

Guía Informe de Proyecto

Unidad 2: Arquitectura de la solución, almacenamiento y plan de pruebas

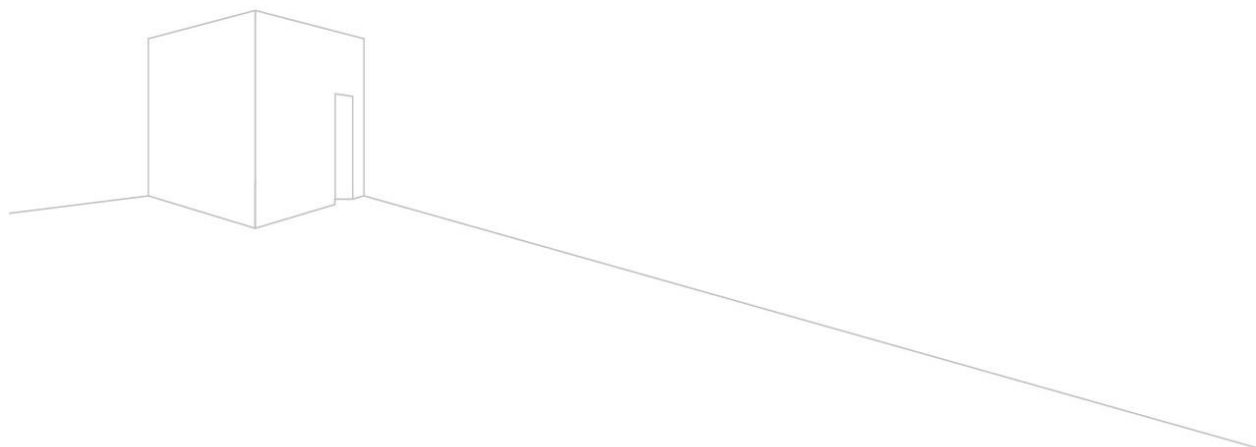
Asignatura: Taller de Diseño y desarrollo de Soluciones

Sección: AP-171-N4

Nombre del docente: Nelson Ganga Calderón

Nombre de los integrantes del grupo: Alejandro Ancamilla, Alen Chávez, Allan Sandoval

21 de noviembre, 2024



Contenido

Introducción	4
Objetivos	5
Instrumentos para el levantamiento de los requerimientos	6
Benchmarking.....	7
Objetivos	7
Competidor.....	7
Funciones y características.....	7
Vistas	9
Modelo de negocios	9
Historias de usuario.....	10
Metodologías y herramientas	12
Priorización de historias de usuarios y Sprints.....	16
Sprint Backlog.....	17
Paradigma 4 + 1	18
Diagrama de clases:.....	18
Diagrama de secuencia:	19
Diagrama de componentes:	20
Diagrama de paquetes:	20
Diagrama de actividades:	21
Diagramas de casos de uso:	22
Interfaces.....	24
Interfaz de inicio de sesión.....	24
Interfaz de página principal.....	25
interfaz de historial de usuario	26
Modelo de negocios	27
Patrones de arquitectura de software	28
Perfiles y roles del sistema	31
Buenas prácticas de programación segura	31
Servicio de almacenamiento Cloud.....	32

Google Cloud Platform (Google)	32
AWS (Amazon Web Services)	33
Azure (Microsoft)	33
Plan de pruebas	35
Criterios de aceptación.....	35
Identificación de funcionalidades	36
Estrategia de pruebas.....	36
Casos de prueba	37
Cronograma de pruebas.....	37
Conclusiones.....	38
Referencias bibliográficas.....	39
Anexo A	39
Encuesta a residentes.....	39
Entrevista a usuarios	40
Entrevista a administrador	41
Benchamarking.....	43
Vistas	43

Introducción

Este informe, titulado "**Arquitectura de la solución, almacenamiento y plan de pruebas**", es una continuación del trabajo previamente desarrollado bajo el título "**Levantamiento de requerimientos Plataforma Web para manejo de gestiones en condominios**". En este documento se retoman y consolidan los contenidos ya presentados, como las historias de usuario, el levantamiento de requerimientos y las funcionalidades clave de la plataforma.

Además, se amplía la información incorporando las cuatro actividades solicitadas en esta nueva etapa del proyecto:

1. **Definición de los patrones de arquitectura** para la organización de la solución, justificando su selección.
2. **Incorporación de estándares de programación segura**, asegurando la confiabilidad del sistema.
3. **Definición del almacenamiento en cloud**, evaluando alternativas y seleccionando la más adecuada para el proyecto.
4. **Diseño del plan de pruebas**, incluyendo la identificación de criterios de aceptación, las funcionalidades asociadas, los tipos de pruebas y la estrategia para su implementación.

En la actualidad, muchos condominios y edificios son gestionados de manera manual (lápiz y papel), lo que implica procesos lentos, ineficiencias y riesgos de errores en la administración de pagos, cobros y reclamos. Estos problemas se agravan cuando las comunidades son grandes o tienen un alto volumen de transacciones.

El informe describe el desarrollo de una plataforma web para resolver un problema del entorno, como la digitalización de procesos como una gestión integral de pagos de gastos comunes y otros cobros asociados a condominios y edificios. Esta solución busca modernizar los procesos tradicionales, permitiendo que tanto administradores como residentes puedan gestionar y visualizar sus pagos de manera ágil, efectiva y sin necesidad de recurrir a métodos manuales.

Además de la gestión de pagos, la plataforma ofrece la posibilidad de **emitir recibos electrónicos**, manejar **reclamos** y **reservar espacios comunes**, brindando una experiencia optimizada tanto para los usuarios generales como para los administradores. La implementación de un **sistema de inteligencia artificial** para la clasificación y etiquetado de reclamos permite mejorar la eficiencia en la gestión de estos, detectando infractores recurrentes y optimizando los procesos administrativos.

En este documento, nos referiremos a esta herramienta con términos como 'plataforma', 'aplicación web', 'sistema', o 'solución'. Asimismo, se utilizarán de manera intercambiable los términos 'condominio', 'edificio' y 'comunidad' para referirnos a este tipo de entorno.

Para el desarrollo de la solución contaremos con la participación de don **Juan Riquelme**, administrador de un condominio con 140 domicilios en Temuco, conoceremos su opinión y comentarios que esperamos nos ayuden de manera constructiva durante este proceso.

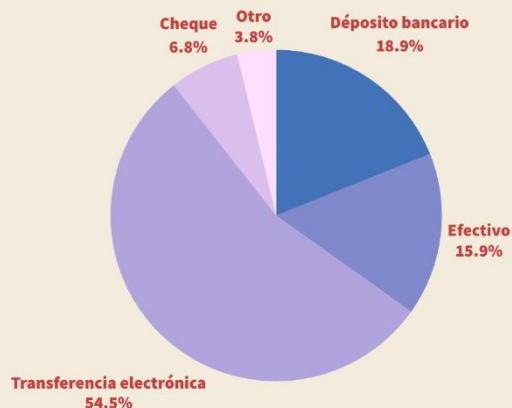
Objetivos

- Identificar las necesidades del negocio mediante el levantamiento de los requerimientos.
- Priorizar las historias de usuario y generar los primeros sprint.
- Diseñar los diagramas UML según las distintas vistas de acuerdo con el paradigma 4 + 1.
- Diseñar las interfaces.
- Definir los patrones de arquitectura de la solución.
- Incluir estándares de programación segura.
- Definir almacenamiento de datos en cloud.
- Diseñar el plan de pruebas.

Instrumentos para el levantamiento de los requerimientos

Se aplicó una encuesta a un condominio de 140 domicilios en la ciudad de Temuco, de los que 132 respondieron; la encuesta consta de 7 preguntas de selección múltiple y puede ser encontrada en el **Anexo A** de este informe. A continuación, se detallan algunos resultados:

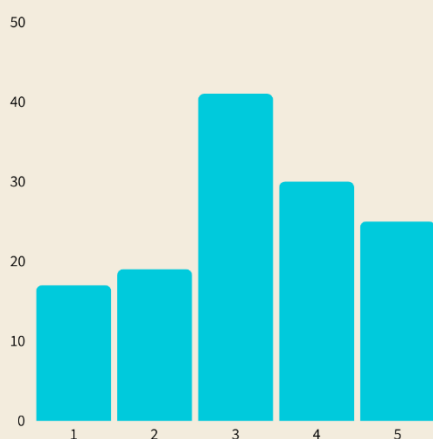
Actualmente, ¿Que método de pago utiliza más frecuentemente para cancelar los gastos comunes?



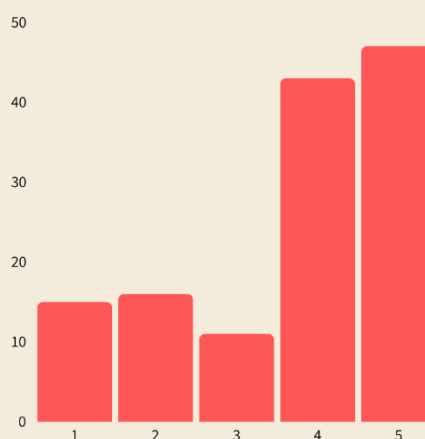
¿Usted conoce el proceso y costos asociados para arrendar espacios comunes en su condominio o edificio?



Del 1 al 5, donde 1 es muy en desacuerdo y 5 muy de acuerdo. ¿Usted ha desistido alguna vez de arrendar espacios comunes disponibles en su condominio/edificio por desconocer el proceso de arriendo, los montos y/o condiciones?



Del 1 al 5, donde 1 es muy en desacuerdo y 5 muy de acuerdo. ¿Usted ha desistido alguna vez de interponer un reclamo por una infracción al reglamento de convivencia y copropiedad o por sentirse pasada/o a llevar como persona en su condominio/edificio por desconocer el proceso y/o condiciones?



Gráficos de elaboración propia usando la herramienta Canva.

En el anterior mencionado **Anexo A** también se adjunta una entrevista a un residente de este condominio para conocer en más detalle su opinión y la entrevista al administrador del mismo condominio, don Juan Riquelme. En la **Tabla 2** Historias de usuario se recopilaron las características aquí recogidas.

Benchmarking

Objetivos

- Aspectos para evaluar: Funcionalidades, características y precios.
- Identificar buenas prácticas, conocer a los líderes en la industria.

Competidor

“ComunidadFeliz es un software de administración de condominios que permite a los administradores de condominios y a los propietarios de apartamentos administrar sus condominios de manera eficiente. La plataforma tiene una variedad de características que incluyen gestión de documentos, reservaciones de amenidades, seguimiento de mantenimiento, gestión de incidencias y cobro de cuotas de mantenimiento.” (Tomado de “¿Qué es ComunidadFeliz?”)

Funciones y características

Administra condominios con tu propio asistente digital

Con el mejor software de administración de edificios



Avisos de pago y cobro

En el módulo de recaudación puedes enviar e incluso imprimirlos avisos de pago y cobro.



Ingreso de pagos sin reconocer

Cuando tengas un depósito que no reconoces en la cuenta, puedes ingresarlo y el sistema te indica qué unidad tiene esa deuda.



Exportación a Excel

Respaldar todos tus datos en Excel, tanto egresos, ingresos, multas y mucho más.



Conexión con bancos

Podrás revisar el libro de Banco (ingresos, egresos y deudas) de manera fácil. Asigna los movimientos que se reflejan en la cuenta corriente y los que no. Obtén saldo cartola y saldo contable.



Módulo de gastos comunes

Generación de cálculo preciso del cobro de los gastos por unidad.



Módulo de egreso

Puedes revisar el registro y control de gastos a proveedores. Además, tienes la posibilidad de imprimir los cheques de la comunidad.



Reportes

Generar reportes financieros de cobranza, ingresos, egresos, cuentas bancarias y de cajas, presupuesto, morosidad, adeudos, entre otros.

Imagen 1. Funciones administrador.

Funciones para residentes de ComunidadFeliz

Gana autonomía con trámites dentro del condominio usando la app.

 <p>Reserva de espacios comunes</p> <p>Mediante la aplicación móvil para copropietarios y residentes del edificio, se pueden reservar áreas comunes.</p>	 <p>Noticias oficiales del condominio</p> <p>En el muro de la comunidad podrás observar los avisos y últimas novedades.</p>	 <p>Encuestas y votaciones online</p> <p>Tenemos una sección de encuestas y votaciones. Se pueden responder incluso, desde la app para residentes.</p>	 <p>Calendario de eventos en el edificio</p> <p>Verifica los próximos eventos que se realizarán en los espacios comunes del condominio.</p>
 <p>Recepción de paquetes</p> <p>Cuando te llegue una encomienda, el conserje que la reciba te notificará mediante la aplicación.</p>	 <p>Restricción de visitantes</p> <p>Puedes solicitar la restricción de visitas específicas a tu departamento.</p>	 <p>Recordatorios de gastos comunes</p> <p>Recibirás notificaciones de cobros de multas a tu unidad y tu última boleta de gastos comunes.</p>	 <p>Historial de pagos al instante</p> <p>Conoce detalladamente el historial de tus pagos mediante la aplicación.</p>

Imagen 2. Funciones residentes.

Funcionalidades más destacables de ComunidadFeliz

ComunidadFeliz.cl posee muchas características que pueden ayudarte a reducir tu gestión administrativa y queremos que las conozcas.

