

Hoomie: Tu espacio, tu elección

Elaborado Por:

Fabian Steven Alfonso Pastrana

Deivyt Santiago García Oviedo

Valery Ovalle Cañon

Tutor:

Jhon Jairo Porras Vega

Universidad EAN
Facultad de Ingeniería
Bogotá D.C

Contenido

| | |
|--|----|
| Tabla de Figuras..... | 4 |
| Lista de Tablas | 6 |
| Resumen..... | 7 |
| 1. Introducción | 8 |
| 2. Objetivos | 9 |
| Objetivo General..... | 9 |
| Objetivos Específicos..... | 9 |
| 3. Definición del Problema | 10 |
| 4. Justificación..... | 11 |
| 5. Análisis de Requerimientos..... | 11 |
| 6. Marco Teórico | 17 |
| 7. Análisis de requerimientos | 20 |
| 7.1. Intención del producto | 20 |
| 7.2. Verificación de Parámetros de Diseño..... | 21 |
| 7.2.1. Parámetros Funcionales..... | 21 |
| 7.2.2. Parámetros de Seguridad | 21 |
| 7.3. Especificaciones del Producto | 21 |
| 7.3.1 Características de Diseño | 21 |
| 7.3.2. Métricas de Rendimiento | 22 |
| 7.3.3. Indicadores de éxito | 22 |
| 8. Análisis de restricciones..... | 22 |

| | |
|--|----|
| | 3 |
| 8.2. Análisis de riesgos y mitigación | 23 |
| 8.3. Análisis DOFA | 25 |
| 8.4. Benchmarking con Soluciones Similares | 26 |
| 8.5. Normativas Específicas Aplicables | 27 |
| 9. Metodología para la selección y desarrollo..... | 28 |
| 10. Análisis Costos | 29 |
| 10.1. Costos fijos..... | 30 |
| 10.2. Costos Directos | 31 |
| 10.3. Costos Generales..... | 34 |
| 11. Arquitectura del desarrollo | 35 |
| 11.1. Diagrama de Contexto C4..... | 35 |
| 11.2. Diagrama de contenedores C4..... | 36 |
| 11.3. Diagrama de componentes C4..... | 37 |
| 12. Prototipado..... | 38 |
| 12.1. Consideraciones UX / UI..... | 38 |
| 12.2. Mockups..... | 39 |
| 13. Modelo de datos..... | 51 |
| 13.1. Diccionario de datos | 51 |
| 14. Implementación | 56 |
| 15. Test..... | 60 |
| 16. Conclusiones..... | 61 |
| Bibliografía | 62 |

Tabla de Figuras

| | | |
|------------------|---|----|
| Figura 1 | <i>Matriz de Riesgos</i> | 24 |
| Figura 2 | <i>Análisis DOFA</i> | 25 |
| Figura 3 | <i>Análisis comparativo</i> | 26 |
| Figura 4 | <i>Diagrama de contexto</i> | 35 |
| Figura 5 | <i>Diagrama de contenedores</i> | 36 |
| Figura 6 | <i>Diagrama de componentes</i> | 37 |
| Figura 7 | <i>Mockup de Iniciar Sesión</i> | 39 |
| Figura 8 | <i>Mockups de Recuperación de Contraseña</i> | 40 |
| Figura 9 | <i>Mockup Iniciar Sesión</i> | 41 |
| Figura 10 | <i>Mockup Cabecera y Página Principal</i> | 42 |
| Figura 11 | <i>Mockup (Formulario Match) y Sección (Acciones Principales)</i> | 42 |
| Figura 12 | <i>Mockup Secciones Ciudades Disponibles, Carrusel Opiniones y Footer</i> | 43 |
| Figura 13 | <i>Mockup Nosotros. Sección Misión</i> | 44 |
| Figura 14 | <i>Mockup Nosotros. Sección Valores</i> | 44 |
| Figura 15 | <i>Mockup Formulario Match</i> | 45 |
| Figura 16 | <i>Mockup Editar Perfil</i> | 46 |
| Figura 17 | <i>Mockup Vista de Publicaciones</i> | 47 |
| Figura 18 | <i>Mockup Publicar - Paso 1 Tipo Inmueble</i> | 48 |
| Figura 19 | <i>Mockup Publicar - Paso 2 Ubicación Inmueble</i> | 48 |
| Figura 20 | <i>Mockup Publicar - Paso 3 Costos Inmueble</i> | 49 |
| Figura 21 | <i>Mockup Publicar - Paso 4 Características Físicas</i> | 49 |
| Figura 22 | <i>Mockup Publicar - Paso 5 Preferencias y normas</i> | 50 |

| | |
|---|----|
| | 5 |
| Figura 23 <i>Mockup Publicar - Paso 6 Fotos</i> | 50 |
| Figura 24 <i>Modelo de la base de datos de Hoomie</i> | 51 |
| Figura 25 <i>Implementación Railway MySQL</i> | 59 |
| Figura 26 <i>Implementación Render - Despliegue en la Nube</i> | 59 |

Lista de Tablas

| | | |
|-----------------|---|----|
| Tabla 1 | <i>Historia de Usuario - Registro de Usuario</i> | 12 |
| Tabla 2 | <i>Historia de Usuario - Recuperar Contraseña</i> | 12 |
| Tabla 3 | <i>Historia de Usuario - Perfil de Usuario</i> | 13 |
| Tabla 4 | <i>Historia de Usuario - Publicación de Anuncios</i> | 14 |
| Tabla 5 | <i>Historia de Usuario - Sistema de Coincidencia (Matching)</i> | 14 |
| Tabla 6 | <i>Historia de Usuario - Sistema de calificación</i> | 15 |
| Tabla 7 | <i>Historia de Usuario - Sistema de geolocalización</i> | 16 |
| Tabla 8 | <i>Historia de Usuario - Comunicación entre usuarios</i> | 16 |
| Tabla 9 | <i>Tabla de Riesgos</i> | 23 |
| Tabla 10 | <i>Herramientas utilizadas</i> | 29 |
| Tabla 11 | <i>Costos fijos mensual</i> | 30 |
| Tabla 12 | <i>Costos directos(variables) Mensuales</i> | 31 |
| Tabla 13 | <i>Costos desarrollador Frontend</i> | 32 |
| Tabla 14 | <i>Costos desarrollador Backend</i> | 33 |
| Tabla 15 | <i>Costos desarrollador API</i> | 34 |
| Tabla 16 | <i>Costos Generales - gasto único</i> | 34 |
| Tabla 17 | <i>Tabla Usuarios</i> | 52 |
| Tabla 18 | <i>Tabla Preferencias</i> | 53 |
| Tabla 19 | <i>Tabla Inmueble</i> | 54 |
| Tabla 20 | <i>Tabla de Publicación</i> | 55 |
| Tabla 21 | <i>Tabla de relaciones</i> | 56 |
| Tabla 22 | <i>Implementación Frontend</i> | 56 |
| Tabla 23 | <i>Implementación Backend</i> | 58 |

Resumen

Para la realización de este proyecto, se llevará a cabo el desarrollo de una plataforma digital que facilitará la búsqueda de arriendos y compañeros de habitación. Esta aplicación web permitirá a los usuarios publicar ofertas de arriendo y conectarse con posibles roomies, priorizando la afinidad y la confianza para lograr una convivencia armoniosa. La herramienta contará con un sistema de coincidencia basado en preferencias y necesidades, garantizando la seguridad mediante la verificación de perfiles y un sistema de calificación. La plataforma será accesible desde cualquier dispositivo móvil, optimizando así el proceso de búsqueda y mejorando la experiencia de los usuarios en la búsqueda de alojamiento.

Palabras clave: Plataforma Digital, Arriendos, Compañeros de Habitación, Coincidencia (Matchmaking), Seguridad, Verificación de Perfiles, Sistema de Calificación, Interfaz Intuitiva, Experiencia de Usuario, Optimización de Búsqueda.

Abstract

For the realization of this project, a digital platform will be developed that will facilitate the search for tenants and roommates. This web application will allow users to publish rental offers and connect with possible roomies, prioritizing affinity and trust to achieve a harmonious coexistence. The tool will have a matching system based on preferences and needs, ensuring security through profile verification and a rating system. The platform will be accessible from any mobile device, thus optimizing the search process and improving the experience of users in search for accommodation.

Keywords: Digital Platform, Rentals, Roommates, Matching (Matchmaking), Security, Profile Verification, Rating System, Intuitive Interface, User Experience, Search Optimization.

1. Introducción

En la actualidad, encontrar un lugar en arriendo o un compañero de habitación adecuado es un desafío en muchas ciudades. El proceso suele ser largo y complicado, ya que los arrendatarios tienen dificultades para acceder a información clara sobre disponibilidad, costos y condiciones de alquiler. Al mismo tiempo, los arrendadores enfrentan el problema de seleccionar inquilinos confiables.

Por otro lado, compartir vivienda se ha convertido en una práctica común, especialmente entre jóvenes y estudiantes, pero elegir un roomie compatible sigue siendo un reto. La falta de herramientas que permitan conocer mejor a posibles compañeros de cuarto aumenta las probabilidades de una convivencia difícil o poco armoniosa.

Este proyecto busca desarrollar una aplicación que facilite la búsqueda de arriendos y compañeros de habitación, optimizando el proceso a través de una plataforma intuitiva que conecte a personas con intereses y estilos de vida similares.

La dificultad para acceder a información clara y confiable sobre arriendos y la falta de compatibilidad entre compañeros de habitación han sido identificadas como las principales barreras en este proceso.

Para abordar estos desafíos, se ha diseñado una solución integral que permite a los usuarios publicar sus necesidades de arriendo y búsqueda de compañeros de habitación, así como filtrar y evaluar a posibles inquilinos o roomies mediante un sistema de perfiles detallados. Esto no solo agiliza el proceso de búsqueda, sino que también proporciona un entorno seguro y confiable para ambas partes.

En este documento, el lector encontrará una descripción detallada de la aplicación, incluyendo sus características principales, el proceso de desarrollo, y cómo se planea implementar la solución en el mercado. Además, se presentarán estudios de caso que ilustran la efectividad de la plataforma en la mejora de la experiencia de búsqueda de arriendos y compañeros de habitación.

2. Objetivos

Objetivo General

Crear una plataforma digital que permita la búsqueda de vivienda y la conexión entre roommates.

Objetivos Específicos

- Investigar las leyes locales y nacionales que regulan el arrendamiento de propiedades, asegurando el cumplimiento legal de la plataforma.
- Crear wireframes que representen la disposición de los elementos y la navegación de la interfaz de usuario, asegurando una experiencia amigable.
- Implementar un prototipo interactivo en herramientas como Figma o Adobe XD, permitiendo a los usuarios potenciales realizar pruebas y proporcionar retroalimentación.
- Desarrollar algoritmos de emparejamiento, utilizando técnicas de aprendizaje automático para optimizar afinidad entre usuarios.

3. Definición del Problema

Según un estudio realizado en 2023 por el DANE (Departamento Administrativo Nacional de Estadística), el 40,3 % de los hogares colombianos viven en arriendo, evidenciando un incremento significativo en comparación con 2021, cuando la cifra era del 38,6 %. Al mismo tiempo, la proporción de colombianos que poseen vivienda propia ha disminuido en el país.

De igual forma, otro estudio realizado por el DANE y publicado en enero de 2025 reveló que el Índice de Precios al Consumidor (IPC) cerró en 2024 con un incremento del 5,2 %. Este aumento ha llevado a los colombianos a implementar nuevas estrategias para reducir costos.

En este contexto, un estudio de la ACIS (Asociación Colombiana de Informática, Sistemas y Tecnologías Afines) señala que cada vez más jóvenes en Colombia optan por compartir su lugar de residencia, una tendencia conocida como "Roommate" o "Roomie", que ofrece ventajas como la reducción del costo del arriendo. (Resnick & Zeckhauser, 2002a)

Además, según una encuesta realizada por Livinn, promotora de residencias universitarias en Colombia, el 90 % de los jóvenes valoraría poder elegir a su compañero de vivienda. Sin embargo, solo el 48 % tuvo la oportunidad de hacerlo, y apenas el 53 % afirma tener afinidad con su compañero actual.

Con base en lo anterior, el proyecto “Hoomie” está orientado con la pregunta ¿Es posible mejorar la fiabilidad y afinidad al buscar un roommate utilizando una aplicación web?

4. Justificación

A partir de la problemática identificada, se evidencia la necesidad de una herramienta que facilite la búsqueda de vivienda compartida, permitiendo a los usuarios encontrar tanto compañeros de cuarto como lugares que se ajusten a sus preferencias, afinidades y propósito de residencia. La falta de opciones personalizadas y de un proceso estructurado para elegir roomates adecuados puede generar experiencias de convivencia poco satisfactorias, afectando la calidad de vida de quienes buscan este tipo de solución habitacional.

Por esta razón, se propone la creación de Hoomie, una aplicación web diseñada para optimizar la búsqueda de compañeros de vivienda y residencias, proporcionando filtros personalizados que permitan a los usuarios conectar con personas afines en términos de estilo de vida, intereses y necesidades económicas. De este modo, Hoomie no solo contribuirá a mejorar la experiencia de convivencia, sino que también facilitará la reducción de costos de arrendamiento y promoverá entornos más armoniosos y compatibles entre quienes deciden compartir hogar.

5. Análisis de Requerimientos

Para garantizar el correcto desarrollo y funcionamiento de la plataforma, es fundamental establecer de manera clara los requerimientos del sistema. Estos definen las funcionalidades esenciales, las necesidades del usuario y las condiciones técnicas que debe cumplir Hoomie para facilitar la búsqueda de roomies y espacios de arrendamiento.

