# Sistema de Gestión Administrativa de las Propiedades Horizontales

Moises David Chapeton Albarracin

Juan Camilo Pantoja Ojeda

Gustavo Alonzo Pedrozo Matute

Dominic Merchan Silva

Instructor

Andrés Chavez

Servicio Nacional de Aprendizaje

Centro de Diseño y Metrologia

Técnologo de Análisis y Desarrollo de Software

2024

Tabla de contenido

[Sistema de Gestión Administrativa de las Propiedades Horizontales 1](#_Toc171640343)

[Lista de Tablas 3](#_Toc171640344)

[Lista de Figuras 4](#_Toc171640345)

[Introducción 5](#_Toc171640346)

[Planteamiento del Problema 6](#_Toc171640347)

[Descripción del problema 6](#_Toc171640348)

[Objetivos del Proyecto 8](#_Toc171640349)

[Objetivo General 8](#_Toc171640350)

[Objetivos específicos 8](#_Toc171640351)

[Alcance y Delimitación 9](#_Toc171640352)

[Delimitación del proyecto 9](#_Toc171640353)

[Alcance 9](#_Toc171640354)

[Requisistos 11](#_Toc171640355)

[Requisitos Funcionales 11](#_Toc171640356)

[Requisitos No Funcionales 14](#_Toc171640357)

[Arquitectura 16](#_Toc171640358)

[Ventajas de Microservicios para este Proyecto 16](#_Toc171640359)

[Patrón Arquitectonico 17](#_Toc171640360)

[Lenguaje de Programación 19](#_Toc171640361)

[Framework 20](#_Toc171640362)

[Blazor para el desarrollo del Back- end y Fron- end 20](#_Toc171640363)

[Base de Datos 21](#_Toc171640364)

[Metodologia de Desarrollo 22](#_Toc171640365)

# Lista de Tablas

# Lista de Figuras

# Introducción

En la gestión administrativa de las propiedades horizontales, la eficiencia administrativa juega un papel crucial para asegurar un ambiente armonioso y bien gestionado. Sin embargo, los procesos manuales y desconectados pueden generar problemas como falta de transparencia en la información de los acontecimientos de las propiedades, y en la inversión de fondos con los pagos administrativos. Para abordar estas complejidades, se está desarrollando un software administrativo. Este tipo de software ofrece una solución integral para gestionar y optimizar todas las operaciones administrativas de las propiedades horizontales. Al automatizar procesos, centralizar datos y ofrecer herramientas de seguimiento y análisis, este software tiene el potencial de transformar la gestión de la propiedad horizontal, proporcionando una plataforma única y accesible que simplifica la vida tanto de los residentes como de los administradores.

# Planteamiento del Problema

## Descripción del problema

En las propiedades horizontales, se puede evidenciar un problema en la administración, específicamente en la comunicación e información. Los administradores y residentes no tienen una comunicación eficiente y directa donde se puedan expresar y hablar sobre dudas e inquietudes que puedan tener cada uno, esto incluye al reporte de errores y sugerencias que tengan los residentes sobre los establecimientos de las propiedades horizontales.

Así mismo, la información de estas propiedades es algo ineficiente, los residentes no tienen una plataforma donde puedan informarse sobre las reuniones, asambleas y eventos que tengan estas, tampoco un espacio que les notifique los acontecimientos y las emergencias que puedan presentar las propiedades.

Por otro lado, la información sobre la inversión que se le tiene a los pagos de administración que realizan los residentes es escasa, no hay información del manejo de esos pagos y en cómo se ven reflejados en el mantenimiento y condiciones de las áreas pertenecientes a las propiedades en general.

Esto conlleva a diversas consecuencias administrativas, por ejemplo, comunicación defectuosa y poco asertiva. Por otro lado, se puede generar desconfianza de los residentes por la escasa información que se tiene en cómo se están invirtiendo y manejando sus pagos administrativos. Por otro lado, están muy inconscientes de lo que pasa en las propiedades, no están lo suficientemente alertas de los acontecimientos, emergencias, reuniones y asambleas que pueden tener estas.

# Objetivos del Proyecto

## Objetivo General

Construir un sistema de información centralizado, conveniente y transparente que mejore el desempeño en los procesos administrativos que se encuentran en las propiedades horizontales.

## Objetivos específicos

Realizar el levantamiento de información para identificar las necesidades de los residentes y las fallas presentadas a nivel administrativo.

Analizar la información recolectada y determinar qué requisitos son necesarios para el óptimo desarrollo de nuestro software.

