidad real del mercado inmobiliario, especialmente en un contexto donde compartir piso se ha convertido en una alternativa cada vez más frecuente debido al incremento del precio del alquiler. HomeMate no solo facilita el contacto entre arrendadores e inquilinos, sino que también proporciona un entorno confiable y organizado para gestionar las búsquedas de vivienda compartida.

En conclusión, HomeMate es un paso adelante en la digitalización del alquiler compartido. Su alto rendimiento y rapidez mejoran la experiencia de los usuarios, convirtiéndolo en una herramienta clave en el sector.

7. Trabajo futuro

A pesar de que la plataforma HomeMate cumple con los objetivos planteados y ofrece una solución funcional para la búsqueda de habitaciones y compañeros de piso, se identifican posibles mejoras que podrían implementarse en futuros desarrollos para adaptar el sistema a nuevas necesidades tecnológicas. Entre las posibles mejoras, se destacan las siguientes:

- **Transformación y optimización de imágenes:** Una mejora clave a considerar es la implementación de un sistema automatizado que transforme las imágenes subidas por los usuarios al formato .webp antes de almacenarlas en la plataforma. Este formato, desarrollado por Google, ofrece una compresión más eficiente en comparación con los formatos tradicionales como .jpg o .png, logrando reducir significativamente el tamaño de los archivos sin comprometer la calidad visual. La adopción de esta medida permitiría optimizar el espacio de almacenamiento en la nube, reducir los tiempos de carga de las imágenes y mejorar la experiencia del usuario, especialmente en dispositivos con conexiones lentas.
- **Implementación de algoritmos de recomendación:** Para mejorar la personalización de la experiencia del usuario, sería beneficioso implementar algoritmos de recomendación que sugieran habitaciones y compañeros de piso según los intereses y comportamientos previos de cada usuario. Estos algoritmos podrían basarse en técnicas de aprendizaje automático y análisis de patrones de búsqueda, teniendo en cuenta variables como la ubicación, el precio y las características del alojamiento. Con ello, se facilitaría a los usuarios encontrar opciones que se ajusten mejor a sus necesidades y se incrementaría la eficiencia de las búsquedas dentro de la plataforma, lo que podría llegar a un aumento del nivel de satisfacción de los usuarios.
- **Reseñas verificadas de alojamientos:** Esta posible mejora sería la incorporación de un sistema de reseñas de los alojamientos en los que los usuarios se han hospedado. Este sistema debería contar con un mecanismo de verificación que asegure que únicamente puedan valorar aquellos usuarios que realmente hayan estado en un alojamiento concreto. Esto fomentaría la transparencia y la confianza entre los usuarios, ya que las valoraciones serían más fiables, ayudando así a futuros inquilinos a tomar decisiones mejor informadas.

Estas propuestas tienen el potencial de hacer de HomeMate una plataforma más competitiva, aumentando la satisfacción de los usuarios y consolidándola como una alternativa destacada en el mercado del alojamiento compartido.