Esta sección recoge tanto los requerimientos funcionales como los no funcionales, sirviendo como base para el diseño, implementación y validación del prototipo.

Tabla 1

Historia de Usuario - Registro de Usuario

| HISTORIA DE USUARIO | | |
|---|--|---------------------|
| Registro de usuario | | |
| Identificador | Nombre | |
| HU001 | Usuario nuevo (arrendador/arrendatario) | |
| Puntos de historia estimados | | |
| Prioridad | Alta | |
| Responsable | Equipo de desarrollo | |
| Descripción | | |
| Como usuario nuevo (arrendador/arrendatario) quiero poder registrarme con un nombre de usuario, correo electrónico, número de celular y contraseña para garantizar la autenticidad de mi cuenta y evitar perfiles duplicados. | | |
| Número de criterio | Criterio de aceptación | Atributo de calidad |
| 1 | Validar formato de correo electrónico y unicidad del nombre de usuario | Funcionalidad |
| 2 | Almacenar datos sensibles | Seguridad |
| 3 | Exigir contraseña con mínimo 8 caracteres (letras, números y símbolos) | Seguridad |
| 4 | Enviar correo de confirmación para activar la cuenta | Usabilidad |

Tabla 2

Historia de Usuario - Recuperar Contraseña


| HISTORIA DE USUARIO | | |
|------------------------------|----------------------|--|
| Recuperar contraseña | | |
| Identificador | Nombre | |
| HU002 | Usuario | |
| Puntos de historia estimados | | |
| Prioridad | Alta | |
| Responsable | Equipo de desarrollo | |
| Descripción | | |

Como usuario nuevo (arrendador/arrendatario) me gustaría tener la opción de poder recuperar mi contraseña en caso de que se me olvide para poder acceder nuevamente a mi cuenta sin perder mi información.

| Número de criterio | Criterio de aceptación | Atributo de calidad |
|--------------------|---|---------------------|
| 1 | El usuario puede ingresar su correo electrónico asociado a la cuenta. El sistema envía un correo con un enlace para restablecer la contraseña. | Funcionalidad |
| 2 | Al hacer clic en el enlace, el usuario puede ingresar una nueva contraseña. La nueva contraseña se valida (mínimo de caracteres, combinación de letras/números, etc.). | Seguridad |
| 3 | El sistema confirma que la contraseña se actualizó exitosamente. El usuario puede iniciar sesión con la nueva contraseña. | Seguridad |

Tabla 3

Historia de Usuario - Perfil de Usuario

| HISTORIA DE USUARIO | | |
|--|---|---------------------|
| Perfil de usuario | | |
| Identificador | Nombre | |
| HU003 | Usuario registrado (arrendador/arrendatario) | |
| Puntos de historia estimados | | |
| Prioridad | Media | |
| Responsable | Equipo de desarrollo | |
| Descripción | | |
| Como un usuario ya registrado quiero poder editar mi perfil, ajustar la visibilidad de mis datos y configurar preferencias de vivienda con el fin de personalizar mi experiencia y controlar qué información comparten con otros usuarios. | | |
| Número de criterio | Criterio de aceptación | Atributo de calidad |
| 1 | Permitir edición de datos personales y preferencias | Funcionalidad |
| 2 | Implementar opciones de visibilidad (público/privado)  | Usabilidad |

| | | |
|---|---|------------|
| 3 | Validar cambios con contraseña actual para evitar modificaciones no autorizadas | Seguridad |
| 4 | Mostrar vista previa del perfil antes de guardar cambios | Usabilidad |

Tabla 4

Historia de Usuario - Publicación de Anuncios

| HISTORIA DE USUARIO | | |
|--|--|---------------------|
| Publicación de anuncios | | |
| Identificador | Nombre | |
| HU004 | Usuario arrendador | |
| Puntos de historia estimados | | |
| Prioridad | Alta | |
| Responsable | Equipo de desarrollo | |
| Descripción | | |
| Poder Publicar y explorar anuncios de viviendas o roomates. Para encontrar inquilinos o roomates que cumplan con mis requisitos de manera eficiente. | | |
| Número de criterio | Criterio de aceptación | Atributo de calidad |
| 1 | Permitir subir fotos, descripción y detalles de la propiedad | Funcionalidad |
| 2 | Filtrar anuncios por ubicación, precio y preferencias | Usabilidad |
| 3 | Incluir sistema de etiquetas para categorizar anuncios (ej: "amueblado", "mascotas") | Usabilidad |

Tabla 5

Historia de Usuario - Sistema de Coincidencia (Matching)

| HISTORIA DE USUARIO | | |
|------------------------------------|----------------------|--|
| Sistema de coincidencia (Matching) | | |
| Identificador | Nombre | |
| HU005 | Usuario arrendatario | |
| Puntos de historia estimados | | |
| Prioridad | Critica | |
| Responsable | Equipo de backend | |
| Descripción | | |

Quiero poder recibir sugerencias de roomates o viviendas basadas en mis preferencias. Por qué lo quiere: Para agilizar la búsqueda y encontrar opciones compatibles con mi estilo de vida.

| Número de criterio | Criterio de aceptación | Atributo de calidad |
|--------------------|--|---------------------|
| 1 | Priorizar resultados según hábitos, presupuesto y ubicación | Funcionalidad |
| 2 | Actualizar sugerencias en tiempo real al modificar preferencias | Usabilidad |
| 3 | Incluir opción para excluir resultados no deseados (ej: fumadores) | Usabilidad |
| 4 | Generar informe de compatibilidad detallado para cada sugerencia | Funcionalidad |

Tabla 6

Historia de Usuario - Sistema de calificación

| HISTORIA DE USUARIO | | |
|---|---|---------------------|
| Sistema de calificación– reputación | | |
| Identificador | Nombre | |
| HU006 | Usuario con experiencia (arrendador/arrendatario) | |
| Puntos de historia estimados | | |
| Prioridad | Media | |
| Responsable | Equipo de backend | |
| Descripción | | |
| Calificar a roomates/arrendadores después de 1 mes de convivencia. Para ayudar a futuros usuarios a tomar decisiones informadas basadas en experiencias reales. | | |
| Número de criterio | Criterio de aceptación | Atributo de calidad |
| 1 | Habilitar calificación solo tras 30 días de registro | Funcionalidad |
| 2 | Mostrar puntuación promedio en perfiles | Usabilidad |
| 3 | Permitir comentarios anónimos para proteger la privacidad del evaluador | Seguridad |
| 4 | Generar alertas por calificaciones repetitivas o sospechosas | Seguridad |

Tabla 7

Historia de Usuario - Sistema de geolocalización

| HISTORIA DE USUARIO | | |
|--|---|---------------------|
| Sistema de geolocalización | | |
| Identificador | Nombre | |
| HU007 | Usuario arrendatario | |
| Puntos de historia estimados | | |
| Prioridad | Alta | |
| Responsable | Equipo de backend | |
| Descripción | | |
| Ver la ubicación exacta de propiedades y roomates mediante geolocalización. Para optimizar mi búsqueda basada en proximidad y ahorrar tiempo en desplazamientos. | | |
| Número de criterio | Criterio de aceptación | Atributo de calidad |
| 1 | Integrar API de mapas (Google Maps/OpenStreetMap) | Funcionalidad |
| 2 | Mostrar radio de búsqueda ajustable (1-10 km) | Usabilidad |
| 3 | Permitir guardar ubicaciones favoritas para accesos rápidos | Funcionalidad |
| 4 | Optimizar carga de mapas para reducir consumo de datos en móviles | Rendimiento |

Tabla 8

Historia de Usuario - Comunicación entre usuarios

| HISTORIA DE USUARIO | | |
|--|---------------------------------|--|
| Comunicación entre usuarios | | |
| Identificador | Nombre | |
| HU008 | Usuario arrendador/arrendatario | |
| Puntos de historia estimados | | |
| Prioridad | Media | |
| Responsable | Equipo de frontend | |
| Descripción | | |
| Comunicarme directamente con otros usuarios mediante un chat integrado. Para negociar acuerdos de manera ágil y transparente sin depender de terceras plataformas. | | |

| Número de criterio | Criterio de aceptación | Atributo de calidad |
|--------------------|--|---------------------|
| 1 | Implementar chat en tiempo real con historial | Funcionalidad |
| 2 | Notificar mensajes nuevos por correo o push | Usabilidad |
| 3 | Permitir adjuntar documentos (ej: contrato) en el chat | Funcionalidad |
| 4 | Ofrecer opción de bloqueo de usuarios no deseados | Seguridad |

6. Marco Teórico

El desarrollo de la aplicación Hommie combina conocimientos técnicos y sociales para ofrecer una solución eficiente y segura ante la creciente tendencia de vivienda compartida en la sociedad actual. La adecuada gestión del ciclo de vida del software, junto con el uso de metodologías ágiles y una arquitectura escalable, permitirá la creación de una plataforma adaptable a las necesidades del mercado.

La implementación de algoritmos de coincidencia, sistemas de calificación y medidas de seguridad garantizará una experiencia confiable y transparente para los usuarios, cumpliendo con normativas sobre protección de datos y alquiler de inmuebles, lo que brinda mayor seguridad a los usuarios. Comprender el comportamiento humano en la búsqueda de compañeros de habitación permitirá optimizar la experiencia del usuario, fomentando una convivencia armoniosa y reduciendo los riesgos de incompatibilidad. Con estos fundamentos, Hommie se proyecta como una herramienta innovadora y eficaz para mejorar la manera en que las personas encuentran y comparten su hogar.

Según Randhawa (2022), el ciclo de vida del software se divide en ocho fases: planificación, análisis de requerimientos, diseño, desarrollo, pruebas, implementación y mantenimiento. Diferentes modelos de procesos, como el modelo en cascada, iterativo, espiral,

ágil y Scrum, se utilizan en el desarrollo. El desarrollo dirigido por comportamiento destaca por garantizar pruebas de aceptación que describen las funcionalidades del software desde la perspectiva del usuario.

La documentación es esencial para registrar de manera detallada el desarrollo del software, facilitando la comprensión para terceros y la resolución de errores (Pressman, 2010). Debe incluirse en cada fase del desarrollo, abarcando requerimientos, diseño, especificaciones, codificación y pruebas (Bravo et al., 2022).

Las metodologías ágiles, como Scrum y Kanban, han reemplazado modelos clásicos como el de cascada, permitiendo a los clientes ver el desarrollo funcional desde el inicio (Martínez & López Comino, 2018). Scrum se centra en la formación de equipos y revisiones diarias del trabajo mediante entregables llamados sprints, mientras que Kanban se enfoca en la eliminación de desperdicios y en mostrar lo necesario. La metodología Lean mejora la calidad del software al eliminar elementos innecesarios (Martínez & López Comino, 2018).

La arquitectura de software integra decisiones de diseño y codificación para ofrecer un resultado final que cumpla con las expectativas del cliente. Las decisiones deben considerar problemas, soluciones, argumentos e implicaciones (Pressman, 2010). Algunas arquitecturas relevantes incluyen:

- Arquitectura centrada en los datos
- Arquitectura de flujo de datos
- Arquitecturas orientadas a objetos
- Arquitecturas en capas

El alquiler de inmuebles en Colombia está regulado por la Ley 820 de 2003, que establece el régimen jurídico para los contratos de arrendamiento de vivienda urbana. Esta ley garantiza equidad y transparencia en las relaciones entre arrendadores y arrendatarios (Ley 820 de 2003).

La protección de datos personales es crucial para plataformas digitales que manejan información sensible. La Ley Estatutaria 1581 de 2012 establece el marco legal para la protección de datos personales en Colombia, alineándose con estándares internacionales como el GDPR de la Unión Europea (Ley 1581 de 2012).

El diseño de interfaces y la experiencia del usuario son fundamentales en el desarrollo de aplicaciones web y móviles. Una buena interfaz permite la navegación intuitiva y eficiente, siguiendo principios de simplicidad, claridad, feedback inmediato, accesibilidad y adaptabilidad (Nielsen & Norman, 2018a; 2018b).

Los sistemas de calificación y reseñas son esenciales para fomentar la confianza entre usuarios en plataformas digitales. Estos sistemas pueden basarse en modelos de reputación, donde las calificaciones y comentarios verificables mejoran la calidad del servicio (Resnick & Zeckhauser, 2002; Marek et al., 2004).

La convivencia entre compañeros de cuarto es un fenómeno social que influye en la experiencia universitaria. Las interacciones iniciales positivas son cruciales para mantener relaciones satisfactorias (Marek et al., 2004). La teoría del Valor Predicho del Resultado (POV) explica que las personas se motivan a desarrollar relaciones basadas en impresiones iniciales, lo que puede resultar en una convivencia armoniosa.

Los algoritmos de coincidencia son fundamentales para conectar a los usuarios con servicios o personas basándose en criterios específicos. Existen diferentes tipos de algoritmos, incluyendo:

- Algoritmos basados en reglas
- Algoritmos de coincidencia exacta
- Algoritmos de coincidencia difusa

- Algoritmos basados en afinidad
- Algoritmos de recomendación
- Algoritmos de optimización
- Algoritmos basados en machine learning

En este proyecto, se propone utilizar algoritmos basados en afinidad por proximidad mediante el uso de la librería sklearn con la función de cocine similarity, esta función se encarga de comparar la preferencia e importancia que les da cada usuario a las diferentes situaciones mostrando una mayor compatibilidad entre más cercanos son, que se enfocan en compatibilidades personales y estilos de vida (Cormen et al., 2022).

