

DISEÑO DE UN PROTOTIPO DE APLICACIÓN WEB PARA LA GESTIÓN DE CORRESPONDENCIA EN CONJUNTOS RESIDENCIALES

Presentado por:
Sergio Andres Nova Aguilar

Tutor:
Beitmantt Geovanni Cárdenas Quintero

Universidad Distrital Francisco José de Caldas.
Especialización en ingeniería de software.
Facultad de ingeniería.

ÍNDICE GENERAL

INTRODUCCIÓN 1
PARTE I CONTEXTUALIZACIÓN 3
1. CAPÍTULO I. DESCRIPCIÓN DE LA INVESTIGACIÓN 4
1.1. PLANTEAMIENTO DEL PROBLEMA 4
1.2. FORMULACIÓN DEL PROBLEMA 4
1.3. SISTEMATIZACIÓN DEL PROBLEMA 4
1.4. OBJETIVOS DE LA INVESTIGACIÓN 5
1.4.1. OBJETIVO GENERAL 5
1.4.2. OBJETIVOS ESPECÍFICOS 5
1.5. JUSTIFICACIÓN DE LA INVESTIGACIÓN 6
1.6. HIPÓTESIS DE TRABAJO 7
1.7. MARCO REFERENCIAL 7
1.7.1. MARCO TEÓRICO 7
1.7.1.1. PROPIEDAD HORIZONTAL EN COLOMBIA 7
1.7.1.2. INNOVACIÓN TECNOLOGICA 8
1.7.1.3. AUTOMATIZACIÓN DE PROCESOS 10
1.7.2. MARCO ESPACIAL 11
1.7.2.1. CONJUNTO RESIDENCIAL PARQUE PRIMAVERAL 11
1.7.3. MARCO LEGAL 12
1.7.3.1. LEY 675 DEL 2001 12
1.7.3.2. LEY 1582 DE 2012 12
1.8. MARCO METODOLOGICO 13
1.8.1. SCRUM 13
1.8.1.1. EQUIPO SCRUM 13
1.8.1.2. DESARROLLADORES 13
1.8.1.3. PRODUCT OWNER 13
1.8.1.4. SCRUM MASTER 14
1.8.1.5. ARTEFACTOS DE SCRUM 14
1.8.1.6. PRODUCT BACKLOG 14
1.8.1.7. SPRINT BACKLOG 14
1.8.1.8. INCREMENT 15
1.8.2. DJANGO 15
1.8.3. API REST 16
1.8.4. ORM 16
1.8.5. MySQL 17
1.8.6. TWILIO 19
1.8.7. VUE.JS 19
1.9. ALCANCES, LIMITACIONES Y RESULTADOS ESPERADOS 20
1.9.1. ALCANCES 20
1.9.2. LIMITACIONES 20
1.9.3. RESULTADOS ESPERADOS 20
PARTE II DESARROLLO 21
2. CAPÍTULO II INVESTIGACIÓN 22

2.1. RECOLECCIÓN DE INFORMACIÓN 22iii
2.1.1. ENTREVISTA 22
2.1.2. ENCUESTAS 23
2.2. ORDENAMIENTO DE LA INFORMACIÓN 24
2.3. ANÁLISIS DE LA INFORMACIÓN 24
2.4. MODULOS A IMPLEMENTAR 25
2.4.1. MODULO RESIDENTES 25
2.4.2. MODULO CORRESPONDENCIA 26
3. CAPÍTULO III ARQUITECTURA 27
3.1. DIAGRAMA DE CASOS DE USO 28
3.2. DIAGRAMA DE CLASES 29
3.3. DIAGRAMAS DE SECUENCIA 30
3.4. DIAGRAMAS DE DESPLIEGUE 33
4. CAPÍTULO IV METODOLOGIA 34
5. CAPÍTULO V MODELO DE DATOS 48
5.1. RELACIÓN ENTRE ENTIDADES 48
5.1.1. TABLA RESIDENTES 48
5.1.2. TABLA CORRESPONDENCIA 49
6. CAPÍTULO VI PROTOTIPO GESTIÓN DE CORRESPONDENCIA 50
6.1. HERRAMIENTAS 50
6.1.1. VISUAL STUDIO CODE 50
6.1.2. POSTMAN 51
6.1.3. BOOTSTRAP 52
6.2. MODULOS 53
6.2.1. MODULO REGISTRO RESIDENTES 53
6.2.2. MODULO CONSULTA RESIDENTES 54
6.2.3. MODULO RECEPCIÓN CORRESPONDENCIA 55
6.2.4. MODULO ENTREGA CORRESPONDENCIA 57
PARTE III CIERRE 60
7. CAPÍTULO VII RESULTADOS Y DISCUSIÓN 61
8. CAPÍTULO VIII TRABAJO FUTURO 62
9. CAPÍTULO IX CONCLUSIONES 63
ANEXOS 64
A ANEXO I: ENCUESTAS 65
B ANEXO II: TABULACIÓN DE LA INFORMACIÓN 67
10. REFERENCIAS 70