Diseñar el modelo de solución a partir de la arquitectura propuesta teniendo en cuenta cada uno de los elementos necesarios.

Desarrollar y construir el software a partir de esa arquitectura.

Implementar la solución informática requerida de acuerdo a las necesidades de los residentes.

Ejecutar pruebas para asegurar el correcto funcionamiento del software y sus correctas actualizaciones.

# Alcance y Delimitación

## Delimitación del proyecto

Integraciones complejas con sistemas externos que no estén directamente relacionados con la gestión residencial.

Funcionalidades avanzadas no mencionadas explícitamente en los requisitos.

Desarrollo de características que no sean parte integral de la gestión residencial, a menos que se acuerde lo contrario con los interesados.

## Alcance

El proyecto se llevará a cabo en las propiedades horizontales, específicamente en las actividades y procesos relacionados con la gestión administrativa de estas propiedades.

1.Portal de comunicación: Un sistema de mensajería integrado que permitirá una comunicación directa entre administradores y residentes para resolver problemas y recibir retroalimentación.

2.Gestión de áreas comunes: El sistema permitirá realizar reservas de las áreas comunes con las que cuenten las propiedades horizontales, también reportar algún mantenimiento que necesite alguno de estos espacios, y recibir el seguimiento de este mismo.

3. Sistema de reporte de errores y sugerencias: Una plataforma donde los residentes podrán reportar problemas en los establecimientos del conjunto y hacer sugerencias para mejoras.

4. Transparencia informativa: Informes periódicos que mostrarán cómo se están utilizando los pagos de administración, así como actualizaciones sobre proyectos y mejoras planificadas. Además, se informará sobre eventos, asambleas y reuniones que puedan tener las propiedades.

5. Notificación de emergencia: Notificaciones que alerten e informen sobre acontecimientos imprevistos que se presenten en las propiedades horizontales.

6. Soporte técnico: Un espacio donde los residentes y administradores puedan comunicar errores y fallas que pueda tener el Software.

# Requisistos

## Requisitos Funcionales

### RF1: Registro y autenticación de usuarios

Permitir a los residentes registrarse y crear cuentas en la plataforma.

### RF2: Gestión de perfiles

Facilitar a los usuarios actualizar y administrar su información personal, como nombre, correo electrónico, teléfono y número de apartamento.

### RF3: Definición de roles

Establecer diferentes roles de usuario con distintos niveles de acceso y permisos, por ejemplo, administrador, miembro del consejo de administración, propietario y residente.

### RF4: Tablero de anuncios

Implementar un tablero de anuncios virtual para publicar avisos y comunicados relevantes para la comunidad de propietarios.

### RF5: Mensajería interna

Habilitar un sistema de mensajería interna para facilitar la comunicación entre residentes, miembros del consejo de administración y administradores.

### RF6: Foros y chats

Crear espacios virtuales para que los residentes puedan interactuar, discutir temas de interés común y resolver dudas.

### RF7: Reserva de espacios comunes

Permitir a los residentes reservar espacios comunes, como salas de reuniones, zonas de ocio o instalaciones deportivas, a través de la plataforma.

### RF8: Calendario de eventos

Implementar un calendario de eventos para informar sobre actividades comunitarias, reuniones, talleres y eventos organizados dentro de la propiedad horizontal.

### RF9: Mantenimiento de espacios comunes

Facilitar la solicitud de reparaciones, mantenimiento preventivo y correctivo de espacios comunes, permitiendo adjuntar fotos y descripciones del problema.

### RF10: Seguimiento de incidencias

Implementar un sistema de seguimiento para gestionar las solicitudes de mantenimiento, desde la recepción hasta la resolución, informando a los residentes sobre el estado de cada solicitud.

### RF11: Convocatoria de reuniones

Facilitar la convocatoria de reuniones del consejo de administración o asambleas de propietarios, definiendo fecha, hora, lugar y agenda

### RF12: Registro de asistentes

Permitir a los residentes registrar su asistencia a las reuniones online o presenciales.

### RF13: Sistema de votación

Implementar un sistema de votación online seguro y transparente para realizar votaciones durante las asambleas de las propiedades.

### RF14: Espacio de denuncias

Implementar un espacio para que los residentes puedan registrar quejas, denuncias o incidencias relacionadas con la convivencia en la propiedad horizontal, permitiendo adjuntar fotos y descripciones.

### RF15: Seguimiento de denuncias

Establecer un proceso de seguimiento para gestionar las denuncias recibidas, desde la recepción hasta la resolución, informando a los residentes sobre el estado de cada denuncia.

