



UNIVERSIDAD POLITÉCNICA DE PACHUCA

INGENIERÍA EN SOFTWARE

PROYECTO

E-COMMERCE

ADMINISTRADORES:

CodeMakeSoft

CATEDRÁTICO:

Miguel Angel Montoya Cerro

Mayo – Agosto 2025

ESPECIFICACIÓN PARA EL DISEÑO DEL SISTEMA “CASA FÁCIL”

INFORMACIÓN DEL DOCUMENTO

Nombre del proyecto:	Casa Fácil
Fecha:	4 de agosto de 2025
Versión:	3.0
Creado por:	CodeMakeSoft
Revisado por:	Miguel Angel Montoya Cerro
Aprobado por:	Miguel Angel Montoya Cerro

Índice

1. Introducción	1
2. Objetivo general	2
2.1. Objetivos específicos	2
3. Alcance del Sistema	3
4. Funcionalidades del Sistema	4
4.1. Autenticación y Gestión de Usuarios	4
4.2. Gestión de Propiedades	4
4.3. Búsqueda y Exploración	5
4.4. Interacción del Usuario	5
4.5. Pagos, Transacciones y Facturas	5
4.6. Panel de Administración	6
4.7. Servicios Complementarios	6
5. Roles del sistema	7
6. Modelo de negocio	9
6.1. Clasificación del Modelo de Negocio	9
6.1.1. Tipo Principal	9
6.2. Segmentación de Mercado Específica	9
6.2.1. Análisis Detallado por Segmento	9
6.3. Análisis Financiero Específico	10
6.3.1. Estructura de Ingresos Recurrentes	10
6.3.2. Mecanismos de Conversión por Limitaciones	10
6.4. Análisis de Ventajas Competitivas	11
6.4.1. Diferenciación Tecnológica Específica	11
6.4.2. Sistema de Posicionamiento Jerárquico	12
6.5. Análisis de Riesgos y Factores Críticos	12
6.5.1. Factores Críticos de Éxito Específicos	12
6.5.2. Riesgos Específicos del Modelo	12
6.6. Membresías	13
7. Requisitos Funcionales	14
8. Requisitos No Funcionales	28
9. Análisis de Componentes del Sistema <i>Casa Fácil</i>	31
9.1. Cliente Móvil/Web	31
9.2. Backend	31
9.3. Base de Datos	32
9.4. API de Pagos	32
9.5. Servicios de Mapas	32
9.6. Diagrama de Comunicación	33
10. Estilos Arquitectónicos	34
10.1. Microservicios	34

11.Patrones Arquitectónicos	35
12.Patrones de Diseño	40
13.Diagramas	50
13.1. Diagrama de componenes.	50
13.2. Diagrama de patrones y estilo.	50
13.3. Diagrama de Clases Inicial con Entidades Principales y Relaciones.	51
14.Resumen del Sistema	52
15.Identificación de atributos de calidad	56
15.1. Rendimiento (Performance)	56
15.2. Disponibilidad	56
15.3. Seguridad	57
15.4. Escalabilidad	57
15.5. Mantenibilidad	57
15.6. Usabilidad	58
15.7. Accesibilidad	58
16.Evaluación de Riesgos	59
16.1. Identificación de riesgos	59
16.2. Análisis de Riesgos	60
16.3. Planes de Mitigación	61
16.4. Monitoreo y Control	62
16.5. Documentación	62
17.Estrategia de Escalabilidad	63
18.Plan de Pruebas	65
19.Diagrama de Escalabilidad	67
20.Mapa de riesgos	69

Índice de figuras

1.	Diagrama de Comunicación.	33
2.	Componentes API Gateway.	35
3.	Componentes CQRS.	36
4.	Componentes Event Sourcing.	36
5.	Componentes Service Registry & Discovery.	37
6.	Componentes Backend for Frontend.	37
7.	Componentes Saga.	38
8.	Componentes Strangler Fig.	39
9.	Patrón de diseño Strategy.	40
10.	Patrón de diseño Observer.	41
11.	Patrón de diseño Facade.	42
12.	Patrón de diseño Decorator.	43
13.	Patrón de diseño Singleton.	44
14.	Patrón de diseño Factory.	45
15.	Patrón de diseño Repository.	46
16.	Patrón de diseño Composite.	47
17.	Patrón de diseño Lazy Loading.	48
18.	Diagrama de Componentes.	50
19.	Diagrama de Patrones y Estilo Arquitectónico.	50
20.	Diagrama de Clases.	51
21.	Diagrama de escalabilidad del sistema CasaFácil	67

Índice de cuadros

2.	Ingresos recurrentes.	10
3.	Posicionamiento de propiedades por plan.	12
4.	Membresías.	13
5.	Entradas, validaciones y comportamientos.	23
6.	Identificación de Riesgos	59
8.	Análisis de Riesgos	60
10.	Planes de Mitigación	61
12.	Matriz de Riesgos IPER	69

1. Introducción

Este documento presenta el análisis, diseño y planificación de un sistema integral para la renta de propiedades, dirigido a cualquier persona que busque alojamiento. La plataforma conectará a usuarios con arrendadores, ofreciendo herramientas de búsqueda avanzada, mensajería interna, contratos digitales y servicios complementarios.

Actualmente, existen diversas páginas y aplicaciones que abordan esta necesidad; sin embargo, muchas presentan deficiencias, como su escasa presencia en países hispanohablantes o su enfoque limitado a un público específico.

Este proyecto busca resolver problemas como la falta de información centralizada y la baja visibilidad de propiedades, priorizando una interfaz amigable, así como seguridad y transparencia en los procesos de renta.

A lo largo del documento se detallan las actividades a realizar, los requerimientos funcionales y no funcionales, las metodologías de recolección de información, los diagramas necesarios para comprender la estructura del sistema, y un análisis de los alcances y oportunidades. Todo ello con el objetivo de mejorar la experiencia de renta mediante una solución tecnológica accesible, confiable y escalable.

2. Objetivo general

- Desarrollar una aplicación web que funcione como punto de encuentro entre arrendadores y arrendatarios de inmuebles, que permita a los usuarios interactuar en tiempo real, mediante una interfaz accesible, y una lógica de negocio basada en arquitectura de microservicios y orientada a eventos.

2.1. Objetivos específicos

- Facilitar la publicación y gestión de propiedades por parte de los propietarios para lograr un mayor alcance, a través de un panel de control intuitivo que permita registrar, editar, visualizar y organizar inmuebles, incorporando imágenes, ubicación geográfica en mapa, precios, reglas de convivencia y servicios disponibles.
- Permitir a los usuarios buscar alojamiento según filtros personalizados para cubrir sus necesidades de vivienda, mediante un motor de búsqueda que incluya criterios como rango de precio, tipo de propiedad, distancia a puntos de interés, reglas de la vivienda (horarios de visitas, no mascotas, etc.), y que muestre los resultados tanto en forma de lista como en un mapa interactivo.
- Garantizar la seguridad y transparencia en los procesos de renta para mayor confiabilidad, mediante la implementación de validación de cuentas, validación de credenciales, contratos digitales, historial de actividad, sistema de calificaciones y reseñas, además de notificaciones automáticas.
- Integrar servicios locales complementarios (como lavandería, paradas de autobús o tiendas cercanas), mediante la conexión con APIs de mapas y servicios externos, y la inclusión de filtros que permitan resaltar inmuebles con estas ventajas en su cercanía o ya incluidos.

3. Alcance del Sistema

El sistema se desarrollará como una plataforma web y móvil (SPA) para conectar a personas que buscan alojamiento con propietarios que desean rentar sus inmuebles. El público objetivo está compuesto, por un lado, por los propietarios de inmuebles que buscan promocionar sus espacios para renta de forma eficiente y segura, y por otro, por usuarios que requieren alojamiento, tales como estudiantes foráneos, trabajadores desplazados, emprendedores, profesionistas y cualquier persona que necesite rentar una propiedad. El alcance contempla funcionalidades esenciales para permitir la interacción y comunicación entre ambos actores y facilitar el proceso de publicación, búsqueda y reserva de propiedades.

4. Funcionalidades del Sistema

El sistema *Casa Fácil* incluye una serie de funcionalidades esenciales que permiten a los usuarios interactuar eficientemente entre sí y con el sistema. A continuación, se detallan todas las funcionalidades clave:

4.1. Autenticación y Gestión de Usuarios

- **Registro de usuarios:** Permite a nuevos usuarios registrarse mediante un formulario en línea o mediante OAuth.
- **Inicio de sesión:** Los usuarios pueden iniciar sesión con sus credenciales.
- **Recuperación de contraseña:** Los usuarios pueden recuperar el acceso a su cuenta mediante un enlace enviado a su correo electrónico.
- **Gestión de roles y permisos:** El sistema permite diferenciar funcionalidades según el rol (usuario autenticado, usuario no autenticado, propietario o administrador).
- **Verificación de identidad del propietario:** El sistema debe verificar la identidad del propietario como dueño o administrador del inmueble.
- **Cierre de sesión seguro:** El sistema permite a los usuarios cerrar su sesión de forma segura.
- **Gestión de perfil:** Los usuarios pueden actualizar su información personal, foto y preferencias desde su perfil.
- **Eliminación o desactivación de cuenta:** Los usuarios pueden eliminar o desactivar su cuenta conforme a la política de privacidad.

4.2. Gestión de Propiedades

- **Visualización de propiedades:** Los usuarios pueden ver las propiedades disponibles en la plataforma con información básica.
- **Detalles de propiedad:** Al seleccionar una propiedad, se muestran su información adicional, incluyendo galería, descripción, precio, ubicación, subcategorías y reseñas.
- **Creación de propiedades:** Los usuarios con rol propietario pueden registrar nuevas propiedades para rentar.
- **Edición y eliminación de propiedades:** Los propietarios o administradores pueden modificar o eliminar propiedades previamente publicadas.
- **Publicación programada de propiedades:** Los propietarios pueden seleccionar una fecha futura para la publicación automática de sus propiedades.
- **Revisión/moderación de propiedades por administrador:** Las propiedades registradas son revisadas por un administrador antes de ser visibles en la plataforma.

4.3. Búsqueda y Exploración

- **Búsqueda de propiedades:** El sistema permite a los usuarios buscar propiedades mediante filtros avanzados (precio, tipo, ubicación, categoría).
- **Búsqueda en mapa interactivo:** Los resultados de búsqueda se visualizan geográficamente sobre un mapa interactivo.
- **Búsqueda por cercanía a servicios clave:** El sistema permite filtrar propiedades según su proximidad a lugares de interés (universidades, transporte, hospitales, etc.).

4.4. Interacción del Usuario

- **Favoritos:** Los usuarios pueden marcar propiedades como favoritas para acceder fácilmente a ellas.
- **Solicitud de reserva:** Los usuarios autenticados pueden enviar una solicitud para reservar una propiedad.
- **Agendamiento de citas:** Los usuarios pueden agendar visitas presenciales o virtuales con los propietarios.
- **Reserva de propiedad:** Los usuarios pueden reservar temporalmente una propiedad mediante pago anticipado.
- **Mensajería interna:** Los usuarios pueden enviar mensajes a los propietarios directamente desde la plataforma.
- **Calificación y reseña:** Los usuarios pueden calificar y dejar reseñas después de completar una transacción.
- **Reporte de usuarios o publicaciones:** Los usuarios pueden reportar publicaciones sospechosas o usuarios con comportamiento inadecuado.
- **Notificaciones internas y por correo:** El sistema envía notificaciones relevantes dentro de la aplicación y por medio de correo electrónico.

4.5. Pagos, Transacciones y Facturas

- **Opciones de pago:** El sistema permite a los usuarios realizar pagos mediante la pasarela de pago de Mercado Pago.
- **Generación de comprobante:** Tras una transacción exitosa, se genera un comprobante en PDF con los detalles.
- **Generación de factura:** Tras una operación exitosa se podrá emitir la factura correspondiente (si se requiere).
- **Historial de rentas:** Los usuarios pueden consultar su historial de rentas.
- **Gestión de reembolsos:** Los usuarios pueden solicitar reembolsos según las políticas establecidas.

4.6. Panel de Administración

- **Panel de administración:** El sistema permite a los administradores gestionar usuarios, propiedades, reservas y transacciones.
- **Gestión de reportes de usuarios:** Los administradores pueden revisar reportes realizados por los usuarios sobre contenidos o comportamientos inadecuados.
- **Estadísticas por periodo de tiempo:** El sistema permite visualizar métricas clave como usuarios activos, ingresos y publicaciones.
- **Exportación de datos y reportes:** Los administradores pueden exportar reportes en formatos como CSV, PDF, Excel, etc.

4.7. Servicios Complementarios

- **Integración de servicios locales:** El sistema conecta con APIs de mapas y servicios externos para mostrar información sobre lavanderías, estacionamientos, transporte y otros servicios cercanos a las propiedades.

5. Roles del sistema

El sistema Casa Fácil se divide en varios roles para garantizar la seguridad, el control de acceso y la funcionalidad adecuada según el tipo de usuario. Los roles son los siguientes:

Usuario no autenticado (Guest)

- Puede explorar propiedades sin registrarse.
- No puede interactuar con otros usuarios ni realizar acciones críticas (como reservas o publicaciones).
- Solo tiene acceso a información pública y visualización de propiedades.

Usuario autenticado (User)

- Puede buscar y visualizar propiedades según filtros definidos (ubicación, precio, servicios).
- Puede agregar propiedades a favoritos.
- Puede enviar mensajes a los propietarios para solicitar más información.
- Puede realizar reservas de inmuebles, siempre que estén disponibles.
- Puede ver su historial de reservas y transacciones.
- Puede calificar y dejar reseñas sobre propiedades al finalizar una reserva.
- Puede agendar visitas presenciales o virtuales a propiedades.
- Puede recibir notificaciones internas y por correo electrónico.

Propietario (Owner)

- Puede crear y gestionar sus propiedades, incluyendo edición y eliminación.
- Puede subir imágenes, descripciones y especificaciones de cada propiedad.
- Puede programar la fecha de publicación de sus propiedades.
- Puede responder a mensajes de usuarios interesados en sus propiedades.
- Puede ver el estado de disponibilidad de sus propiedades.
- El número de propiedades que puede publicar está limitado según el paquete adquirido:
 - Paquete Básico: 5 propiedades activas.
 - Paquete Estándar: 10 propiedades activas.
 - Paquete Premium: 15 propiedades activas.
- Puede ver el historial de reservas y contactos relacionados con sus propiedades.

Administrador (Admin)

- Tiene permisos completos para gestionar usuarios, propiedades, reservas y transacciones.
- Puede realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) sobre usuarios, propiedades y transacciones.
- Puede gestionar permisos y roles dentro del sistema.
- Puede descargar información en diferentes formatos (CSV, PDF, Excel).
- Puede revisar y moderar reportes de usuarios o propiedades.
- Puede generar estadísticas y métricas del sistema.
- El rol de administrador puede ser segmentado en niveles de permiso para escalabilidad:
 - **Administrador General:** Acceso total a todas las funciones del sistema.
 - **Administrador de Contenido:** Solo gestiona propiedades, usuarios y reportes.
 - **Administrador de Seguridad:** Gestionar usuarios, roles y permisos.
 - **Administrador de Reportes:** Solo visualiza y genera informes.