 <p>Asambleas en línea</p> <p>Conecta con el condominio mediante conferencias virtuales y conoce todo lo que acontece en la comunidad.</p>	 <p>Encuestas y votaciones</p> <p>Tenemos una sección de encuestas y votaciones. Estas se pueden responder desde la app para Copropietarios.</p>	 <p>Monitoreo de correos electrónicos</p> <p>Revisa si los correos con la información fueron enviados, si los correos están bien escritos o no y si los copropietarios recibieron y abrieron los correos.</p>	 <p>Conciliación bancaria automática</p> <p>Podrás revisar el libro de Banco (Ingresos, Egresos y Deudas) de manera fácil. Asignando los movimientos que se reflejan en la cuenta corriente y los que no. Obtén saldo cartola y saldo contable.</p>
 <p>Proveedores</p> <p>En cuanto al registro, control de gastos y proveedores, están ubicados en el módulo de egresos.</p>	 <p>Creación de permisos a usuarios ilimitados</p> <p>Puedes agregar encargados de módulo para ingresar datos o para sólo ver. Funcionamos con usuarios ilimitados, todo tu equipo de trabajo podrá aportaren el software.</p>	 <p>Reportes</p> <p>El administrador podrá realizar reportes financieros de cobranza, ingresos, egresos, cuentas bancarias y de cajas, presupuesto, morosidad, adeudos, entre otros.</p>	 <p>Mantenimiento de instalaciones</p> <p>Con el módulo de incidencias se pueden reportar desperfectos o problemáticas en las instalaciones del condominio.</p>

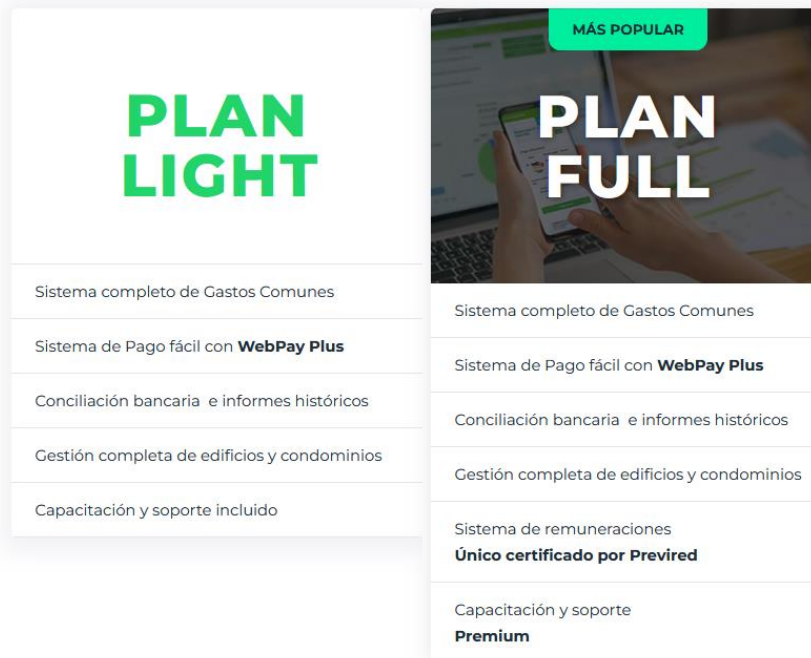
Imagen 3. Funciones comités administrativos.

Vistas

En el **Anexo A** se adjuntan vistas de usuario para las funciones descritas en el punto anterior

Modelo de negocios

Tipos de planes



*Precio válido en Comunidades hasta 500 Unidades, sobre este número está sujeto a cotización.

Imagen 4. Planes ofrecidos por ComunidadFeliz.

El software ComunidadFeliz ofrece 2 planes que cuyos valores fueron resumidos en la **tabla 1**. ComunidadFeliz también cobra una comisión por cada pago digital del **2.5% + IVA**.

Plan	Cobro mensual	Pagado anualmente
Light	2 UF	20 UF
Full	3 UF	30 UF

Tabla 1.

Historias de usuario

RECOPIACIÓN DE HISTORIAS DE USUARIO				
ID	Rol	Característica / Funcionalidad	Razón / Resultado	Criterios de aceptación
HU01	Como usuario de la plataforma	Necesito visualizar la página en un navegador, que esta sea intuitiva, fácil de comprender y con un diseño coherente a través de todas las páginas.	Con la finalidad de que la página sea sencilla de utilizar por el público general	<ul style="list-style-type: none"> El tema de la aplicación web debe mantenerse a través de las distintas paginas El diseño no debe ser muy complejo, optar por lo minimalista.
HU02	Como usuario de la plataforma	Necesito poder iniciar sesión en el sistema con mis credenciales.	Con la finalidad de poder cancelar deudas por gasto común, multa u otro cobro y poder acceder al historial de pago	<ul style="list-style-type: none"> Debo tener permitido ingresar mi nombre de usuario y contraseña dado por la administración. Si mis credenciales son válidas debe poder acceder al sistema, de no ser así, aparecerá un mensaje de error. Una vez autenticadas mis credenciales, debo poder acceder a todas las funciones correspondiente a mi usuario
HU03	Como usuario de la plataforma	Necesito poder pagar mis gastos comunes y acceder al registro histórico de estos.	Con la finalidad de facilitar el pago y mantener un registro online	<ul style="list-style-type: none"> Debo tener habilitado el pago del último mes de gastos comunes una vez se llega a la fecha correspondiente Si tengo pagos pendientes estos deben ser cancelados primeros (solo gastos comunes) Multas y cobros extras también deben estar disponibles para pago, el no pago de estos cobros no debe dificultar el pago de gastos comunes.
HU04	Como usuario de la plataforma	Necesito poder agendar el arriendo de espacios comunes (Estacionamientos, salón de eventos, gimnasio, piscina, quincho, etc.)	Con la finalidad de digitalizar, acelerar el proceso y mantener un registro.	<ul style="list-style-type: none"> Debo tener permitido revisar el calendario de uso y arriendo de un determinado espacio común. Si deseo realizar una reserva debo seleccionar una fecha y horas en un calendario. Para realizar la reserva se debe acreditar un pago inicial o pie. El sistema debe informarme sobre la continuación del proceso (ej: pago del resto del arriendo, retiro de llave, firma de toma de responsabilidad, etc.), en forma de notificación en la plataforma y de mensaje de correo.
HU05	Como usuario de la plataforma	Necesito poder ingresar un reclamo	Con la finalidad de digitalizar el proceso (hasta el momento es con lápiz y	<ul style="list-style-type: none"> Debo tener permitido ingresar un reclamo el que consistirá en un bloque de texto con la posibilidad de adjuntar imágenes y videos Debo poder acceder a mis reclamos anteriores y poder dejar más antecedentes (texto, imágenes y/o videos) Debo poder también ver los reclamos/multas en contra. Los reclamos ingresados pueden no estar dirigidos a un domicilio en específico.x

			papel), mantener un registro y respaldo.	
HU06	Como usuario de la plataforma	Necesito contar con una barra de navegación.	Con la finalidad de acceder a todas las funciones que me permita el sistema.	<ul style="list-style-type: none"> Después de iniciar sesión, a partir de la pagina “home” o de inicio y en adelante, debo tener acceso a una barra de navegación que resuma el acceso a las funciones de la aplicación web. La barra debe incluir una sección de notificaciones. La barra debe incluir una función de cerrar sesión.
HU07	Como usuario de la plataforma	Necesito contar con el reglamento de convivencia.	Con la finalidad de distribuir y acceder de manera fácil y rápida a el reglamento.	<ul style="list-style-type: none"> Debo de ser capaz de ver y descargar el reglamento de convivencia.
HU08	Como usuario de la plataforma	Necesito recibir notificaciones.	Con la finalidad de recibir información.	<ul style="list-style-type: none"> Debo recibir una notificación/alerta cuando se cumplan 3 meses de morosidad de gastos comunes. Debo recibir una notificación/alerta cuando realice cualquier tipo de pago. Debo recibir una notificación/alerta cuando cometa una infracción y deba pagar una multa.
HU09	Como administrador de la plataforma	Necesito listar el estado de los gastos comunes de todos los domicilios.	Con la finalidad de organizar, monitorear y descargar información correspondiente a los gastos comunes.	<ul style="list-style-type: none"> Debo poder listar todos los residentes en una página estilo planilla y una serie de columnas que incluya toda la información correspondiente. Debo ser capaz de hacer clic en un domicilio y poder acceder a todos sus datos. Al hacer clic en un domicilio también debo ser capaz de poder cancelar alguna deuda por si el propietario decide pagar con otros medios que no sea la plataforma.
HU10	Como administrador de la plataforma	Necesito acceder a todos los reclamos.	Con la finalidad de proceder con el cobro	<ul style="list-style-type: none"> Si un reclamo cumple con las condiciones para proceder con el cobro se envía una notificación y un correo electrónico al infractor con la información correspondiente junto con la opción de apelar. La multa asociada a un reclamo se cargará automáticamente al cobro de gastos comunes del mes siguiente, a menos que el infraccionado lo cancele antes.
HU11	Como administrador de la plataforma	Quiero que el sistema etiquete y clasifique los reclamos automáticamente.	Con la finalidad de poder gestionar los reclamos mas eficientemente y detectar infractores recurrentes	<ul style="list-style-type: none"> El sistema impulsado con IA debe clasificar los reclamos en categorías (por ejemplo: mal uso de estacionamiento, ruidos molestos, mala gestión de residuos, etc.) El sistema debe identificar y asociar infractores recurrentes (por ejemplo: si un vehículo está mal estacionado y no se conoce el domicilio el sistema analizará si incurrió anteriormente en una multa/reclamo)

Tabla 2. Historias de usuario.

Metodologías y herramientas

Para el desarrollo de este Sistema de gestión de condominios y edificios se utilizará el marco de trabajo ágil Scrum, **principalmente debido** a su **alta flexibilidad y adaptabilidad** a la hora de cumplir con los requisitos del producto, también se estudió la posibilidad de utilizar otros marcos de trabajo para el desarrollo de la aplicación tales como Crystal, Extreme Programming y Kanban e incluso otras metodologías como lo es el modelo de cascada.

Metodología	Ventajas	Desventajas
Ágil	<ul style="list-style-type: none"> Se adapta rápidamente a los cambios inesperados. Se centra en la satisfacción del cliente. Alta participación del equipo de trabajo. 	<ul style="list-style-type: none"> Puede aumentar el alcance y presupuesto del proyecto de forma inesperada. Interacción con el cliente puede ser difícil dada la frecuencia de estas. Equipo centrado exclusivamente al proceso de sprint dejando de lado otras iniciativas.
Cascada	<ul style="list-style-type: none"> Planificación Anticipada. Fácil seguimiento del progreso. Desarrollo compatible con varios proyectos a la vez. 	<ul style="list-style-type: none"> Poca o nula participación del cliente durante el desarrollo. Posibles retrasos inesperados si controles de calidad se realizan tarde. Baja adaptación al cambio.

Tabla 3. Comparativa de ventajas y desventajas entre la metodología de desarrollo ágil y la gestión en cascada.

Marco de trabajo	Ventajas	Desventajas
Scrum	<ul style="list-style-type: none"> Estructura de equipo definida. Colaboración y comunicación constante del equipo. Retroalimentación continua. Fácil adaptación al cambio. 	<ul style="list-style-type: none"> Rigidez en cuanto a roles. El equipo debe contar con entrenamiento en scrum. Se adapta mejor a equipos pequeños (menos de 10).
Crystal	<ul style="list-style-type: none"> Flexibilidad en el tamaño del proyecto. Énfasis en la comunicación. Entrega Frecuente. 	<ul style="list-style-type: none"> Estructura de equipo poco clara. Poca documentación Alta participación del cliente. Es necesario un equipo muy disciplinado y autónomos.
Extreme Programming	<ul style="list-style-type: none"> Comunicación continua entre el equipo. Retroalimentación frecuente. Ciclos de desarrollo cortos. Adaptación rápida a cambios. 	<ul style="list-style-type: none"> Alta participación del cliente. Difícil de aplicar a equipos grandes. Equipo altamente especializado en la programación y testing continuo.
Kanban	<ul style="list-style-type: none"> Fácil visualización. Flexibilidad en ajustes 	<ul style="list-style-type: none"> Falta de estructura. Requiere disciplina y experiencia.

Tabla 4. Comparativa de ventajas y desventajas entre marcos de trabajo ágiles.

Lenguaje de programación	Nivel de dominio por parte del equipo de desarrollo
Python	Medio.
JavaScript	Medio.
PHP	Bajo.
Java	Bajo.
C#	Bajo.

Tabla 5. Lenguajes de programación más populares el 2023 según usuarios de Stack Overflow y nivel de dominio del equipo de desarrollo.

Nombre	Rol
Alejandro Ancamilla	Scrum Master.
Alen Chávez	Desarrollador.
Allan Sandoval	Desarrollador.
Juan Riquelme	Product Owner.

Tabla 6. Roles Scrum

	Lenguaje	Licencia	Tipo de aplicación ideal	Ejemplos	Familiaridad en el equipo
React (+ Node.js)	JavaScript	Licencia MIT	Single-Page Application (SPA), Progressive Web Application (PWA), Server-Side Rendering (SSR).	Spotify, Discord, Airbnb, Instagram.	Media-alta.
Django	Python	BSD License	Multi-Page Application (MPA)	Udemy, Platzi, Disquss.	Media.
Flask	Python	BSDLicense	MPA ligera	Pinterest, Patreon, Brilliant.	Media-baja.
Ruby on Rails	Ruby	Licencia MIT	MPA tradicional	Shopify, AriBnB.	Baja.
Laravel	PHP	Licencia MIT	MPA	Vanity Fair, Variety.	Baja.

Tabla 7. Marcos de desarrollo evaluados y 5 criterios que los diferencian.

React es una biblioteca de **JavaScript** (a menudo clasificada como framework) diseñada para desarrollar con facilidad aplicaciones de una sola página (**SPA**). Debido a su capacidad para crear interfaces de usuario dinámicas y la familiaridad de nuestro equipo con JavaScript, **React** fue seleccionado como la tecnología ideal para desarrollar esta aplicación que puede ser considerada una SPA, pues no es una aplicación que será abierta al público y espere demasiadas visitas (Baja escalabilidad), presenta pocos roles de usuario y el sistema necesita respuestas rápidas y una interfaz dinámica garantizando una experiencia de usuario fluida, sencilla e intuitiva. **React**, sin embargo, solo nos permite diseñar interfaces de usuario (**Front end**), para manejar la lógica de la aplicación será necesario un back end adecuado.