7. Análisis de requerimientos

El desarrollo exitoso de Hoomie se medirá a través de diversos factores, donde el diseño final se logrará mediante la consecución de los objetivos en el tiempo establecido y el cumplimiento de todos los requerimientos funcionales. A continuación, se presenta el análisis detallado:

7.1. Intención del producto

Propósito Principal: Desarrollar una plataforma digital que facilite la búsqueda y conexión entre personas que buscan compartir vivienda.

El alcance del proyecto contempla la cobertura inicial en las principales ciudades de Colombia, con un enfoque específico en la población joven y estudiantes universitarios, presentando una solución efectiva a la problemática del 40.3% de hogares en arriendo.

Como valor agregado, la plataforma incorporará un sistema de matchmaking basado en afinidades, implementará verificación de perfiles para garantizar la seguridad y optimizará el proceso de búsqueda de vivienda compartida.

7.2. Verificación de Parámetros de Diseño

7.2.1. Parámetros Funcionales

El sistema de registro y autenticación comprenderá la validación de identidad mediante cédula, verificación de correo electrónico y una gestión completa de perfiles de usuario.

Para el sistema de publicaciones se implementará la gestión de anuncios de vivienda, incluyendo la carga y administración de fotografías, además de filtros de búsqueda personalizados según las necesidades del usuario.

El sistema de matchmaking integrará algoritmos de compatibilidad sofisticados, establecerá criterios de emparejamiento precisos y desarrollará métricas de afinidad para optimizar las conexiones entre usuarios.

7.2.2. Parámetros de Seguridad

La seguridad del sistema se fundamentará en la protección robusta de datos personales, verificación rigurosa de identidad, implementación de un sistema de reportes y denuncias, y la encriptación efectiva de toda la información sensible.

7.3. Especificaciones del Producto

7.3.1 Características de Diseño

La arquitectura del sistema se construirá sobre una aplicación web responsive, soportada por una base de datos relacional, implementando una API RESTful y microservicios escalables para garantizar su crecimiento.

El desempeño del sistema garantizará tiempos de respuesta inferiores a 3 segundos, mantendrá capacidad para múltiples usuarios concurrentes, optimizará los recursos del servidor y asegurará una alta disponibilidad 24/7.

La interfaz de usuario se desarrollará bajo principios de diseño intuitivo y amigable, asegurando adaptabilidad a diferentes dispositivos, cumpliendo estándares de accesibilidad web y optimizando la experiencia de usuario para el público objetivo.

7.3.2. Métricas de Rendimiento

- La velocidad de procesamiento se medirá a través del tiempo de carga de páginas, la velocidad de búsqueda y la respuesta del sistema de matchmaking.
- La capacidad del sistema se evaluará mediante el número de usuarios simultáneos soportados, la gestión eficiente del almacenamiento de datos y la capacidad de procesamiento de transacciones.
- La eficiencia del sistema se monitoreará a través del uso de recursos del servidor, la optimización de bases de datos y la implementación de sistemas de gestión de caché.

7.3.3. Indicadores de éxito

Los indicadores funcionales medirán la tasa de emparejamientos exitosos, la precisión del sistema de matchmaking y la efectividad de los filtros de búsqueda implementados.

Los indicadores técnicos evaluarán el tiempo de respuesta del sistema, la disponibilidad de la plataforma y la tasa de errores presentada.

Los indicadores de usuario analizarán la satisfacción general, la tasa de adopción de la plataforma y los niveles de retención de usuarios activos.

8. Análisis de restricciones

El análisis de restricciones para el proyecto Hoomie contempla diversos factores que podrían impactar el desarrollo y la implementación de la plataforma. A continuación, se presenta un análisis detallado de estos elementos:

8.2. Análisis de riesgos y mitigación

La siguiente tabla presenta los principales riesgos identificados durante el desarrollo del proyecto Hoomie, clasificados según su naturaleza (técnicos, operativos y de mercado). Cada riesgo ha sido evaluado en términos de su probabilidad de ocurrencia, el impacto que podría generar en el sistema y su nivel de criticidad. Además, se proponen estrategias de mitigación para reducir su efecto en caso de materializarse, con el objetivo de garantizar la estabilidad, seguridad y escalabilidad de la plataforma.

Niveles de riesgo

- **Bajo:** Riesgo poco probable y/o de bajo impacto
- **Medio:** Requiere atención, pero no es urgente
- **Alto:** Necesita mitigación activa
- **Crítico:** Alto impacto y alta probabilidad. Prioridad Máxima

Tabla 9

Tabla de Riesgos

| ID | Categoría | Riesgo | Probabilidad | Impacto | Nivel de Riesgo | Plan de mitigación |
|----|-----------|---|--------------|---------|-----------------|--|
| R1 | Técnico | Fallos de seguridad en la protección de datos personales | Media | Alta | Alta | Implementar un sistema robusto de encriptación y realizar auditorías de seguridad periódicas. |
| R2 | Técnico | Problemas de escalabilidad ante crecimiento acelerado de usuarios | Media | Alta | Alta | Diseñar una arquitectura escalable basada en microservicios y establecer planes de contingencia. |
| R3 | Técnico | Tiempos de respuesta lentos que afectan la experiencia de usuario | Media | Media | Media | Implementar monitoreo continuo y optimización de recursos. |

| | | | | | | |
|----|-----------|--|-------|-------|---------|--|
| R4 | Operativo | Verificación de identidad compleja o lenta | Alta | Media | Alta | Establecer alianzas con proveedores especializados en verificación de identidad digital. |
| R5 | Operativo | Resistencia de usuarios a compartir información personal | Media | Alta | Alta | Aplicar una política clara de privacidad y explicar detalladamente el uso de los datos. |
| R6 | Mercado | Entrada de competidores con mayor capital | Alta | Alta | Crítica | Desarrollar características diferenciadoras y fortalecer la fidelización de usuarios. |

Figura 1

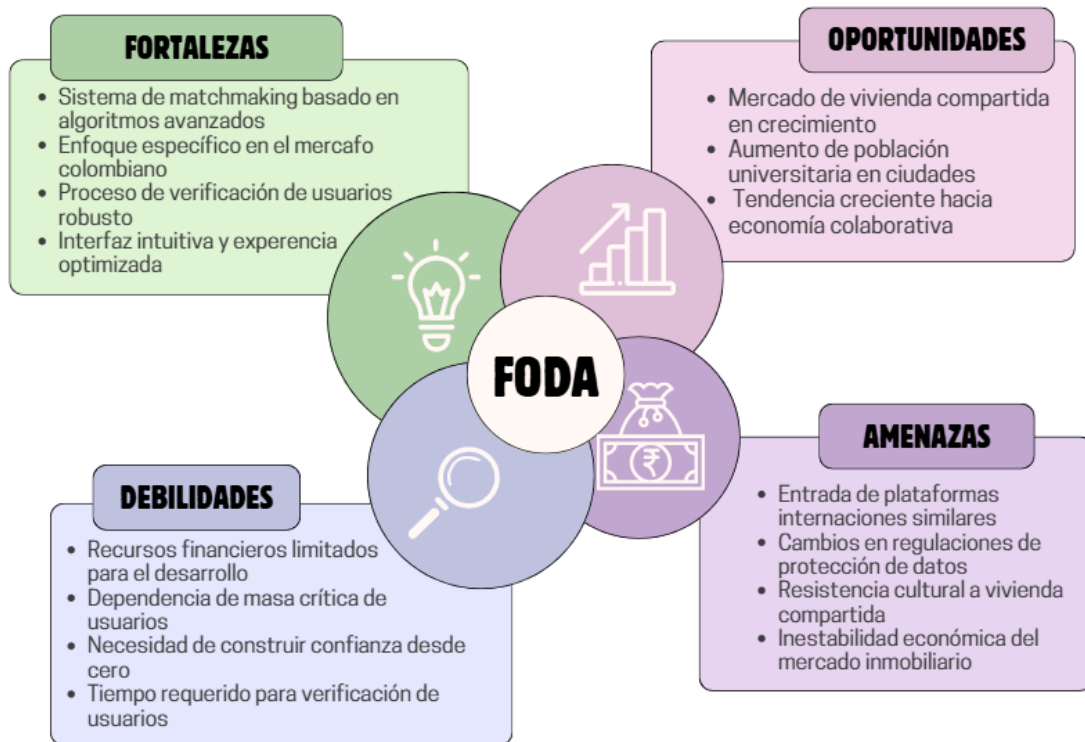
Matriz de Riesgos



8.3. Análisis DOFA

Figura 2

Análisis DOFA



La Figura presenta el análisis DOFA desarrollado para el proyecto Hoomie, una herramienta estratégica que permite visualizar de manera integral los factores críticos que influyen en el éxito potencial de la plataforma. En el cuadrante superior izquierdo, se identifican las Fortalezas fundamentales del proyecto, destacando la implementación de algoritmos avanzados de compatibilidad y su enfoque especializado en el mercado colombiano.

Las Oportunidades, ubicadas en el cuadrante superior derecho, evidencian un entorno favorable caracterizado por el crecimiento del mercado de vivienda compartida y la creciente digitalización del sector inmobiliario. En cuanto a las Debilidades, situadas en el cuadrante inferior izquierdo, se reconocen los desafíos internos como las limitaciones en recursos financieros iniciales y la

necesidad de alcanzar una masa crítica de usuarios para garantizar la efectividad de la plataforma. Finalmente, el cuadrante inferior derecho expone las Amenazas del entorno, entre las que se destacan la posible entrada de competidores internacionales y los cambios regulatorios en materia de protección de datos. Esta matriz DOFA permite una comprensión holística de la posición estratégica del proyecto, facilitando la toma de decisiones y la planificación de acciones estratégicas para maximizar las fortalezas y oportunidades mientras se mitigan las debilidades y amenazas identificadas.

8.4. Benchmarking con Soluciones Similares

Análisis Comparativo: El mercado actual presenta soluciones como Roomgo y Aptuno, que ofrecen servicios básicos de búsqueda de compañeros de vivienda.

Figura 3

Análisis comparativo



Hoomie se diferenciará por:

- **Sistema de Verificación:** Implementación de un proceso más riguroso de verificación de identidad en comparación con la competencia.
- **Algoritmo de Matchmaking:** Desarrollo de un sistema más sofisticado de emparejamiento basado en múltiples variables de compatibilidad.

- **Experiencia de Usuario:** Interfaz más intuitiva y proceso de búsqueda optimizado en comparación con las soluciones existentes.

8.5. Normativas Específicas Aplicables

Esta sección identifica las leyes, estándares y buenas prácticas aplicables a nivel nacional e internacional que deben cumplirse para garantizar el cumplimiento legal, la transparencia del servicio y la confianza de los usuarios.

- **Protección de Datos:** Cumplimiento estricto con la Ley 1581 de 2012 de Protección de Datos Personales en Colombia y el Decreto 1377 de 2013.
- **Comercio Electrónico:** Adherencia a la Ley 527 de 1999 que regula el comercio electrónico en Colombia.
- **Normativa Inmobiliaria:** Consideración de las regulaciones establecidas por la Superintendencia de Industria y Comercio para la publicación de ofertas inmobiliarias.
- **Seguridad Digital:** Cumplimiento de los estándares de seguridad digital establecidos por el MinTIC y las normativas internacionales de protección de datos.

Restricciones Adicionales:

Tecnológicas:

- Limitaciones de ancho de banda en ciertas regiones del país
- Necesidad de compatibilidad con diversos dispositivos y navegadores
- Requerimientos de almacenamiento para el crecimiento de la base de datos

Financieras:

- Presupuesto inicial limitado para desarrollo y marketing
- Costos de implementación de sistemas de seguridad robustos
- Inversión necesaria en infraestructura tecnológica

Operativas:

- Tiempo requerido para la verificación manual de perfiles
- Capacidad de atención al cliente durante el crecimiento inicial
- Recursos humanos necesarios para moderación de contenido

9. Metodología para la selección y desarrollo

Para el desarrollo de la plataforma Hoomie se seleccionó una **metodología ágil**, con base en el marco de trabajo **Scrum**, debido a su enfoque flexible, iterativo y orientado a la entrega continua de valor. Esta elección permitió adaptar el proceso a los cambios de requerimientos durante el desarrollo, garantizar una mejor comunicación entre los integrantes del equipo y enfocarse en las necesidades reales de los usuarios.

El trabajo se organizó en **sprints semanales**, donde se realizaron revisiones constantes tanto internas como con el docente, permitiendo validar los avances y hacer ajustes oportunos. Se mantuvo una **lista de tareas pendientes** gestionada de manera colaborativa, lo que permitió tener un mayor control sobre el progreso del proyecto.

El equipo de trabajo estuvo conformado por tres integrantes, con responsabilidades compartidas de la siguiente manera:

10. **Valery Ovalle, Deivyt Garcia y Fabián Alfonso:** Responsables de la documentación del proyecto.
10. **Deivyt García:** Encargado del desarrollo del backend.
10. **Valery Ovalle:** Encargada del desarrollo del frontend.

Además, los desarrolladores realizaron en el levantamiento de requerimientos, el diseño de mockups, la programación de funcionalidades y la realización de pruebas sobre el sistema.