### RF16: Buzón de sugerencias

Habilitar un buzón de sugerencias para que los residentes puedan aportar ideas o propuestas para mejorar la convivencia y la gestión de la propiedad horizontal.

### RF17: Notificación de emergencia

El sistema debe alertar lo que suceda en el conjunto que sea de suma emergencia, como un accidente o acontecimiento que pase en el por medio de una notificación. Y cómo reaccionar a dicha emergencia.

### RF18: Espacio de información de la inversión de fondos.

Proporciona herramientas para registrar y seguir el destino de los fondos de administración, permitiendo una transparencia en la información y asegurando que los pagos se utilicen eficientemente para el mantenimiento y mejora de las propiedades horizontales.

### RF19: Soporte técnico

El sistema tener un espacio que resuelva o comunique posibles fallos y problemas que tenga el Software al realizar una tarea administrativa o residencial.

## Requisitos No Funcionales

### RNF1: Seguridad de Datos

Garantiza la seguridad de los datos de los usuarios mediante técnicas de encriptación y medidas de seguridad robustas, protegiendo la confidencialidad e integridad de la información

### RNF2: Interfaz Intuitiva

La interfaz de usuario debe ser fácil de entender y navegar para los residentes y administradores, facilitando la interacción con el software y mejorando la experiencia del usuario.

### RNF3: Nivel de usuario

Facilidades y controles para permitir el acceso a la información al personal autorizado a través de Internet, con la intención de consultar y subir información pertinente para cada una de ellas.

### RNF4: Alta disponibilidad

El sistema debe mantener una alta disponibilidad para garantizar que los usuarios puedan acceder al software en cualquier momento, evitando interrupciones

### RNF5: Respaldo y Recuperación de Datos

Debe contar con mecanismos de respaldo y recuperación de datos para prevenir la pérdida de información en caso de fallos del sistema o errores humanos, garantizando la integridad y disponibilidad de los datos almacenados.

### RNF6: Optimización de Rendimiento

El sistema debe estar optimizado para ofrecer tiempos de carga rápidos y respuestas ágiles a las solicitudes de los usuarios, asegurando una experiencia fluida y eficiente durante la interacción con la plataforma.

### RNF7: Manejo de Múltiples Usuarios Concurrentes

Debe ser capaz de manejar múltiples usuarios concurrentes sin experimentar degradación en el rendimiento, garantizando una experiencia consistente y sin interrupciones incluso en momentos de alta demanda.

### RNF8: Escalabilidad Futura

El sistema debe ser escalable para adaptarse al crecimiento futuro de las propiedades horizontales y la cantidad de usuarios, permitiendo una expansión sin problemas y garantizando que la plataforma pueda mantener su rendimiento y funcionalidad a medida que aumenta la demanda

# Arquitectura

La arquitectura a implementar será Microservicios.

## Ventajas de Microservicios para este Proyecto

### Modularidad y Escalabilidad:

La arquitectura de microservicios permite dividir el sistema en servicios pequeños y autónomos. Esto es útil para manejar diferentes funcionalidades como registro de usuarios, gestión de espacios comunes, comunicación interna, etc., de manera independiente.

### Desarrollo y Despliegue Independiente:

Los servicios pueden ser desarrollados, probados y desplegados de manera independiente, lo que facilita la implementación de nuevas características y la corrección de errores sin afectar a todo el sistema.

### Flexibilidad Tecnológica:

Diferentes equipos pueden utilizar diferentes tecnologías y lenguajes de programación para desarrollar cada servicio, lo que puede aumentar la eficiencia y la utilización de tecnologías adecuadas para cada funcionalidad específica.

### Tolerancia a Fallos:

Si un microservicio falla, el resto del sistema puede seguir funcionando, mejorando así la resiliencia del sistema.

# Patrón Arquitectonico

El patrón que se va a utilizar es el MVC (Modelo de Vista Controlador).

El patrón arquitectónico Modelo-Vista-Controlador (MVC) se basa en el principio de separación de responsabilidades, lo cual resulta fundamental para el desarrollo de aplicaciones robustas y mantenibles

El patrón MVC divide la aplicación en tres componentes clave: el Modelo, la Vista y el Controlador. Cada uno de estos componentes tiene responsabilidades específicas y no se entremezclan entre sí. Esto facilita la comprensión del código, ya que cada parte se enfoca en una tarea concreta.

En el caso de tu software de gestión para una comunidad de propietarios, el patrón MVC podría organizarse de la siguiente manera:

- Modelo (Model): Aquí se encontraría la lógica de negocio y la interacción con la base de datos. Sería donde se manejarían las reglas de negocio relacionadas con la gestión de perfiles, roles, eventos, mantenimiento, votaciones, entre otros.