Una **API RESTful** es un servicio web que sigue los principios REST (Representational State Transfer), permite la comunicación entre un servidor y un cliente (como una aplicación React) a través de HTTP usando métodos como GET, POST, PUT, y DELETE. Para el desarrollo del **back end**, hemos optado por **Node.js**, un entorno de ejecución de **JavaScript** que nos permite construir **API RESTful** escalables y eficientes. Estas API, se encargan de gestionar la lógica del negocio, la interacción con la base de datos y la comunicación con otros servicios. En este contexto, **Express**, un framework minimalista para Node.js, se convierte en una herramienta clave, ya que facilita la creación de servidores y la gestión de rutas de manera sencilla y eficiente, permitiendo una integración fluida entre React (front end) y la API (back end). Para el correcto cumplimiento de los requerimientos es necesario que la base de datos (BD) almacene imágenes y videos breves, para cumplir con estos objetivos se optó por una BD no estructurados (NoSQL) donde se destaca **MongoDB** al ser un Software Libre y con un nivel de conocimiento medio por parte del equipo de desarrollo.

Otro beneficio del uso de **Node.JS** y un entorno basado en JavaScript para el desarrollo de la aplicación es el acceso a **Node Package Manager (npm)**, un repositorio y gestor de paquetes de JavaScript que cuenta con una gran variedad de herramientas sin coste, el equipo deberá tener la precaución de revisar las licencias (MIT, Apache 2.0, GPL, etc.) y seguir las condiciones de otros desarrolladores con especial cuidado en las APIs y Software Development Kit (**SDK**) que gestionan pagos.

Todas estas herramientas conforman lo que se denomina como **MERN Stack** (por las siglas de MongoDB, Express, React y Node.js)

Tecnología	Rol
MongoDB	Base de datos NoSQL
Express	Framework para Node.js, simplifica creación de aplicaciones web y APIs del lado del servidor
React	biblioteca de JavaScript para construir interfaces de usuario en el lado del cliente (UI)
Node.js	Entorno de ejecución de JavaScript del lado del servidor que permite ejecutar código JavaScript fuera del navegador
GitHub	Control de versiones.

Tabla 8. Resumen de Stack a utilizar.

Para los pagos se podrá implementar Transbank (Webpay) y Mercadopago cuyos **SDK** están en **NPM** y posteriormente se solicitan keys para las APIs RESTful.

transbank-sdk

5.0.0 • Public • Published 8 months ago

[Readme](#) [Code](#) [Beta](#) [1 Dependency](#) [1 Dependents](#) [16 Versions](#)

tag v5.0.0 license BSD-3-Clause contributors 10 build failing

Transbank SDK Node.js

Este es el SDK oficial de Transbank para Node.js

Requisitos:

- Node.js 8+

Instalación

Instalar con npm

Puedes instalar SDK en tu proyecto usando NPM

```
npm install transbank-sdk
```

Instalar con yarn

Install

```
> npm i transbank-sdk
```

Repository

[github.com/TransbankDevelopers/trans...](#)

Homepage

[github.com/TransbankDevelopers/tran...](#)

Weekly Downloads

1164

Version	License
5.0.0	BSD-3-Clause

Unpacked Size	Total Files
1.85 MB	323

Imagen 5. Transbank SDK en npm.

mercadopago

2.0.15 • Public • Published 24 days ago

[Readme](#) [Code](#) [Beta](#) [2 Dependencies](#) [33 Dependents](#) [84 Versions](#)

tag v2.0.15 downloads 1.2M 40+ badge not found

Mercado Pago SDK for NodeJS

This library provides developers with a simple set of bindings to help you integrate Mercado Pago API to a website and start receiving payments.

Install

```
> npm i mercadopago
```

Repository

[github.com/mercadopago/sdk-nodejs](#)

Homepage

[github.com/mercadopago/sdk-nodejs#r...](#)

Weekly Downloads

19.677

Version	License
2.0.15	none

Imagen 6. Mercadopago SDK en npm.

Para la implementación del sistema de categorización impulsado por inteligencia artificial podemos tomar 2 rutas, desarrollar un modelo de **machine learning supervisado** (algoritmos como Naive Bayes, Support Vector Machine (SVM) o redes neuronales), utilizar alguna herramienta preentrenada. Como equipo nos inclinamos por el uso de una herramienta preentrenada considerando los costos y excesivo tiempo necesario para entrenar, desplegar y mantener una función así por el lado del servidor, en cuanto al tipo de herramienta se cuentan con opciones basadas en la nube que procesan lenguaje natural tales como

IBM Watson, Azure Text Analytics, Google Cloud Natural Language API u OpenAI que ofrecen servicios basados en tokens (por lo tanto, basado en uso).

Por último, para el despliegue e implementación de la aplicación se optará por soluciones basadas en la nube por su **bajo costo** respecto a la instalación de servidores físicos, los cuales a su vez significan un alto costo a la hora de mantener (Espacio, energía, personal, etc.). Existiendo servicios reconocidos a nivel mundial como AWS, Azure, Google Cloud Platform y proveedores locales como Vultr, Defontana o GTD que cobran por carga de trabajo.

Priorización de historias de usuarios y Sprints

En conjunto con el Product owner se propuso el uso de la técnica **MoSCoW** para la priorización de las historias de usuario. Esta técnica clasifica las historias en cuatro categorías:

- **Must have:** Son las funcionalidades absolutamente necesarias para el lanzamiento del producto.
- **Should have:** Son funcionalidades importantes, pero no críticas para el lanzamiento inicial.
- **Could have:** Son funcionalidades deseables, pero que pueden posponerse para futuras iteraciones.
- **Won't have:** Son funcionalidades que pueden no ser incluidas en el lanzamiento del producto, pero agregadas a posterioridad.

Adicionalmente a la categorización se ordenaron de mayor prioridad a menor prioridad.

ID	Rol	Característica / Funcionalidad	Razón / Resultado	Prioridad MoSCoW
HU02	Usuario	Iniciar sesión	Acceder al sistema	Must have
HU03	Usuario	Pagar gastos comunes y acceder al historial	Facilitar pago y mantener un registro en línea	Must have
HU05	Usuario	Registrar un reclamo	Digitalizar el proceso y mantener un registro	Must have
HU09	Administrador	Listar el estado de los gastos comunes de todos los domicilios	Monitorear y gestionar pagos	Must have
HU11	Administrador	Clasificación automática de reclamos mediante IA	Gestionar reclamos más eficientemente y detectar infractores recurrentes	Must have
HU04	Usuario	Agendar el arriendo de espacios comunes	Acelerar el proceso y mantener un registro	Should have
HU10	Administrador	Acceder a los reclamos y gestionar cobros	Autorizar el cobro de reclamos	Should have
HU07	Usuario	Contar con el reglamento de convivencia	Acceso fácil y rápido al reglamento	Should have
HU08	Usuario	Recibir notificaciones	Recibir información sobre pagos y multas	Should have
HU06	Usuario	Barra de navegación	Facilitar acceso a las funciones del sistema	Could have
HU01	Usuario	Diseño minimalista de la interfaz	Simplificar el uso de la plataforma	Won't have

Tabla 9. Priorización de historias de usuario

Sprint Backlog

Sprint	Start	Days	End	Estimated Size	Real Size	Status	Release Date
1	25-09-2024	28	08-11-2024	224		No Started	08-11-2024
2	12-11-2024	28	25-12-2024	224		No Started	25-12-2024
3	26-12-2024	28	11-02-2025	224		No Started	11-02-2025
4	12-02-2025	28	02-04-2025	224		No Started	02-04-2025

Tabla 10. Sprints.

ID	Sprint	Inicio	Dias	Final	Product Backlog Item	Descripción de la tarea	Estado	Owner	Horas estimadas	Horas reales
1.1	1	25-09-2024	5	02-10-2024	R1	Crear la estructura principal del front-end.	No Started	Developer	40	
1.2	1	03-10-2024	10	17-10-2024	R1	Implementar el diseño coherente en todas las páginas.(pagina de navegacion,pagina de inicio de sesion)	No Started	Developer	80	
1.3	1	18-10-2024	7	29-10-2024	R1	Configurar la barra de navegación con las secciones necesarias.	No Started	Developer	56	
1.4	1	30-10-2024	5	06-11-2024	R1	Añadir la opción de visualizar y descargar el reglamento.	No Started	QA	40	
1.5	1	07-11-2024	1	08-11-2024	R1	Realizar pruebas de funcionamiento	No Started	QA	8	
2.1	2	12-11-2024	5	19-11-2024	R2	Desarrollar el sistema de autenticación de usuarios.	No Started	Developer	40	
2.2	2	18-11-2024	5	25-11-2024	R2	Implementar el diseño coherente en todas las páginas.(pagina de pago,pagina de arriendo de espacios comunes, pagina de reclamos)	No Started	Developer	40	
2.3	2	26-11-2024	6	04-12-2024	R2	Crear base de datos	No Started	Developer	48	
2.4	2	05-12-2024	5	12-12-2024	R2	Configurar la base de datos para almacenar usuarios y credenciales.	No Started	Developer	40	
2.5	2	13-12-2024	6	23-12-2024	R2	Implementar el control de acceso basado en roles (usuario/administrador).	No Started	QA	48	
2.6	2	24-12-2024	1	25-12-2024	R2	Realizar pruebas de funcionamiento	No Started	QA	8	
3.1	3	26-12-2024	10	09-01-2025	R4	Integrar un sistema de pagos en línea.	No Started	Developer	80	
3.2	3	10-01-2025	5	17-01-2025	R4	Implementar el diseño coherente en todas las páginas de administrador.(pagina de listado de gastos comunes, pagina para acceder a reclamos)	No Started	Developer	40	
3.3	3	20-01-2025	4	24-01-2025	R4	Desarrollar el historial de pagos.	No Started	Developer	32	
3.4	3	27-01-2025	4	31-01-2025	R4	Configurar notificaciones automáticas para recordatorios de deudas, confirmaciones de pago y sobre el proceso de reserva	No Started	Developer	32	
3.5	3	30-01-2025	2	03-02-2025	R4	Desarrollar un calendario interactivo para la reserva de espacios.	No Started	Developer	16	
3.6	3	04-02-2025	3	07-02-2025	R4	Integrar pagos iniciales para las reservas.	No Started	Developer	24	
3.7	3	10-02-2025	1	11-02-2025	R4	Realizar pruebas de funcionamiento	No Started	QA	8	
4.1	4	12-02-2025	5	19-02-2025	R5	Desarrollar el módulo para que los usuarios ingresen reclamos con archivos adjuntos.	No Started	Developer	40	
4.2	4	20-02-2025	5	27-02-2025	R5	Implementar una sección para que los administradores gestionen los reclamos.	No Started	Developer	40	
4.3	4	28-02-2025	3	05-03-2025	R5	Automatizar la asociación de multas a las deudas de gastos comunes.	No Started	Developer	24	
4.4	4	10-03-2025	5	17-03-2025	R5	Ampliar diseño de base de datos que permita guardar los datos de fechas de solicitud,etc	No Started	Developer	40	
4.5	4	17-03-2024	4	21-03-2024	R6	Implementar el sistema de IA para la clasificación de reclamos.	No Started	Developer	32	
4.6	4	24-03-2025	5	31-03-2025	R6	Configurar reglas para la identificación de infractores recurrentes.	No Started	Developer	40	
4.7	4	01-04-2025	1	02-04-2025	R7	Realizar pruebas de funcionamiento	No Started	QA	8	

Tabla 11. Sprint backlog.

Paradigma 4 + 1

Diagrama de clases:

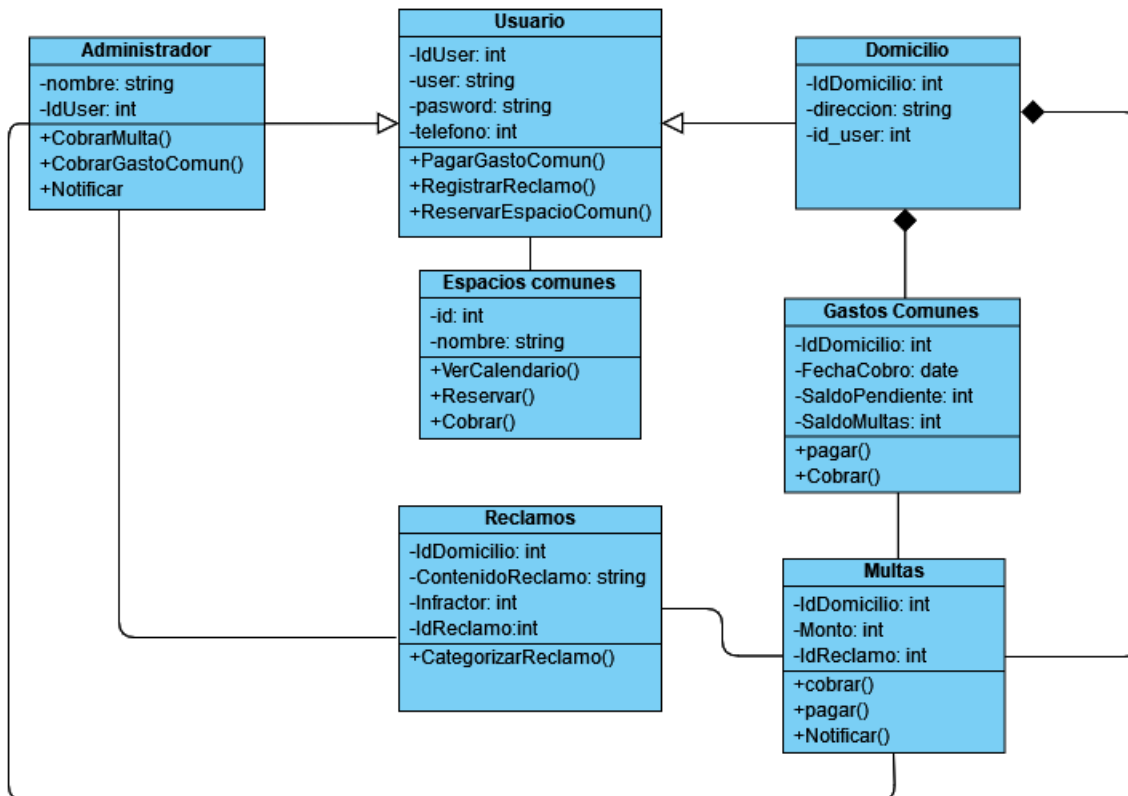


Imagen 7.