Para apoyar la organización y ejecución del proyecto, se utilizaron diversas herramientas tecnológicas, entre ellas:

Tabla 10

Herramientas utilizadas

| Tipo de Herramienta | Herramienta Utilizada |
|----------------------|---|
| Gestión de tareas | Notion |
| Diseño de interfaces | Miro |
| Desarrollo | HTML, CSS, JavaScript, PHP, MySQL, Render |
| Control de versiones | Git, GitHub |
| Comunicación | WhatsApp y reuniones presenciales |

Gracias a esta metodología, se logró mantener un flujo de trabajo constante, colaborativo y enfocado en la entrega progresiva de funcionalidades, facilitando la validación temprana de ideas y reduciendo los riesgos asociados al desarrollo.

10. Análisis Costos

Para todo proyecto es fundamental realizar el análisis de costos para tener una buena alternativa económica y de esta manera generar rentabilidad que es lo esperado de todo proyecto, además de generar control y una estimación para realizar el proyecto teniendo en cuenta el presupuesto a utilizar dentro de los cuales encontramos 3 principales los cuales son: costos fijos, costos directos y generales dentro de los que se profundizara generando un presupuesto general desglosado en tablas los cuales permiten tener dicho estimado.

10.1. Costos fijos

Son costos que permanecen fijos dentro de un periodo o rango específico y permanecen de manera constantes. Para los cuales se encontraron dos principales categorías, infraestructura básica y oficina y servicios.

Dentro de la infraestructura básica es utilizado para todo el tema del software lo cual se encontró el hosting base para lo cual para poder publicar el aplicativo web y alojarla dentro de un servidor.

Dominio y SSL: Permite garantizar la seguridad del aplicativo y privacidad transmitidos de un sitio web además de contar con un certificado que autentica la identidad del sitio web, esto con el fin de evitar fraudes cibernéticos.

Licencias de Software: uno de los principales es el pago de un antivirus con el fin de tener seguridad dentro de los equipos a desarrollar.

Otra categoría para adjuntar dentro de estos costos son la de oficina y servicios las cuales serán detalladas a continuación.

Servicios que se tendrá en cuenta que sería luz, agua, internet y aseo dentro de los cuales se hace un consumo promedio para una oficina de 50 m² se realizó con el consumo básico.

Como se muestran en la siguiente imagen para el detalle de los costos fijos:

Tabla 11

Costos fijos mensual

| Categoría | Descripción | Costo(COP) |
|------------------------|----------------------------|----------------------|
| Infraestructura Básica | Hosting base | \$ 7.270.833 |
| | Dominio y SSL (anual) | \$ 4.000.000 |
| | Licencias de software | \$ 1.500.000 |
| Servicios | Luz y agua | \$ 182.061 |
| | Internet empresarial fibra | \$ 146.990 |
| | Seguros | \$ 678.000 |
| Total | | \$ 13.777.884 |

10.2. Costos Directos

Estos costos van directos a toda la mano de obra y sistemas de verificación con el fin de presentar un estimado promedio de la ganancia de un desarrollador en las diferentes categorías las cuales entrarían backend, frontend y desarrollador api y junto a esto se presenta algunos servicios para la aplicación que son los servicios de verificación lo que nos permitiría contar con la búsqueda de antecedentes y contar con una api que genere que cada usuario puede verificar su perfil con el fin de evitar cualquier tipo de falsedad en perfil.

Tabla 12

Costos directos(variables) Mensuales

| Categoría | Descripción | Costo(COP) |
|---------------------------|-----------------------------------|----------------------|
| Personal del equipo | Desarrollador Backend | \$ 8.610.000 |
| | Desarrollador Frontend | \$ 6.440.000 |
| | Desarrollador API | \$ 2.800.000 |
| Servicios de verificación | Api de verificación (por usuario) | \$ 15.000 |
| | Verificaciones antecedentes | \$ 25.000 |
| Total | | \$ 17.890.000 |

Detalle costos personal del equipo

Desarrollador Frontend: En la siguiente tabla se detallan las funcionalidades implementadas por el desarrollador frontend, incluyendo el número estimado de horas dedicadas y el costo asociado.

Con base en el total del tiempo invertido (92 horas) , se estima que el desarrollador frontend tuvo una tarifa promedio de **\$70.000 COP por hora**. Este cálculo permite tener una visión clara de los recursos invertidos en el diseño e implementación de las interfaces de usuario, que incluyen procesos como registro, edición de perfil, publicación de anuncios y comunicación entre usuarios.

Tabla 13

Costos desarrollador Frontend

| Actividad | Estimacion de horas | Costo(COP) |
|--------------------------------------|---------------------|---------------------|
| Registro de usuario | 12 | \$ 840.000 |
| Perfil de usuario | 10 | \$ 700.000 |
| Edición de perfil | 8 | \$ 560.000 |
| Implementar opciones de visibilidad | 6 | \$ 420.000 |
| Mostrar vista previa de perfil | 4 | \$ 280.000 |
| Publicación de anuncios | 15 | \$ 1.050.000 |
| Cargue de fotos y detalles | 10 | \$ 700.000 |
| Incluir sistema de etiquetas | 5 | \$ 350.000 |
| Incluir radio de búsqueda ajustable | 6 | \$ 420.000 |
| Interfaz de comunicación de usuarios | 12 | \$ 840.000 |
| Permitir adjuntar documentos | 4 | \$ 280.000 |
| Total | 92 | \$ 6.440.000 |

Desarrollador Backend: El desarrollador backend es esencial para garantizar el funcionamiento interno de la plataforma, gestionando la lógica del sistema, el manejo seguro de la información, la comunicación con la base de datos y la ejecución de procesos clave como el registro, publicación de anuncios, sistema de coincidencias y calificación entre usuarios. La siguiente tabla presenta las actividades realizadas por el desarrollador backend, con un total de 123 horas trabajadas y un valor total de \$8.610.000 COP, se calcula un valor promedio de \$70.000 COP por hora, lo que refleja la complejidad y el nivel técnico requerido para implementar las funcionalidades del lado del servidor.

Tabla 14

Costos desarrollador Backend

| Actividad | Estimación de horas | Costo(COP) |
|--|---------------------|---------------------|
| Validar formato de correo y unicidad | 8 | \$ 560.000 |
| Almacenar datos sensibles | 10 | \$ 700.000 |
| Exigir contraseña segura | 6 | \$ 420.000 |
| Enviar correo de confirmación | 4 | \$ 280.000 |
| Validar cambios de detalles del perfil | 5 | \$ 350.000 |
| Filtrar anuncios | 8 | \$ 560.000 |
| Limitar publicaciones a 3 por usuario | 4 | \$ 280.000 |
| Sistema de coincidencia (Matching) | 20 | \$ 1.400.000 |
| Priorizar resultados | 10 | \$ 700.000 |
| Actualizar sugerencias en tiempo real | 8 | \$ 560.000 |
| opción para excluir resultados no deseados | 5 | \$ 350.000 |
| Generar informe de compatibilidad | 7 | \$ 490.000 |
| Sistema de calificación | 12 | \$ 840.000 |
| Habilitar calificación tras 30 días | 6 | \$ 420.000 |
| Comentarios anónimos | 4 | \$ 280.000 |
| Generar alertas por calificaciones | 6 | \$ 420.000 |
| Total | 123 | \$ 8.610.000 |

Desarrollador API: El desarrollador encargado de las APIs en Hoomie tuvo un rol clave al integrar servicios externos y facilitar la comunicación en tiempo real entre los diferentes módulos del sistema. Dentro de sus responsabilidades se incluyeron funcionalidades como el sistema de geolocalización, la integración con APIs de mapas y la implementación del chat en tiempo real.

Tabla 15

Costos desarrollador API

| Actividad | Estimacion de horas | Costo(COP) |
|---------------------------------|---------------------|---------------------|
| Sistema de geolocalización | 15 | \$ 1.050.000 |
| Integrar API de mapas | 10 | \$ 700.000 |
| Implementar chat en tiempo real | 15 | \$ 1.050.000 |
| Total | 40 | \$ 2.800.000 |

10.3. Costos Generales

Dentro de los costos generales encontramos aquellos que son de inversión única las cuales entrarían equipos de cómputo para el desarrollo de la aplicación, monitores de apoyo, reguladores de energía las cuales permitirán tener una seguridad energética al momento de usar estos dispositivos electrónicos y evitar cualquier falla y junto a esto todos los equipos de red necesarios para mantener una óptima red y la calidad de este.

Tabla 16

Costos Generales - gasto único

| Equipo | Especificaciones | Costo unitario | Cantidad | Costo(COP) |
|-------------------------|------------------------|-----------------|----------|----------------------|
| Computadores Desarrollo | MacBook Pro M4 pro 14" | \$ 14.119.000 | 3 | \$ 42.357.000 |
| Monitores | Dell 27" 4K | \$ 2.000.000,00 | 3 | \$ 6.000.000 |
| UPS | 1000VA | \$ 519.900,00 | 3 | \$ 1.559.700 |
| Total | | | | \$ 49.916.700 |

Teniendo en cuenta lo anterior se estima un presupuesto general de \$ 81.584.584 para la compra de equipos, servicios y herramientas.

11. Arquitectura del desarrollo

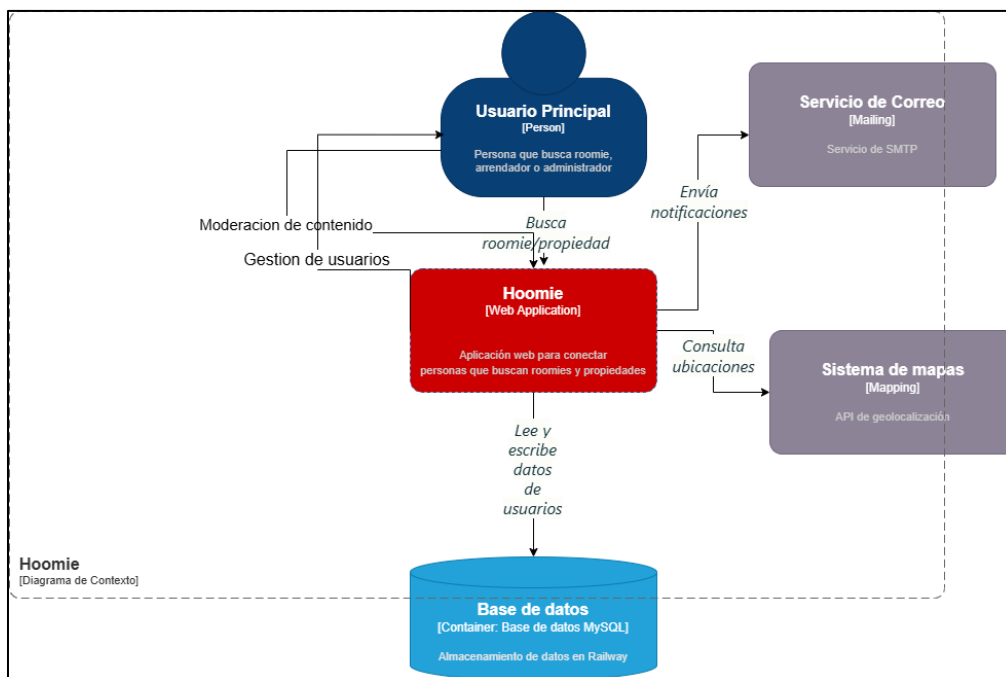
Para representar de manera estructurada los distintos niveles del sistema, se empleará el **modelo C4**, una metodología visual que permite descomponer la arquitectura en cuatro niveles: contexto, contenedores, componentes y código. Esta representación facilita la comprensión del sistema tanto para perfiles técnicos como no técnicos, mostrando cómo interactúan los usuarios, las aplicaciones y los servicios involucrados en el funcionamiento de Hoomie.

11.1. Diagrama de Contexto C4

El diagrama de contexto muestra una visión general del sistema Hoomie y cómo interactúa con los actores externos que lo rodean. Este nivel del modelo C4 permite identificar a los usuarios principales, los sistemas externos con los que se comunica y el propósito general de la plataforma dentro de su ecosistema digital. En este caso, se destaca la relación de Hoomie con los usuarios y los servicios externos como la API de mapas y el servicios de correo electrónico.