6. Modelo de negocio

6.1. Clasificación del Modelo de Negocio

6.1.1. Tipo Principal

Modelo Freemium-Premium con Suscripción Recurrente Escalonada

Componentes Específicos

1. **Plataforma de Intermediación Bilateral:** Conecta propietarios con inquilinos
2. **Monetización por Suscripción:** Ingresos recurrentes mensuales
3. **Diferenciación por Servicios:** Cada plan ofrece funcionalidades específicas
4. **Estrategia de Conversión Ascendente:** Incentivos para upgrade entre planes

6.2. Segmentación de Mercado Específica

6.2.1. Análisis Detallado por Segmento

Segmento 1: Usuarios Ocasionales

- **Perfil:** Personas físicas que rentan 1-2 propiedades
- **Planes dirigidos:** Prueba Gratuita (15 días) → Bronce (\$79.99 MXN/mes)
- **Capacidad específica:** 1-2 propiedades activas, 2-3 imágenes por propiedad
- **Comunicación limitada:** 3-10 mensajes/día, 100-200 caracteres/mensaje
- **Propuesta de valor:** Acceso económico con funcionalidades básicas suficientes

Segmento 2: Propietarios Frecuentes

- **Perfil:** Inversores pequeños, personas con 1-12 propiedades
- **Plan dirigido:** Plata (\$99.00 MXN/mes)
- **Capacidad específica:** Hasta 12 propiedades activas, 6 imágenes por propiedad
- **Comunicación mejorada:** 20 mensajes/día, 400 caracteres/mensaje
- **Automatización:** Chatbot avanzado con información específica (precio, disponibilidad, dirección)
- **Herramientas profesionales:** Panel de métricas personalizadas incluido

Segmento 3: Profesionales Inmobiliarios

- **Perfil:** Inmobiliarias, corredores, empresas del sector inmobiliario
- **Plan dirigido:** Oro (\$129.99 MXN/mes)
- **Capacidad específica:** Hasta 40 propiedades activas, 12 imágenes y 2 videos por propiedad
- **Comunicación ilimitada:** Sin límite de mensajes, 1000 caracteres/mensaje
- **Tecnología premium:** Chatbot conversacional con diálogo natural
- **Ventajas competitivas:** Posicionamiento destacado, verificación, métricas con IA

6.3. Análisis Financiero Específico

6.3.1. Estructura de Ingresos Recurrentes

Plan	Precio Mensual	Precio Anual	Ingreso/Propiedad
Bronce	\$79.99 MXN	\$959.88 MXN	\$39.99 MXN
Plata	\$99.00 MXN	\$1,188.00 MXN	\$6.60 MXN
Oro	\$129.99 MXN	\$1,559.88 MXN	\$3.25 MXN

Cuadro 2: Ingresos recurrentes.

Análisis de Eficiencia por Propiedad

1. **Plan Bronce:** Menor eficiencia (\$39.99/propiedad) pero mayor margen
2. **Plan Plata:** Eficiencia media (\$6.60/propiedad) con servicios balanceados
3. **Plan Oro:** Mayor eficiencia (\$3.25/propiedad) orientado a volumen

6.3.2. Mecanismos de Conversión por Limitaciones

Prueba Gratuita Estandar

- **Limitación temporal:** 15 días de acceso
- **Limitación de capacidad:** Solo 1 propiedad activa
- **Limitación de comunicación:** 2 mensajes/día, 50 caracteres
- **Limitación de exposición:** Sin chatbot, posicionamiento bajo

De Prueba Gratuita a Bronce

- **Limitación de capacidad:** Solo 1-2 propiedades activas
- **Limitación de comunicación:** 3 mensajes/día, 100 caracteres
- **Limitación de exposición:** Sin chatbot, posicionamiento estándar

De Bronce a Plata

- **Escalabilidad:** Hasta 12 propiedades (incremento 600 %)
- **Comunicación:** De 10 a 20 mensajes/día, de 200 a 400 caracteres
- **Automatización:** Chatbot básico vs. avanzado con datos específicos
- **Análisis:** Acceso a métricas personalizadas
- **Posicionamiento:** Prioridad sobre planes inferiores

De Plata a Oro

- **Escalabilidad premium:** De 12 a 40 propiedades (incremento 333 %)
- **Comunicación ilimitada:** Sin límite de mensajes, 1000 caracteres
- **Tecnología avanzada:** Chatbot conversacional con IA
- **Máxima visibilidad:** Posicionamiento destacado (Top)
- **Servicios profesionales:** Verificación, soporte prioritario, métricas con IA

6.4. Análisis de Ventajas Competitivas

6.4.1. Diferenciación Tecnológica Específica

Evolución del Chatbot

1. **Nivel 0 (Gratuito):** Sin chatbot
2. **Nivel 1 (Bronce):** Chatbot básico - Saludo automático y notificación de respuesta
3. **Nivel 2 (Plata):** Chatbot avanzado - Información específica (precio, disponibilidad, dirección)
4. **Nivel 3 (Oro):** Chatbot premium - Conversacional, respuestas específicas, diálogo natural

6.4.2. Sistema de Posicionamiento Jerárquico

Plan	Posición	Ventaja Específica
Gratuito	Estándar	Sin prioridad
Bronce	Estándar	Sobre Gratuito
Plata	Prioritario	Sobre Bronce y Gratuito
Oro	Destacado (Top)	Máxima visibilidad

Cuadro 3: Posicionamiento de propiedades por plan.

6.5. Análisis de Riesgos y Factores Críticos

6.5.1. Factores Críticos de Éxito Específicos

Métricas de Conversión Clave

1. **Tasa de conversión Gratuito → Bronce:** Meta >15 %
2. **Tasa de conversión Bronce → Plata:** Meta >25 %
3. **Tasa de conversión Plata → Oro:** Meta >20 %
4. **Tasa de retención mensual:** Meta >85 %

6.5.2. Riesgos Específicos del Modelo

Riesgo de Saturación por Plan

- **Plan Bronce:** Límite de 2 propiedades puede ser insuficiente
- **Plan Plata:** 12 propiedades puede satisfacer completamente el segmento
- **Plan Oro:** 40 propiedades puede ser excesivo para algunos usuarios

Riesgo de Comunicación Limitada

- **Límite de caracteres:** Puede frustrar comunicación efectiva
- **Límite de mensajes:** Puede interrumpir negociaciones importantes
- **Competencia con WhatsApp:** Usuarios pueden migrar comunicación fuera de la plataforma

6.6. Membresías

Cuadro 4: Membresías.

Característica	Prueba	Bronce	Plata	Oro
Duración/ Precio	15 días gratuitos	\$79.99 MXN/mes	\$99.00 MXN/mes	\$129.99 MXN/mes
Propiedades activas	1 propiedad	Hasta 2	Hasta 12	Hasta 40
Imágenes por propiedad	2 imágenes	3 imágenes	6 imágenes	12 imágenes y 2 videos
Chat entre usuarios - Mensajes por día	Hasta 2 mensajes	Hasta 10 mensajes	Hasta 20 mensajes	Mensajes ilimita- dos
Chat entre usuarios - Ca- racteres por mensaje	50 caracteres má- ximo	200 caracteres má- ximo	400 caracteres má- ximo	1000 caracteres máximo
Chatbot	No disponible	Básico: Saludo au- tomático. Indica que el propietario responderá	Avanzado: Saludo + características básicas (precio, disponibilidad, dirección)	Premium: Conver- sacional, responde preguntas específi- cas, diálogo natu- ral
Posiciona- miento	Estándar	Bajo	Prioritario sobre Bronce y Gratis	Destacado (Top)
Soporte	No disponible	General	General	Prioritario
Verificación de propiedad	No disponible	Si	Si	Sí (insignia de ve- rificación)
Métricas per- sonalizadas	No disponible	No	Sí (incluido)	Sí (con IA)
Propiedades recomendadas	No	No	No	Sí (con mayor prioridad)

7. Requisitos Funcionales

SECCIÓN 1: AUTENTICACIÓN Y GESTIÓN DE USUARIOS

RF01. Registro de usuarios

- **Descripción:** El sistema permitirá a los nuevos usuarios registrarse mediante un formulario o servicios como Google o Facebook.
- **Rol:** Usuario no autenticado.
- **Entradas esperadas:**
 - Solicitud de Permisos (Scopes para obtener datos de 'profile' y 'email') en caso de utilizar OAuth.
 - Nombre de usuario (texto, obligatorio).
 - Correo electrónico (formato válido, obligatorio).
 - Contraseña (mínimo 8 caracteres que contengan un caracter especial y una mayúscula, obligatorio).

RF02. Inicio de Sesión

- **Descripción:** El sistema permitirá a los usuarios iniciar sesión con sus credenciales. Al autenticarse, la aplicación generará un token único y temporal que permitirá el acceso a funcionalidades según su rol. Este token se eliminará al cerrar sesión para garantizar la seguridad, también el inicio de sesión se puede hacer mediante Facebook o Google.
- **Rol:** Usuario autenticado.
- **Entradas esperadas:**
 - Correo electrónico (obligatorio).
 - Contraseña (obligatorio).
 - Token de inicio de sesión.

RF03. Recuperación de contraseña

- **Descripción:** El sistema permitirá a los usuarios registrados recuperar el acceso a su cuenta mediante un enlace enviado a su correo electrónico registrado.
- **Rol:** Usuario autenticado.
- **Entradas esperadas:**
 - Correo electrónico (obligatorio).

RF04. Gestión de roles y permisos

- **Descripción:** El sistema permitirá diferenciar funcionalidades según el rol y el permiso (usuario autenticado, usuario no autenticado, propietario o administrador). Los administradores tendrán acceso completo, los propietarios gestionarán propiedades pero no usuarios ni configuración, los usuarios autenticados solo verán, gestionarán sus reservas y perfil, y los usuarios no autenticados solo podrán mirar propiedades. Los usuarios con rol de administrador pueden tener diferentes permisos para promover la escalabilidad del sistema.
- **Rol:** Todos los usuarios.
- **Entradas esperadas:**
 - Acceso al sistema (obligatorio).

RF05. Verificación de identidad del propietario

- **Descripción:** El sistema debe verificar la identidad del propietario como dueño o administrador del inmueble. La documentación será validada de forma inmediata por un sistema basado en inteligencia artificial que detectará y validará automáticamente la autenticidad y coincidencia de los datos ingresados. El sistema emitirá una respuesta en menos de 5 minutos. En caso de que los documentos sean ilegibles o no coincidan, se notificará al usuario con observaciones específicas para permitir la re-subida de la documentación.
- **Rol:** Usuario con rol del Propietario.
- **Entradas esperadas:**
 - Identificación oficial (INE, pasaporte, cartilla militar o cédula profesional).
 - Título de propiedad o documento que acredite la posesión legal del inmueble.

RF06. Cierre de sesión seguro

- **Descripción:** El sistema permitirá a los usuarios cerrar su sesión de forma segura, eliminando el token de sesión activa del servidor y finalizando el acceso a recursos protegidos. Tras cerrar sesión, el usuario será redirigido a la pantalla home del sistema.
- **Rol:** Usuario autenticado.
- **Entradas esperadas:**
 - Solicitud de cierre de sesión (por clic en botón de cerrar sesión o inactividad).

SECCIÓN 2: GESTIÓN DE PROPIEDADES

RF07. Visualización de propiedades

- **Rol:** Usuario autenticado o no autenticado.
- **Descripción:** El sistema permitirá a los usuarios ver las propiedades disponibles en la plataforma con información básica (precio, disponibilidad).
- **Entradas esperadas:**
 - Parámetros de consulta (ubicación, tipo, orden, precio).

RF08. Detalles de propiedad

- **Descripción:** El sistema permitirá seleccionar una propiedad, se mostrará su información, incluyendo galería (extendida), descripción, precio, ubicación, subcategorías y reseñas.
- **Rol:** Usuario autenticado o no autenticado.
- **Entradas esperadas:**
 - Identificador de la propiedad.

RF09. Creación de propiedades

- **Descripción:** El sistema permitirá a los usuarios con los permisos adecuados registrar una nueva propiedad para rentar.
- **Rol:** Usuario con rol de propietario.
- **Entradas esperadas:**
 - Datos de la propiedad: título, descripción, tipo, categoría, ubicación, reglas, fotos, precio, título de la propiedad (o documento que valide al propietario, solo para validación).

RF10. Edición y eliminación de propiedades

- **Descripción:** El sistema permitirá a los propietarios o administradores modificar o eliminar propiedades previamente publicadas (únicamente las de su pertenencia).
- **Rol:** Usuario con rol de propietario o administrador.
- **Entradas esperadas:**
 - Identificador de la propiedad, nuevos datos.

RF11. Publicación programada de propiedades

- **Descripción:** El sistema permitirá que los propietarios seleccionen una fecha futura para la publicación automática de sus propiedades.
- **Rol:** Usuario con rol propietario.
- **Entradas esperadas:**
 - Datos de la propiedad y fecha programada.

RF12. Revisión/moderación de propiedades por administrador

- **Descripción:** El sistema permitirá que las propiedades registradas sean revisadas por un administrador antes de ser visibles en la plataforma.
- **Rol:** Usuario con rol Administrador.
- **Entradas esperadas:**
 - Lista de propiedades en estado “pendiente”.
- **Validaciones:**

SECCIÓN 3: BÚSQUEDA Y EXPLORACIÓN

RF13. Búsqueda de propiedades

- **Descripción:** El sistema permitirá a los usuarios buscar propiedades mediante filtros avanzados (precio, tipo, ubicación, categoría).
- **Rol:** Usuario autenticado o no autenticado.
- **Entradas esperadas:**
 - Términos de búsqueda.
 - Parámetros de filtrado.

RF14. Búsqueda en mapa interactivo

- **Descripción:** El sistema permitirá al usuario visualizar los resultados de búsqueda geográficamente sobre un mapa interactivo.
- **Rol:** Usuario autenticado o no autenticado.
- **Entradas esperadas:**
 - Ubicación base.
 - Criterios de búsqueda.

RF15. Búsqueda por cercanía a servicios clave

- **Descripción:** El sistema ofrecerá la funcionalidad de búsqueda con un estilo minimalista el cual filtrará con base en su proximidad a lugares de interés (universidades, transporte, hospitales, farmacias, supermercados, tiendas).
- **Rol:** Usuario autenticado o no autenticado.
- **Entradas esperadas:**
 - Tipo de servicio clave.
 - Radio o distancia máxima deseada.

SECCIÓN 4: INTERACCIÓN DEL USUARIO

RF16. Favoritos

- **Descripción:** El sistema permitirá a los usuarios marcar propiedades como favoritas para acceder fácilmente a ellas.
- **Rol:** Usuario autenticado.
- **Entradas esperadas:**
 - Identificador de la propiedad.

RF17. Solicitud de reserva

- **Descripción:** El sistema permitirá a los usuarios autenticados (no propietarios ni administradores) enviar una solicitud para reservar una propiedad.
- **Rol:** Usuario autenticado.
- **Entradas esperadas:**
 - Propiedad seleccionada.
 - Fecha deseada y observaciones (opcional).

RF18. Agendamiento de citas

- **Descripción:** El sistema permitirá a los usuarios agendar visitas presenciales o virtuales con los propietarios.
- **Rol:** Usuario autenticado.
- **Entradas esperadas:**
 - Fecha, hora, tipo de cita (presencial o virtual).

RF19. Reserva de propiedad

- **Descripción:** El sistema permitirá a los usuarios reservar temporalmente una propiedad mediante pago anticipado.
- **Rol:** Usuario autenticado.
- **Entradas esperadas:**
 - Selección de propiedad.
 - Medio de pago.

RF20. Mensajería interna

- **Descripción:** El sistema permitirá a los usuarios enviar mensajes a los propietarios directamente desde la plataforma y viceversa atendiendo al modelo de negocio y las mebresías.
- **Rol:** Usuario autenticado y usuario con rol de administrador.
- **Entradas esperadas:**
 - Texto del mensaje.
 - Usuario destinatario (propietario).

RF21. Calificación y reseña

- **Descripción:** El sistema permitirá a los usuarios calificar y dejar reseñas después de completar una transacción.
- **Rol:** Usuario autenticado.
- **Entradas esperadas:**
 - Valoración (número de estrellas).
 - Comentario textual (opcional).

RF22. Reporte de usuarios o publicaciones

- **Descripción:** El sistema permitirá a los usuarios reportar publicaciones sospechosas o usuarios con comportamiento inapropiado mediante un formulario.
- **Rol:** Usuario autenticado.
- **Entradas esperadas:**
 - Motivo del reporte.
 - ID del usuario o propiedad reportada.

RF23. Notificaciones internas y por correo

- **Descripción:** El sistema permitirá enviar notificaciones relevantes dentro de la aplicación y por medio de correo electrónico.
- **Rol:** Usuario autenticado.
- **Entradas esperadas:**
 - Acciones generadoras (mensajes, reservas, aprobaciones, cancelaciones, denuncias).

SECCIÓN 5: PAGOS, TRANSACCIONES Y FACTURAS

RF24. Opciones de pago

- **Descripción:** El sistema permitirá a los usuarios realizar pagos mediante la pasarela de pago de Mercado Pago.
- **Rol:** Usuario autenticado.
- **Entradas esperadas:**
 - Selección del método de pago.

RF25. Generación de comprobante

- **Descripción:** El sistema permitirá tras una transacción, generar un comprobante en PDF con los detalles el cuál puede ser descargado.
- **Rol:** Usuario autenticado.
- **Entradas esperadas:**
 - ID de la transacción.