Diagrama de secuencia:

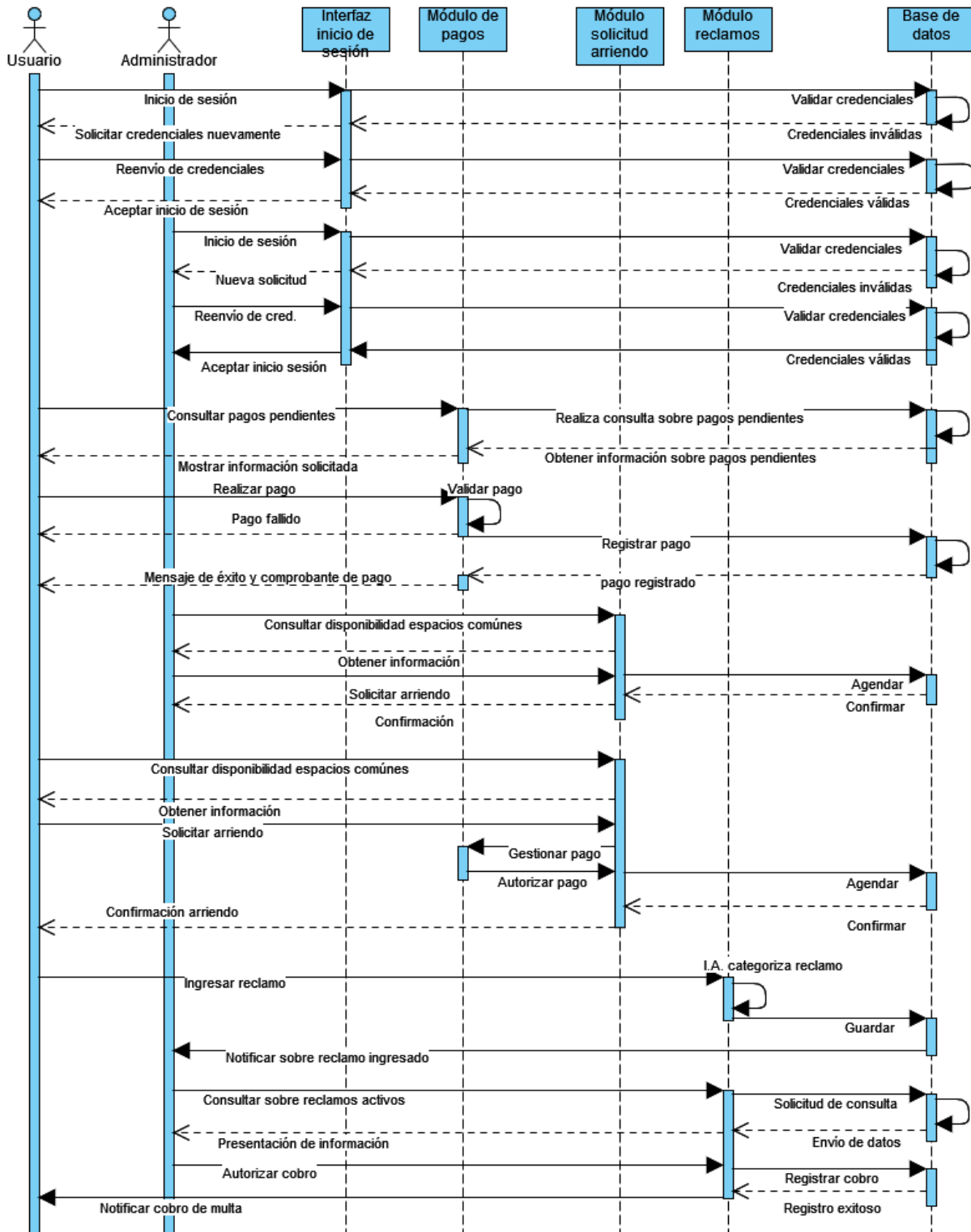


Imagen 8.

Diagrama de componentes:

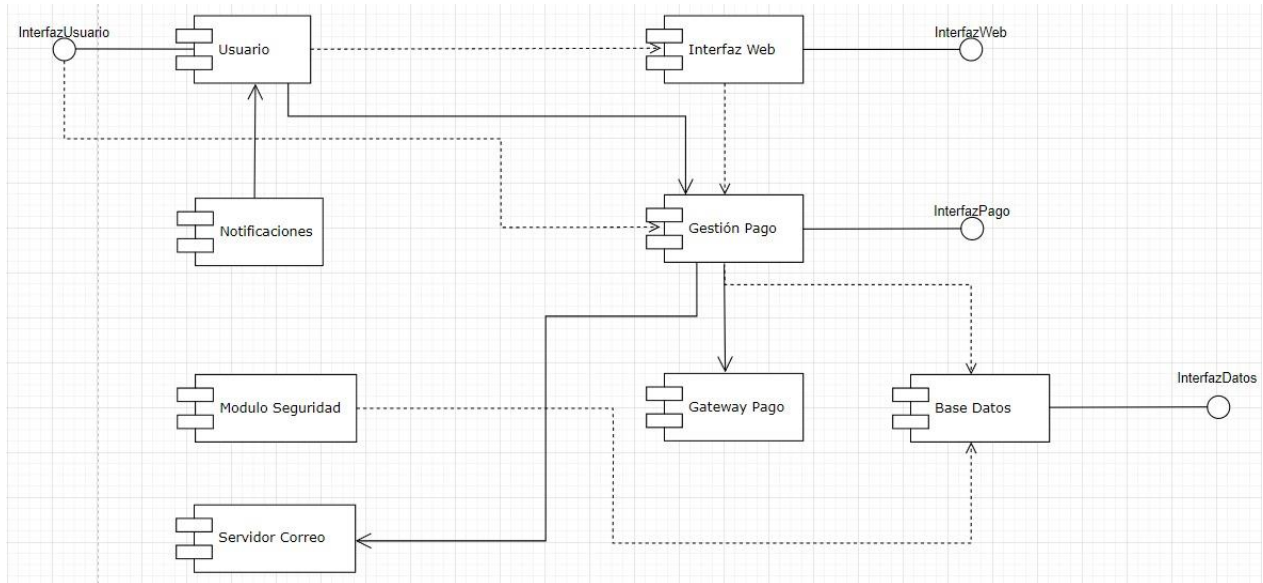


Imagen 9.

Diagrama de paquetes:

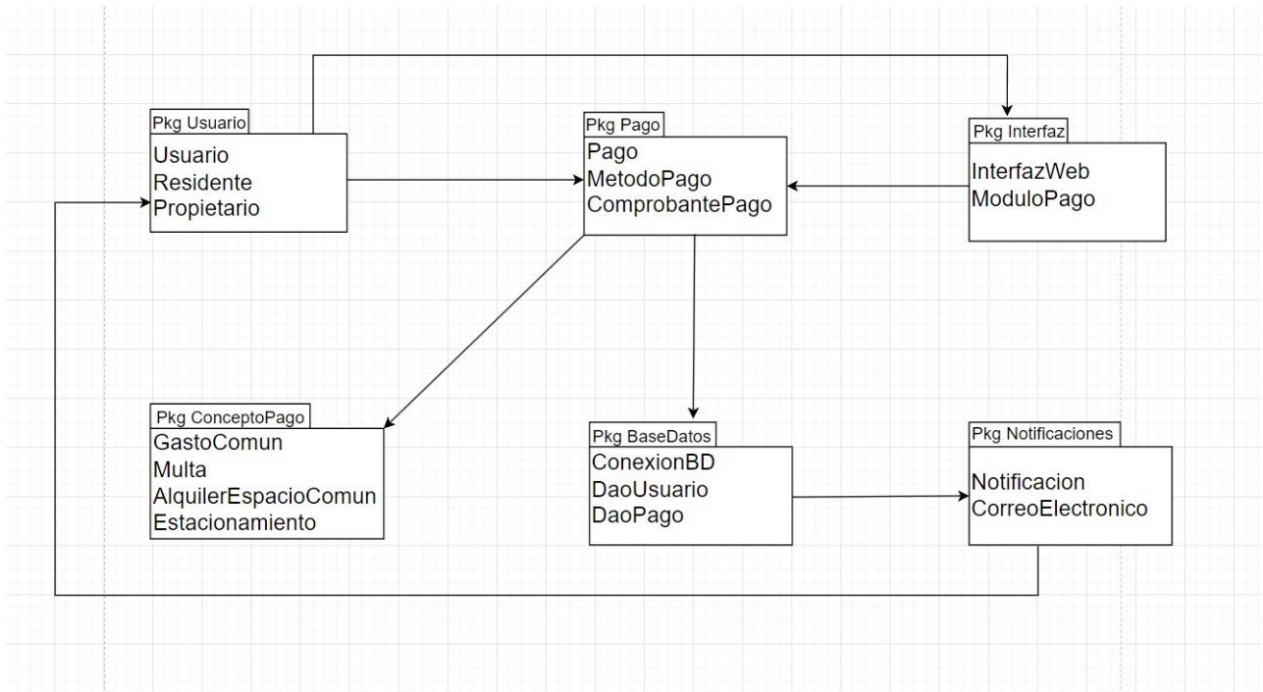


Imagen 10.

Diagrama de actividades:

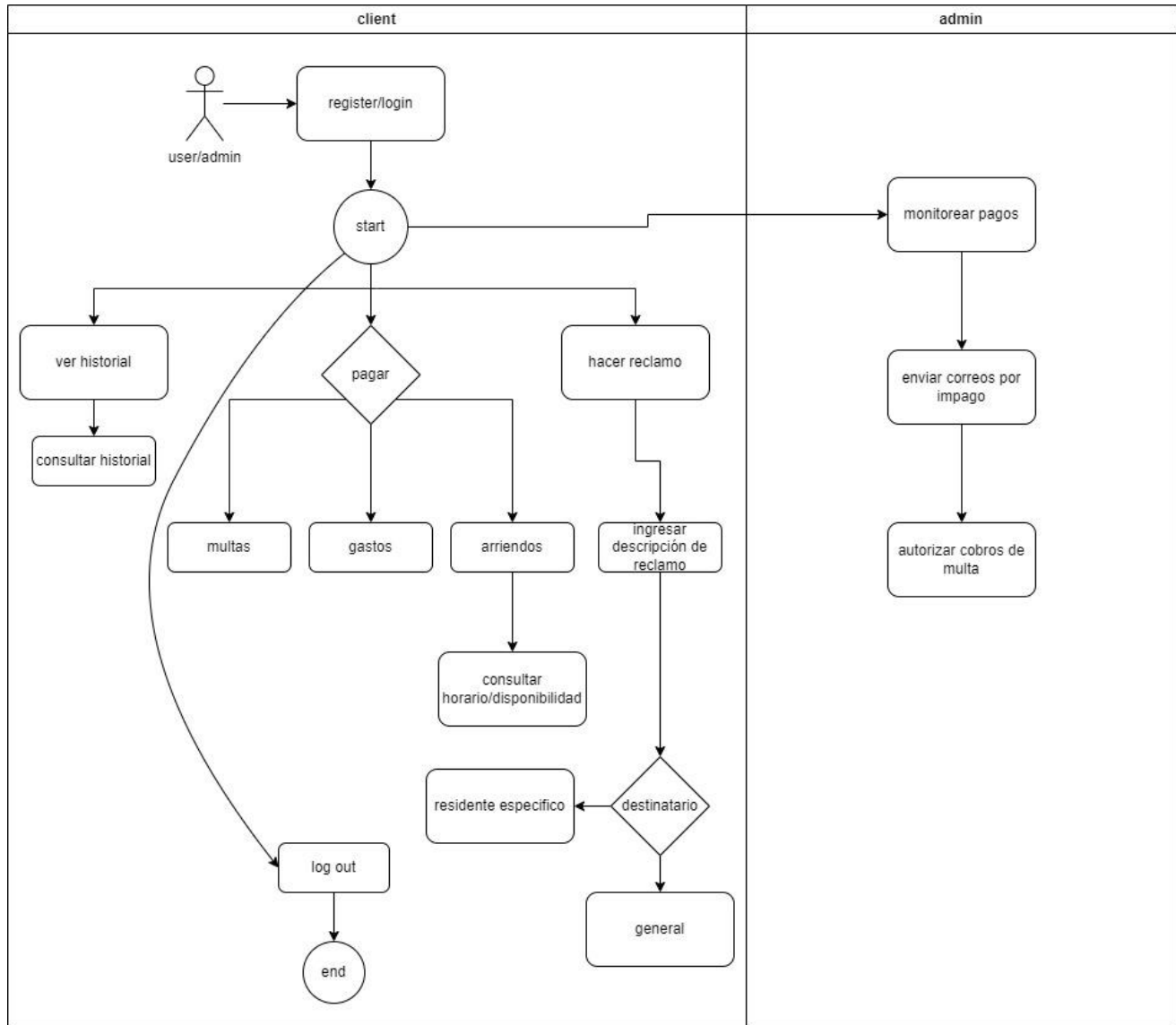


Imagen 11.