Figura 4

Diagrama de contexto

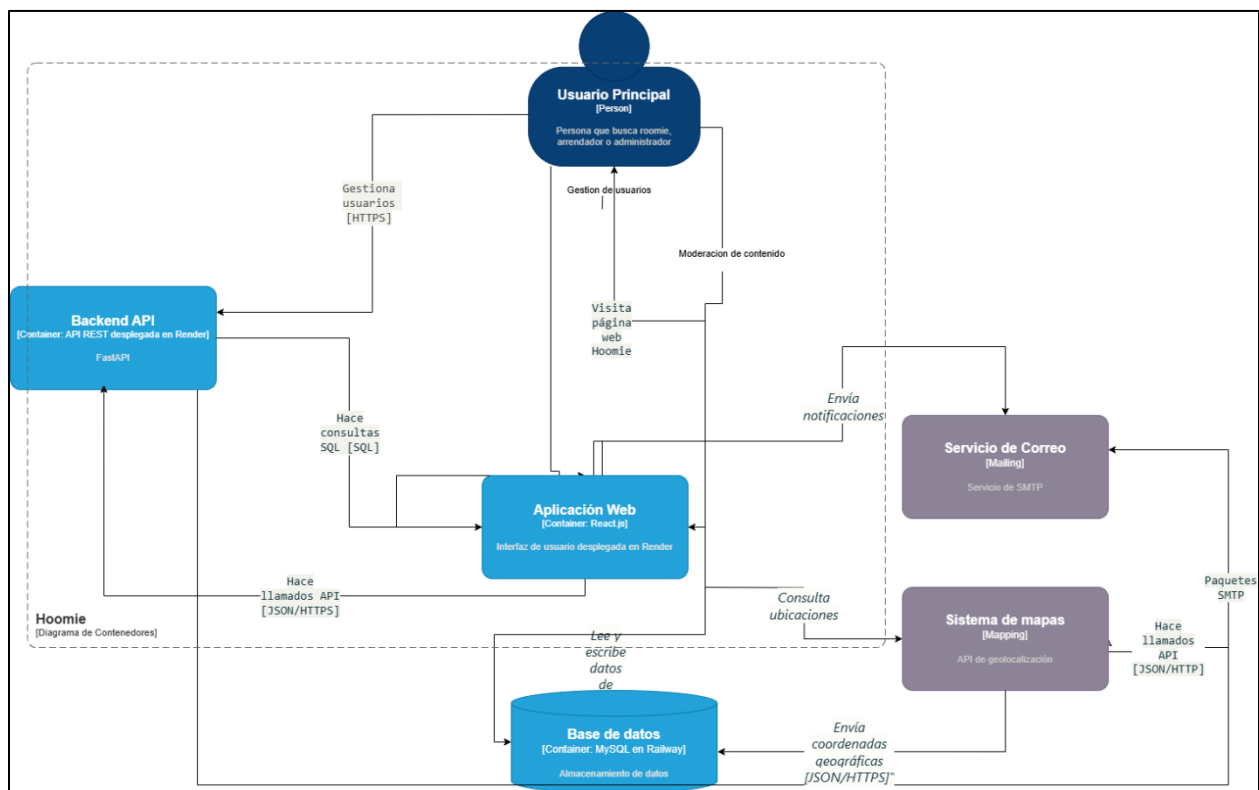


11.2. Diagrama de contenedores C4

El diagrama de contenedores descompone el sistema Hoomie en sus principales bloques funcionales o "contenedores", mostrando cómo se organizan y comunican las diferentes partes del software a nivel de ejecución. En este nivel del modelo C4, se visualizan componentes clave como la aplicación web (frontend), la API backend, la base de datos y servicios externos como la API de geolocalización y el sistema de notificaciones por correo. Este diagrama permite entender qué tecnologías intervienen, cómo se distribuyen las responsabilidades entre los contenedores, y cómo fluye la información entre ellos para brindar una experiencia completa y funcional al usuario final.

Figura 5

Diagrama de contenedores

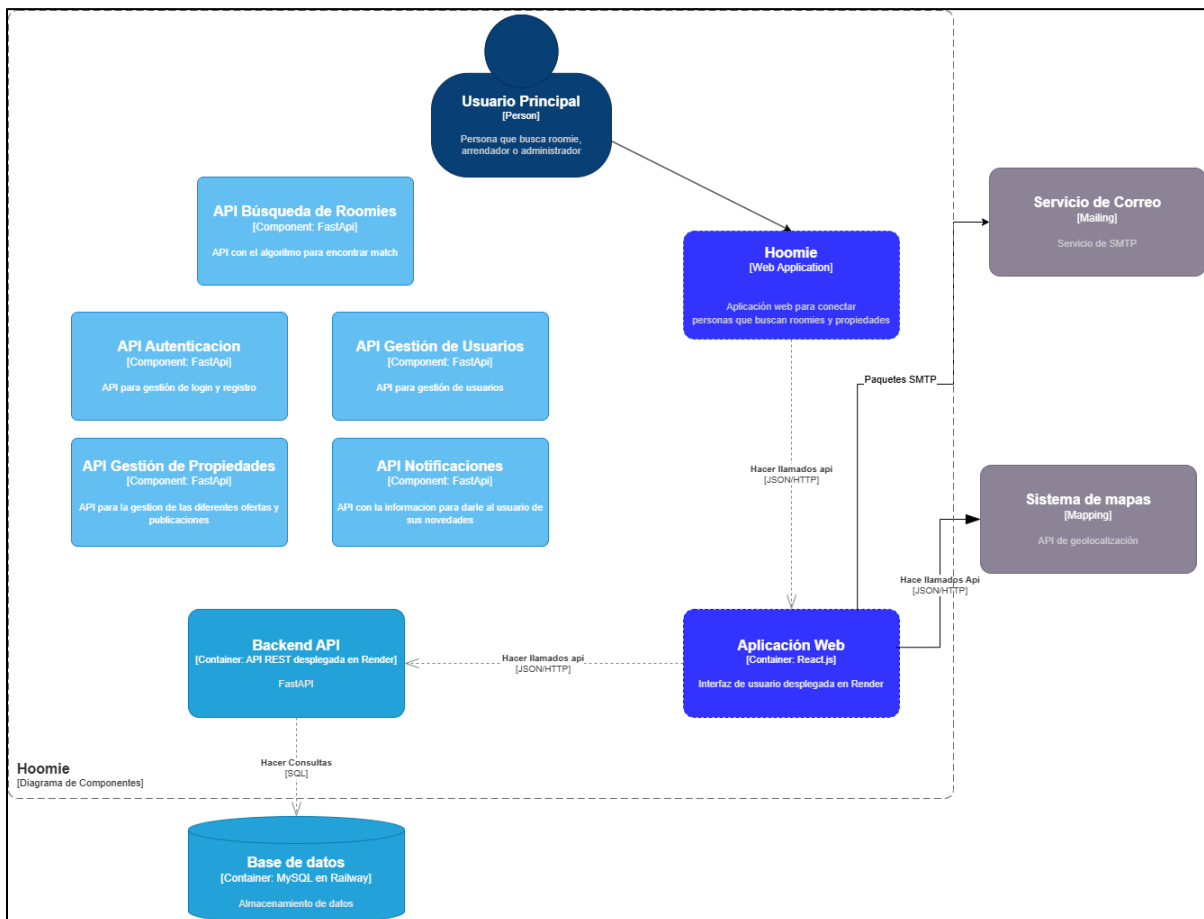


11.3. Diagrama de componentes C4

El diagrama de componentes ofrece una vista detallada de la estructura interna de cada contenedor del sistema, mostrando cómo se dividen en módulos o componentes individuales que trabajan juntos para cumplir con las funcionalidades del proyecto. En el caso de Hoomie, este diagrama detalla los componentes principales del backend, como el módulo de autenticación, el gestor de publicaciones, el sistema de coincidencia (matching), el módulo de geolocalización y el sistema de calificaciones. También se incluyen los componentes del frontend, encargados de gestionar la navegación, las vistas de perfil, los formularios y la interfaz de chat.

Figura 6

Diagrama de componentes



12. Prototipado

12.1. Consideraciones UX / UI

Durante el diseño de la plataforma Hoomie se tomaron en cuenta diversos principios de **experiencia de usuario (UX)** y **diseño de interfaz (UI)** con el fin de crear una herramienta intuitiva, accesible y visualmente atractiva para los usuarios. La prioridad fue facilitar la navegación y ofrecer una experiencia fluida en cada interacción. A continuación, se describirán las principales consideraciones:

- **Diseño minimalista y centrado al usuario:** Se optó por una interfaz limpia que evita la sobrecarga de información, permitiendo que las funciones principales (como buscar roomies, publicar anuncios o editar perfil) sean fáciles de encontrar y usar.
- **Jerarquía visual clara:** Se utilizaron tamaños, colores y espaciado adecuados para destacar botones, formularios y secciones importantes, mejorando la comprensión y usabilidad de la plataforma.
- **Uso de colores neutros y acentos llamativos:** Se seleccionó una paleta de colores amigable, moderna y coherente con la identidad visual del proyecto. Los colores de acento se usaron para destacar acciones importantes.
- **Diseño responsivo:** La plataforma fue pensada para adaptarse a diferentes dispositivos, incluyendo móviles, tablets y computadores, asegurando una buena experiencia de uso sin importar el medio.

12.2. Mockups

A continuación, se presentan los mockups diseñados para la plataforma Hoomie. Estas vistas permiten visualizar la apariencia y funcionalidad de la interfaz antes del desarrollo, facilitando la validación del diseño y la experiencia del usuario.

- **Iniciar sesión (Login):** La vista de inicio de sesión permite a los usuarios acceder a la plataforma ingresando su correo electrónico y contraseña. Incluye un botón de “Ingresar”, un botón de “Regresar” que los lleva a la página principal, y enlaces para “¿Olvidaste tu contraseña?” y “Crear sesión”, en caso de que aún no tengan una cuenta.

Figura 7

Mockup de Iniciar Sesión



The mockup shows a login interface for the Hoomie platform. At the top, there is a logo consisting of two smiling faces (one yellow, one blue) under a simple house outline, with the word "HOOMIE" in bold black letters below it. The main heading is "¡Bienvenido Nuevamente!". Below this, the text "Identificate para ingresar" is followed by two input fields: "Correo Electrónico" and "Contraseña". A link "¿Olvidaste tu Contraseña?" is positioned below the password field. At the bottom, there are two buttons: a blue "Regresar" button and an orange "Ingresar" button. A final link "¿No tienes cuenta? Crear Cuenta" is located at the very bottom of the form area.

- **Contraseña Olvidada:** Permite a los usuarios recuperar el acceso a su cuenta en caso de haber olvidado la contraseña. Al enviarlo, el sistema genera un enlace de recuperación que se envía por correo.

Figura 8

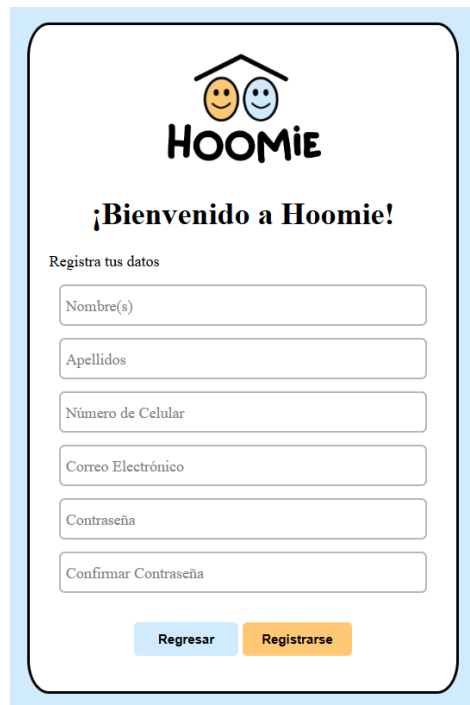
Mockups de Recuperación de Contraseña



- **Crear sesión (Sign up):** Contara con el formulario de registro con la información personal del usuario conociendo su nombre, apellido, celular, correo electrónico, contraseña y la autorización de los datos al momento de dar al botón registro.

Figura 9

Mockup Iniciar Sesión



- **Inicio:** La página de inicio de Hoomie está diseñada para dar la bienvenida a los usuarios y guiarlos de forma clara hacia las principales funcionalidades de la plataforma.
 - **Cabecera:** En la parte superior se encuentra la cabecera, que incluye el logo de Hoomie, un menú con dos opciones: “**Publicaciones**”, que redirige a la vista de anuncios disponibles, y “**Publicar**”, que lleva al formulario para crear un nuevo anuncio. En el extremo derecho se encuentran los botones de “**Crear sesión**” e “**Iniciar sesión**”.
 - **Banner Principal:** Se explica brevemente qué es Hoomie y se invita a los usuarios a conocer más sobre el proyecto mediante un botón que redirige a la sección de “**Nosotros**”.

Figura 10

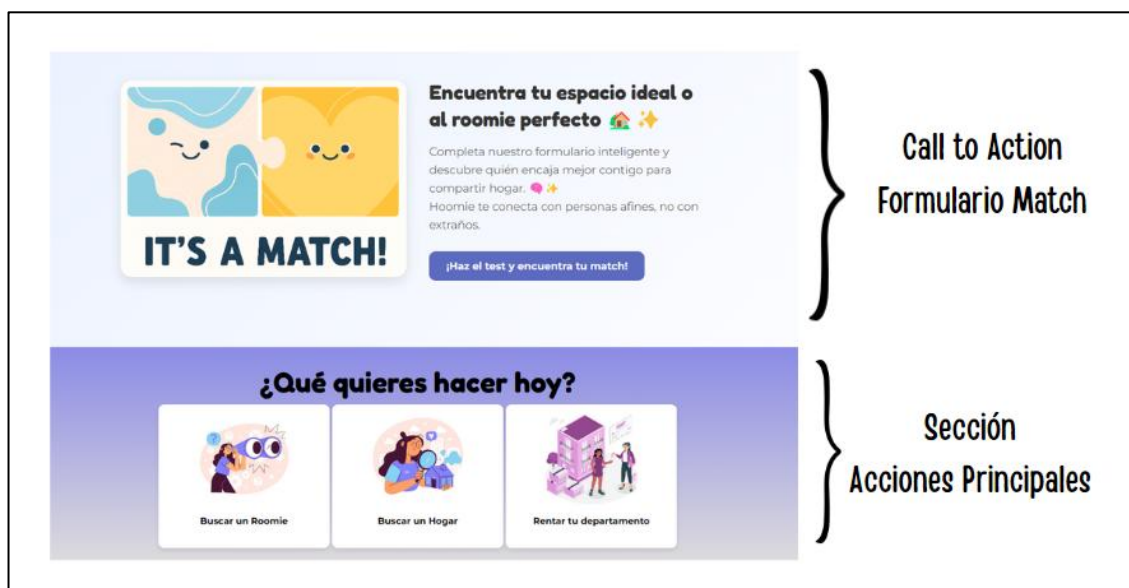
Mockup Cabecera y Página Principal



- **Call to Action (Formulario Match):** Más abajo, se encuentra un **call to action**, enfocado en motivar a los usuarios a diligenciar el formulario de Match, destacando la funcionalidad principal de la plataforma.
- **Sección Acciones Principales:** Luego se destacan las tres acciones principales que los usuarios pueden realizar: **Buscar Roomie**, **Buscar un hogar** y **Rentar un apartamento**, cada una representada de forma visual y accesible.