RF26. Historial de rentas

- **Descripción:** El sistema permitirá al usuario consultar su historial de rentas.
- **Rol:** Usuario autenticado.
- **Entradas esperadas:**
 - Acceso al perfil o sección de historial.

RF27. Gestión de reembolsos

- **Descripción:** El sistema permitirá a los usuarios solicitar reembolsos según las políticas establecidas.
- **Rol:** Usuario autenticado.
- **Entradas esperadas:**
 - Motivo del reembolso.
 - Transacción relacionada.

SECCIÓN 6: PERFIL DEL USUARIO

RF28. Gestión de perfil

- **Descripción:** El sistema permitirá a los usuarios actualizar su información personal, foto y preferencias desde su perfil.
- **Rol:** Usuario autenticado.
- **Entradas esperadas:**
 - Datos personales editables.
 - Este proceso podrá realizarse si el usuario no se registro mediante OAuth.

RF29. Eliminación o desactivación de cuenta

- **Descripción:** El sistema permitirá a los usuarios eliminar o desactivar su cuenta conforme a la política de privacidad.
- **Rol:** Usuario autenticado.
- **Entradas esperadas:**
 - Confirmación de identidad.
 - Motivo (opcional).

SECCIÓN 7: PANEL DE ADMINISTRACIÓN

RF30. Panel de administración

- **Descripción:** El sistema permitirá visualizar un panel de administración centralizado para gestionar usuarios, propiedades, pagos y reservas.
- **Rol:** Administrador.
- **Entradas esperadas:**
 - Acceso con credenciales de administrador.

RF31. Gestión de reportes de usuarios

- **Descripción:** El sistema permitirá a los usuarios con rol de administrador revisar reportes realizados por los usuarios sobre contenidos o comportamientos inadecuados.
- **Rol:** Usuario con rol administrador.
- **Entradas esperadas:**
 - Lista de reportes con detalles (usuario, motivo, fecha).

RF32. Estadísticas por periodo de tiempo

- **Descripción:** El sistema permitirá visualizar un panel de administración para estadísticas filtradas por rangos de fechas.
- **Rol:** usuario con rol administrador.
- **Entradas esperadas:**
 - Fechas de inicio y fin.
 - Categorías de métricas (usuarios activos, ingresos, publicaciones).

RF33. Exportación de datos y reportes

- **Descripción:** El sistema permitirá exportar reportes administrativos en formatos como XML, JSON, CSV o PDF.
- **Rol:** Usuario con rol administrador.
- **Entradas esperadas:**
 - Selección del tipo de reporte y formato deseado.

Cuadro 5: Entradas, validaciones y comportamientos.

ID	Nombre	Entradas esperadas	Validaciones	Comportamiento del sistema
RF01	Registro de usuarios	Nombre, correo, contraseña, OAuth scopes	Formato de correo, contraseña de más de 8 caracteres, campos obligatorios	Valida datos, crea cuenta, envía correo de verificación
RF02	Inicio de sesión	Correo electrónico, contraseña o token de acceso	Correo registrado, contraseña válida, campos completos	Verifica autenticidad, genera token, inicia sesión, redirige según rol
RF03	Recuperación de contraseña	Correo electrónico	Correo debe estar registrado	Genera token, envía correo con enlace de restablecimiento
RF04	Gestión de roles y permisos	Acceso al sistema	Validación de credenciales	Determina acceso según rol y permisos
RF05	Verificación de identidad del propietario	Identificación oficial, título de propiedad	Documentos válidos y correspondientes	Valida autenticidad y permite publicación si es verificado
RF06	Cierre de sesión seguro	Solicitud de cierre de sesión	Debe haber sesión activa	Invalida sesión, destruye token, redirige a pantalla inicial

ID	Nombre	Entradas esperadas	Validaciones	Comportamiento del sistema
RF07	Visualización de propiedades	Parámetros de consulta	Existencia de propiedades	Muestra listado con paginación o scroll infinito
RF08	Detalles de propiedad	Identificador de propiedad	Existencia y estado activo	Muestra galería, descripción, precio y opciones de contacto
RF09	Creación de propiedades	Datos de propiedad, título legal, fotos, reglas	Campos obligatorios, tipo de datos, peso/formato de imagen	Registra propiedad como activa o pendiente
RF10	Edición y eliminación de propiedades	ID de propiedad, nuevos datos	Validación de permisos (admin/propietario)	Permite editar/eliminar y notifica si aplica
RF11	Publicación programada	Datos de propiedad y fecha futura	Fecha debe ser futura	Programa publicación automática en fecha indicada
RF12	Revisión por administrador	Lista de propiedades pendientes	Solo admin puede aprobar/rechazar	Muestra lista y permite decisión con notificación al propietario
RF13	Búsqueda de propiedades	Términos, filtros avanzados	Filtros válidos y combinables	Resultados paginados/autocompletado

ID	Nombre	Entradas esperadas	Validaciones	Comportamiento del sistema
RF14	Búsqueda en mapa interactivo	Ubicación base, criterios	Coordenadas y rango válidos	Muestra propiedades geolocalizadas como pines
RF15	Búsqueda por cercanía a servicios	Tipo de servicio clave, distancia	Existencia del lugar de interés y validez del radio	Filtra por cercanía, destaca visualmente
RF16	Favoritos	Identificador de propiedad	Evitar duplicados	Agrega/quita favoritos, muestra lista en perfil
RF17	Solicitud de reserva	Propiedad, fecha, observaciones	Disponibilidad de propiedad	Envía solicitud, confirma visual y por correo
RF18	Agendamiento de citas	Fecha, hora, tipo de citas	Disponibilidad del propietario	Registra cita, permite cancelar/reprogramar
RF19	Reserva de propiedad	Selección de propiedad, medio de pago	Disponibilidad y autenticidad del pago	Confirma reserva y notifica al propietario
RF20	Mensajería interna	Texto del mensaje, usuario destino	Longitud del mensaje	Guarda mensajes, muestra conversación cronológica
RF21	Calificación y reseña	Valoración, comentario	Una reseña por transacción, modificable	Registra reseña, la muestra públicamente

ID	Nombre	Entradas esperadas	Validaciones	Comportamiento del sistema
RF22	Reporte de usuarios o publicaciones	Motivo, ID de usuario o propiedad	Evitar duplicados	Registra y notifica a administrador
RF23	Notificaciones internas y por correo	Acciones generadoras (mensajes, reservas, etc.)	Preferencias de notificación activas	Notifica en panel y por correo
RF24	Opciones de pago	Método de pago (Mercado Pago, transferencia)	Validar método y fondos	Procesa pago, genera comprobante
RF25	Generación de comprobante	ID de transacción	Usuario debe ser parte de la operación	Genera PDF, lo envía por correo y habilita descarga
RF26	Historial de rentas	Acceso desde perfil	Autenticación válida	Muestra transacciones con comprobantes
RF27	Gestión de reembolsos	Motivo, transacción	Dentro del plazo permitido	Registra solicitud y notifica para revisión
RF28	Gestión de perfil	Datos personales, imagen	Formato y requisitos mínimos	Guarda cambios, confirma notificación
RF29	Eliminación/desactivación de cuenta	Confirmación, motivo (opcional)	Confirmación doble, autenticación válida	Ejecuta acción, ofrece cancelación en 30 días

ID	Nombre	Entradas esperadas	Validaciones	Comportamiento del sistema
RF30	Panel de administración	Credenciales de administrador	Validar rol antes de permitir acceso	Muestra panel, permite gestión CRUD y métricas
RF31	Gestión de reportes de usuarios	Lista de reportes	Verificar si fue resuelto antes	Permite sanciones y registra historial
RF32	Estadísticas por periodo	Fechas, categorías métricas	Fechas en formato válido	Muestra gráficos, permite exportación
RF33	Exportación de datos y reportes	Tipo de reporte y formato	Permiso de exportación activo	Genera archivo, audita la acción

8. Requisitos No Funcionales

RNF01. Compatibilidad del Sistema

- **Descripción:** El sistema debe ser compatible y ejecutarse correctamente en plataformas basadas en Linux, Mac y Windows, garantizando su funcionamiento en versiones ampliamente usadas.
- **Criterios de aceptación:**
 - El sistema debe funcionar correctamente en distribuciones Linux modernas.
 - El sistema debe funcionar correctamente en versiones recientes de Windows.
 - El sistema debe funcionar correctamente en versiones recientes de Mac.

RNF02. Seguridad del Sistema

- **Descripción:** El sistema debe contar con mecanismos de seguridad que impidan el acceso a información crítica a usuarios no autorizados. Se deben implementar medidas de autenticación y autorización para proteger los datos.
- **Criterios de aceptación:**
 - Implementar autenticación segura para todos los usuarios.
 - Restringir el acceso a información sensible mediante autorización basada en roles.
 - Cifrado de contraseñas y datos sensibles almacenados.

RNF03. Arquitectura y Tecnología

- **Descripción:** El sistema debe desarrollarse con una arquitectura moderna que permita la separación clara entre lógica de negocio y presentación, utilizando frameworks actualizados y eficientes.
- **Criterios de aceptación:**
 - Backend implementado con un framework robusto y escalable.
 - Frontend compatible y bien integrado con el backend.
 - Comunicación eficiente y segura entre las capas.

RNF04. Disponibilidad del Sistema

- **Descripción:** El sistema debe estar disponible y operativo las 24 horas del día, los 7 días de la semana.
- **Criterios de aceptación:**
 - Disponibilidad mínima mensual del 99.9%.
 - Plan de contingencia activo ante caídas o fallos.

RNF05. Rendimiento del Sistema

- **Descripción:** El sistema debe ser capaz de procesar grandes volúmenes de transacciones manteniendo tiempos de respuesta óptimos.
- **Criterios de aceptación:**
 - Tiempo de respuesta promedio menor a 2 segundos.
 - Soporte para múltiples usuarios concurrentes sin pérdida de datos.

RNF06. Responsividad y Adaptabilidad

- **Descripción:** El sistema debe ajustarse automáticamente a diferentes tamaños de pantalla y resoluciones.
- **Criterios de aceptación:**
 - Interfaz usable en smartphones, tablets y desktops.
 - Diseño responsivo y consistente en todos los dispositivos.

RNF07. Mantenibilidad

- **Descripción:** El sistema debe estar bien documentado, con código claro y comentado, y soporte para monitoreo y diagnóstico.
- **Criterios de aceptación:**
 - Documentación técnica actualizada.
 - Comentarios en al menos el 80 % del código.

RNF08. Escalabilidad del Sistema

- **Descripción:** El sistema debe poder adaptarse al crecimiento del modelo de negocio, número de usuarios, propiedades y transacciones sin degradar su rendimiento.
- **Criterios de aceptación:**
 - Arquitectura preparada para escalado horizontal y vertical.
 - Soporte para aumento de cargas sin rediseño profundo.

RNF09. Accesibilidad

- **Descripción:** El sistema debe cumplir con estándares de accesibilidad para que cualquier usuario, incluidas personas con discapacidad, pueda utilizarlo.
- **Criterios de aceptación:**
 - Compatibilidad con lectores de pantalla y navegación por teclado.
 - Contraste adecuado y etiquetas descriptivas en formularios.

RNF10. Usabilidad

- **Descripción:** La interfaz debe ser fácil de entender y usar, con flujos de usuario claros y retroalimentación inmediata.
- **Criterios de aceptación:**
 - Menús y botones intuitivos.
 - Asistencia o guías dentro de la plataforma.

RNF11. Portabilidad

- **Descripción:** El sistema debe poder desplegarse fácilmente en distintos entornos sin modificaciones significativas.
- **Criterios de aceptación:**
 - Soporte para contenedores o scripts de despliegue automático.
 - Compatibilidad con servicios en la nube o servidores locales.

RNF12. Auditoría y Registro de Actividades

- **Descripción:** El sistema debe registrar acciones relevantes (logins, errores, transacciones) para propósitos de auditoría y seguridad.
- **Criterios de aceptación:**
 - Registro de eventos con fecha, usuario y tipo de acción.
 - Visualización o exportación de logs desde el panel administrativo.

RNF13. Localización e Internacionalización

- **Descripción:** El sistema debe ser capaz de adaptarse a distintos idiomas y configuraciones regionales.
- **Criterios de aceptación:**
 - Soporte para múltiples idiomas desde el frontend.
 - Formatos de fecha, moneda y dirección adaptables por región.

9. Análisis de Componentes del Sistema *Casa Fácil*

9.1. Cliente Móvil/Web

Función

Interfaz principal para usuarios (*inquilinos, propietarios, administradores*).

Permite:

- Búsqueda de propiedades con filtros avanzados (RF13, RF14).
- Gestión de perfiles, reservas y mensajería interna (RF16, RF20).
- Visualización de mapas interactivos (RF14) y detalles de propiedades (RF08).

Tecnologías: Angular 19 (frontend) para una SPA responsiva (RNF06).

Comunicación

- Se conecta al Backend mediante APIs REST/HTTP (RF24).
- Consume servicios externos:
 - **MapLibre** para geolocalización (RF14).
 - **MercadoPago** para transacciones (RF24).

9.2. Backend

Función

Procesa la lógica de negocio y gestiona los datos.

Incluye:

- Autenticación (JWT/OAuth) y autorización por roles (RF01, RF04).
- Gestión de propiedades (CRUD, validación) (RF09, RF10).
- Procesamiento de pagos (RF24) y generación de comprobantes (RF25).
- Mensajería en tiempo real con Socket.IO (RF20).

Tecnologías: Django 4.2 (Python) para APIs REST (RNF03).

Comunicación

- API Gateway (Nginx + Django) centraliza solicitudes del cliente.
- Interactúa con:
 - Bases de datos (PostgreSQL, MongoDB, Redis).
 - Servicios externos (MercadoPago, MapLibre).

9.3. Base de Datos

Función

Almacenamiento estructurado y no estructurado.

- **PostgreSQL:** Datos relacionales (usuarios, propiedades, transacciones).
- **MongoDB:** Datos no estructurados (mensajes de chat, logs).
- **Redis:** Caché y sesiones (RNF05).

Comunicación

El Backend accede mediante ORM de Django o drivers específicos.

9.4. API de Pagos

Función

Procesa transacciones seguras (RF24) y gestiona reembolsos (RF27).

- Integración con MercadoPago como pasarela externa.

Comunicación

El Backend envía solicitudes de pago (HTTPS) y recibe confirmaciones.

9.5. Servicios de Mapas

Función

Geolocalización de propiedades y búsqueda por cercanía (RF14, RF15).

- **Tecnología:** MapLibre GL JS (open-source).

Comunicación

- El cliente consume directamente la API de mapas (JavaScript).
- El Backend valida coordenadas y direcciones.