Diagramas de casos de uso:

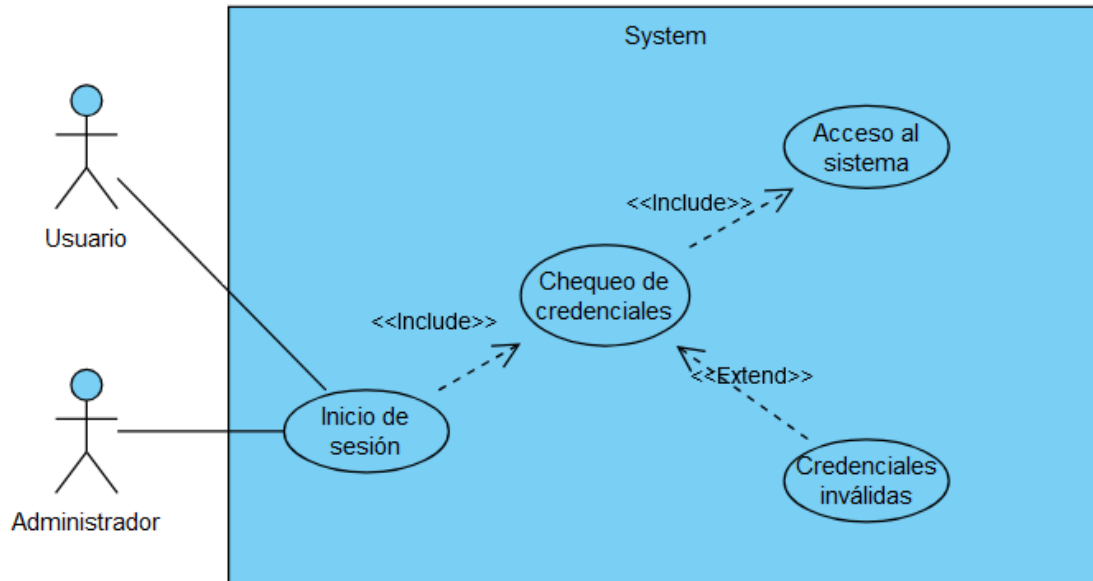


Imagen 12. CU-01

Identificador de caso de uso	CU-01
Nombre	Inicio de sesión
Descripción	Se realiza una validación de credenciales por parte del sistema
Actores	Usuarios, Administrador
Secuencia Normal	
Actor	Software
1. Los usuarios ingresan los datos requeridos	2. El sistema solicita al usuario su nombre de usuario y contraseña.
	3. Validar información solicitada en la base de datos
	4. El caso de uso termina
Excepciones	Software
Error al ingresar los datos	Se muestra mensaje de error indicando credenciales invalidas
	Se regresa al paso 2.
CU relacionados	
Precondición	El actor debe estar registrado en el sistema
Post condición	El actor accede a las funcionalidades del sistema correspondientes al rol de usuario o de administrador.

Tabla 12. CU-01

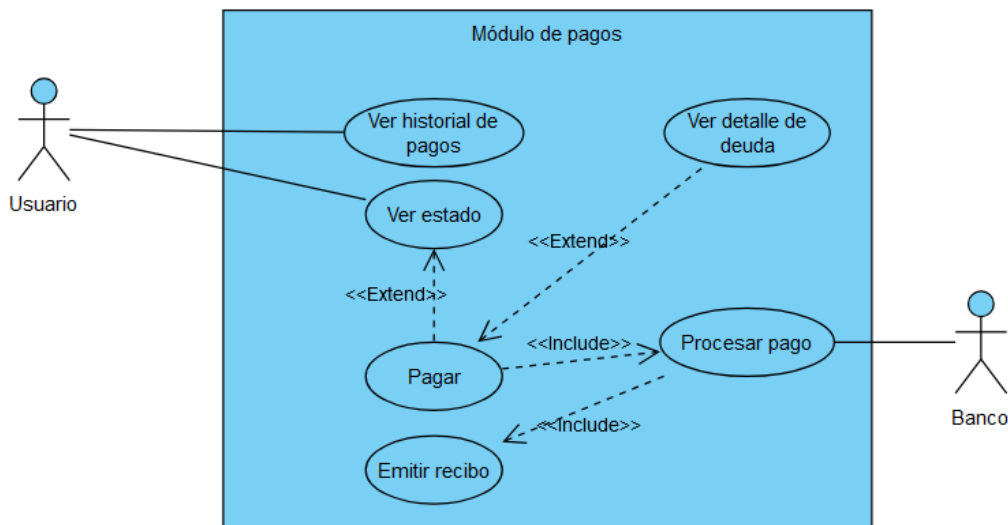


Imagen 12. CU-02

Identificador de caso de uso	CU-02
Nombre	Módulo sistema de pagos
Descripción	El usuario puede ver si tiene mantiene deuda, puede pagar escogiendo un medio de pago que una vez verificado se emitirá un comprobante de pago, adicionalmente el usuario puede revisar su historial de pagos.
Actores	Usuarios
Secuencia Normal	
Actor	Software
1. Usuario ingresa al módulo de pagos	2. El sistema le presenta al usuario el estado de su cuenta y si presenta deudas y opciones de pago.
	3. Al seleccionar pagar se redirige al sitio del sistema de pago escogido.
	4. Tras completar el pago se regresa al paso 6.
5. Usuario ingresa al historial de pagos.	6. El sistema presenta la información solicitada después de consultar la base de datos.
Excepciones	Software
Error al ingresar autenticar el pago	Se muestra mensaje de error indicando el error
	Se regresa al paso 6.
CU relacionados	CU-01
Precondición	El actor debe estar registrado en el sistema
Post condición	El actor accede a las funcionalidades del sistema correspondientes al rol de usuario o de administrador.

Tabla 13. CU-02.

Interfaces

Interfaz de inicio de sesión

The image shows a web browser window displaying a login page. The page has a light orange background. At the top, there is a navigation bar with a 'Login' link and a gear icon on the left, and a search bar with the text 'Buscar' on the right. The navigation bar also includes links for 'Inicio', 'Acerca', 'Features', 'Precios', and 'Historial de Pagos'. The main content area is a white card with the title 'Iniciar Sesión'. It contains two input fields for 'Nombre de usuario' and 'Contraseña'. Below these fields is a radio button labeled 'Recordarme?'. A red button labeled 'Iniciar sesión' is positioned below the radio button. Below the button, there are two links: 'Olvidaste tu contraseña?' and 'No tienes una cuenta? Regístrate aquí'. At the bottom of the card, there is a small text link: 'Al registrarte aceptas nuestra Términos y condiciones'.

Imagen 13.

Interfaz de página principal

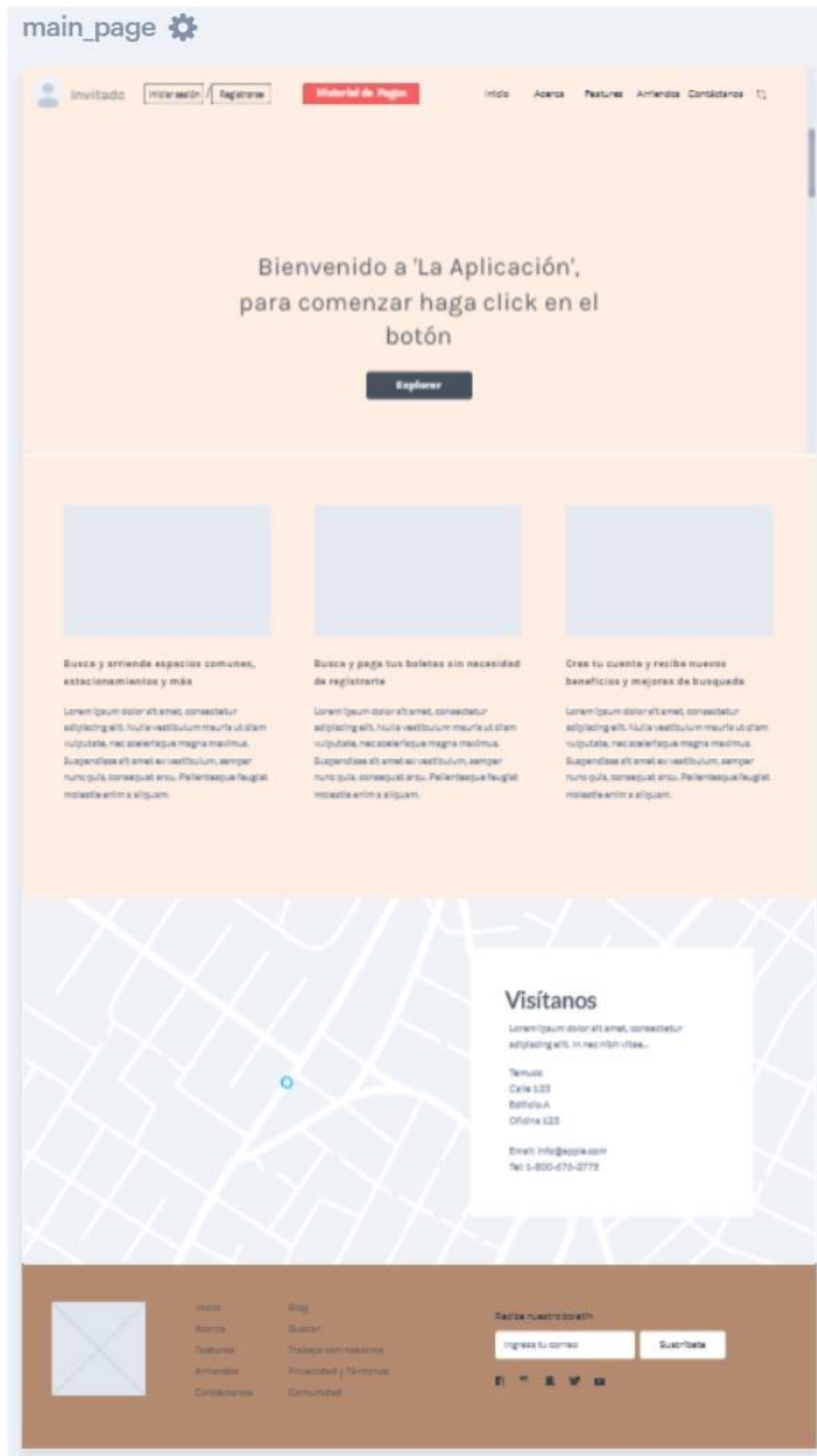




Imagen 14.

interfaz de historial de usuario

history_screen ⚙️

 Inicio Acerca Features Precios **Historial de Pagos** 🔍 Buscar

HISTORIAL DE PAGOS

 RUT
Ingrese su RUT

N°	Condominio/Dpto.	Precio	Fecha	N° Boleta	Filtrar ▼	
1	Honda	15.000.0 CLP	01/11/2025	N°123456789	Ver Detalles	Pagar
2	Toyota	249.990.0 CLP	23/02/2018	N°123456789	Ver Detalles	Pagar
3	Hyundai	990.000.000.0 CLP	25/09/2024	N°123456789	Ver Detalles	Pagar
4	Honda	499.000.0 CLP	06/02/2019	N°123456789	Ver Detalles	Pagar
5	Toyota	9.990.0 CLP	18/12/2023	N°123456789	Ver Detalles	Pagar

1 2 3 ... 20 21 22

Imagen 15.

Modelo de negocios

Propuesta de Valor La plataforma es una solución web diseñada para la gestión integral de condominios y edificios, que centraliza tareas administrativas como la gestión de pagos de gastos comunes, emisión de recibos, manejo de reclamos, y reservas de espacios comunes. Además, la plataforma cuenta con un sistema de inteligencia artificial para la clasificación automática de reclamos, lo que mejora la eficiencia y permite identificar infractores recurrentes.	Segmento de Clientes Esta plataforma está diseñada para distintos usuarios: los propietarios de unidades, que deben gestionar sus pagos de gastos comunes y multas; los inquilinos, que requieren pagar por servicios adicionales y alquileres de espacios; y los administradores de edificios, responsables de la gestión financiera y supervisión de todas las transacciones del condominio.	Canales Se basa en una app web como canal principal para que los usuarios puedan acceder y realizar sus pagos de manera eficiente. Utilizamos el correo electrónico para enviar confirmaciones de pago y recordatorios de vencimientos, asegurando que los usuarios estén siempre informados. Además, ofrecemos soporte técnico mediante chat en vivo o correo electrónico para resolver cualquier duda o problema que puedan tener.	Relación con el Cliente Nuestros principales clientes son administradores de condominios que a su vez son un nexo entre nosotros y los usuarios (Residentes y copropietarios). Nuestra relación con los clientes se refuerza con una atención al cliente dedicada, disponible para solucionar inconvenientes
Fuentes de Ingresos Nuestro modelo de ingresos se basa en comisiones por transacción, aplicando un pequeño porcentaje o una tarifa fija por cada pago realizado a través de la plataforma.		Estructura de Costos Los principales costos de nuestra plataforma incluyen licencias y permisos para utilización de APIs de IA, mantención de servicios basados en nube y auditorías externas al menos una vez al año (gestor de pagos).	
Actividades Clave Nuestro enfoque incluye el desarrollo y mantenimiento constante de la plataforma, lo que implica realizar actualizaciones regulares y resolver problemas técnicos para asegurar un funcionamiento óptimo. Además, nos enfocamos en la gestión de relaciones con los clientes mediante una interacción continua que nos permite comprender mejor sus necesidades y mejorar el servicio de manera proactiva. Finalmente, destinamos esfuerzos a marketing y ventas, promoviendo el sistema en condominios y edificios para ampliar nuestra base de usuarios y aumentar la adopción del servicio.			
Recursos Clave Nuestro equipo de desarrollo está compuesto por programadores (Stack MERN) y diseñadores que se encargan de la creación y optimización de la aplicación, especialistas en DevOps, infraestructura tecnológica como plataformas en la nube, pasarelas de pago y sistemas de seguridad y cifrado.		Socios Clave Colaboramos con proveedores de servicios de pago para transacciones seguras y confiables en nuestra plataforma. Además, establecemos alianzas con administradoras de edificios, quienes juegan un papel clave en la promoción del uso del sistema entre propietarios e inquilinos. Por último, trabajamos importantes empresas a nivel mundial de computación en la nube y proveedores de soluciones de inteligencia artificial	

Tabla 14. Propuesta de negocio.

Patrones de arquitectura de software

Todas las tecnologías mencionadas en la sección **Metodologías y herramientas** facilitan la separación de la lógica siguiendo los patrones **cliente-servidor** y **Modelo-Vista-Controlador (MVC)**, incluso otras tecnologías no mencionadas, pero ampliamente utilizadas en el mercado favorecen la aplicación de los mencionados patrones, tales como: Spring Boot, ASP.NET Core o Symfony.

Ian Sommerville, en su libro *Ingeniería de Software* (2011), describe el patrón cliente-servidor a continuación en la tabla 15.