Figura 11

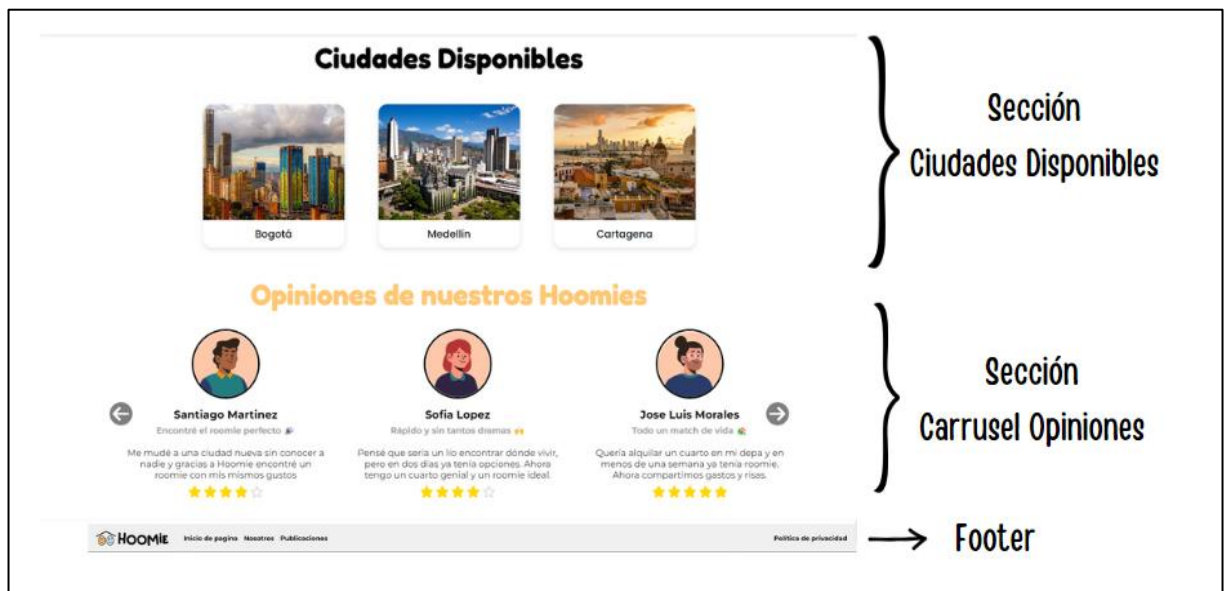
Mockup (Formulario Match) y Sección (Acciones Principales)



- **Sección Ciudades Disponibles:** La siguiente sección corresponde a “Ciudades disponibles”, donde se muestran tres ciudades destacadas. Al hacer clic sobre cualquiera de ellas, el usuario es redirigido automáticamente a la página de publicaciones con el filtro aplicado según la ciudad seleccionada.
- **Sección Carrusel Opiniones:** Después, se incluye un carrusel de opiniones que muestra comentarios y valoraciones de usuarios que han utilizado la plataforma, reforzando la credibilidad del servicio a través de testimonios reales.
- **Footer:** Finalmente, el footer o pie de página contiene accesos rápidos a secciones clave como: Inicio, Nosotros, Publicaciones y Política de privacidad, facilitando la navegación y acceso a información importante desde cualquier punto de la página.

Figura 12

Mockup Secciones Ciudades Disponibles, Carrusel Opiniones y Footer



- **Nosotros:** La página “Nosotros” presenta la esencia de Hoomie, transmitiendo la visión y los valores que guían el propósito de la plataforma. Se divide en dos secciones principales:
 - **Sección de Misión:** La primera sección está dedicada a la misión, donde se destaca el compromiso de Hoomie con facilitar la búsqueda del espacio ideal para vivir. El

texto resalta que la plataforma fue creada pensando en quienes desean encontrar un hogar o un roomie sin complicaciones, en un ambiente cómodo y seguro.

Figura 13

Mockup Nosotros. Sección Misión



- **Sección Valores:** La segunda sección está enfocada en los valores de la plataforma, presentados de manera visual y clara. Estos valores son: Confianza, Comunidad y Innovación:

Figura 14

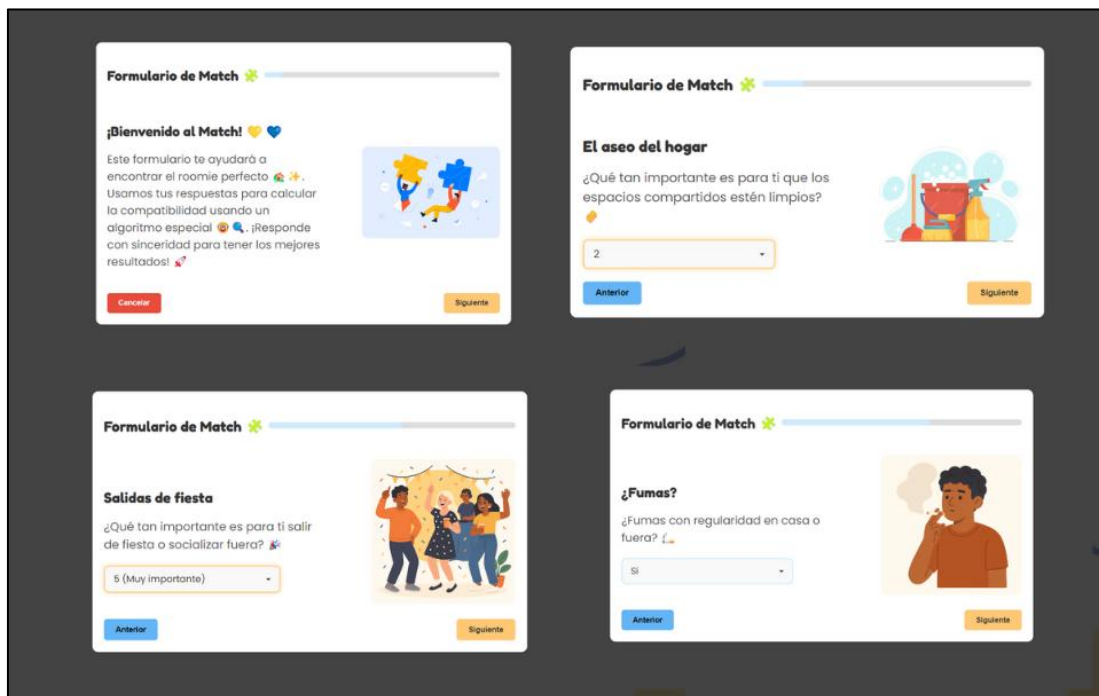
Mockup Nosotros. Sección Valores



- **Formulario Match:** El formulario de Match es una de las funcionalidades principales de Hoomie y ha sido diseñado como un formulario multistep (por pasos), con el objetivo de recopilar información clave del usuario de manera organizada y sin abrumarlo.

Figura 15

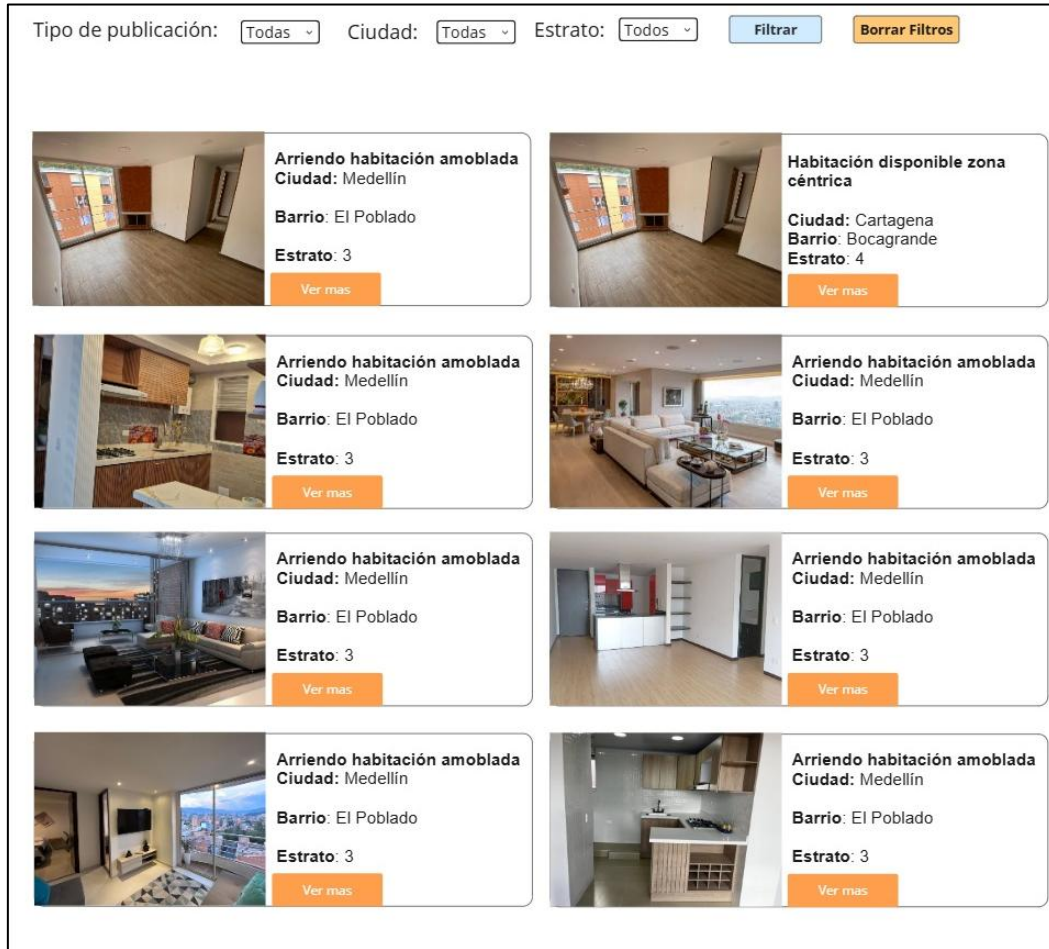
Mockup Formulario Match



- **Editar Perfil:** La vista de Editar Perfil permite a los usuarios modificar su información persona. El diseño de esta pantalla presenta un formulario estructurado en secciones donde el usuario puede actualizar datos como su nombre, descripción personal, preferencias, También incluye la opción de cambiar la foto de perfil y editar las respuestas del formulario de Match si desean ajustar sus criterios de búsqueda. Así mismo el usuario podrá visualizar las publicaciones que haya realizado

Figura 17

Mockup Vista de Publicaciones

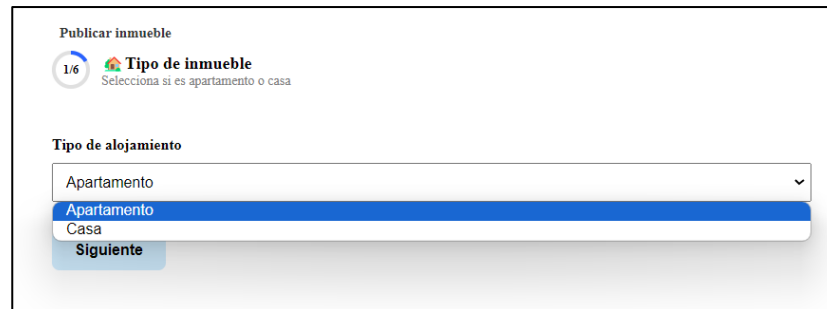


- Publicar:** La vista de Publicar permite a los usuarios crear un nuevo anuncio con toda la información necesaria para ofrecer un espacio o buscar un roomie. Entre los campos a diligenciar se encuentran: título del anuncio, tipo de propiedad, ubicación, precio, descripción, fotos del lugar, etiquetas relacionadas, y condiciones específicas (como si se aceptan mascotas o si incluye servicios). También se permite seleccionar el radio de búsqueda para ayudar a quienes visualizan el anuncio a ubicarlo geográficamente con mayor precisión.

- **Paso 1 (Tipo de Inmueble):** El usuario selecciona que tipo de inmueble desea publicar (Apartamento o Casa).

Figura 18

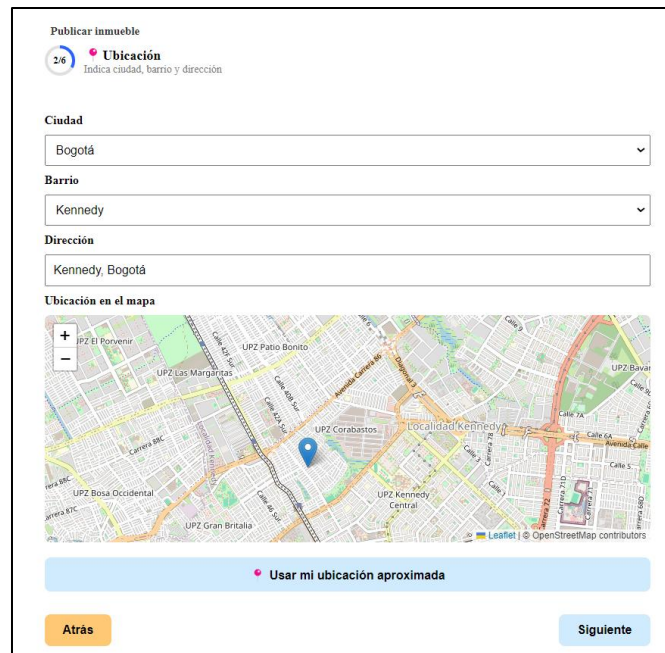
Mockup Publicar - Paso 1 Tipo Inmueble



- **Paso 2 (Ubicación):** El usuario indica los datos geográficos del inmueble como lo son ciudad, barrio y dirección. Así mismo encuentra la ubicación en el mapa o tiene la opción de utilizar su ubicación en tiempo real y la aplicación marcará automáticamente la ubicación en el mapa.