9.6. Diagrama de Comunicación

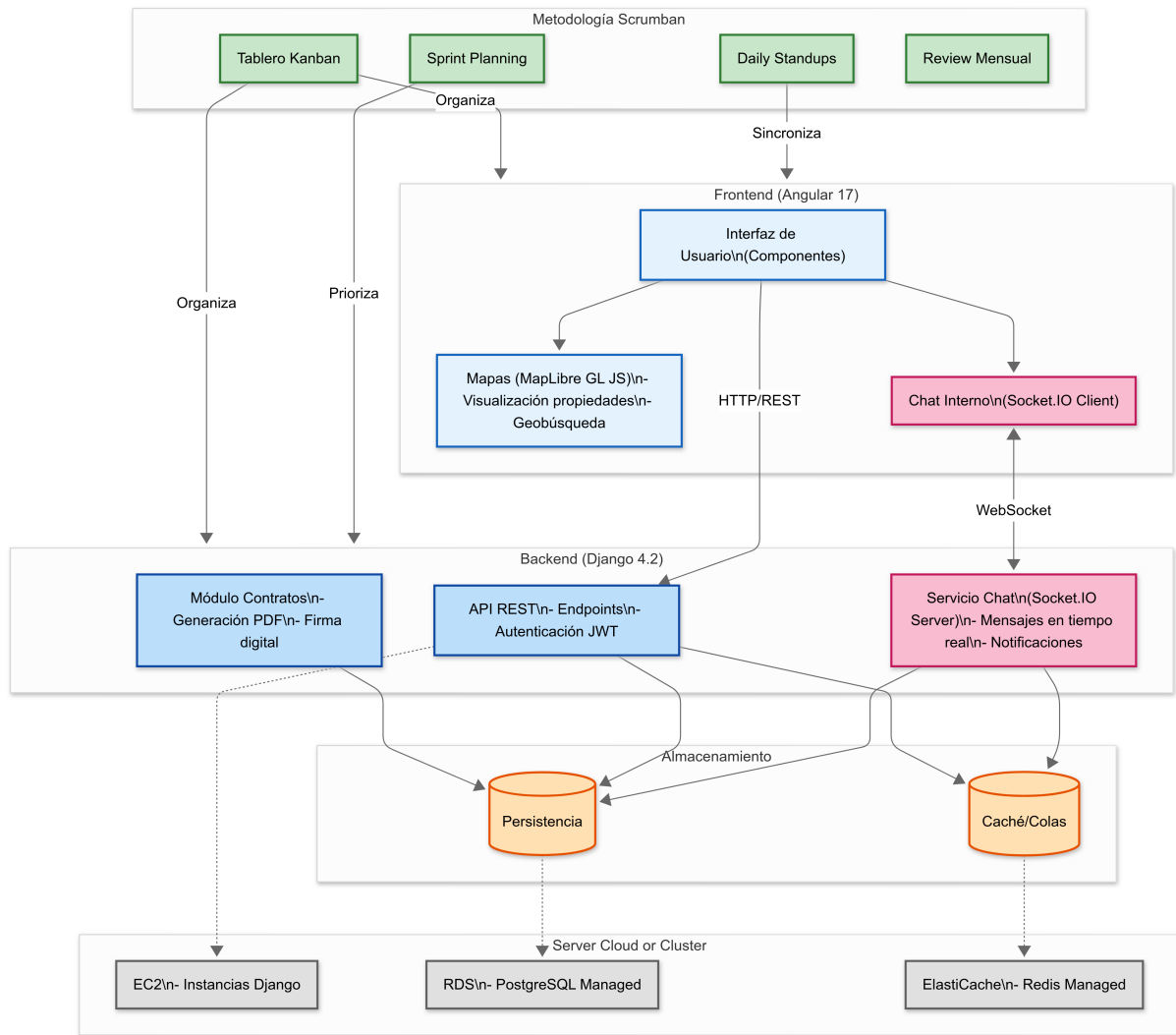


Figura 1. Diagrama de Comunicación.

10. Estilos Arquitectónicos

10.1. Microservicios

El estilo arquitectónico basado en **microservicios** consiste en dividir una aplicación en múltiples servicios pequeños, autónomos y especializados, que se comunican entre sí generalmente a través de APIs. Esto permite que cada uno de estos servicios pueda desarrollarse, implementarse, operarse y actualizarse sin afectar el funcionamiento de otros servicios del sistema. Una aplicación basada en microservicios facilita el mantenimiento y testeado de cada servicio, permitiendo implementar cambios y revertirlos en caso de ser necesario. Para una plataforma como *Casa Fácil*, donde el sistema requiere diversas funcionalidades que pueden operar de forma autónoma, la arquitectura de microservicios permite dividir el sistema en módulos especializados, tales como:

- **Servicio de cuentas:** Gestión de usuarios, roles, autenticación y perfil.
- **Servicio de pago:** Procesamiento de pagos e integración con Mercado Pago.
- **Servicio de carrito de compra:** Manejo de reservas e historial de inmuebles guardados.
- **Servicio de mapas:** Búsqueda geolocalizada y ubicación de propiedades.
- **Servicio de mensajería:** Comunicación entre arrendadores y arrendatarios.

El uso del modelo de microservicios en *Casa Fácil* permite una organización modular, escalable y mantenible del sistema. Además, al combinarse con una **arquitectura orientada a eventos (EDA)**, se potencia aún más la reactividad, flexibilidad y eficiencia del desarrollo, permitiendo que los servicios interactúen de forma desacoplada y fluida, adaptándose al crecimiento futuro del proyecto.

11. Patrones Arquitectónicos

Para complementar el diseño basado en Microservicios y la Arquitectura Orientada a Eventos (EDA), se proponen los siguientes patrones arquitectónicos. Cada uno resuelve necesidades específicas relacionadas con escalabilidad, comunicación entre servicios, separación de responsabilidades y resiliencia.

1. API Gateway

- **Descripción:** Patrón que establece una única puerta de entrada para acceder a múltiples microservicios.
- **Descripción:** Patrón que establece una única puerta de entrada para acceder a múltiples microservicios.
- **Aplicación en Casa Fácil:** El API Gateway (ej. Nginx o Kong) será el primer punto de contacto para el cliente. Centraliza la gestión de autenticación, autorización, enrutamiento de solicitudes y limitación de tráfico.
- **Ventajas:**
 - Oculta la complejidad interna del sistema.
 - Reduce el acoplamiento entre cliente y servicios individuales.
 - Permite control centralizado de seguridad y monitoreo.

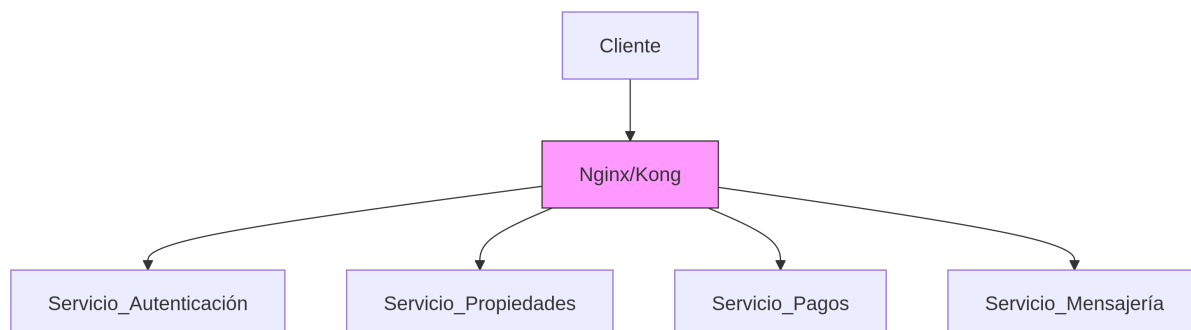


Figura 2. Componentes API Gateway.

2. CQRS (Command Query Responsibility Segregation)

- **Descripción:** Separa las operaciones de lectura (consultas) de las de escritura (comandos), permitiendo optimizar y escalar cada una por separado.
- **Aplicación en Casa Fácil:** En funcionalidades como gestión de reservas o propiedades, el modelo de lectura puede ser distinto y más optimizado que el modelo de escritura, por ejemplo, consultas masivas de propiedades vs. edición de una publicación.
- **Ventajas:**
 - Mejora el rendimiento y escalabilidad.
 - Facilita el uso de diferentes tecnologías para leer y escribir.

- Ideal para sistemas con alto volumen de datos y lógica compleja.

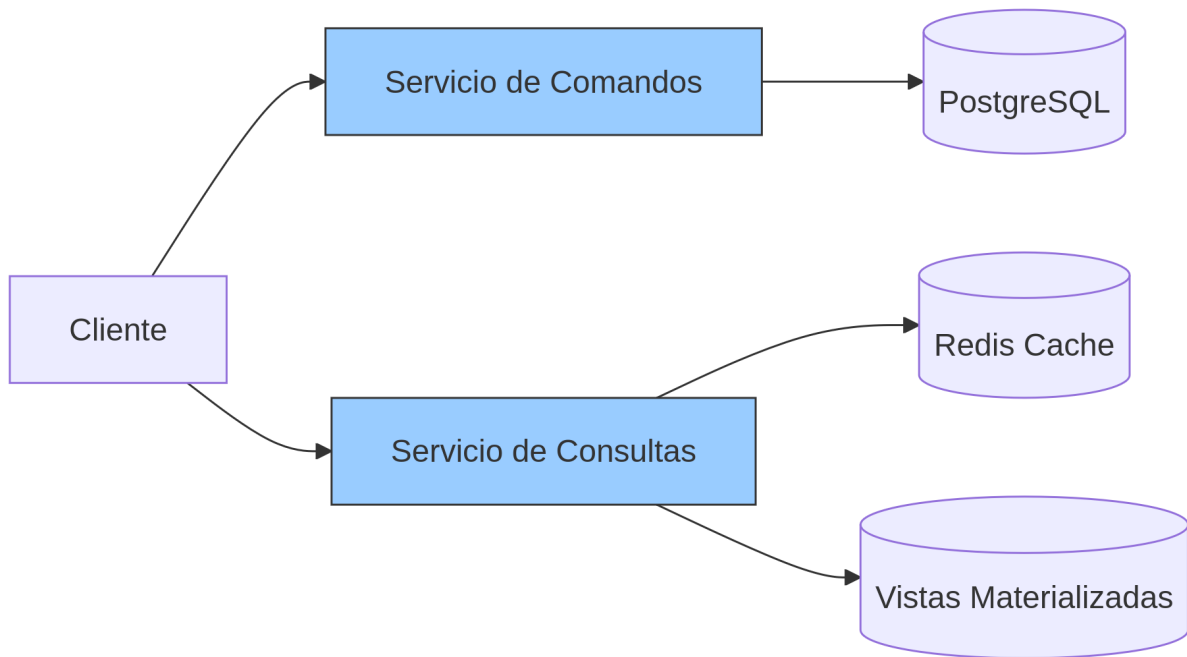


Figura 3. Componentes CQRS.

3. Event Sourcing

- **Descripción:** En lugar de almacenar el estado actual de una entidad, se almacena una secuencia de eventos que modificaron su estado.
- **Aplicación en Casa Fácil:** Para gestionar el historial de reservas, cambios de disponibilidad o pagos, se puede almacenar cada evento como "propiedad reservada", "pago confirmado".
- **Ventajas:**
 - Trazabilidad completa de acciones pasadas.
 - Reversión de cambios y auditoría sencilla.
 - Se complementa naturalmente con EDA.

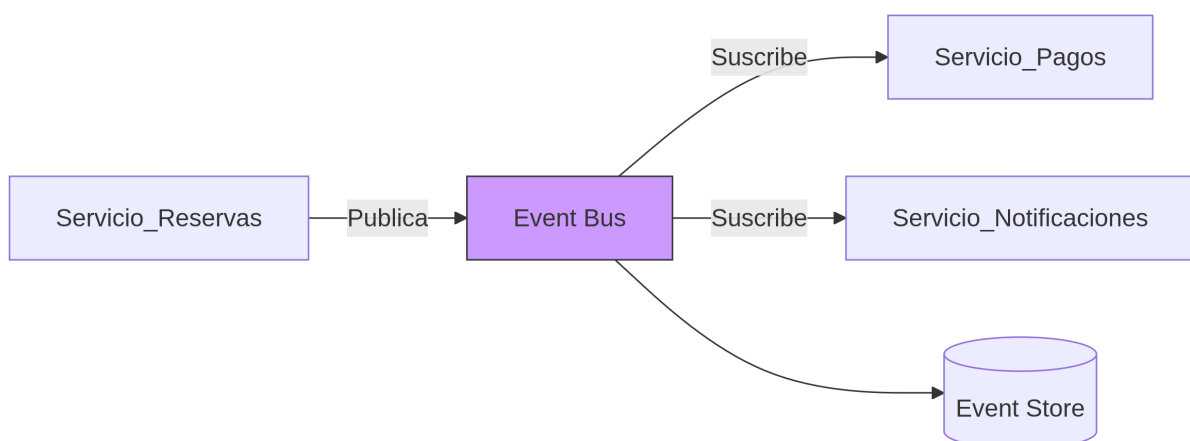


Figura 4. Componentes Event Sourcing.

4. Service Registry & Discovery

- **Descripción:** Permite que los microservicios se registren en un directorio común para ser descubiertos dinámicamente.
- **Aplicación en Casa Fácil:** En entornos con múltiples servicios desplegados dinámicamente (como en contenedores Docker), el registro facilita la comunicación entre servicios sin direcciones IP fijas.
- **Ventajas:**
 - Reducción de la configuración manual.
 - Mejor soporte para escalado automático.
 - Necesario para balanceo de carga dinámico.

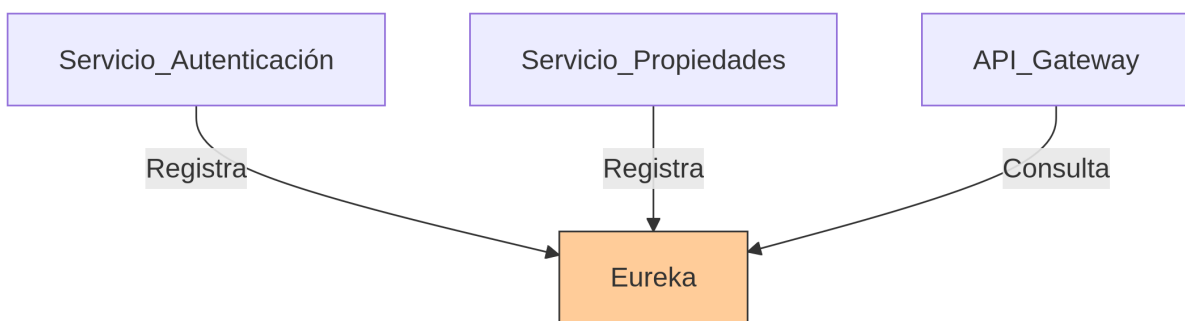


Figura 5. Componentes Service Registry & Discovery.

5. Backend for Frontend (BFF)

- **Descripción:** Crea una capa intermedia específica para cada tipo de cliente (web, móvil, etc.), adaptando las respuestas del backend a sus necesidades.
- **Aplicación en Casa Fácil:** La versión móvil puede tener un BFF que priorice velocidad y respuesta ligera, mientras que la web puede entregar más datos.
- **Ventajas:**
 - Reduce carga y complejidad en el frontend.
 - Mejora la experiencia de usuario según el dispositivo.
 - Facilita la evolución independiente de interfaces.

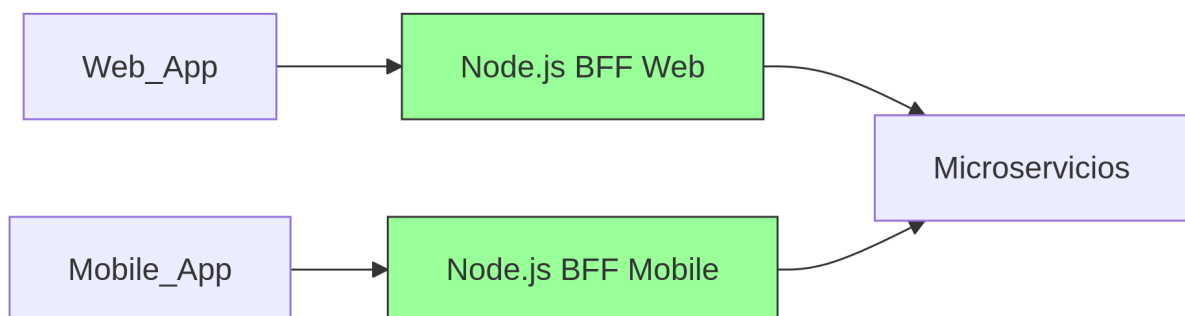


Figura 6. Componentes Backend for Frontend.

6. Saga

- **Descripción:** Coordina transacciones distribuidas entre múltiples servicios, garantizando consistencia eventual a través de una serie de pasos y compensaciones si algo falla.
- **Aplicación en Casa Fácil:** Una reserva de propiedad puede requerir verificar disponibilidad, generar pago y enviar notificación. Si el pago falla, debe deshacerse todo.
- **Ventajas:**
 - Maneja flujos de negocio complejos sin transacciones bloqueantes.
 - Aumenta la confiabilidad en flujos asincrónicos.
 - Escalable y desacoplado.