ÍNDICE DE FIGURAS

1.1 DIAGRAMA FUNCIONAMIENTO DE ORM. 177

3.1 DIAGRAMA CASOS DE USO. 29
3.2 DIAGRAMA DE CLASES. 30
3.3 DIAGRAMA DE SECUENCIA AUTENTICACIÓN DEL SISTEMA. 31
3.4 DIAGRAMA DE SECUENCIA CREACIÓN DE RESIDENTES. 32
3.5 DIAGRAMA DE SECUENCIA RECEPCIÓN DE CORRESPONDENCIA..... 33
3.6 DIAGRAMA DE SECUENCIA ENTREGA DE CORRESPONDENCIA. 33
3.7 DIAGRAMA DE DESPLIEGUE DEL SISTEMA. 34

5.1 DIAGRAMA ENTIDAD/RELACIÓN MODELO DE DATOS. 50

6.1 INTERFAZ DE USUARIO VS CODE. 52
6.2 NTERFAZ DE USUARIO POSTMAN. 523
6.3 INTERFAZ DE USUARIO MÓDULO CREACIÓN DE RESIDENTES. 54
6.4 INTERFAZ DE USUARIO MÓDULO CONSULTA DE RESIDENTES. 55
6.5 INTERFAZ DE USUARIO MÓDULO RECEPCIÓN DE CORRESPONDENCIA. 56
6.6 NOTIFICACIÓN ENVIADA AL RESIDENTE VÍA WHATSAPP. 57
6.7 INTERFAZ DE USUARIO MÓDULO ENTREGA DE CORRESPONDENCIA. ... 58
6.8 INTERFAZ DE USUARIO MÓDULO ENTREGA DE CORRESPONDENCIA. ... 59
6.9 NOTIFICACIÓN DE ENTREGA DE CORRESPONDENCIA ENVIADA AL RESIDENTE VÍA WHATSAPP. 60

ÍNDICE DE TABLAS

4.1 PLANEACIÓN DE LAS HISTORIAS DE USUARIO POR SPRINT 36
4.2 HISTORIA PG-1 36
4.3 HISTORIA PG-2 38
4.4 HISTORIA PG-3 39
4.5 HISTORIA PG-4 40
4.6 HISTORIA PG-5 41
4.7 HISTORIA PG-6 42
4.8 HISTORIA PG-7 43
4.9 HISTORIA PG-8 44
4.10 HISTORIA PG-9 45
4.11 HISTORIA PG-10 46
4.12 HISTORIA PG-11 47
4.13 HISTORIA PG-12 48

INTRODUCCIÓN

Los conjuntos residenciales son conocidos por brindar de forma adicional a la vivienda, una serie de beneficios que surgen de la vida en comunidad, con el objetivo de maximizar la calidad de vida en la mayor medida posible con el presupuesto brindado por todos los miembros de la propiedad, por esto, la administración como ente central, debe aprovechar las herramientas tecnológicas disponibles para agilizar y facilitar los procesos internos ya que a menudo existen procesos manuales, cuyos errores crean tensión entre los diferentes actores de la comunidad.