Referencias bibliográficas

- Appscrip. (2024). Los mejores marcos frontend para aplicaciones web en 2024. *Actualización de la industria*. Descargado de <https://appscrip.com/es/blog/best-frontend-frameworks-for-web-apps-in-2024/> (Consultado el 11 de diciembre de 2024)
- CEI. (2024). ¿Qué es figma y para qué sirve? *Figma*. Descargado de <https://cei.es/que-es-figma/> (Consultado el 2 enero de 2025)
- Firebase. (2024). Firebase authentication. *Authentication*. Descargado de <https://firebase.google.com/docs/auth?hl=es-419/> (Consultado el 11 de diciembre de 2024)
- FormadoresIT. (2023). ¿Qué es postman? ¿Cuáles son sus principales ventajas? *FormadoresIT*. Descargado de <https://formadoresit.es/que-es-postman-cuales-son-sus-principales-ventajas/#:~:text=Las%20principales%20caracter%C3%ADsticas%20y%20funcionalidades,Gesti%C3%B3n%20de%20entornos>. (Consultado el 20 de enero de 2025)
- Fotocasa. (2024). El porcentaje de españoles que comparte piso porque no puede pagar un alquiler completo aumenta. *Fotocasa Blog*. Descargado de <https://blogprofesional.fotocasa.es/el-porcentaje-de-espanoles-que-comparte-piso-porque-no-puede-pagar-un-alquiler-completo-aumenta> (Consultado el 19 de noviembre de 2024)
- Hostinger. (2024a). Qué es node.js: Casos de uso comunes y cómo instalarlo. Descargado de <https://www.hostinger.es/tutoriales/que-es-node-js> (Consultado el 11 de diciembre de 2024)
- Hostinger. (2024b). Qué es react: definición, características y funcionamiento. Descargado de <https://www.hostinger.es/tutoriales/que-es-react> (Consultado el 11 de diciembre de 2024)
- Jobbloo. (2024). 2024 stack overflow developer survey. *Empleo, Noticias, Tecnología*. Descargado de <https://jobbloo.com/2024-stack-overflow-developer-survey/#:~:3Atext=Seg%C3%A1n%20la%20encuesta%20de%202024,y%20casos%20de%20uso%20espec%C3%ADficos>. (Consultado el 11 de diciembre de 2024)
- Leaflet. (2023). Una biblioteca javascript de código abierto para mapas interactivos compatibles con dispositivos móviles. Descargado de <https://leafletjs.com/> (Consultado el 2 de enero de 2025)
- Lumion. (2023). Qué es openstreetmaps. *OpenStreetMap*. Descargado de <https://lumion.es/que-es-openstreetmap/#:~:3Atext=Openstreetmaps%20es%20un\protect\penalty-\@M%20proyecto%20internacional,gratuitos%20de%20todo%20el%20mundo>. (Consultado el 2 enero de 2025)
- Mora, S. L. (2020). Firebase: qué es, para qué sirve, funcionalidades y ventajas. *Blog digital55*. Descargado de <https://digital55.com/blog/que-es-firebase-funcionalidades-ventajas-conclusiones/> (Consultado el 11 de diciembre de 2024)
- Mundo, E. (2024). El mercado de la vivienda continúa su desaceleración, pero la demanda sigue alta. *El Mundo*. Descargado de <https://www.elmundo.es/economia/vivienda/2024/06/05/666052f0e85ece8e158b45df.html> (Consultado el 19 de noviembre de 2024)

- Rondón, M. (2024). Review de cloudinary: Tu solución definitiva para la gestión de medios. *Review de Cloudinary*. Descargado de <https://negociosoptimizados.com/review-de-cloudinary/> (Consultado el 2 de enero de 2025)
- Terreros, D. (2024). Qué es trello, para qué sirve y cómo funciona. *Trello*. Descargado de <https://blog.hubspot.es/marketing/que-es-trello> (Consultado el 2 enero de 2025)
- Xataka. (2024). Qué es github y qué es lo que le ofrece a los desarrolladores. *Github*. Descargado de <https://www.xataka.com/basics/que-github-que-que-le-ofrece-a-desarrolladores> (Consultado el 11 de diciembre de 2024)

A. Anexo I

Este Anexo presenta una descripción detallada de las historias de usuario que han orientado el desarrollo de la aplicación. Para cada historia, se incluyen una descripción, las validaciones necesarias, la prioridad asignada, la estimación de tiempo y las dependencias con otras historias.

| Publicar habitación | | | |
|--|------------------|-------------------|--------------------|
| Descripción: Como usuario quiero publicar una nueva habitación a partir de una interfaz. | | | |
| Validación: | | | |
| <ul style="list-style-type: none"> ■ Los datos obligatorios han sido completados ■ Los datos introducidos tienen un formato válido ■ Usuario ha sido autenticado previamente en la aplicación | | | |
| Id | Prioridad | Estimación | Dependencia |
| 3 | 1 | 22h | Ninguna |

Tabla A.1: [Historia de usuario - Publicar habitación](#)

| Actualizar habitación | | | |
|---|------------------|-------------------|--------------------|
| Descripción: Como usuario quiero actualizar mi habitación a partir de una interfaz. | | | |
| Validación: | | | |
| <ul style="list-style-type: none"> ■ Se permite la edición solo de las habitaciones creadas por el usuario ■ Los datos obligatorios han sido completados ■ Los datos introducidos tienen un formato válido ■ Usuario ha sido autenticado previamente en la aplicación | | | |
| Id | Prioridad | Estimación | Dependencia |
| 4 | 2 | 20h | 3 |