Nombre	Cliente-servidor
Descripción	En una arquitectura cliente-servidor, la funcionalidad del sistema se organiza en servicios, y cada servicio lo entrega un servidor independiente. Los clientes son usuarios de dichos servicios y para utilizarlos ingresan a los servidores.
Cuando se usa	Se usa cuando, desde varias ubicaciones, se tiene que ingresar a los datos en una base de datos compartida. Como los servidores se pueden replicar, también se usan cuando la carga de un sistema es variable.
Ventajas	La principal ventaja de este modelo es que los servidores se pueden distribuir a través de una red. La funcionalidad general (por ejemplo, un servicio de impresión) estaría disponible a todos los clientes, así que no necesita implementarse en todos los servicios.
Desventajas	Cada servicio es un solo punto de falla, de modo que es susceptible a ataques de rechazo de servicio o a fallas del servidor. El rendimiento resultará impredecible porque depende de la red, así como del sistema. Quizás haya problemas administrativos cuando los servidores sean propiedad de diferentes organizaciones.
Ejemplos	La imagen 16 es un ejemplo de una filmoteca y videoteca (videos/DVD) organizada como un sistema cliente-servidor.

Tabla 15. Patrón cliente-servidor. Sommerville, I. (2011). *Ingeniería de Software* (9na ed.)

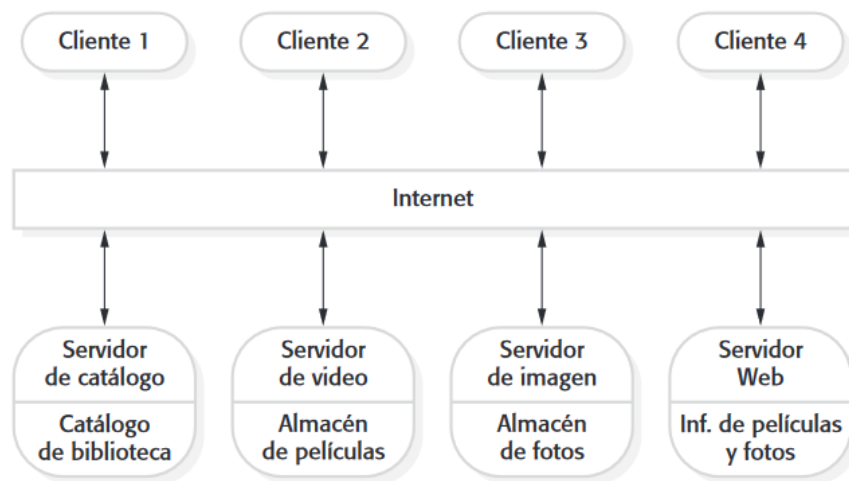


Imagen 16. Arquitectura cliente-servidor para una filmoteca. Sommerville (2011)

El patrón cliente-servidor separa las responsabilidades entre dos tipos principales de componentes: Clientes (que solicitan servicios) y servidores (que responden a estas solicitudes proporcionando servicios), los **clientes** suelen ser interfaces de usuario o aplicaciones que interactúan con el usuario final, este patrón es altamente escalable porque se pueden agregar más clientes o servidores según sea necesario.

Sommerville (2011) también describe el Modelo-vista-controlador tal y como se resume en la tabla 16.

Nombre	MVC (modelo de vista del controlador)
Descripción	Separa presentación e interacción de los datos del sistema. El sistema se estructura en tres componentes lógicos que interactúan entre sí. El componente Modelo maneja los datos del sistema y las operaciones asociadas a esos datos. El componente Vista define y gestiona cómo se presentan los datos al usuario. El componente Controlador dirige la interacción del usuario (por ejemplo, teclas oprimidas, clics del mouse, etcétera) y pasa estas interacciones a Vista y Modelo. Véase la imagen 17 .
Cuando se usa	Se usa cuando existen múltiples formas de ver e interactuar con los datos. También se utiliza al desconocerse los requerimientos futuros para la interacción y presentación.
Ventajas	Permite que los datos cambien de manera independiente de su representación y viceversa. Soporta en diferentes formas la presentación de los mismos datos, y los cambios en una representación se muestran en todos ellos.
Desventajas	Puede implicar código adicional y complejidad de código cuando el modelo de datos y las interacciones son simples.
Ejemplos	La imagen 18 muestra la arquitectura de un sistema de aplicación basado en la Web, que se organiza con el uso del patrón MVC

Tabla 16. Patrón modelo de vista del controlador (MVC). Sommerville (2011).

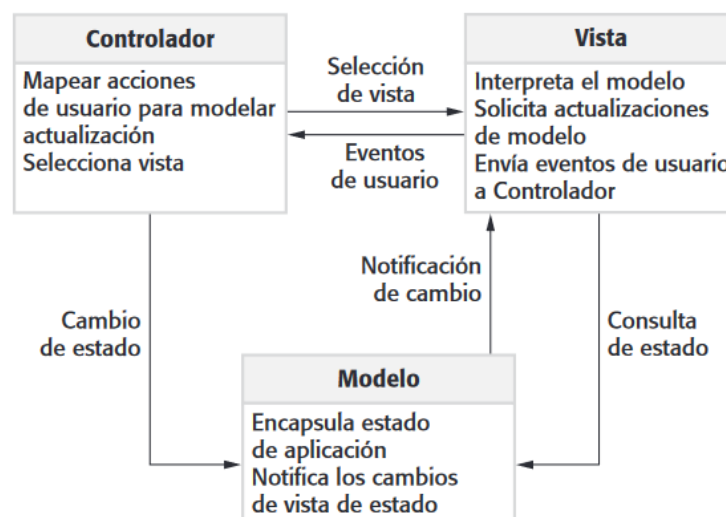


Imagen 17. La organización del MVC. Sommerville (2011).

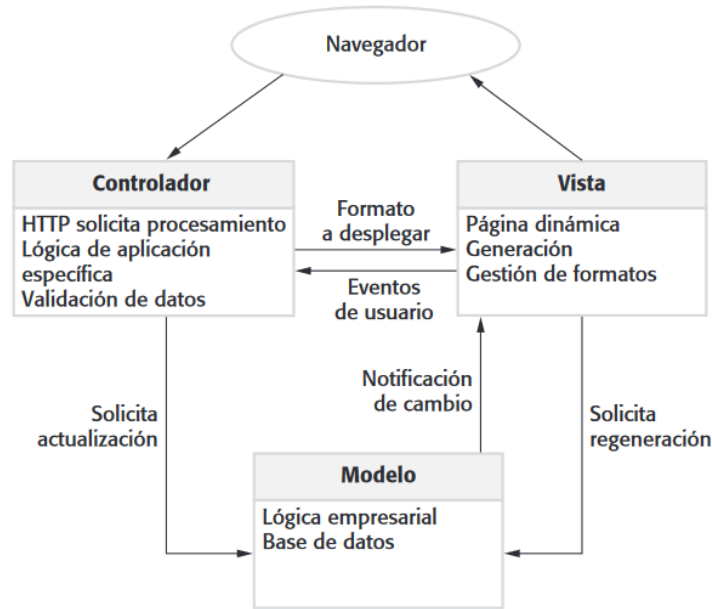


Imagen 18. Arquitectura de aplicación web con el patrón MVC. Sommerville (2011).

Ambos patrones pueden usarse juntos, aunque tienen objetivos y niveles de abstracción distintos, por ejemplo, el MVC es un patrón de arquitectura interna, mientras que el cliente-servidor es una arquitectura de red o distribución. Sin embargo, ambos permiten una mayor escalabilidad, flexibilidad y modularidad.

Componente	Herramienta
Modelo	MongoDB
Vista	React
Controlador	Node.js, Express

Tabla 17. Aplicación del MVC.

Perfiles y roles del sistema

Rol	Descripción	Privilegios de acceso
Administrador del sistema	Sera el usuario con mayor privilegio ya que estará a cargo de gestionar el software a nivel técnico y de configuración general.	<ul style="list-style-type: none"> Gestionar roles y permisos Configurar seguridad y autenticación. Acceder a todos los datos y logs del sistema
Administrador del condominio	Es el responsable de la gestión operativa del condominio esto incluye la revisión y aprobación de los pagos, la resolución de reclamos y la administración de reclamos	<ul style="list-style-type: none"> Gestionar pagos Administrar reclamos (clasificar, cerrar, sancionar) gestionar espacios comunes
Residente registrado	Son los propietarios o arrendatarios con usuarios en el sistema, estos usuarios pueden realizar pagos, visualizar su historial de transacciones, generar reclamos y reservar espacios comunes.	<ul style="list-style-type: none"> Realizar y visualizar pagos Generar y gestionar reclamos personales Reservar y cancelar espacios comunes
Usuario invitado	Son usuarios no registrados por lo tanto solo pueden realizar pagos únicos y recibir el comprobante de pago en su correo electrónico.	<ul style="list-style-type: none"> Realizar pagos puntuales Realizar comprobantes de pago

Tabla 18. Resumen de roles.

Buenas prácticas de programación segura

Requisitos de seguridad:

- Implantar autenticación basada en tokens JWT para usuarios registrados y administradores. *OWASP Foundation, 2023.*
- Usar HTTPS para todas las comunicaciones para garantizar la confidencialidad de los datos. *Schneier, 2015.*
- Asegurar la protección contra ataques comunes como SQL Injection, XSS y CSRF *OWASP Foundation, 2023.*

Validaciones de entrada:

- Validar y limpiar todos los datos ingresados por el usuario para prevenir inyecciones de código *OWASP Foundation, 2023.*
- Establecer reglas claras para los campos obligatorios, como formato de correo electrónico, contraseñas seguras etc. *Nilsen, 1995.*

Control de acceso:

- Aplicar el principio de menos privilegio en todas las operaciones según los roles definidos *ISO/IEC, 2013.*
- Implementar políticas de cloqueo tras múltiples intentos fallidos de inicio de sesión. *ISO/IEC, 2013.*

Simplicidad den el diseño:

- Crear interfaces intuitivas y flujos simples para los usuarios finales. *Nielsen, 1995.*
- Dividir responsabilidades en componentes modulares y bien documentados en el código. *Sommerville, 2015.*

Manejo de errores:

- Registrar errores críticos en el servicio para análisis sin revelar la información sensible al usuario. *Schneier, 2015.*
- Proporcionar mensajes de error claros y específicos. *Nielsen, 1995.*

Auditorias y logs:

- Registrar las actividades críticas, como inicio de sesión, pagos y gestión de reclamos. *ISO/IEC, 2013.*
- Monitorear y analizar los logs periódicamente en busca de posibles anomalías. *ISO/IEC, 2013.*

Actualizaciones y mantenimiento:

- Mantener actualizadas las dependencias y librerías del proyecto. *Sommerville, 2015.*
- Realizar Pruebas periódicas y de penetración y revisión de código para identificar vulnerabilidades. *OWASP Foundation, 2023.*

Servicio de almacenamiento Cloud

El almacenamiento en la nube o Cloud Storage podrá permitir almacenar datos y archivos en internet a través de proveedores de computación en la nube que brinden estos servicios, a la cual se accede mediante red pública o red privada dedicada. El Cloud o almacenamiento en la nube permite mayor flexibilidad, fiabilidad, aumentando el rendimiento y la eficacia, reduciendo los costes TI, según el almacenamiento de nube de Google; “mejora la innovación, ya que las empresas pueden agilizar el tiempo de lanzamiento e incorporar casos prácticos de IA y aprendizaje automático en sus estrategias.” (Google Cloud). Algunos de los proveedores de estos servicios tal como el Cloud de Google tienen diferentes características y algunos suelen ser de pago o gratuitos, pero también de que ofrecen al usuario distintos planes para mejorar aún más el servicio.

Los principales proveedores del mundo del Cloud, además del anteriormente mencionado Google Cloud, tenemos a AWS de Amazon, Azure de Microsoft, entre otros. Para escoger el servicio adecuado para el proyecto se ha realizado una investigación de los servicios de cloud que ofrecen los proveedores.

Google Cloud Platform (Google)

El GCP proporciona una amplia gama de servicios: IaaS o Infrastructure as a Service en el cual permite la creación de máquinas virtuales montadas dentro de la nube de Google, la ejecución de aplicaciones utilizando servicio GKE (Google Kubernetes Engine).

PaaS o Platform as a Service que proporciona una variedad de bases de datos relacionales y no relacionales (Cloud SQL), almacena datos de forma segura y escalable con los servicios de BD y extracción de valor de datos usando servicios como BigQuery.

SaaS o Software as a Service que implementa herramientas de desarrollo y inteligencia artificial desarrollando modelos de machine learning, aprovechando APIs para tareas de visión por computadora, procesamiento de lenguaje natural y más.

Google nos da una suite completa de servicios en la nube que está diseñada para ayudar a las empresas o microempresas, desarrolladores y organizaciones a ser más eficientes proporcionando acceso a la misma infraestructura y tecnología utilizada por el mismo Google para operar servicios propios, de estos están: Motor de Búsqueda, Gmail y YouTube, con los cuales se pueden aprovechar la escalabilidad, la fiabilidad y la seguridad de Google para aplicaciones propias.

AWS (Amazon Web Services)

Amazon Web Service o comúnmente conocida como AWS, es una de las plataformas de servicio cloud más grande y completa existente en el mundo ofrecida por la misma Amazon que te permite tener un propio centro de datos sin necesidad de mantenciones o equipos y que proporciona recursos de forma virtual y bajo demanda. AWS cuenta con una variada cantidad de servicios y características a disponer tanto como una base de datos, cómputo, almacenamiento en general y tales como la misma inteligencia artificial (IA) o análisis e internet de las cosas, “AWS ofrece la más amplia variedad de bases de datos que están diseñadas especialmente para diferentes tipos de aplicaciones, de modo que usted puede elegir la herramienta adecuada para el trabajo a fin de obtener el mejor costo y rendimiento.” (AWS, Amazon)

Este proveedor de cloud ofrece tanto como:

- Escalabilidad
- Flexibilidad
- Fiabilidad
- Seguridad
- Alta gama de servicios

Azure (Microsoft)

El cloud de red publica o privada de Microsoft, Azure, permite a los usuarios compilar, implementar y administrar de forma rápida aplicaciones de red global de datacenters de Microsoft ofreciendo variados servicios que están destinados a ayudar a cualquier tipo de empresa para confrontar cualquier tipo de adversidades actuales o futuras. Azure dispone de servicios de infraestructura y plataforma (IaaS y PaaS) compatibles para tecnologías como Oracle, Linux, php, MySQL, entre otros. Azure es, según, la única nube híbrida coherente Ofrece una productividad de desarrollo sin parangón, proporciona una seguridad completa en varias capas, incluida una cobertura de cumplimiento normativo mayor que la que ofrece cualquier otro proveedor de nube y, además, Azure es más barato, además de ofrecer una prueba gratuita, ya que AWS es cinco veces más caro que Azure para Windows Server y SQL Server, y también la seguridad y la privacidad son fundamentales para Azure. Microsoft se compromete a ofrecer los mayores niveles de confianza, transparencia y cumplimiento de estándares y normativo, con un conjunto de ofertas de cumplimiento más completo que el que pueda ofrecer cualquier otro proveedor de servicios en la nube.