A medida que la tecnología y los sistemas evolucionan, el proceso de gestión de correspondencia se puede automatizar mejorando significativamente la eficiencia y experiencia de los residentes, el personal de seguridad y la administración del conjunto. Las ventajas de automatizar el sistema de gestión de correspondencia en los conjuntos residenciales son:

- Notificar ágilmente al residente sobre su correspondencia.
- Evitar el diligenciamiento de bitácoras físicas.
- Centralizar la información de los residentes.

PARTE I

CONTEXTUALIZACIÓN

1. CAPÍTULO I. DESCRIPCIÓN DE LA INVESTIGACIÓN

1.1. PLANTEAMIENTO DEL PROBLEMA

La mayoría de los conjuntos residenciales no cuentan con sistemas de gestión de correspondencia automatizados, por lo que los residentes deben acercarse a la recepción del conjunto, validar si tienen correspondencia, recibirla y diligenciar una bitácora física de recepción de correspondencia custodiada por el personal de seguridad.

1.2. FORMULACIÓN DEL PROBLEMA

¿Puede la automatización del sistema de gestión de correspondencia resolver los problemas comunes que surgen de la ejecución de procesos manuales, minimizando errores, reprocesos, notificando ágilmente, centralizando la información del residente y digitalizando la recepción y entrega de correspondencia?

1.3. SISTEMATIZACIÓN DEL PROBLEMA

- ¿Cuáles son los beneficios de la automatización del sistema de gestión de correspondencia para los residentes, personal de seguridad y la administración del conjunto residencial?
- ¿Qué métodos de notificación son los más usados por los residentes?

1.4. OBJETIVOS DE LA INVESTIGACIÓN

1.4.1. OBJETIVO GENERAL

Diseñar un prototipo de aplicación web que permita automatizar la gestión de correspondencia en conjuntos residenciales.

1.4.2. OBJETIVOS ESPECÍFICOS

- Realizar un análisis del proceso actual y de la experiencia de los residentes, el personal de seguridad y la administración del conjunto residencial referente a la gestión de la correspondencia.
- Diseñar una base de datos para almacenar la información del residente y de la correspondencia.
- Diseñar la arquitectura de la aplicación web del sistema de gestión de correspondencia.
- Implementar los componentes del sistema de gestión de correspondencia dentro de un ambiente local.

1.5. JUSTIFICACIÓN DE LA INVESTIGACIÓN

Este proyecto busca optimizar y centralizar los procesos relacionados con la recepción, entrega y seguimiento de la correspondencia en conjuntos residenciales. Este enfoque garantiza un manejo eficiente, seguro y transparente, ofreciendo beneficios tanto para los residentes como para el personal administrativo.

El prototipo se basa en un sistema modular que cubre todas las funciones de un sistema de correspondencia, como la gestión de residentes y la gestión de correspondencia, resuelve errores en la entrega y recepción, evita procesos manuales e ineficientes y elimina brechas de seguridad en la información.

En general, el sistema representa una solución robusta, moderna y adaptada a las necesidades de la comunidad. Gracias al proceso de automatización la eficiencia y la calidad de los servicios mejoran significativamente y brindan a los actores una nueva y mejorada experiencia en la gestión de la correspondencia.

1.6. HIPÓTESIS DE TRABAJO

La automatización del sistema de gestión de correspondencia dentro de los conjuntos residenciales generará beneficios a los residentes, personal de seguridad y a la administración del conjunto notificando ágilmente, optimizando tiempos y centralizando la información del residente.

1.7. MARCO REFERENCIAL

1.7.1. MARCO TEÓRICO

1.7.1.1. PROPIEDAD HORIZONTAL EN COLOMBIA

La Propiedad Horizontal se entiende como un derecho sobre un inmueble que se encuentra en el horizonte o en una edificación. también puede referirse al sistema o entidad legal para administrar un conjunto de construcciones privadas. Una copropiedad pertenece a dos o más personas denominadas “copropietarios”. Ellos tienen propiedad compartida sobre las zonas comunes y se distribuye según la proporción acordada. Es decir, nadie tiene control exclusivo sobre el sitio; adicional comparten derechos y obligaciones. Así, se busca la función social, la seguridad y la convivencia pacífica de todos los que viven allí. (Moret, 2019).