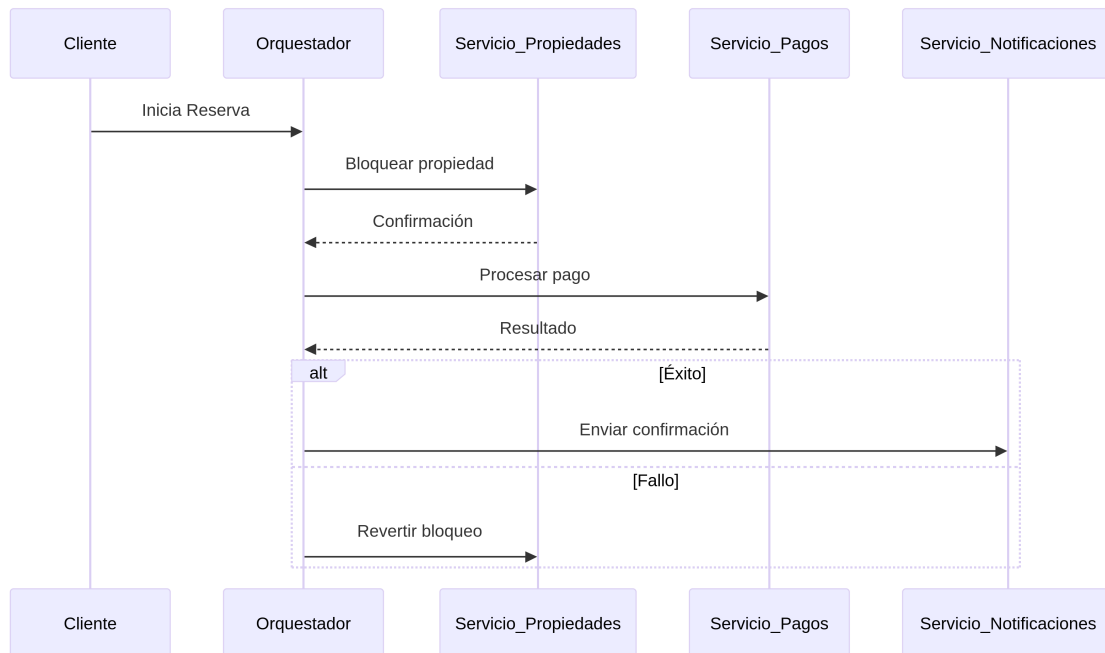


Figura 7. Componentes Saga.

7. Strangler Fig

- **Descripción:** Permite reemplazar gradualmente un sistema monolítico por microservicios. Se encapsula el sistema antiguo y se introducen nuevos componentes uno por uno.
- **Aplicación en Casa Fácil:** Si partes del sistema comienzan siendo monolíticas (como reservas o pagos), pueden migrarse paulatinamente sin interrumpir la operación.
- **Ventajas:**
 - Reducción del riesgo en la migración.
 - Permite evolución controlada del sistema.

- Compatible con despliegue incremental.

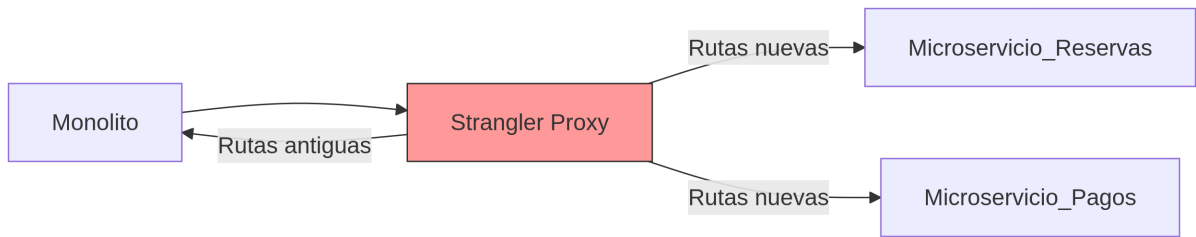


Figura 8. *Componentes Strangler Fig.*

12. Patrones de Diseño

Los patrones de diseño permiten organizar el código de forma flexible, reutilizable y mantenible. A continuación se describen los patrones más relevantes que se aplicarán al sistema Casa Fácil:

1. Strategy Pattern

Este patrón permite definir diferentes algoritmos para una misma tarea y cambiarlos dinámicamente según la necesidad, manteniendo la misma interfaz. **Aplicación:** En la búsqueda de inmuebles, se pueden aplicar múltiples estrategias: por precio, por cercanía a una universidad o por reglas del inmueble (como "no fumar", "sin mascotas"), sin modificar la lógica principal del buscador.

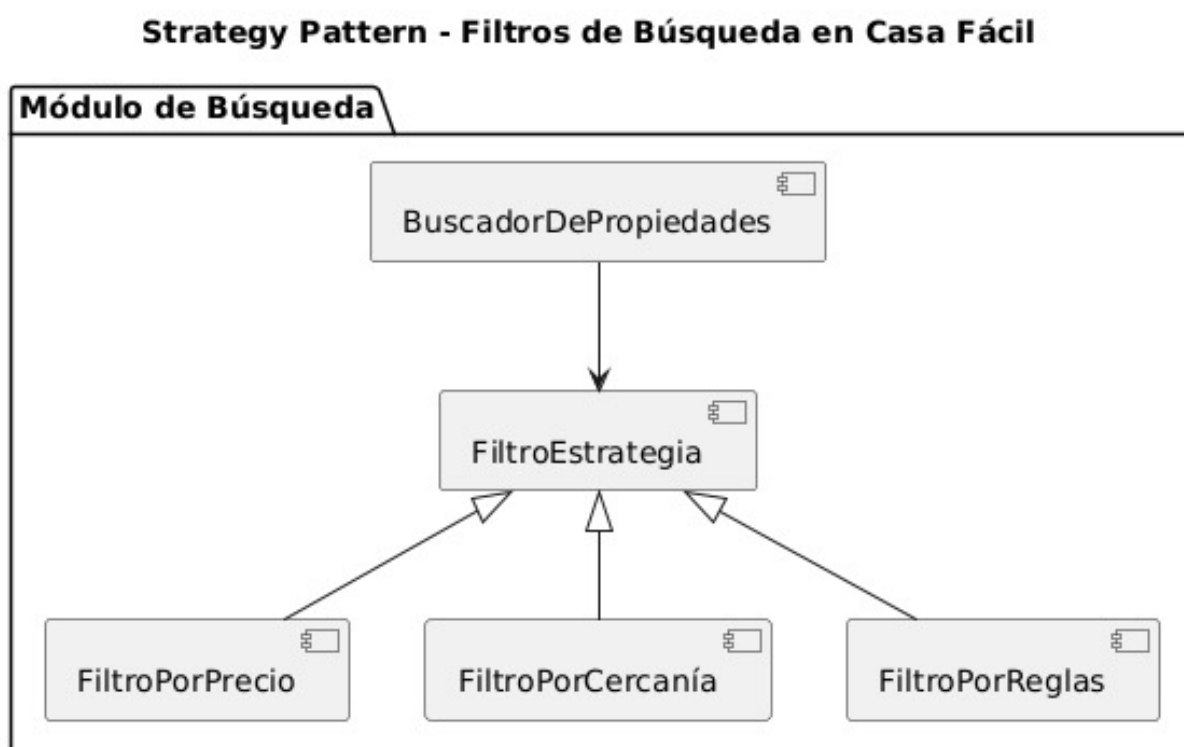


Figura 9. Patrón de diseño Strategy.

2. Observer Pattern

Define una relación de dependencia uno-a-muchos, de modo que cuando un objeto cambia de estado, todos los observadores reciben notificaciones automáticamente. **Aplicación:** Los usuarios que se suscriban a una propiedad recibirán notificaciones automáticas cuando cambie su disponibilidad o precio.

Observer Pattern - Notificaciones a Usuarios en Casa Fácil

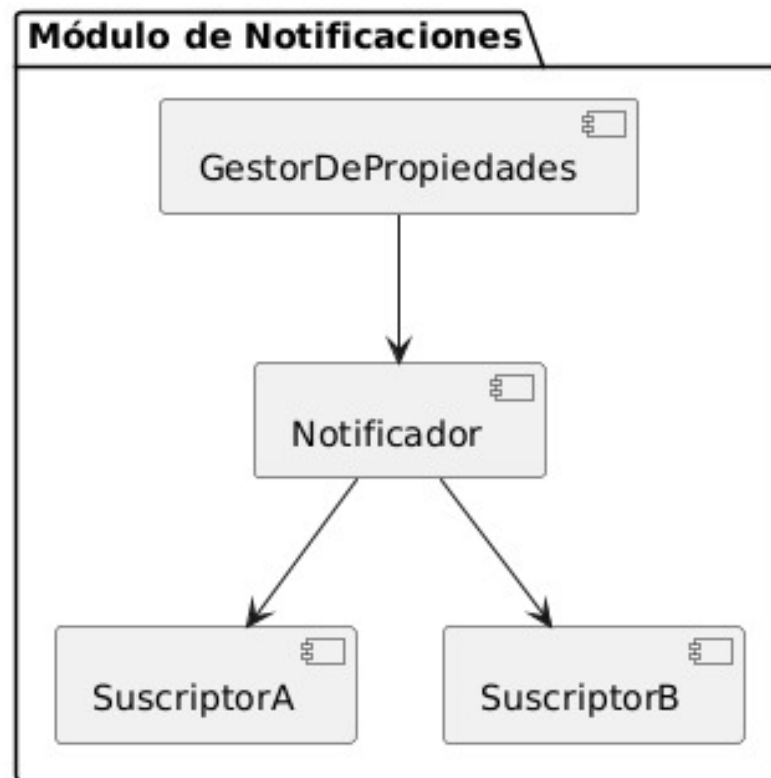


Figura 10. Patrón de diseño Observer.

3. Facade Pattern

Proporciona una interfaz unificada y simplificada a un conjunto de funcionalidades complejas. **Aplicación:** Puede encapsular toda la lógica de reservas, pagos o validaciones en una clase fachada, de modo que el frontend interactúe con un solo punto de entrada, simplificando las integraciones.

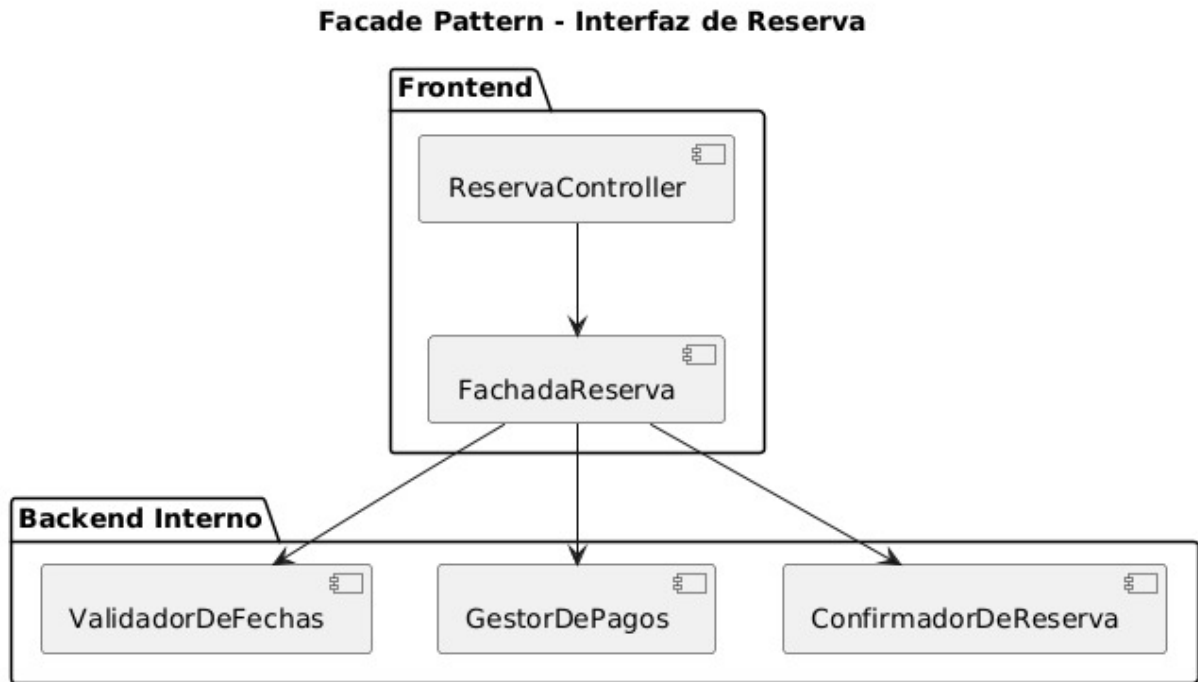


Figura 11. *Patrón de diseño Facade.*

4. Decorator Pattern

Permite extender dinámicamente el comportamiento de un objeto envolviéndolo con nuevos componentes, sin modificar su clase original. **Aplicación:** El sistema de notificaciones puede comenzar con envíos por correo electrónico, y luego extenderse fácilmente a SMS o WhatsApp mediante decoradores.

Decorator Pattern - Sistema de Notificaciones

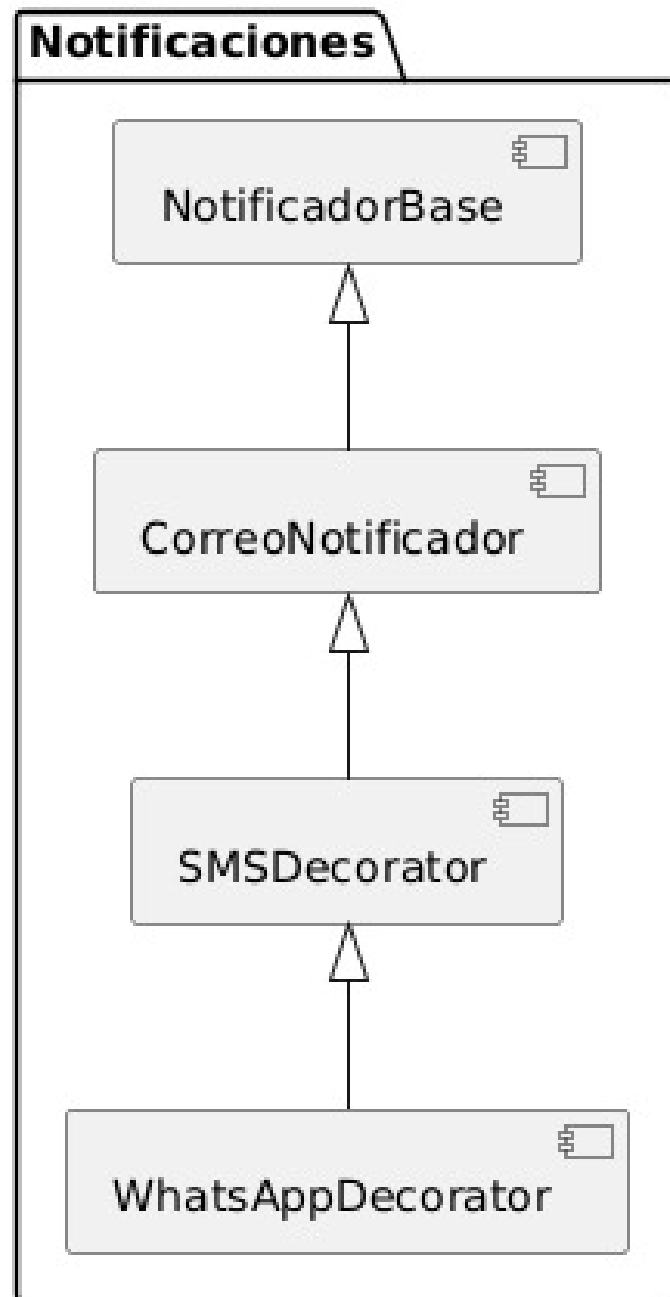


Figura 12. Patrón de diseño Decorator.

5. Singleton Pattern

Garantiza que una clase tenga una única instancia durante todo el ciclo de vida del sistema, y proporciona un acceso global a ella. **Aplicación:** Puede utilizarse para gestionar una única instancia del sistema de configuración o autenticación del sistema.