Característica	Amazon Web Services	Microsoft Azure	Google Cloud Platform
Mercado	Líder del mercado, amplia gama de servicios y mayor madurez.	Segundo mayor proveedor, fuerte integración con productos Microsoft.	Tercer mayor proveedor, enfoque en machine learning y análisis de datos.
Fortalezas	Amplia gama de servicios, precios competitivos, gran comunidad y ecosistema de partners.	Integración con productos Microsoft, seguridad robusta, soluciones híbridas.	Excelencia en machine learning, análisis de big data, contenedores y Kubernetes.
Debilidades	Interfaz puede ser compleja para principiantes, gestión de costos puede ser complicada.	Menos maduro que AWS en algunos servicios, precios pueden ser más altos.	Menor cuota de mercado, ecosistema de partners más pequeño.
Público objetivo	Empresas de todos los tamaños, desarrolladores, startups.	Empresas que utilizan productos Microsoft, desarrolladores .NET.	Empresas que buscan soluciones de análisis de datos y machine learning, startups tecnológicas.
Servicios destacados	EC2 (cómputo), S3 (almacenamiento), RDS (bases de datos), Lambda (serverless), AI Services.	Azure Virtual Machines, Azure SQL Database, Azure Kubernetes Service, Azure Cognitive Services.	Google Compute Engine, Google Cloud Storage, Cloud SQL, Kubernetes Engine, TensorFlow.
Precios	Modelo de pago por uso, precios competitivos, múltiples opciones de reserva.	Modelos de pago por uso y por suscripción, precios competitivos, ofertas especiales para clientes de Microsoft.	Modelos de pago por uso, precios competitivos, descuentos por uso sostenido.

Tabla 19. Tabla comparativa servicios cloud.

La mejor opción sería **Google Cloud Platform (GCP)**, ya que ofrece herramientas avanzadas de machine learning, escalabilidad eficiente y costos competitivos, aspectos que se alinean perfectamente con los requerimientos del proyecto, especialmente en la automatización y análisis de datos para la clasificación de reclamos. **GCP** también se encuentra dentro de los conocimientos y experiencias de los integrantes del equipo de desarrollo.

Plan de pruebas

Criterios de aceptación

Historia de usuario	Criterio de aceptación
HU01	El tema de la aplicación web debe mantenerse a través de las distintas paginas
	El diseño debe ser minimalista y coherente para facilitar su uso.
HU02	El sistema debe permitir ingresar credenciales válidas para acceder al sistema.
	Se debe mostrar un mensaje de error si las credenciales son inválidas.
	Los usuarios autenticados deben acceder a las funciones de acuerdo con su rol.
HU03	El sistema debe habilitar el pago del último mes de gastos comunes cuando corresponda.
	Las deudas pendientes de gastos comunes deben bloquear el pago de meses actuales.
	Multas y cobros adicionales deben estar disponibles para pago, pero su falta de pago no debe impedir el pago de gastos comunes.
HU04	El sistema debe permitir revisar la disponibilidad de espacios comunes a través de un calendario.
	Los usuarios deben poder seleccionar fecha y hora para realizar una reserva.
	Se debe acreditar un pago inicial para confirmar la reserva.
	El sistema debe enviar notificaciones sobre los pasos a seguir (retiro de llave, firma de documentos).
HU05	Los usuarios deben poder ingresar un reclamo con texto e imágenes/videos adjuntos.
	Se debe permitir a los usuarios revisar y ampliar información de reclamos anteriores.
	Los usuarios deben poder ver los reclamos/multas en su contra.
HU06	La barra de navegación debe estar accesible en todas las páginas después del inicio de sesión.
	La barra debe incluir una sección de notificaciones y una opción para cerrar sesión.
HU07	Los usuarios deben poder ver y descargar el reglamento de convivencia desde la plataforma.
HU08	Los usuarios deben recibir alertas al alcanzar tres meses de morosidad.
	Se deben enviar alertas cada vez que los usuarios realicen un pago o cometan una infracción que conlleve multa.
HU09	Los administradores deben poder listar todos los domicilios con su estado de pago en un formato tipo planilla.
	Al hacer clic en un domicilio, se debe mostrar toda la información correspondiente y permitir la cancelación de deudas si es necesario.
HU10	Los administradores deben poder clasificar reclamos y enviar notificaciones/correos al infractor si procede el cobro.
	Las multas asociadas deben cargarse automáticamente al mes siguiente salvo pago previo.

HU11	El sistema debe clasificar los reclamos en categorías automáticamente (e.g., estacionamiento, ruidos).
	El sistema debe identificar infractores recurrentes basándose en historial de reclamos.

Tabla 20. Criterios de aceptación.

Identificación de funcionalidades

Ítem	Funcionalidades	Historias de usuario
Gestión de usuario	Autenticación, roles y acceso al resto de funciones.	HU2
Gestión de pagos	Pagos pendientes, históricos y bloqueos de transacciones según reglas de negocio.	HU3
Reservas	Consulta de disponibilidad, creación y gestión de reservas	HU4
Reclamos	Registro, seguimiento, y clasificación automática mediante IA	HU5, HU10, HU11
Notificaciones	Alertas sobre pagos, infracciones, y procesos de reclamos	HU08
Gestión administrativa	Listado y control de estados financieros y deudas	HU09

Tabla 21. Funcionalidades.

Estrategia de pruebas

Objetivos	Tipo de prueba	Descripción
<ul style="list-style-type: none"> identificar errores verificar la funcionalidad 	Funcionales	Validan el cumplimiento de los criterios de aceptación para cada funcionalidad.
	De integración	Evalúan la interacción entre los módulos del sistema (por ejemplo: frontend-backend, API-base de datos)
	De regresión	Garantizan que los cambios realizados no afecten funcionalidades existentes.
	De usabilidad (no funcional)	Verifican que la interfaz sea intuitiva y cumpla con los requisitos visuales y de navegación.
	De rendimiento (no funcional)	Miden tiempos de respuesta y comportamiento bajo carga simulada.
	De seguridad (no funcional)	Detectan vulnerabilidades como inyección de datos y validan la protección de información sensible.

Tabla 22. Tipos de pruebas a realizar.

Casos de prueba

ID Prueba	Historia de usuario	Descripción	Tipo de prueba
P01	HU02	Validar inicio de sesión con credenciales correctas.	Funcional
P02	HU01	Comprobar que el tema de la aplicación web se conserva de forma coherente en las distintas páginas y esta sea fácil e intuitivo de usar.	Usabilidad
P03	HU02	Intentar inicio de sesión con credenciales incorrectas y verificar mensaje de error.	Funcional
P04	HU03	Comprobar que el pago de gastos comunes esté habilitado solo cuando no existan deudas pendientes.	Funcional, Integración
P05	HU04	Reservar un espacio común seleccionando fecha y hora disponibles en el calendario.	Funcional
P06	HU05	Ingresar un reclamo con texto e imágenes adjuntas y verificar almacenamiento correcto.	Funcional
P07	HU06	Verificar que la barra de navegación esté accesible en todas las vistas después del inicio de sesión.	Funcional
P08	HU08	Simular tres meses de morosidad y comprobar que se envíe una alerta.	Funcional
P09	HU09	Listar el estado de gastos comunes de todos los domicilios y acceder a los detalles de un domicilio específico.	Funcional, integración
P10	HU10	Procesar un reclamo administrativo y enviar una notificación al infractor correspondiente.	Funcional, integración
P11	HU11	Verificar la clasificación automática de reclamos en categorías y detección de infractores recurrentes.	Funcional, seguridad
P12	HU03, HU08	Simular múltiples usuarios realizando pagos simultáneos y medir tiempos de respuesta.	Rendimiento

Tabla 23. Casos de prueba.

Cronograma de pruebas

Fase	Actividades	Duración
Fase 1: Pruebas unitarias	Validación de componentes individuales.	2 semanas.
Fase 2: Pruebas de integración	Evaluación de interacción entre módulos.	1 semana.
Fase 3: Pruebas de sistema	Pruebas funcionales y regresión.	1 semana.
Fase 4: Pruebas de rendimiento	Simulación de carga y estrés en la plataforma.	1 semana.

Tabla 24. Cronograma de pruebas.

Conclusiones

El desarrollo de esta plataforma proporciona una solución integral para la gestión de pagos, cobros y reclamos en condominios y edificios, mejorando la eficiencia administrativa y optimizando la experiencia de los residentes. La inclusión de inteligencia artificial para la clasificación de reclamos representa un avance importante en la automatización de tareas repetitivas y el manejo de infracciones recurrentes, permitiendo a los administradores actuar de manera más ágil y precisa. Con esta plataforma, los problemas inherentes a los métodos manuales son minimizados, facilitando un sistema moderno, transparente y fácil de usar para todos los usuarios.

Además, se definió la arquitectura de la solución con un enfoque en los patrones **Modelo-Vista-Controlador (MVC)** y **Cliente-Servidor**, garantizando modularidad, escalabilidad y mantenimiento. Se seleccionó **Google Cloud Platform (GCP)** como proveedor de servicios en la nube, destacando por su eficiencia en análisis de datos, escalabilidad dinámica y costos competitivos.

El diseño del plan de pruebas asegura que cada funcionalidad cumpla con los criterios de aceptación establecidos, evaluando la calidad del sistema mediante pruebas funcionales, de integración, rendimiento y seguridad. De esta manera, la plataforma se establece como un sistema moderno, robusto y adaptable a las necesidades de las comunidades que la utilicen.

Referencias bibliográficas

- Asana. (n.d.). **Waterfall vs. Agile vs. Kanban vs. Scrum.** Asana. <https://asana.com/es/resources/waterfall-agile-kanban-scrum>
- IBM. (n.d.). **REST APIs.** IBM. <https://www.ibm.com/es-es/topics/rest-apis>
- Kinsta. (2023). **Qué es NPM.** Kinsta. <https://kinsta.com/es/base-de-conocimiento/que-es-npm/>
- MercadoPago. (n.d.). **Developers - MercadoPago.** MercadoPago. <https://www.mercadopago.cl/developers/es>
- MongoDB. (n.d.). **MERN Stack.** MongoDB. <https://www.mongodb.com/resources/languages/mern-stack>
- Transbank. (n.d.). **Documentación de Transbank.** Transbank Developers. https://www.transbankdevelopers.cl/documentacion/como_empezar#b-usando-un-sdk
- Sommerville, I. (2011). Ingeniería de Software 9ª ed.

Anexo A

Encuesta a residentes

1. Actualmente, ¿Que método de pago utiliza más frecuentemente para cancelar los gastos comunes?

- a. Depósito bancario
 - b. Efectivo
 - c. Transferencia electrónica
 - d. Cheque
 - e. Otro
2. ¿Usted conoce el proceso y costos asociados para arrendar espacios comunes en su condominio o edificio?
 - a. Sí, he arrendado espacios anteriormente
 - b. Sí, pero nunca he arrendado espacios
 - c. No
3. ¿Usted tiene a mano o sabe dónde encontrar el reglamento de convivencia en donde se explicitan que acciones constituyen multas y las sanciones correspondientes?
 - a. Si
 - b. No
 - c. No sabía que había un reglamento
4. ¿Usted está en conocimiento de la existencia de aplicaciones que gestionan el pago de gastos comunes y otras funciones en un condominio/edificio?
 - a. Si
 - b. No
5. Del 1 al 5, donde 1 es muy en desacuerdo y 5 muy de acuerdo. ¿Usted ha desistido alguna vez de arrendar espacios comunes disponibles en su condominio/edificio por desconocer el proceso de arriendo, los montos y/o condiciones?
 - a. 1 b. 2 c. 3 d. 4 e. 5
6. Del 1 al 5, donde 1 es muy en desacuerdo y 5 muy de acuerdo. ¿Usted ha desistido alguna vez de interponer un reclamo por una infracción al reglamento de convivencia y copropiedad o por sentirse pasada/o a llevar como persona en su condominio/edificio por desconocer el proceso y/o condiciones?
 - a. 1 b. 2 c. 3 d. 4 e. 5
7. Aproximadamente, ¿cuánto tiempo lleva viviendo en comunidad? (Considere el tiempo total acumulado si es que no fueron años o meses continuos y no solo su comunidad actual)
 - a. Menos de 1 año.
 - b. Entre 1 y 2 años.
 - c. Entre 2 y 3 años.
 - d. Entre 3 y 4 años.
 - e. Mas de 4 años.

Entrevista a usuarios

1. ¿Por qué medios prefieres acceder a la plataforma para realizar tus pagos?

Suelo utilizar un notebook en mi día a día por lo que cada vez que hago un pago o una compra “grande” o importante lo hago desde un navegador web, también utilizo bastante mi teléfono celular por lo que sí puedo acceder a la plataforma desde ahí me sería muy cómodo, sobre todo si quiero revisar algo rápido y puntual.

2. ¿Cómo crees debería ser el proceso de pago de los gastos comunes?

Me gustaría poder ver claramente mis pagos pendientes y poder pagarlos sin problemas. Sería útil que el sistema me permita cancelar los gastos comunes del mes y que también pueda ver multas o cobros adicionales, aunque prefiero que estos no interfieran con el pago de los gastos comunes.

3. ¿Qué esperas del sistema de reservas de espacios comunes?

Lo ideal sería poder ver un calendario o una lista donde pueda revisar la disponibilidad de los espacios, como el salón o la piscina, y poder reservar en línea fácilmente. También me gustaría que el sistema me informe de los pasos a seguir, como los pagos adicionales o la retirada de llaves.

4. ¿Qué importancia le das al acceso al reglamento de convivencia?