- Vista (View): En esta capa se ubicarían las interfaces de usuario, como las pantallas de registro, perfiles, tablero de anuncios, calendario de eventos, espacios comunes, entre otros. Cada vista estaría encargada de mostrar la información al usuario y de capturar las interacciones del mismo.

- Controlador (Controller): Los controladores actuarían como intermediarios entre el modelo y la vista. Se encargarían de procesar las solicitudes del usuario desde la vista, interactuar con el modelo para obtener o modificar datos y luego actualizar la vista con los resultados correspondientes.

Este patrón arquitectónico MVC te permitiría separar claramente las responsabilidades dentro del sistema, facilitando la mantenibilidad, escalabilidad y reutilización del código. Además, al tener una estructura bien definida, se facilita el trabajo en equipo y la colaboración entre desarrolladores.

# Lenguaje de Programación

El lenguaje con el que se planea codificar el proyecto es C#.

C# es un lenguaje de programación robusto, orientado a objetos y altamente tipado que permite desarrollar aplicaciones escalables y de alto rendimiento. Su sintaxis clara y estructurada facilita la escritura de código limpio y fácil de mantener.

C# se integra fácilmente con otras tecnologías de Microsoft, como el entorno de desarrollo Visual Studio y el framework .NET, lo que facilita la creación de aplicaciones completas y bien integradas en el ecosistema Microsoft.

# Framework

El Framewor que se va a utilizar para el proyecto es .NET.

El framework .NET fue creado por Microsoft y está estrechamente integrado con el lenguaje C#. De hecho, C# es el lenguaje principal utilizado en el desarrollo con .NET, lo que significa que están diseñados para trabajar de forma conjunta de manera eficiente.

El framework .NET proporciona una amplia gama de funcionalidades y bibliotecas que facilitan el desarrollo de aplicaciones robustas y escalables. Desde la manipulación de datos hasta la creación de interfaces de usuario, .NET ofrece herramientas poderosas para cubrir todas las necesidades del proyecto.

## Blazor para el desarrollo del Back- end y Fron- end

Blazor es un framework que permite desarrollar aplicaciones web interactivas utilizando C#, tanto en el lado del cliente como en el servidor. Esto significa que puedes crear tanto el frontend como el backend de tu aplicación utilizando un único lenguaje, lo cual simplifica el desarrollo y la mantenibilidad del código.

Blazor utiliza WebAssembly para ejecutar código C# en el navegador, lo que permite una mayor velocidad de ejecución y una experiencia de usuario más fluida sin sacrificar la productividad del desarrollador al escribir en un lenguaje conocido como C#.

# Base de Datos

La base de datos que se va a utilizar es SQL Server.

Utilizando el lenguaje C# en conjunto con el framework .NET, se puede establecer conexiones directas a una base de datos SQL Server desde tus aplicaciones. Esto permite realizar consultas, inserciones, actualizaciones y eliminaciones de datos de manera eficiente.

Tanto Visual Studio como SQL Server Management Studio (SSMS) son herramientas desarrolladas por Microsoft que te permiten trabajar de manera integrada en entornos de desarrollo y administración de bases de datos SQL Server, lo que facilita el trabajo colaborativo y la gestión eficiente del proyecto.

SQL Server se complementa perfectamente con el ecosistema de desarrollo de Microsoft, incluyendo el framework .NET, el lenguaje C#, lo que brinda una experiencia unificada y coherente al trabajar en el proyecto.

# Metodologia de Desarrollo

La metodología de desarrollo que se va a implementar e SCRUM.

Scrum es la metodología más adecuada para este proyecto debido a las siguientes razones:

Estructura y Claridad: Dada la complejidad y el amplio alcance del proyecto, una estructura clara ayudará a mantener el enfoque y la dirección del desarrollo.

Entrega Incremental: La naturaleza iterativa e incremental de Scrum permitirá la implementación gradual de las múltiples funcionalidades, permitiendo a los usuarios finales (residentes y administradores) proporcionar feedback constante y ajustar los requisitos según sea necesario.

Transparencia y Mejora Continua: Las reuniones diarias y las retrospectivas fomentan la comunicación y la mejora continua, lo cual es crucial para asegurar que el software cumpla con las expectativas y necesidades cambiantes de los usuarios.

Gestión de Riesgos: La revisión regular de los sprints permite la identificación temprana de problemas y la adaptación del plan de trabajo, minimizando riesgos a lo largo del desarrollo.