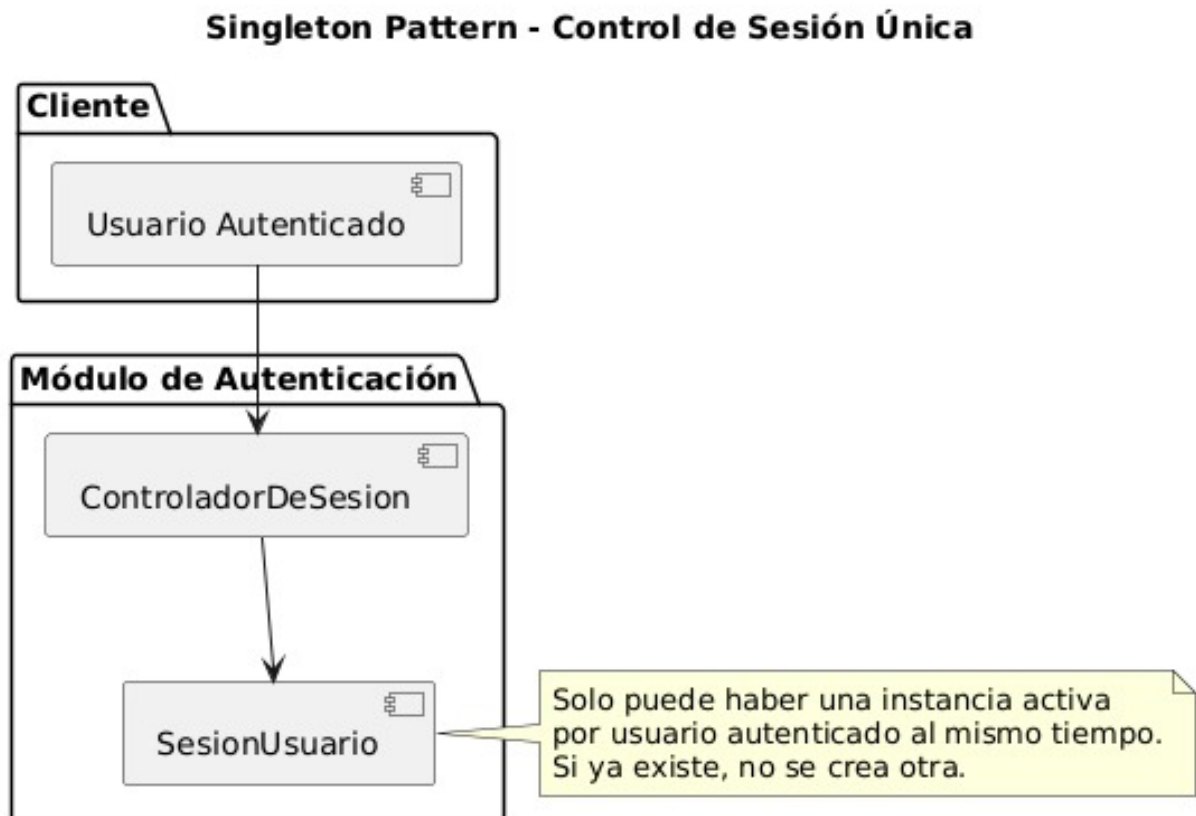


Figura 13. Patrón de diseño Singleton.

6. Factory Pattern

Permite crear objetos sin exponer su lógica de creación al cliente, utilizando una interfaz común. **Aplicación:** Se puede usar para crear diferentes tipos de usuarios (inquilino, propietario, administrador), propiedades o métodos de pago, desde una misma factoría base.

Factory Pattern - Creación de Propiedades

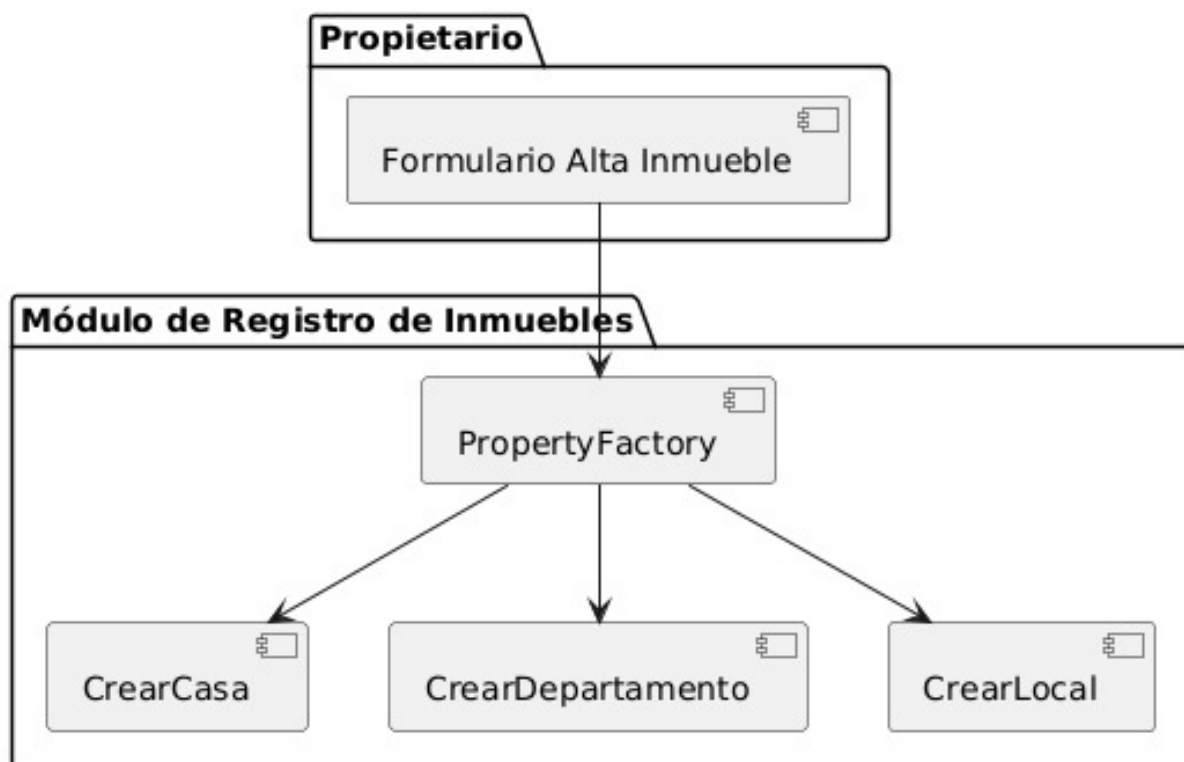


Figura 14. Patrón de diseño Factory.

7. Repository Pattern

Actúa como una capa intermedia entre la lógica de negocio y la base de datos, permitiendo un acceso estructurado y mantenible. **Aplicación:** Se puede aplicar para generar repositorios reutilizables de entidades como propiedades, usuarios o mensajes.

Repository Pattern - Acceso a Propiedades y Usuarios

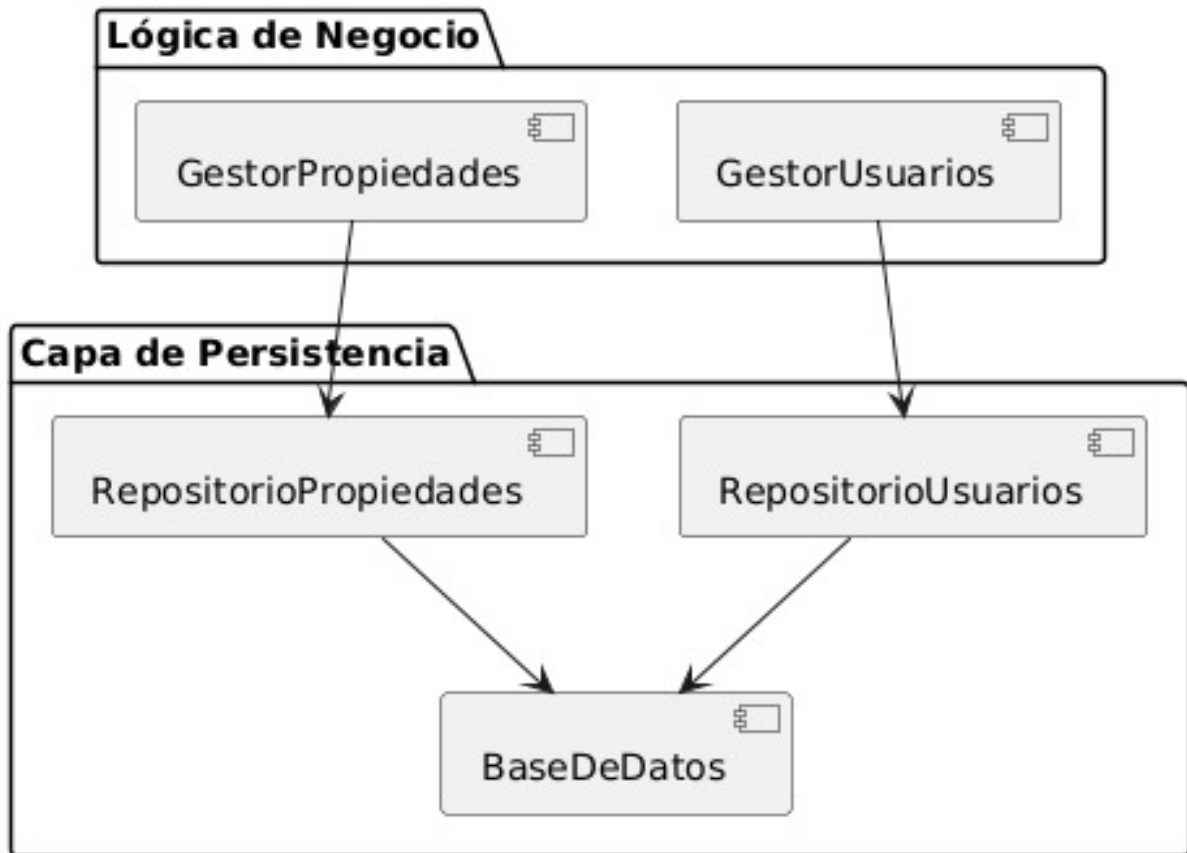


Figura 15. Patrón de diseño Repository.

8. Composite Pattern

Permite tratar objetos individuales y estructuras jerárquicas de manera uniforme. **Aplicación:** Es útil para representar jerarquías de reglas (por ejemplo, reglas generales y subreglas por inmueble), o categorías y subcategorías de propiedades (como “Habitaciones” → “Amuebladas” / “Sin amueblar”).

Composite Pattern - Reglas de Propiedad

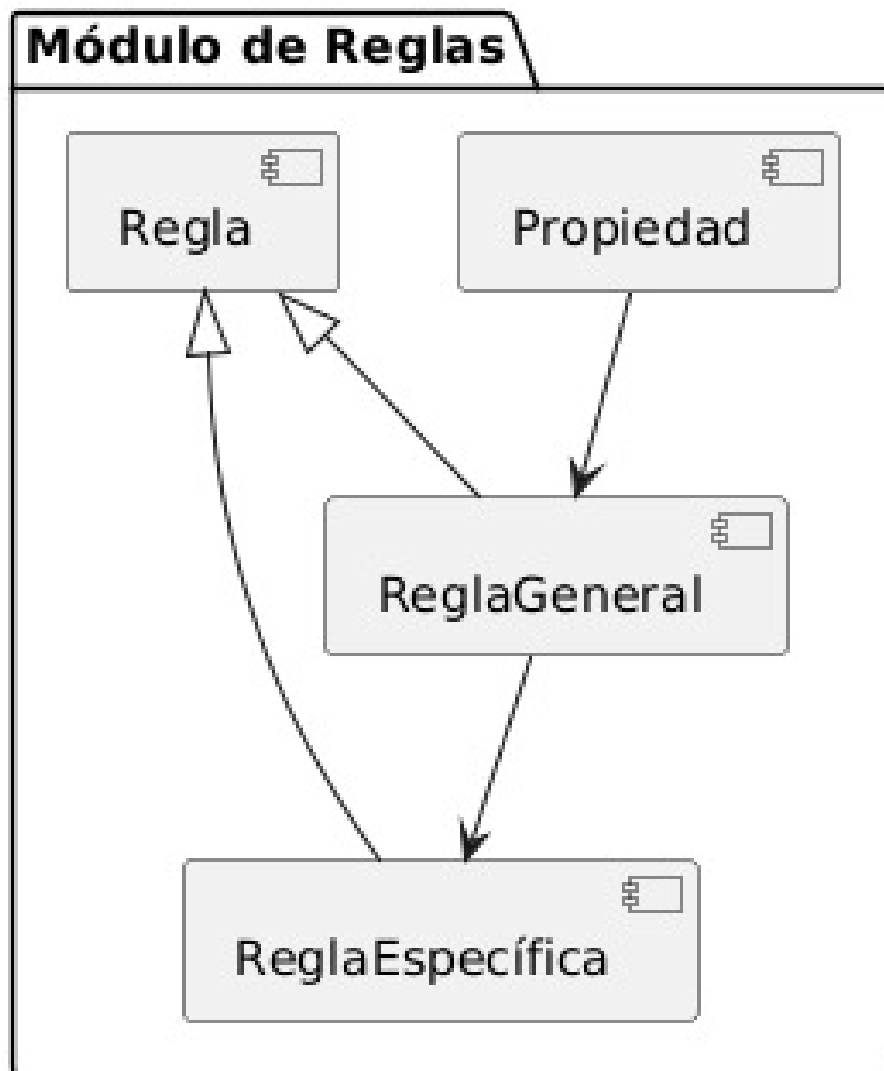


Figura 16. Patrón de diseño Composite.

9. Lazy Loading

Retrasa la carga de recursos hasta que realmente se necesitan. **Aplicación:** Imágenes, videos, mapas o reseñas de propiedades pueden cargarse solo cuando el usuario interactúe con ellos, mejorando así el rendimiento general del sistema.

Lazy Loading - Carga Diferida de Contenido

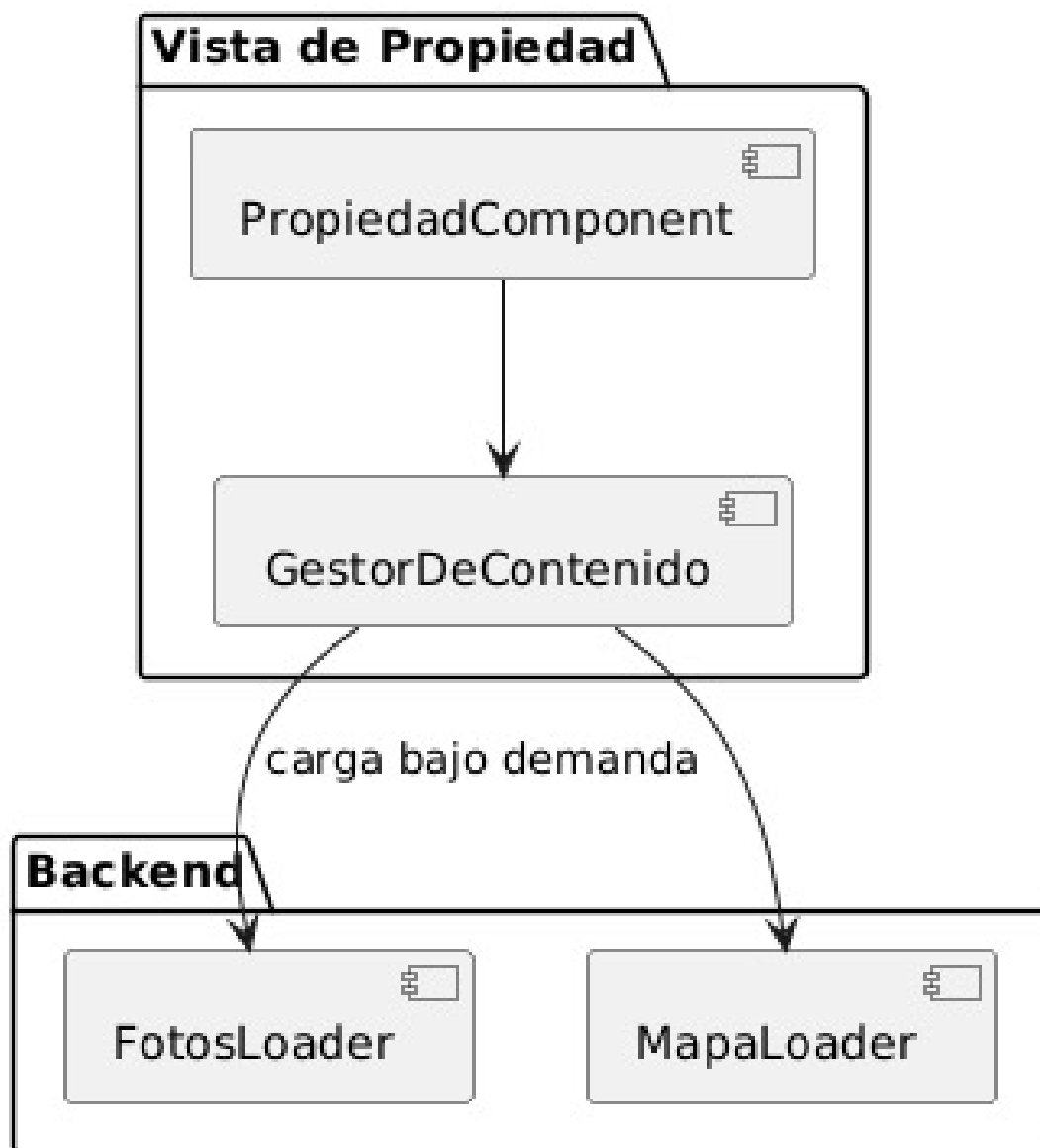


Figura 17. Patrón de diseño Lazy Loading.