Figura 19

Mockup Publicar - Paso 2 Ubicación Inmueble



- **Paso 3 (Costos):** En este paso se definen los valores relacionados al arriendo o, así como si incluye administración

Figura 20

Mockup Publicar - Paso 3 Costos Inmueble



Publicar inmueble

3/6 **Costos**
Arriendo mensual y administración

Valor arriendo

\$ 900.000

¿Tiene costos de administración? Si no tiene valor de administración, desactiva el interruptor.

Atrás Siguiente

- **Paso 4 (Características físicas):** Se describe el espacio a nivel físico: número de habitaciones, baños, si tiene cocina, sala compartida, parqueadero, entre otros.

Figura 21

Mockup Publicar - Paso 4 Características Físicas



Publicar inmueble

4/6 **Características físicas**
Área, habitaciones, baños, parqueaderos y estrato

Área del inmueble (m²)

Número de baños

Número de habitaciones

Número de parqueaderos

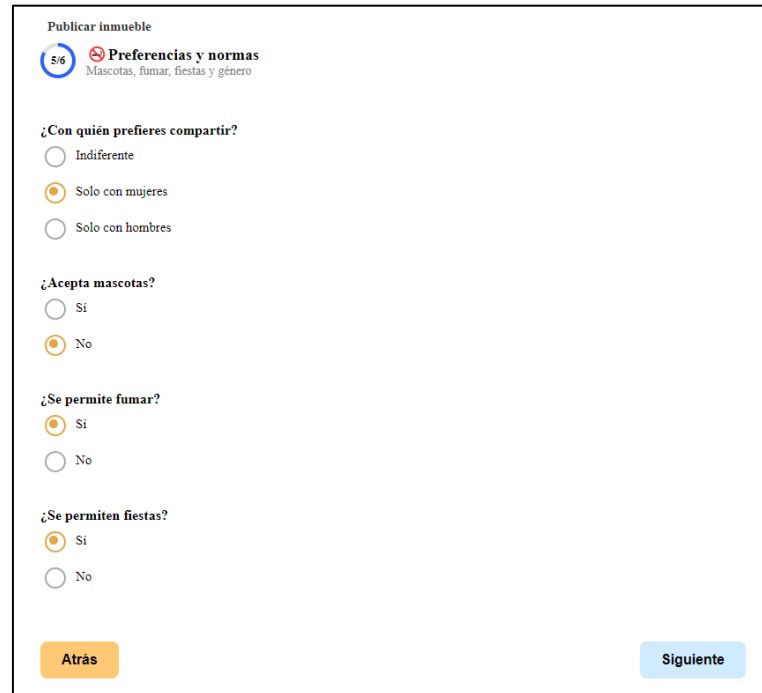
Estrato

Atrás Siguiente


- **Paso 5 (Preferencias y normas):** El usuario puede especificar qué tipo de roomie busca (por ejemplo, que sea estudiante, que no tenga mascotas, etc.) y establecer normas básicas de convivencia.

Figura 22

Mockup Publicar - Paso 5 Preferencias y normas



Publicar inmueble

5/6  **Preferencias y normas**
Mascotas, fumar, fiestas y género

¿Con quién prefieres compartir?

Indiferente

Solo con mujeres

Solo con hombres

¿Acepta mascotas?

Si

No

¿Se permite fumar?

Si

No

¿Se permiten fiestas?

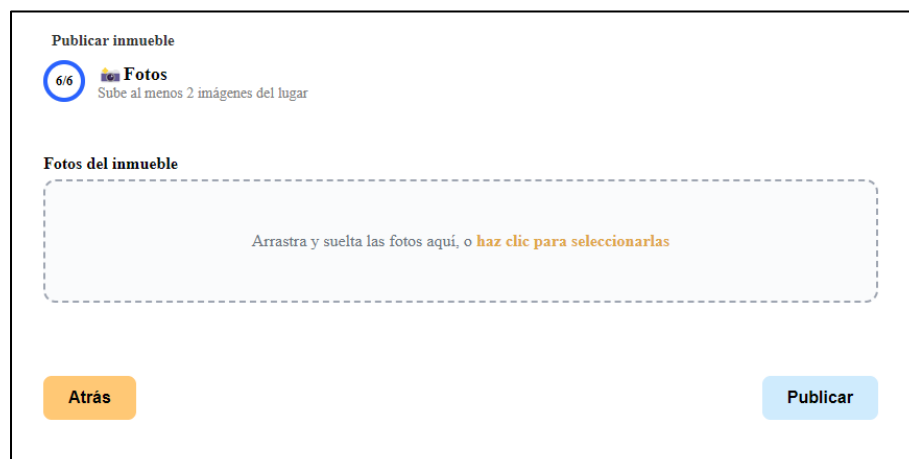
Si

No


- **Paso 6 (Fotos):** Finalmente, se permite al usuario subir imágenes del lugar.

Figura 23

Mockup Publicar - Paso 6 Fotos



Publicar inmueble

6/6  **Fotos**
Sube al menos 2 imágenes del lugar

Fotos del inmueble

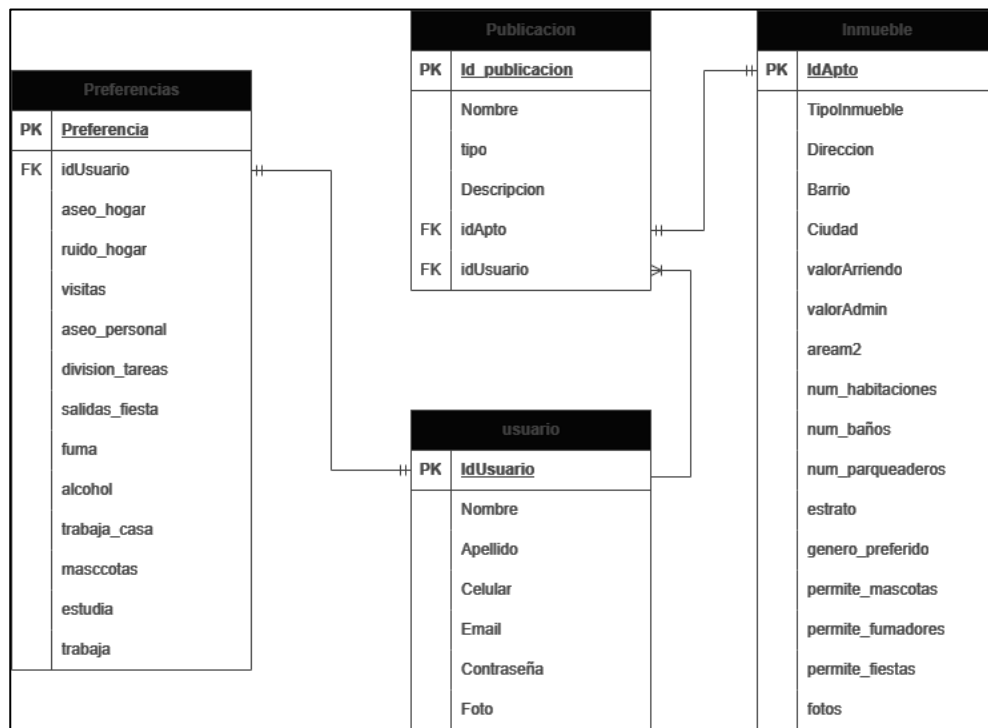
Arrastra y suelta las fotos aquí, o haz clic para seleccionarl

13. Modelo de datos

El modelo de datos de Hoomie define la estructura y organización de la información dentro del sistema, permitiendo almacenar, consultar y gestionar de forma eficiente los datos relacionados con los usuarios, publicaciones, coincidencias, calificaciones y demás elementos clave de la plataforma. Este modelo fue diseñado para garantizar integridad, escalabilidad y relaciones claras entre las entidades, permitiendo soportar funcionalidades como el sistema de match, el cargue de anuncios, la interacción entre usuarios y el historial de calificaciones. A continuación, se presentan las principales entidades y sus relaciones.

Figura 24

Modelo de la base de datos de Hoomie



13.1. Diccionario de datos

- **Tabla Usuario:** Almacena la información personal de los usuarios registrados en la plataforma (buscadores de roomies, arrendadores y administradores).

Tabla 17

Tabla Usuarios

| Campo | Tipo de Dato | Longitud | Restricciones | Descripción | Valores Permitidos |
|-------------------|--------------|----------|---------------------------------------|---|---------------------------------------|
| idUsuario | INT | - | PRIMARY KEY, NOT NULL, AUTO_INCREMENT | Identificador único del usuario en el sistema | Números enteros positivos únicos |
| Nombre | Varchar | 50 | NOT NULL | Nombre(s) completo del usuario | Texto alfabético, sin números |
| Apellido | Varchar | 50 | NOT NULL | Apellido(s) completo del usuario | Texto alfabético, sin números |
| Celular | Varchar | 15 | NOT NULL, UNIQUE | Número de teléfono celular del usuario | Formato: +57XXXXXXXXXX X |
| Email | Varchar | 100 | NOT NULL, UNIQUE | Dirección de correo electrónico del usuario | Formato de email válido |
| Contraseña | Varchar | 255 | NOT NULL | Contraseña encriptada del usuario | Hash bcrypt de la contraseña original |
| Foto | Varchar | 255 | NULL | URL de la foto de perfil del usuario | URL válida o NULL si no tiene foto |

- **Tabla Preferencias:** Define las preferencias de convivencia y estilo de vida de cada usuario para facilitar la compatibilidad entre roomies.

Tabla 18

Tabla Preferencias

| Campo | Tipo de Dato | Longitud | Restricciones | Descripción | Valores Permitidos |
|------------------------|--------------|----------|---|---|--|
| Preferencia | INT | - | PRIMARY KEY, NOT NULL, AUTO_INCREMENT | Identificador único de las preferencias | Números enteros positivos únicos |
| idUsuario | INT | - | FOREIGN KEY, NOT NULL, UNIQUE | Referencia al usuario propietario de las preferencias | Debe existir en tabla usuario |
| aseo_hogar | TINYINT | 1 | NOT NULL, CHECK (aseo_hogar BETWEEN 1 AND 5) | Nivel de importancia del aseo en áreas comunes | 1=Muy bajo, 2=Bajo, 3=Medio, 4=Alto, 5=Muy alto |
| ruido_hogar | TINYINT | 1 | NOT NULL, CHECK (ruido_hogar BETWEEN 1 AND 5) | Tolerancia al ruido en el hogar | 1=Muy tolerante, 2=Tolerante, 3=Neutro, 4=Poco tolerante, 5=Nada tolerante |
| visitas | TINYINT | 1 | NOT NULL, CHECK (visitas BETWEEN 1 AND 5) | Nivel de importancia a las visitas | 1=Muy bajo, 2=Bajo, 3=Medio, 4=Alto, 5=Muy alto |
| aseo_personal | TINYINT | 1 | NOT NULL, CHECK (aseo_personal BETWEEN 1 AND 5) | Nivel de importancia del aseo personal | 1=Muy bajo, 2=Bajo, 3=Medio, 4=Alto, 5=Muy alto |
| division_tareas | TINYINT | 1 | NOT NULL, CHECK (division_tareas BETWEEN 1 AND 5) | Nivel de Importancia de dividir tareas domésticas | 1=Muy bajo, 2=Bajo, 3=Medio, 4=Alto, 5=Muy alto |

- **Tabla Inmueble:** Contiene la información detallada de las propiedades disponibles para arriendo y sus características específicas.

Tabla 19

Tabla Inmueble

| Campo | Tipo de Dato | Longitud | Restricciones | Descripción | Valores Permitidos |
|----------------------|--------------|----------|---|--|---------------------------------------|
| idApto | INT | - | PRIMARY KEY, NOT NULL, AUTO_INCREMENT | Identificador único del inmueble | Números enteros positivos únicos |
| TipoInmueble | Varchar | 30 | NOT NULL | Tipo de propiedad disponible | "Apartamento", "Casa" |
| Direccion | Varchar | 200 | NOT NULL | Dirección física completa del inmueble | Dirección válida en Colombia |
| Barrio | Varchar | 50 | NOT NULL | Barrio donde se ubica el inmueble | Nombre de barrio válido |
| Ciudad | Varchar | 50 | NOT NULL | Ciudad donde se ubica el inmueble | Ciudades principales de Colombia |
| valorArriendo | Decimal | 10,2 | NOT NULL, CHECK (valorArriendo > 0) | Valor mensual del arriendo en pesos colombianos | Números decimales positivos |
| valorAdmin | Decimal | 10,2 | NULL, CHECK (valorAdmin >= 0) | Valor mensual de administración en pesos colombianos | Números decimales no negativos o NULL |

- **Tabla Publicación:** Registra las publicaciones creadas por los usuarios, ya sea para buscar roomies o para ofrecer propiedades en arriendo.