Tabla A.2: [Historia de usuario - Publicar habitación](#)

| Eliminar habitación | | | |
|---|-----------|------------|-------------|
| Descripción: Como usuario quiero eliminar mi habitación a partir de una interfaz. | | | |
| Validación: | | | |
| <ul style="list-style-type: none"> ■ Se permite la eliminación solo de las habitaciones creadas por el usuario ■ Usuario debe confirmar la acción de eliminar ■ Usuario ha sido autenticado previamente en la aplicación | | | |
| Id | Prioridad | Estimación | Dependencia |
| 5 | 2 | 10h | 3 |

Tabla A.3: [Historia de usuario - Eliminar habitación](#)

| Consultar habitación | | | |
|---|-----------|------------|-------------|
| Descripción: Como usuario quiero acceder a los alojamientos que se han publicado en la aplicación. | | | |
| Validación: No existen validaciones | | | |
| Id | Prioridad | Estimación | Dependencia |
| 6 | 1 | 18h | 3, 4, 5 |

Tabla A.4: [Historia de usuario - Consultar habitación](#)

| Filtrar en búsqueda de habitación | | | |
|---|-----------|------------|-------------|
| Descripción: Como usuario quiero aplicar filtros en la búsqueda de habitaciones. | | | |
| Validación: No existen validaciones | | | |
| Id | Prioridad | Estimación | Dependencia |
| 7 | 2 | 20h | 3, 4, 5 |

Tabla A.5: [Historia de usuario - Filtrar en búsqueda de habitación](#)

| Acceso perfil usuario | | | |
|--|------------------|-------------------|--------------------|
| Descripción: Como usuario quiero acceder a través de una interfaz a la información de mi perfil en la aplicación. | | | |
| Validación: | | | |
| <ul style="list-style-type: none"> ■ Usuario ha sido autenticado previamente en la aplicación | | | |
| Id | Prioridad | Estimación | Dependencia |
| 8 | 2 | 15h | 1 |

Tabla A.6: Historia de usuario - Acceso perfil usuario

| Actualizar perfil usuario | | | |
|--|------------------|-------------------|--------------------|
| Descripción: Como usuario quiero actualizar a través de una interfaz los datos de mi perfil. | | | |
| Validación: | | | |
| <ul style="list-style-type: none"> ■ Los datos obligatorios han sido completados ■ Los datos introducidos tienen un formato válido ■ Usuario ha sido autenticado previamente en la aplicación | | | |
| Id | Prioridad | Estimación | Dependencia |
| 9 | 2 | 12h | 1 |

Tabla A.7: Historia de usuario - Actualizar perfil usuario

| Publicar mi perfil de compañero de piso | | | |
|---|------------------|-------------------|--------------------|
| Descripción: Como usuario deseo que mi perfil sea visible en el listado de compañeros de piso para que otros usuarios puedan contactarme directamente. | | | |
| Validación: | | | |
| <ul style="list-style-type: none"> ■ Usuario ha sido autenticado previamente en la aplicación ■ Usuario ha marcado la opción de "Publicar mi perfil de compañero de piso" | | | |
| Id | Prioridad | Estimación | Dependencia |
| 10 | 2 | 10h | 9 |

Tabla A.8: Historia de usuario - Publicar mi perfil de compañero de piso

| Eliminar mi perfil de compañero de piso | | | |
|--|------------------|-------------------|--------------------|
| Descripción: Como usuario deseo eliminar mi perfil en el listado de compañeros de piso. | | | |
| Validación: | | | |
| <ul style="list-style-type: none"> ■ Usuario ha sido autenticado previamente en la aplicación ■ Usuario ha desmarcado la opción de "Publicar mi perfil de compañero de piso" | | | |
| Id | Prioridad | Estimación | Dependencia |
| 11 | 2 | 7h | 10 |

Tabla A.9: Historia de usuario - Eliminar mi perfil de compañero de piso

| Consultar compañero de piso | | | |
|---|------------------|-------------------|--------------------|
| Descripción: Como usuario quiero acceder a los compañeros de piso que se han publicado en la aplicación. | | | |
| Validación: No existen validaciones | | | |
| Id | Prioridad | Estimación | Dependencia |
| 12 | 2 | 18h | 10 |