Para mí, es importante tener acceso rápido al reglamento de convivencia. Me gustaría poder verlo en la plataforma y poder descargarlo si lo necesito en PDF o algo similar, me ha pasado que conversando con vecinos ellos no tenían idea de la existencia de un reglamento así, solo sabían que no debían atrasarse en gastos comunes por que podrían recibir un corte de energía.

5. ¿Qué tipo de notificaciones prefieres recibir desde la plataforma?

Bueno, quisiera recibir notificaciones cuando esté atrasado en mis pagos, por ejemplo, si llevo 2 meses o más de morosidad. También me parecería útil recibir una confirmación cada vez que hago un pago, y notificaciones si cometo alguna infracción que conlleve una multa, o recibir

6. ¿Cómo te gustaría poder revisar el historial de tus pagos?

Me gustaría que el sistema me permita ver fácilmente mi historial de pagos, con un resumen claro de cada mes y los detalles de cualquier multa o cobro adicional que se haya hecho. Sería ideal tener todo esto organizado en una sola sección de la plataforma.

7. ¿Qué herramienta o función consideras que sería importante realizar desde la plataforma?

Me gustaría poder hacer el reclamo de forma fácil, actualmente el proceso es con lápiz y papel y funciona más o menos así: Primero debo dirigirme a la recepción y solicitar un libro de reclamos, anoto fecha, hora, describo la situación en detalle y termino firmando, si quiero ver resultados debo esperar al menos un mes para que se contabilicen todos los reclamos y se procedan a cursar multas, que muchas veces se cursan ya 2 meses después del incidente. Si pudiera llenar un formulario donde pueda detallar el problema y adjuntar fotos o videos si es necesario. Que eso le llegue a administración y que ahí lo gestionen para cursar la multa.

8. ¿Te parece útil recibir notificaciones sobre el estado de tus reclamos?

Sí, definitivamente. Quiero recibir notificaciones sobre cualquier actualización en mis reclamos, ya sea que se esté procesando o si requiere de algún pago si es que fuese en mi contra.

9. ¿Cómo prefieres navegar por las distintas funciones de la plataforma?

Me gustaría que, después de iniciar sesión, la plataforma tenga una barra de navegación clara donde pueda acceder fácilmente a todas las funciones, como los pagos, las reservas y los reclamos. También me parece importante que incluya una opción para cerrar sesión de forma rápida y sencilla.

Entrevista a administrador

1. ¿Cómo debería ser el diseño y la navegación de la plataforma para los usuarios?

Me gustaría que la plataforma sea intuitiva y fácil de entender para todos los residentes. No solicito ningún color en específico solo que el diseño debe ser coherente en todas las páginas, manteniendo un estilo minimalista para no abrumar al usuario.

2. ¿Cómo deberían los residentes acceder al sistema para pagar sus gastos comunes? ¿Cómo se registran?

Una vez el condominio y su junta de vecinos decidan utilizar la plataforma se procedería a la individualización de cada domicilio mediante la toma de datos del propietario o un representante de este, luego se envía por correo electrónico un enlace o credenciales de acceso dando por termino al proceso de registro. Después los residentes deberían poder iniciar sesión con sus credenciales, es decir, el usuario y la contraseña. Sin embargo, también me gustaría que puedan hacer pagos como invitados por sí este lo realiza alguien distinto al propietario, por olvido de contraseña o rapidez.

3. ¿Cómo se deben gestionar los pagos de gastos comunes y otros cobros?

Me gustaría que los residentes puedan pagar los gastos comunes mensuales fácilmente. Si tienen pagos pendientes, como cuotas anteriores, preferiría que se les exija cancelarlos antes de pagar los gastos del mes actual. Las multas y otros cobros deberían poder pagarse, pero su falta de pago no debe bloquear los pagos de gastos comunes.

4. ¿Qué necesitas para la gestión de reservas de espacios comunes?

Me parece importante que los usuarios puedan ver la disponibilidad de los espacios comunes en y hacer una reserva seleccionando fecha y horas. Además, deberían realizar un pago inicial para confirmar la reserva, y recibir notificaciones sobre el siguiente paso del proceso, como retirar llaves o firmar documentos.

5. ¿Qué tipo de notificaciones deben recibir los residentes?

Me gustaría que los residentes reciban notificaciones en varias situaciones. Por ejemplo, cuando tengan tres meses de atraso en los gastos comunes, aviso de reuniones, envío de circulares, cuando realicen cualquier tipo de pago, y cuando cometan una infracción que conlleve una multa.

6. ¿Cómo debería gestionar el estado de los gastos comunes de los domicilios?

Necesito poder listar todos los domicilios en una especie de planilla donde pueda ver el estado de los pagos. Además, me gustaría poder hacer clic en un domicilio para ver detalles y, si el propietario desea pagar por otros medios, poder cancelar la deuda directamente desde esa vista.

7. ¿Qué opinas sobre una función para gestionar los reclamos? Implicaría recibir reclamos (texto e imágenes) de residentes y decidir si proceder con la multa o no ¿cambiarías o modificarías algo?

Suena como una función bastante útil, si un reclamo amerita una multa, poder enviar automáticamente una notificación al infractor, y no solo eso, muchas veces hay “infracciones menores” como por ejemplo estacionarse en lugares indebidos, podríamos juntar unos 3 reclamos para cursar una multa (que asciende a 1 U.T.M.) para esto sería necesario categorizar y agrupar a infractores recurrentes, también sucede que las infracciones las realizan visitas y no los mismos residentes.

8. ¿Qué facilidades necesitas para gestionar los reclamos que ingresan los residentes?

Los residentes deberían poder ingresar sus reclamos de manera digital, especificando fecha y hora aproximada de los hechos, con la opción de adjuntar imágenes y videos y asegurarse que el

infractor no se entere de quien interpuso el reclamo. También me gustaría que puedan y agregar más información si es necesario. Para evitar abusos de la herramienta se debe informar que el mal uso de esta será sancionado, adicionalmente sería una buena idea dejar una ventana de 2 días antes de yo como administrador recibir la notificación porque si la persona que envió el reclamo lo hizo de manera mal intencionada o con la cabeza caliente pueda reflexionar y eliminar dicho reclamo.

Benchmarking

Vistas

Vista del módulo panel de usuarios: Aquí se encuentran los detalles de cobros y pagos relacionados con la propiedad, así como también las publicaciones creadas por la administración y las notificaciones de los gastos comunes generados.



Imagen 19. Priorización de historias de usuario.

Vista del módulo ingresos extraordinarios: En esta parte pueden ver toda la información registrada en la comunidad respecto a los ingresos extraordinarios.



Imagen 20. Ingresos extraordinarios.

Vista del módulo egresos: Este módulo tiene como finalidad mostrar todos los datos registrados en la comunidad relacionado a los egresos.

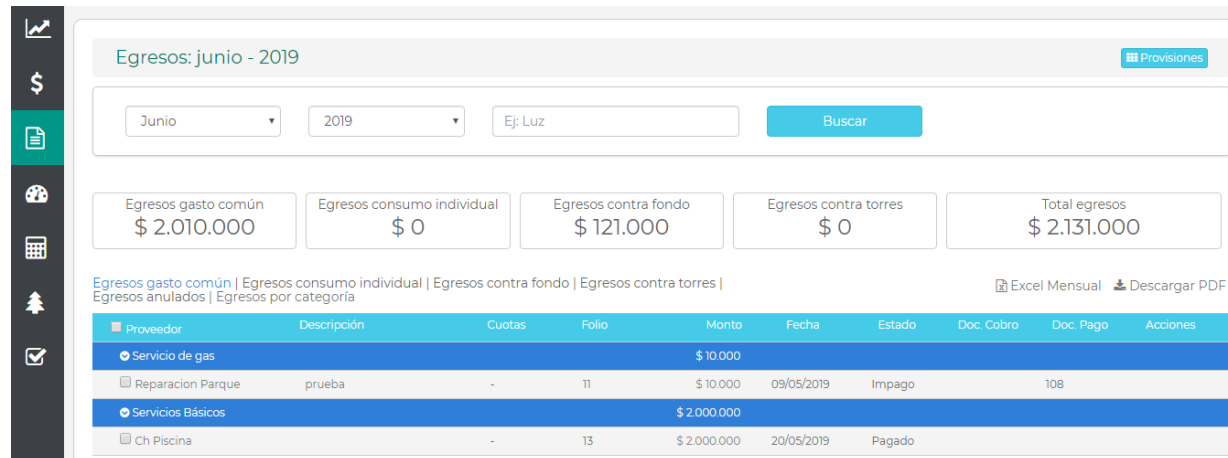


Imagen 21. Egresos.

Vista del módulo medidores: Desde esta parte solo podrán visualizar información relacionada a su unidad.

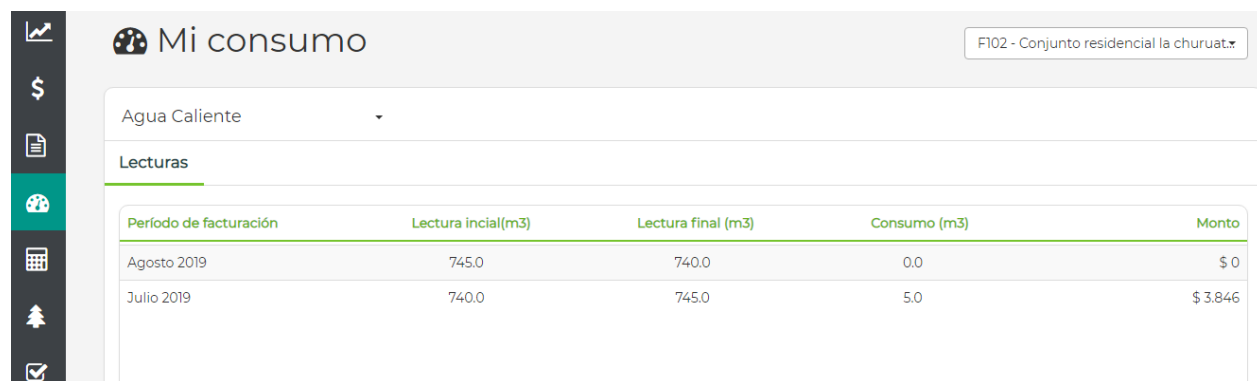


Imagen 22. Consumo de servicios básicos

Vista del módulo gastos comunes: El siguiente elemento busca detallar la relación con respecto al próximo gasto común a emitir en la comunidad.



Imagen 23. Gastos comunes.

Vista del módulo de espacios comunes: En este módulo solo encontrarán la información relacionada a su unidad, así como la posibilidad de agendar un evento si se cuenta con espacios comunes registrados en la comunidad.

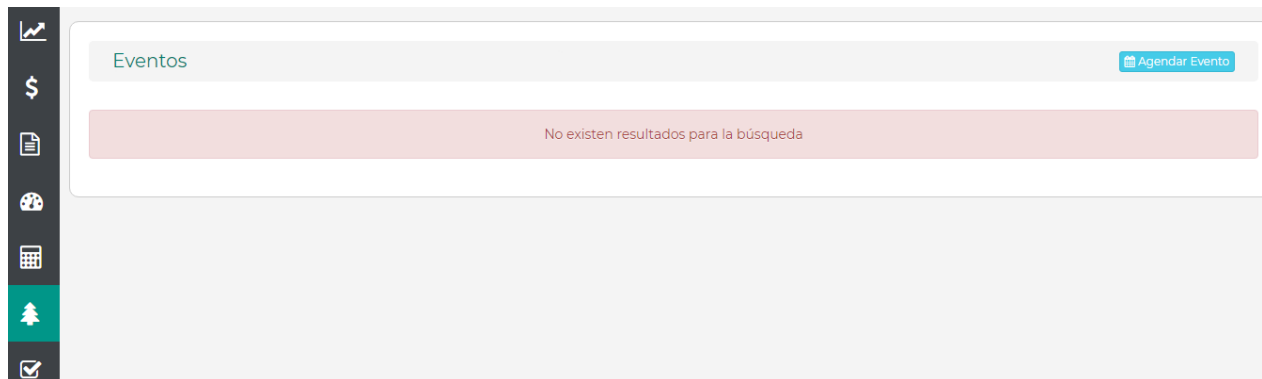


Imagen 24. Espacios comunes.

Vista del módulo de votaciones: En esta parte se verán todas las votaciones registradas, además de tener la posibilidad para dar respuesta en aquellas que estén abiertas

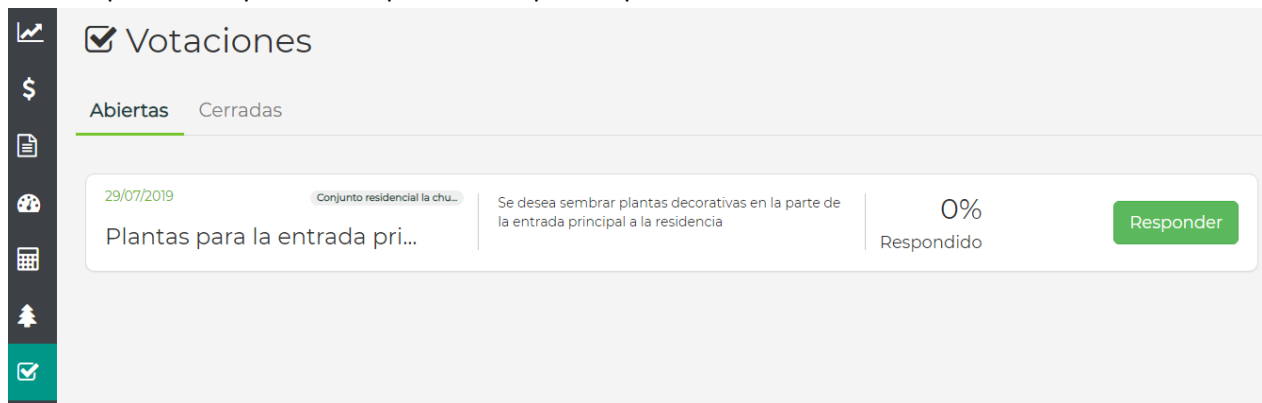


Imagen 25. Votaciones.