Tabla 20

Tabla de Publicación

| Campo | Tipo de Dato | Longitud | Restricciones | Descripción | Valores Permitidos |
|-----------------------|--------------|----------|---------------------------------------|--|---|
| id_publicacion | INT | - | PRIMARY KEY, NOT NULL, AUTO_INCREMENT | Identificador único de la publicación | Números enteros positivos únicos |
| Nombre | Varchar | 100 | NOT NULL | Título descriptivo de la publicación | Texto descriptivo sin caracteres especiales |
| tipo | Varchar | 20 | NOT NULL | Tipo de publicación que se está creando | "Busco_Roomie", "Ofrezco_Habitacion", "Ofrezco_Apartamento" |
| Descripcion | Text | - | NOT NULL | Descripción detallada de la publicación y requisitos | Texto libre con información relevante |
| idApto | Int | - | FOREIGN KEY, NOT NULL | Referencia al inmueble asociado con la publicación | Debe existir en tabla Inmueble |
| idUsuario | Int | - | FOREIGN KEY, NOT NULL | Referencia al usuario que creó la publicación | Debe existir en tabla usuario |
| valorAdmin | Decimal | 10,2 | NULL, CHECK (valorAdmin >= 0) | Valor mensual de administración en pesos colombianos | Números decimales no negativos o NULL |

Relaciones Entre Tablas

Tabla 21

Tabla de relaciones

| Relación | Tipo | Descripción | Integridad Referencial |
|--------------------------------------|------|--|---------------------------------------|
| usuario → Preferencias | 1:01 | Cada usuario tiene un único conjunto de preferencias | CASCADE en DELETE, RESTRICT en UPDATE |
| usuario → Publicación | 1:N | Un usuario puede crear múltiples publicaciones | CASCADE en DELETE, RESTRICT en UPDATE |
| Inmueble → Publicación | 1:N | Un inmueble puede tener múltiples publicaciones | RESTRICT en DELETE y UPDATE |

14. Implementación

La implementación de Hoomie se llevó a cabo bajo un enfoque modular y escalable, utilizando tecnologías web modernas y servicios en la nube para facilitar el acceso, mantenimiento y despliegue. El sistema se divide en tres capas principales: **frontend**, **backend** y **base de datos**, integrando además servicios externos a través de APIs.

- **Frontend:** La interfaz de usuario fue desarrollada con HTML, CSS y JavaScript, cuidando la experiencia visual, la usabilidad y la compatibilidad con dispositivos móviles.

Tabla 22

Implementación Frontend

| Nombre Archivo | Descripción |
|-----------------------|---|
| Archivo base de datos | Se encarga de conectar la aplicación con la base de datos local o de railway. |

| Archivos de modelos | Estos archivos tienen la forma base de las diferentes tablas (inmueble, preferencias, publicación, usuario) y conecta las tablas entre si en la base de datos. |
|-----------------------------|--|
| Archivos estáticos CSS | Estos archivos son los estilos que se utilizan en cada una de las vistas de la aplicación. |
| Archivos estáticos imágenes | Almacenan todas las imágenes utilizadas en el front. |
| Archivos estáticos JS | Son los diferentes archivos de javascript utilizados en las diferentes vistas. Se encargan de funciones como mostrar toasts, alerts y modales. |
| Vistas | |
| Base | En este archivo se encuentra la barra de navegación y el footer que se muestra en las distintas vistas debido a que funciona como una plantilla base. |
| Inicio | Es la pantalla inicial la cual da conexión a las diferentes funciones de la página tal como lo es: buscar roomie, buscar apartamento, realizar publicación, gestionar perfil, entre otras. |
| Perfil | Permite gestionar los datos del usuario, las preferencias y los inmuebles que ha publicado. |
| Publicaciones | Muestra y permite filtrar todas las publicaciones que se han realizado en la página. |
| Registro | Permite el registro del usuario nuevo. |
| Ingreso | Permite ingresar con correo y contraseña. Además, tiene la opción de restablecer contraseña o crear perfil. |
| Olvida | Permite restablecer la contraseña utilizando el correo del usuario. |
| Detalle | Muestra los detalles de la publicación (Datos del publicador, Datos de la publicación, Match) |
| Match | Es el formulario de preferencias que realizan los usuarios para lograr calcular la compatibilidad. |
| Publicar apto | Permite la publicación de apartamento o búsqueda de roomie. |
| Conócenos | En esta página se muestra información sobre que es Hoomie y cómo funciona la aplicación web. |

- **Backend:** La lógica del sistema fue implementada utilizando PHP, estructurada en módulos para manejar usuarios, publicaciones, coincidencias y validaciones. El backend también se encarga de la seguridad, gestión de sesiones, envío de correos y validación de credenciales.

Tabla 23

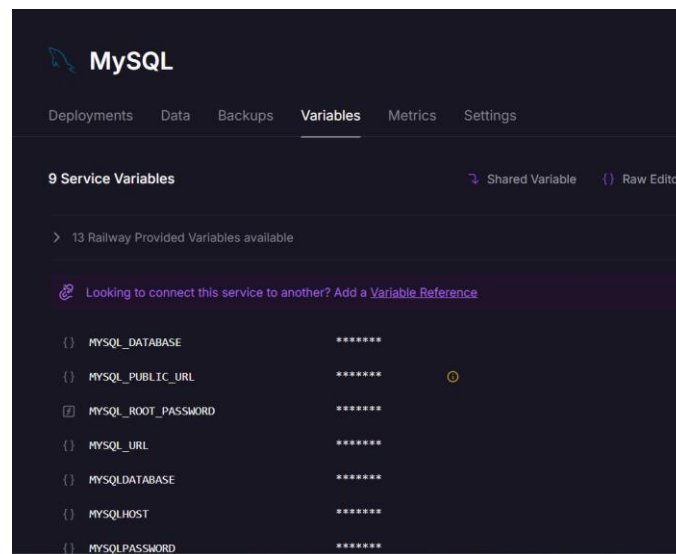
Implementación Backend

| Nombre Archivo | Descripción |
|----------------------|---|
| Main | Es la conexión entre todos los routers y el archivo que inicia la aplicación web. |
| router auth | Se encarga de autenticar el registro e inicio de sesión del usuario |
| router detalle | Este router permite la visualización de los detalles de la publicación |
| router match | Realiza la comparación de las preferencias entre los usuarios por medio de la función “cosine similarity” para obtener el nivel de compatibilidad que se tiene |
| Router perfil | Obtiene los datos del usuario una vez ingresado y los guarda en cookies para su uso |
| Router Preferencias | Se encarga de guardar y editar las preferencias del usuario |
| Router Publicacion | Permite crear, editar, publicar y eliminar publicaciones de inmuebles o búsqueda de roomies. |
| Router publicaciones | Filtra y muestra todas las publicaciones que se han realizado en la aplicación web. |
| Router Usuario | Permite el registro de un nuevo usuario confirmando que el correo no se haya utilizado anteriormente y enviando el correo de bienvenida una vez realizado el registro. |
| Router vistas | Se encarga de conectar los templates con jinja2 mostrando todo el front y utilizando la cookie del usuario cuando se tiene la sesión iniciada. Además, precarga los estilos y archivos estáticos. |
| Archivo Email | Contiene la función utilizada para enviar el correo de registro a los usuarios nuevos y el correo de recuperación en caso de olvidar la contraseña. |
| Archivo Sesión | Obtiene y confirma que la cookie del usuario contenga la información de un usuario ya registrado. |

- **Base de Datos:** El almacenamiento de la información se realiza en una base de datos MySQL, desplegada en la nube mediante **Railway**, lo que permite una gestión confiable y accesible desde el backend. El modelo de datos fue diseñado bajo un enfoque relacional, con relaciones claras entre entidades y procesos de normalización.

Figura 25

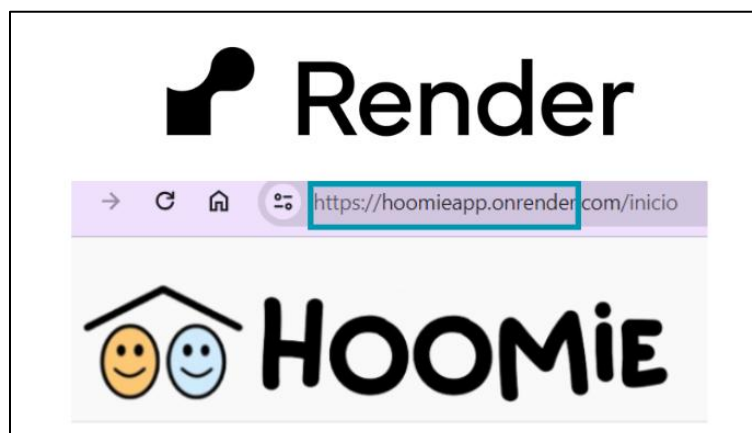
Implementación Railway MySQL



- **APIs Externas:** Hoomie se conecta con servicios externos como una API de mapas para funcionalidades de geolocalización, y servicios de correo para el proceso de registro y recuperación de contraseña.
- **Despliegue en la Nube:** El backend fue desplegado utilizando **Render**, un servicio de hosting en la nube que facilita el despliegue continuo, manejo de entornos y acceso remoto.

Figura 26

Implementación Render - Despliegue en la Nube



15. Test

Con el fin de garantizar la calidad, funcionalidad y estabilidad del sistema, se llevaron a cabo diferentes tipos de pruebas sobre los módulos clave de Hoomie. Estas pruebas permitieron identificar posibles errores, validar el cumplimiento de los requerimientos funcionales y no funcionales, y asegurar una experiencia de usuario fluida y confiable.

Tests funcionales

- Confirmar que al registrar un usuario se añada en la base de datos
- Validar que no existe un usuario con el mismo correo al registrar
- Confirmar que el usuario existe y la contraseña es correcta al iniciar sesión
- Verificar que el cambio de contraseña se realice correctamente
- Verificar que los cambios y ediciones a la información del perfil se actualicen en la base de datos
- Confirmar que el usuario se elimina de la base de datos al eliminar el perfil
- Validar la subida de una publicación realizada por el usuario a la base de datos
- Confirmar el funcionamiento del match de compatibilidad
- Validar que los datos de publicador sean correctos al mostrar una publicación

Test de usabilidad

- Permitir la visualización en dispositivos móviles, tablets y distintos navegadores.
- Verificar que todos los botones y enlaces funcionen correctamente
- Mostrar validaciones al rellenar los formularios

16. Conclusiones

- Se ha demostrado la viabilidad técnica y comercial de una solución tecnológica orientada a resolver las problemáticas asociadas con la búsqueda de vivienda compartida en el contexto colombiano, donde actualmente no existe una plataforma especializada con enfoque en compatibilidad y convivencia.
- La propuesta de valor de Hoomie se fundamenta en la implementación de algoritmos de compatibilidad que trascienden la simple intermediación inmobiliaria, incorporando elementos de análisis psicosocial para optimizar la convivencia entre usuarios. Esto representa una diferenciación significativa frente a las soluciones tradicionales del mercado.
- Se garantiza la escalabilidad del sistema gracias al uso de arquitecturas modernas, servicios en la nube (Render y Railway), y una estructura modular que permite la evolución y crecimiento de la plataforma sin comprometer su estabilidad.
- La implementación de tecnologías como el algoritmo de matchmaking y las APIs de geolocalización demuestra la aplicación práctica de conceptos avanzados de ingeniería de software, incluyendo diseño basado en componentes, separación de responsabilidades y uso de servicios externos.
- La estructura actual permite la incorporación futura de funcionalidades adicionales como verificación de identidad, pagos integrados o un sistema de soporte, sin alterar la lógica del sistema base, gracias a la organización clara de los módulos.
- Se evidencian desafíos importantes relacionados con la privacidad de los usuarios, la seguridad de la información y la estabilidad de la plataforma ante un posible crecimiento exponencial en el número de usuarios. Estos aspectos deberán abordarse con estrategias robustas de ciberseguridad y pruebas de carga progresivas.

Bibliografía

Bravo, E., Cieza, S., Flores, L., & Uceda, L. (2022). Revisión Sistemática de la Literatura sobre Metodologías Ágiles en el Desarrollo de Software aplicadas a la Gestión de Proyectos. *Iberian Journal of Information Systems and Technologies*, E54.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). Introduction to algorithms, 4 Edition. In *The MIT Press Cambridge, Massachusetts London, England*.

Marek, C. I., Wanzer, M. B., & Knapp, J. L. (2004). An exploratory investigation of the relationship between roommates' first impressions and subsequent communication patterns. *Communication Research Reports*, 21(2). <https://doi.org/10.1080/08824090409359982>

Martinez, J. I., & López Comino, M. (2018). El método ágil Scrum, evolución y aplicación en la administración de proyectos.) *Universidad UNR, Argentina, Universidad de Oviedo*.

Nielsen, J., & Norman, D. (2018a). The definition of user experience (UX). Nielsen Norman Group. *Nielsen Norman Group (NN/g)*.

Ley 820 de 2003 - Gestor Normativo. (s. f.). Función Pública. <https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=8738>

Ley 1581 de 2012 - Gestor Normativo. (s. f.). Función Pública. <https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=49981>

Pressman, R. (2010). Ingeniería Del Software I. *Ingeniería Del Software I*.

Randhawa, T. S. (2022). Mobile Applications. In *Mobile Applications*. <https://doi.org/10.1007/978-3-030-02391-1>

Resnick, P., & Zeckhauser, R. (2002a). Trust among strangers in internet transactions: Empirical analysis of eBay' s reputation system. *Advances in Applied Microeconomics*, 11. [https://doi.org/10.1016/S0278-0984\(02\)11030-3](https://doi.org/10.1016/S0278-0984(02)11030-3)