10. Principios SOLID

Los principios SOLID permiten crear software más mantenible, modular y fácil de probar. Son esenciales al trabajar con microservicios y arquitecturas escalables como la de **Casa Fácil**.

- **S - Single Responsibility Principle:** Cada clase debe tener una única responsabilidad. En Casa Fácil, por ejemplo, un servicio dedicado solo a manejar pagos o reservas.
- **O - Open/Closed Principle:** El código debe estar abierto a extensión, pero cerrado a modificación. Por ejemplo, al agregar nuevas formas de pago sin alterar el código original, se puede usar el patrón Strategy.
- **L - Liskov Substitution Principle:** Las clases derivadas deben poder sustituir a sus clases base sin afectar el funcionamiento. Por ejemplo, una clase `UsuarioPremium` debe poder sustituir a `Usuario` sin romper la lógica.
- **I - Interface Segregation Principle:** Es mejor tener interfaces pequeñas y específicas que una general demasiado extensa. Un controlador de publicaciones no debería forzarse a implementar métodos de notificación si no los necesita.
- **D - Dependency Inversion Principle:** Las clases deben depender de abstracciones, no de clases concretas. Esto permite inyectar servicios como repositorios o validadores sin acoplarlos al código del controlador.

13. Diagramas

13.1. Diagrama de componenes.

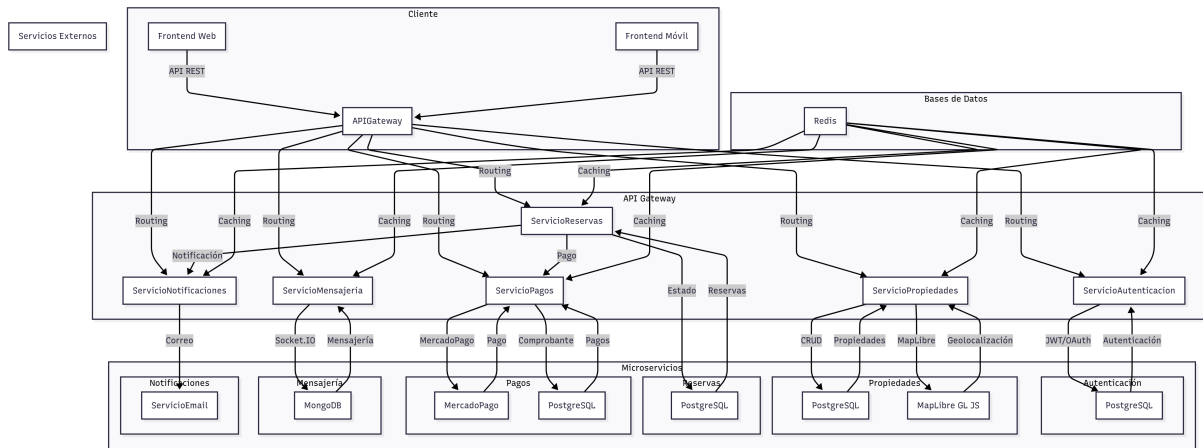


Figura 18. Diagrama de Componentes.

13.2. Diagrama de patrones y estilo.

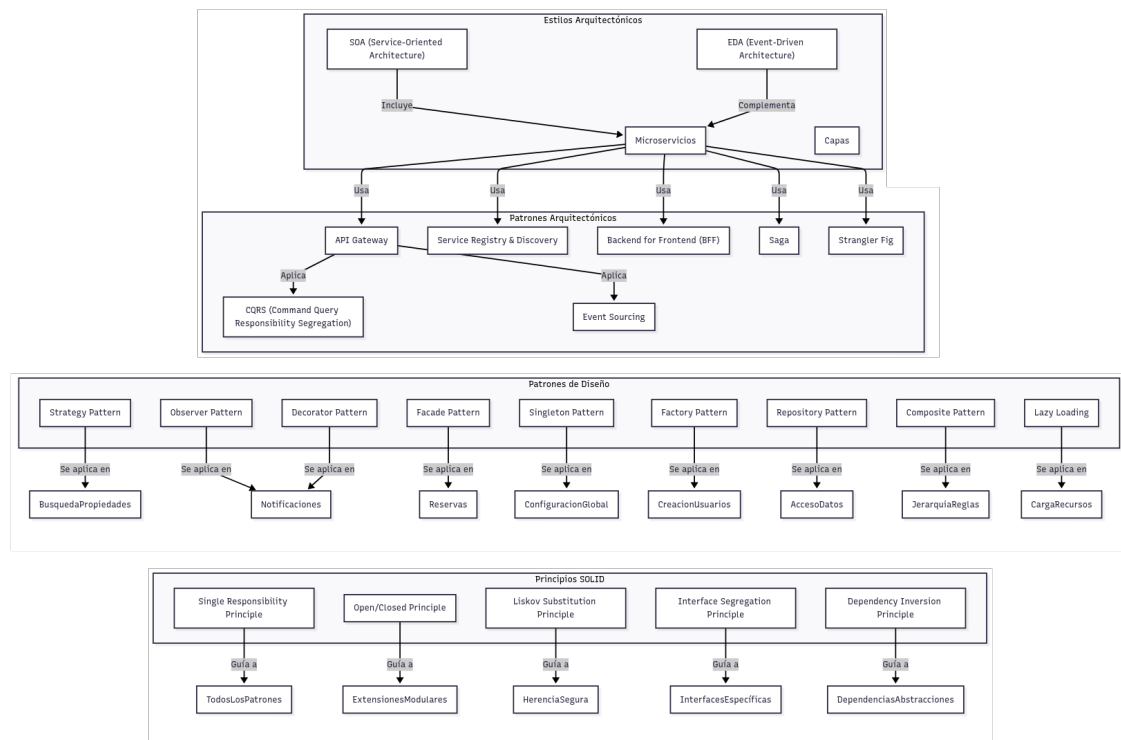


Figura 19. Diagrama de Patrones y Estilo Arquitectónico.

13.3. Diagrama de Clases Inicial con Entidades Principales y Relaciones.

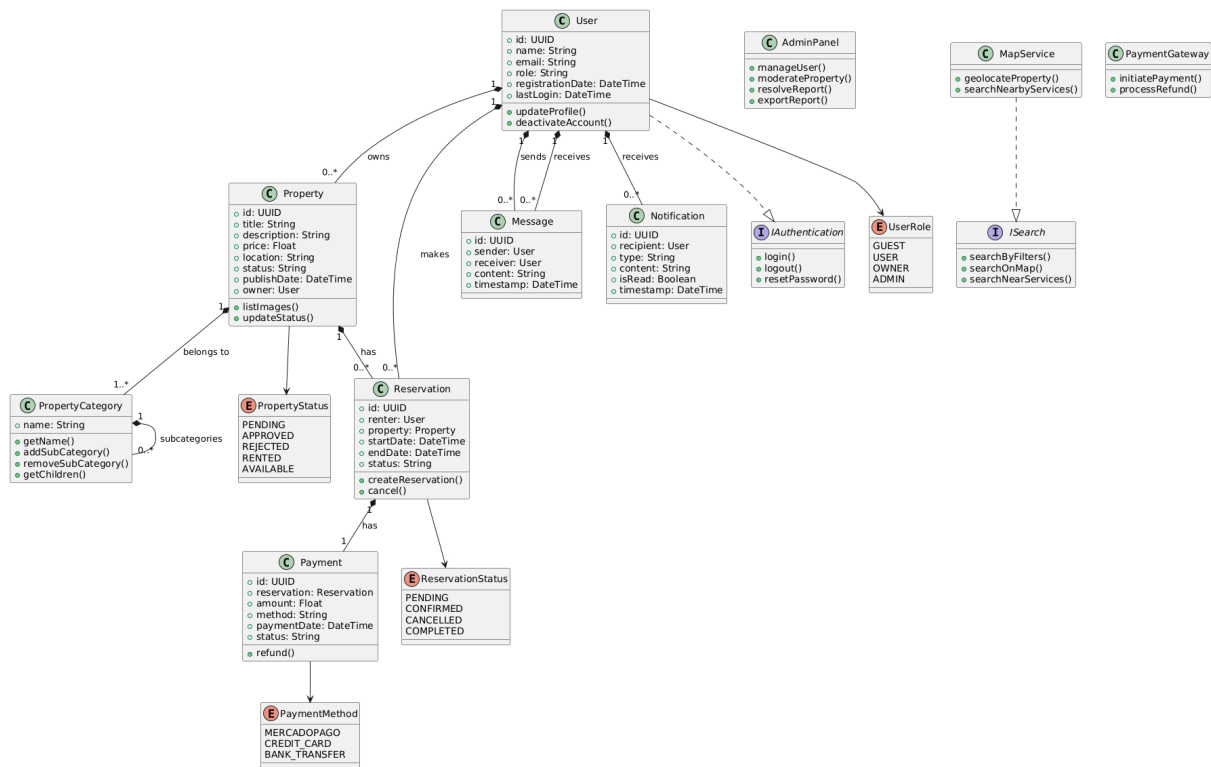


Figura 20. Diagrama de Clases.

14. Resumen del Sistema

El sistema *Casa Fácil* incluye una serie de funcionalidades esenciales que permiten a los usuarios interactuar eficientemente entre sí y con el sistema. A continuación, se detallan todas las funcionalidades clave:

Autenticación y Gestión de Usuarios

- **Registro de usuarios:** Permite a nuevos usuarios registrarse mediante un formulario en línea o mediante OAuth.
- **Inicio de sesión:** Los usuarios pueden iniciar sesión con sus credenciales.
- **Recuperación de contraseña:** Los usuarios pueden recuperar el acceso a su cuenta mediante un enlace enviado a su correo electrónico.
- **Gestión de roles y permisos:** El sistema permite diferenciar funcionalidades según el rol (usuario autenticado, usuario no autenticado, propietario o administrador).
- **Verificación de identidad del propietario:** El sistema debe verificar la identidad del propietario como dueño o administrador del inmueble.
- **Cierre de sesión seguro:** El sistema permite a los usuarios cerrar su sesión de forma segura.
- **Gestión de perfil:** Los usuarios pueden actualizar su información personal, foto y preferencias desde su perfil.
- **Eliminación o desactivación de cuenta:** Los usuarios pueden eliminar o desactivar su cuenta conforme a la política de privacidad.

Gestión de Propiedades

- **Visualización de propiedades:** Los usuarios pueden ver las propiedades disponibles en la plataforma con información básica.
- **Detalles de propiedad:** Al seleccionar una propiedad, se muestran su información adicional, incluyendo galería, descripción, precio, ubicación, subcategorías y reseñas.
- **Creación de propiedades:** Los usuarios con rol propietario pueden registrar nuevas propiedades para rentar.
- **Edición y eliminación de propiedades:** Los propietarios o administradores pueden modificar o eliminar propiedades previamente publicadas.
- **Publicación programada de propiedades:** Los propietarios pueden seleccionar una fecha futura para la publicación automática de sus propiedades.
- **Revisión/moderación de propiedades por administrador:** Las propiedades registradas son revisadas por un administrador antes de ser visibles en la plataforma.

Búsqueda y Exploración

- **Búsqueda de propiedades:** El sistema permite a los usuarios buscar propiedades mediante filtros avanzados (precio, tipo, ubicación, categoría).
- **Búsqueda en mapa interactivo:** Los resultados de búsqueda se visualizan geográficamente sobre un mapa interactivo.
- **Búsqueda por cercanía a servicios clave:** El sistema permite filtrar propiedades según su proximidad a lugares de interés (universidades, transporte, hospitales, etc.).

Interacción del Usuario

- **Favoritos:** Los usuarios pueden marcar propiedades como favoritas para acceder fácilmente a ellas.
- **Solicitud de reserva:** Los usuarios autenticados pueden enviar una solicitud para reservar una propiedad.
- **Agendamiento de citas:** Los usuarios pueden agendar visitas presenciales o virtuales con los propietarios.
- **Reserva de propiedad:** Los usuarios pueden reservar temporalmente una propiedad mediante pago anticipado.
- **Mensajería interna:** Los usuarios pueden enviar mensajes a los propietarios directamente desde la plataforma.
- **Calificación y reseña:** Los usuarios pueden calificar y dejar reseñas después de completar una transacción.
- **Reporte de usuarios o publicaciones:** Los usuarios pueden reportar publicaciones sospechosas o usuarios con comportamiento inadecuado.
- **Notificaciones internas y por correo:** El sistema envía notificaciones relevantes dentro de la aplicación y por medio de correo electrónico.

Pagos, Transacciones y Facturas

- **Opciones de pago:** El sistema permite a los usuarios realizar pagos mediante la pasarela de pago de Mercado Pago.
- **Generación de comprobante:** Tras una transacción exitosa, se genera un comprobante en PDF con los detalles.
- **Historial de rentas:** Los usuarios pueden consultar su historial de rentas.
- **Gestión de reembolsos:** Los usuarios pueden solicitar reembolsos según las políticas establecidas.

Panel de Administración

- **Panel de administración:** El sistema permite a los administradores gestionar usuarios, propiedades, reservas y transacciones.
- **Gestión de reportes de usuarios:** Los administradores pueden revisar reportes realizados por los usuarios sobre contenidos o comportamientos inadecuados.
- **Estadísticas por periodo de tiempo:** El sistema permite visualizar métricas clave como usuarios activos, ingresos y publicaciones.
- **Exportación de datos y reportes:** Los administradores pueden exportar reportes en formatos como CSV, PDF, Excel, etc.

Servicios Complementarios

- **Integración de servicios locales:** El sistema conecta con APIs de mapas y servicios externos para mostrar información sobre lavanderías, estacionamientos, transporte y otros servicios cercanos a las propiedades.

Seguridad y Control de Acceso

- **Control de roles y permisos:** El sistema permite diferenciar funcionalidades según el rol (usuario autenticado, usuario no autenticado, propietario o administrador).
- **Verificación de identidad del propietario:** El sistema debe verificar la identidad del propietario como dueño o administrador del inmueble.
- **Autenticación y autorización:** El sistema gestiona el acceso mediante JWT/OAuth y controla qué funcionalidades puede usar cada usuario según su rol.

Arquitectura y Diseño Técnico

- **Microservicios:** El sistema se divide en múltiples servicios pequeños, autónomos y especializados, como el servicio de cuentas, el servicio de pago, el servicio de mapas, y el servicio de mensajería.
- **API Gateway:** Centraliza las solicitudes del cliente y las enruta a los servicios correspondientes.
- **Base de datos:** Almacena datos relacionales (PostgreSQL) y no estructurados (MongoDB), junto con caché y sesiones (Redis).
- **Event Sourcing:** Almacena eventos en lugar de estados actuales, facilitando la trazabilidad y auditoría.
- **Service Registry & Discovery:** Facilita la comunicación entre microservicios sin necesidad de direcciones IP fijas.
- **Backend for Frontend (BFF):** Crea capas intermedias específicas para cada tipo de cliente (web, móvil), adaptando las respuestas del backend a sus necesidades.

Diseño y Experiencia de Usuario

- **Interfaz intuitiva:** La interfaz debe ser fácil de entender y usar, con flujos de usuario claros y retroalimentación inmediata.
- **Accesibilidad:** El sistema debe cumplir con estándares de accesibilidad para usuarios con discapacidad.
- **Responsividad:** La interfaz debe ajustarse automáticamente a diferentes tamaños de pantalla y resoluciones.
- **Localización e internacionalización:** Soporte para múltiples idiomas y formatos regionales.

Funcionalidades Específicas

- **Chatbot integrado:** Incluye distintos niveles de chatbot (básico, avanzado y premium) para mejorar la interacción con los usuarios y proporcionar información automatizada.
- **Sistema de posicionamiento jerárquico:** Prioriza propiedades según el plan adquirido (estándar, prioritario, destacado).
- **Membresías escalonadas:** Ofrece planes Freemium y Premium con diferentes beneficios y limitaciones.
- **Notificaciones automáticas:** El sistema envía notificaciones relevantes dentro de la aplicación y por medio de correo electrónico.