Tabla A.10: Historia de usuario - Consultar compañero de piso

| Contacto por chat | | | |
|--|------------------|-------------------|--------------------|
| Descripción: Como usuario quiero poder comunicarme en tiempo real a través de un chat tanto con los anunciantes de habitaciones, como con aquellos que se han ofrecido como compañeros de piso. | | | |
| Validación: | | | |
| <ul style="list-style-type: none"> ■ Usuario ha sido autenticado previamente en la aplicación | | | |
| Id | Prioridad | Estimación | Dependencia |
| 13 | 3 | 40h | 3 |

Tabla A.11: Historia de usuario - Contacto por chat

| Cerrar sesión | | | |
|---|-----------|------------|-------------|
| Descripción: Como usuario quiero poder cerrar mi sesión y ser redirigido automáticamente a la pantalla de inicio de sesión | | | |
| Validación: | | | |
| <ul style="list-style-type: none">■ Usuario ha sido autenticado previamente en la aplicación | | | |
| Id | Prioridad | Estimación | Dependencia |
| 14 | 3 | 4h | 2 |

Tabla A.12: [Historia de usuario - Cerrar sesión](#)

| Gestión visitas | | | |
|---|-----------|------------|-------------|
| Descripción: Como usuario quiero programar una visita en un día y hora determinado. | | | |
| Validación: | | | |
| <ul style="list-style-type: none">■ Usuario ha sido autenticado previamente en la aplicación■ Usuario ha iniciado el chat con el propietario de la habitación. | | | |
| Id | Prioridad | Estimación | Dependencia |
| 15 | 4 | 10h | 2, 13 |

Tabla A.13: [Historia de usuario - Gestión visitas](#)

B. Anexo II

A continuación, se puede observar un ejemplo del documento de una habitación en formato JSON. Este formato es comúnmente utilizado para la transmisión de datos entre el cliente y el servidor. El siguiente objeto JSON devuelve los detalles completos de una habitación específica, incluyendo su ubicación, características, información sobre el propietario que la registró y un conjunto de imágenes de la habitación.

```
{  
    "id": "5VKKQqjNbl2RryVPegjc",  
    "ubicationDetails": {  
        "street": "C/ de Venezuela",  
        "zipCode": "46007",  
        "number": "5",  
        "city": "Valencia",  
        "province": "Valencia"  
    },  
    "features": {  
        "name": "Alquiler Dormitorio en Valencia",  
        "price": "270",  
        "m2": "80",  
        "availableFrom": "2025-02-13T23:00:00.000Z",  
        "availableUntil": "2026-02-09T23:00:00.000Z",  
        "numberOfRooms": "2",  
        "numberOfBathrooms": "2",  
        "waterConsumption": "20",  
        "electricityConsumption": "20",  
        "description": "Este estudio en Ruzafa es una pasada. Es una zona llena de vida, con muchos bares, restaurantes y tiendas de diseño. El estudio es muy bonito, tiene una cocina bien equipada, un baño grande y una cama super cómoda. Ideal si te gusta estar en un barrio trendy pero con un ambiente relajado. Además, la zona está súper bien conectada para moverte por Valencia.",  
        "airConditioning": true,  
        "heating": true,  
        "tv": true,  
        "pool": true,  
        "gym": true,  
        "terrace": true  
    },  
    "owner": {  
        "id": "pGzT5nMZk4gjbmtyrscEQLwq36o1",  
        "email": "maicalopezgarcia@gmail.com",  
        "name": "Maica",  
        "surnames": "López García",  
        "phone": null,  
        "image": "https://lh3.googleusercontent.com/a/ACg8ocI5oUVGtHz1g9xb-fPyq47tqEjvjHW52c3VZI962PdsrYyrCw=s96-  
    }  
}
```

```
        c"
    },
    "photos": [
        "https://res.cloudinary.com/de4yrydad/image/upload/v1739179304
         /kzdogquwyfqvnyr3uhtx.webp",
        "https://res.cloudinary.com/de4yrydad/image/upload/v1739180076
         /wpapd3fsvttrmn5szcrz.webp",
        "https://res.cloudinary.com/de4yrydad/image/upload/v1739180090
         /cjpwys3wbdsbogp0y8r.webp",
        "https://res.cloudinary.com/de4yrydad/image/upload/v1739180101
         /fdxsnnzw1m1yyw1nyuk1k.jpg",
        "https://res.cloudinary.com/de4yrydad/image/upload/v1739180107
         /ckaoqxd2ahn7fkzn0cae.jpg"
    ]
}
```