15. Identificación de atributos de calidad

15.1. Rendimiento (Performance)

Definición

Capacidad del sistema para responder en tiempos aceptables bajo distintas condiciones de carga.

Características claves

- Tiempo de respuesta menor a 2 segundos en operaciones críticas.
- Capacidad para manejar múltiples usuarios concurrentes y búsquedas simultáneas.
- Optimización de imágenes y mapas mediante Lazy Loading.

Evidencia en el sistema

- Arquitectura basada en microservicios permite distribuir la carga.
- Uso de Redis para cacheo de sesiones y consultas frecuentes.
- Separación de comandos y consultas con CQRS mejora eficiencia.

15.2. Disponibilidad

Definición

Tiempo en que el sistema se mantiene operativo.

Características claves

- Alta disponibilidad: 99.9 % operativo al mes.
- Soporte 24/7 por ser un sistema de renta de propiedades.

Evidencia en el sistema

- Arquitectura con posibilidad de escalado automático.
- Sistema de failover mediante balanceadores y monitoreo.

15.3. Seguridad

Definición

Protección de datos e integridad del sistema frente a accesos no autorizados.

Características claves

- Uso de OAuth y JWT para autenticación y sesiones seguras.
- Cifrado de datos sensibles como contraseñas y transacciones.
- Validación de identidad con verificación automática vía IA.

Evidencia en el sistema

- Gestión de roles y permisos por usuario.
- Auditoría de logs y eventos relevantes.

15.4. Escalabilidad

Definición

Capacidad de adaptarse a un crecimiento en usuarios, datos y servicios.

Características claves

- Escalado horizontal y vertical según necesidades.
- Uso de contenedores (Docker) y despliegues independientes.

Evidencia en el sistema

- Separación de servicios: búsqueda, pagos, mensajería, etc.
- Soporte para planes de crecimiento (modelo Freemium/Oro).

15.5. Mantenibilidad

Definición

Facilidad para realizar correcciones, mejoras o adaptaciones al sistema.

Características claves

- Código basado en principios SOLID.
- Aplicación de patrones de diseño como Strategy, Singleton, Factory, Repository.
- Documentación técnica clara y actualizada.

15.6. Usabilidad

Definición

Facilidad con la que los usuarios interactúan con el sistema.

Características claves

- Interfaz responsiva y clara (SPA con Angular).
- Asistencia visual, flujos de usuario intuitivos, notificaciones claras.

15.7. Accesibilidad

Definición

Capacidad del sistema para ser utilizado por personas con discapacidades.

Características claves

- Navegación por teclado.
- Compatibilidad con lectores de pantalla.
- Contraste de colores y etiquetas descriptivas.

16. Evaluación de Riesgos

16.1. Identificación de riesgos

Métodos utilizados

- Revisión de documentación del proyecto.
- Análisis de funcionalidades y arquitectura.
- Revisión de experiencias anteriores.

Taxonomía aplicada

- Técnicos, humanos, operativos y de negocio.

Cuadro 6: Identificación de Riesgos

ID	Riesgo	Categoría	Método de identificación
R1	Fallo en pasarela de pagos (MercadoPago)	Técnico	Revisión de dependencias externas
R2	Picos de tráfico en campañas	Técnico / Operativo	Lecciones aprendidas
R3	Fuga de datos de usuarios	Seguridad / Legal	Revisión de requisitos no funcionales
R4	Pérdida de disponibilidad de API de mapas	Técnico / Externo	Consulta con expertos
R5	Cambios inesperados en requerimientos	Negocio	Brainstorming de equipo
R6	Desincronización entre microservicios	Técnico / Arquitectónico	Análisis de diseño
R7	Conflictos por uso excesivo del sistema de mensajes	Operativo / Usuario	Proyectos anteriores

16.2. Análisis de Riesgos

Cuadro 8: Análisis de Riesgos

ID	Probabilidad	Impacto	Nivel de riesgo	Justificación
R1	Media	Alta	Alto	Los errores o caídas impiden realizar transacciones, lo que puede derivar en pérdidas económicas y frustración del usuario.
R2	Alta	Alta	Crítico	Puede provocar la caída del sistema o tiempos de respuesta excesivos en momentos clave.
R3	Media	Muy Alta	Crítico	Implica responsabilidad legal, pérdida de confianza y daño a la reputación del proyecto.
R4	Baja	Media	Moderado	Aunque afecta la visualización, no bloquea funcionalidades clave; puede mitigarse con un fallback.
R5	Alta	Media	Alto	Afectan planificación, costos y alcance; común en proyectos con múltiples partes interesadas.
R6	Media	Alta	Alto	Puede generar inconsistencias en los datos o en la lógica de negocio distribuida.
R7	Alta	Media	Alto	Podría saturar el sistema o ser utilizado de forma abusiva si no se limita o modera correctamente.

16.3. Planes de Mitigación

Cuadro 10: Planes de Mitigación

ID	Estrategia de Mitigación	Plan de Contingencia	Responsable
R1	Validar pagos en tiempo real y con logs	Permitir reintento, enviar aviso al admin	Equipo Backend
R2	Escalabilidad horizontal + CDN	Activar instancias en AWS autoscaling	Arquitecto DevOps
R3	Cifrado + roles estrictos + IA para verificación	Auditoría automatizada + alerta de incidentes	Encargado de seguridad
R4	Fallback visual sin mapa + mensaje informativo	Interfaz sin mapa activa con coordenadas	Equipo Frontend
R5	Contrato de cambios controlados (Scope Creep)	Reuniones semanales con stakeholders	Scrum Master
R6	Implementación de patrón Saga + mensajería asincrónica	Registro de eventos para rollback	Equipo de Arquitectura
R7	Limitar mensajes por membresía	Alertas por spam y monitoreo de abuso	Soporte / Moderación

16.4. Monitoreo y Control

- Revisión semanal de riesgos en reunión de seguimiento.
- Matriz de riesgos IPER actualizada.
- Uso de sistemas de logging y herramientas como Trello, Jira o Notion.

16.5. Documentación

- Información centralizada en un documento en la nube.
- Uso de etiquetas por tipo de riesgo y fase del proyecto.
- Herramientas sugeridas: Notion, Jira, GitHub Projects.

17. Estrategia de Escalabilidad

La arquitectura de CasaFácil combina escalabilidad horizontal como eje principal, vertical para casos específicos, y un enfoque híbrido para balancear rendimiento y costos.

Escalabilidad Horizontal (Scale-Out)

Justificación: Responde a picos de carga impredecibles (por ejemplo, inicio de semestre). Mejora rendimiento al distribuir la carga.

Características que lo requieren:

- Crecimiento elástico ante demanda variable.
- Tolerancia a fallos (elimina puntos únicos de fallo).
- Planes premium generan consultas y operaciones simultáneas.

Componentes escalados horizontalmente:

■ Microservicios:

- Autenticación con JWT/OAuth detrás de API Gateway.
- Búsqueda con Elasticsearch y geolocalización.
- Mensajería usando Redis Pub/Sub y WebSockets.

■ Bases de datos:

- PostgreSQL con réplicas de lectura.
- MongoDB con sharding por colección.

■ Alta disponibilidad:

- Kubernetes con múltiples zonas de disponibilidad.
- PodDisruptionBudgets y Service Discovery dinámico.

Escalabilidad Vertical (Scale-Up)

Justificación: Se usa en componentes con estado o requerimientos de consistencia fuerte.

Características que lo requieren:

- Bases de datos maestras (transacciones ACID).
- Servicios intensivos en RAM o IOPS.

Componentes escalados verticalmente:

- PostgreSQL maestro (más CPU/RAM).
- Redis con mayor capacidad de memoria.
- Kafka con brokers optimizados para eventos.

Enfoque Híbrido (Hybrid Scaling)

Justificación: Combina lo mejor de ambos modelos.

- **API Gateway:** Horizontal (balanceo de tráfico).
- **PostgreSQL:** Vertical + horizontal (maestro y réplicas).
- **Servicio de pagos:** Horizontal (procesamiento paralelo).

Ventajas:

- Flexibilidad ante demanda.
- Costos optimizados.
- Alta disponibilidad garantizada.

Alta Disponibilidad (99.9 %)

- **Multi-AZ Deployment:** Microservicios y DBs en múltiples zonas.
- **Replicación:** PostgreSQL síncrona, MongoDB asíncrona.
- **Health Checks:** Kubernetes y ELB supervisan salud del sistema.
- **Backups:** Snapshots automáticos (RDS, MongoDB Atlas).

18. Plan de Pruebas

Objetivos del Plan de Pruebas

- Verificar funcionalidad de microservicios individuales.
- Detectar errores en etapas tempranas.
- Validar integridad de datos.
- Comprobar comportamiento ante entradas válidas e inválidas.
- Asegurar correcto funcionamiento aislado de cada módulo.

Microprocesos Críticos a Validar

- **Cuentas:** Registro e inicio de sesión (JWT, validación).
- **Propiedades:** Creación, edición y moderación.
- **Reservas:** Disponibilidad, confirmación, rechazo.
- **Pagos:** Integración con MercadoPago, comprobantes.
- **Mensajería:** Envío y recepción, validación de longitud.
- **Mapas:** Búsqueda por cercanía, fallos de API externa.

Herramientas a Utilizar

- **Pytest:** Pruebas unitarias.
- **Django TestCase:** Validación de vistas y modelos.
- **SQLite/PostgreSQL:** Bases de prueba.
- **Mock/Patch:** Simulación de servicios externos.
- **Factory Boy:** Generación de datos.
- **Coverage.py:** Medición de cobertura.

Estructura del Plan de Pruebas

Servicio de Cuentas

- Registro válido → usuario creado y correo enviado.
- Registro duplicado → error de validación.
- Login exitoso → token JWT generado.
- Login fallido → error 401.

Servicio de Propiedades

- Crear → pendiente de aprobación.
- Editar → reflejo de cambios.
- Moderación → cambia a estado visible.

Servicio de Reservas

- Reserva exitosa → disponibilidad actualizada.
- Reserva fallida → acción bloqueada.

Servicio de Pagos

- Pago correcto → HTTP 200, comprobante generado.
- Reembolso válido → estado actualizado.

Servicio de Mensajería

- Mensaje válido → reenviado vía WebSocket.
- Exceso de caracteres → envío bloqueado.

Servicio de Mapas

- Coordenadas válidas → propiedades cercanas.
- Error API → excepción manejada con mensaje.

Cobertura y Técnicas

- **Cobertura esperada:** Mínimo 90 % en microservicios críticos.
- **Técnicas:**
 - Particiones equivalentes.
 - Valores límite.
 - Simulación de fallos externos.
 - Verificación antes/después de operación.
 - Inyección de dependencias con mocks.

Gestión y Ejecución

- Automatización con Pytest.
- Ejecución en CI/CD (GitHub Actions o GitLab CI).
- Reportes con `coverage.py`.
- Alertas automáticas ante fallos.

19. Diagrama de Escalabilidad

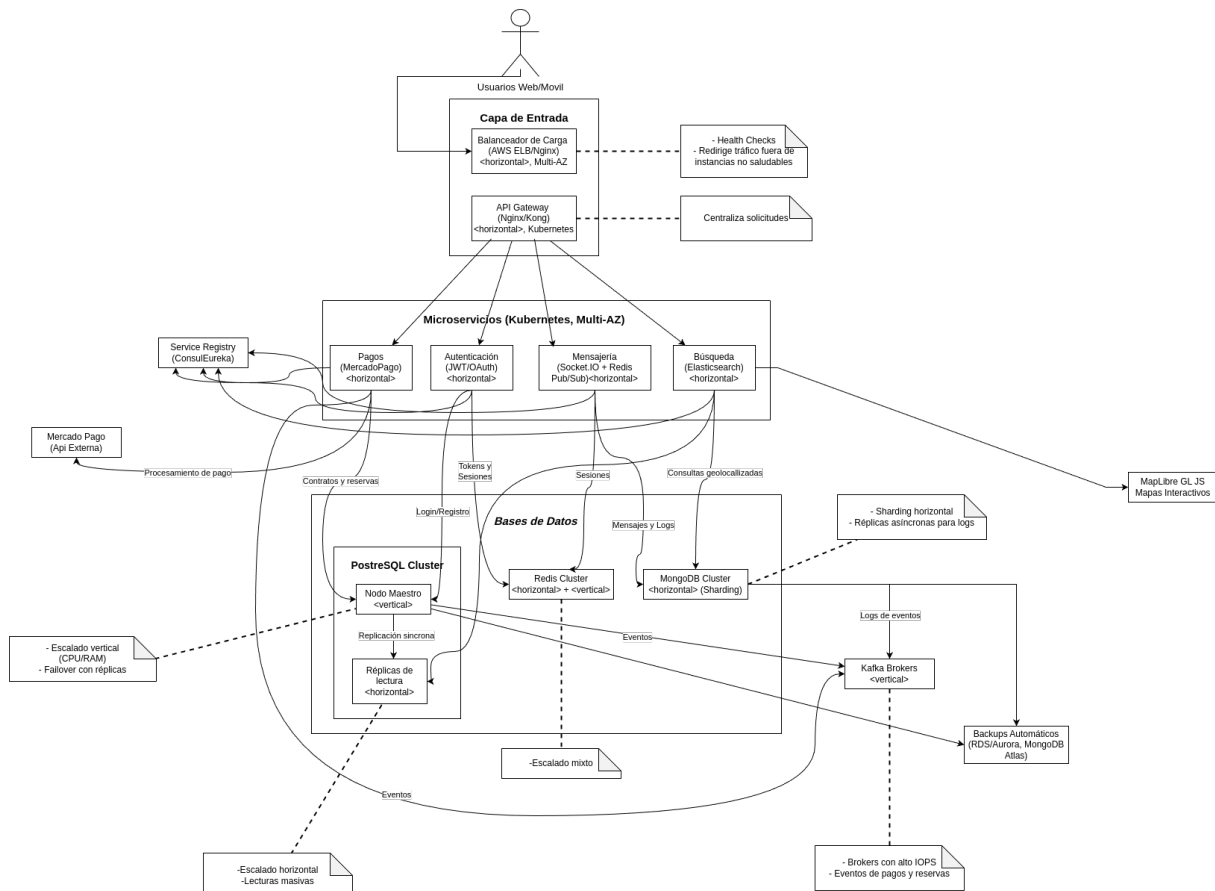


Figura 21. Diagrama de escalabilidad del sistema CasaFácil

Descripción del flujo:

- Usuarios → balanceador → API Gateway → microservicios.
- Microservicios → DBs (PostgreSQL, MongoDB), Redis, Kafka.
- Servicio de pagos conecta a MercadoPago.
- Servicio de búsqueda consulta mapas externos (MapLibre).
- Registro de servicios facilita descubrimiento entre microservicios.
- Backups garantizan persistencia de datos.

Alta disponibilidad y failover:

- Balanceador redirige tráfico a instancias saludables.
- Kubernetes reinicia pods fallidos automáticamente.
- PostgreSQL con réplicas síncronas.
- MongoDB con réplicas asíncronas.
- Redis en cluster.
- Backups automáticos ante fallos catastróficos.

Crecimiento ante demanda:

- API Gateway y balanceador escalan horizontalmente.
- Kubernetes lanza más pods.
- PostgreSQL añade réplicas o CPU.
- MongoDB shardea nuevas colecciones.
- Redis escala en RAM/nodos.
- Kafka usa brokers más potentes.

20. Mapa de riesgos

Cuadro 12: Matriz de Riesgos IPER

ID	Riesgo	Prob.	Exposición	Resultado	Nivel	Acciones
R1	Fallo en pasarela de pagos	Media	Alta	Sin reserva	Alto	Backup de pagos + logs
R2	Pico de tráfico	Alta	Alta	Caída de sistema	Crítico	Escalado + CDN
R3	Fuga de datos	Media	Muy Alta	Daño reputacional	Crítico	Cifrado + roles
R4	API de mapas cae	Baja	Media	No se muestra ubicación	Medio	Vista alternativa
R5	Cambio de requerimientos	Alta	Media	Retraso de entregas	Alto	Control de versiones
R6	Desincronización servicios	Media	Alta	Fallos funcionales	Alto	Patrón Saga
R7	Abuso de mensajes	Alta	Media	Saturación de sistema	Medio	Límite por membresía

Leyenda:

 Riesgo Crítico
  Riesgo Alto
  Riesgo Medio