Además, varios organismos se encargan de su gestión que son:

Asamblea de copropietarios: Está formada por todos los propietarios y es la máxima entidad de los bienes de la propiedad.

Consejo de administración: Conformada por un grupo especial de propietarios.

Administrador del conjunto: Persona que ejerce el derecho de representación legal de la propiedad horizontal.

En Colombia el fenómeno de construcción de viviendas ha aumentado los últimos años. Se estima que el 60% de la población del país vive en propiedades horizontales. El Censo Inmobiliario realizado en 2018 en Bogotá mostró que el 70% de los predios de la capital son propiedad horizontal, esto explica el crecimiento que ha tenido en los años recientes este concepto de convivencia y derecho patrimonial entre los habitantes (Portafolio, 2022).

1.7.1.2. INNOVACIÓN TECNOLOGICA

La innovación es el uso del conocimiento para crear productos, servicios o procesos nuevos o mejorar de otro modo los que ya existen, y en beneficio del bienestar humano. Por tanto, la innovación tecnológica es el procedimiento con el que las empresas crean

nuevos productos o servicios; o mejoran los que ya existen, utilizando los beneficios de la tecnología (García, 2023).

La innovación tecnológica incluye muchas actividades y por tanto se puede segmentar en:

Innovación tecnológica incremental: Se genera a partir de un proyecto existente realizando mejoras muy significativas con valor agregado. El cambio suele lograrse a través de este tipo de innovación porque se centra en mejorar la eficiencia, la productividad o la diferenciación competitiva de los servicios o productos existentes (García, 2023).

Innovación tecnológica disruptiva: Altera profundamente las industrias y mercados existentes y crean nuevas industrias y modelos de negocio. Esta innovación cambia fundamentalmente la forma en que se componen los mercados.

Innovación tecnológica radical: Implica la creación de nuevas tecnologías que revolucionan el mundo. A menudo crea nuevas industrias o estructuras de mercado o realiza cambios significativos en las estructuras existentes.

1.7.1.3.AUTOMATIZACIÓN DE PROCESOS

Muchas organizaciones todavía dependen de procesos manuales engorrosos que no sólo consumen tiempo y recursos, sino que también aumentan el riesgo de errores. En un mercado competitivo donde la eficiencia es primordial, la solución para completar estas tareas rutinarias y repetitivas de manera rápida y precisa es automatizándolas, utilizando herramientas que puedan administrar datos de manera centralizada. Esto permite a los empleados centrarse en proyectos más estratégicos y motivadores, en lugar de perder tiempo de trabajo en tareas monótonas. (Ayerdi, 2024)

Por tanto, la automatización de procesos implica el uso de tecnología para automatizar tareas rutinarias. Esto reduce los plazos y los errores y reduce la carga de trabajo del equipo, lo que les permite centrarse en otros objetivos más estratégicos.

Los beneficios que trae consigo la automatización de procesos son:

Mayor productividad: Se eliminan tareas manuales que consumen mucho tiempo para centrarse en actividades estratégicas de mayor valor.

Costos: Es el beneficio de una mayor productividad y una reducción de los recursos humanos necesarios para tareas repetitivas. Esto no significa necesariamente menos empleados, sino que pueden dedicar su tiempo a trabajar en tareas más estratégicas.

Reducción de errores: Es una herramienta clave para mejorar la calidad del trabajo reduciendo los errores.

1.7.2. MARCO ESPACIAL

1.7.2.1. CONJUNTO RESIDENCIAL PARQUE PRIMAVERAL

El conjunto residencial Parque Primaveral está ubicado en el barrio Bosa Islandia de Bogotá, su dirección es: Calle 71a Sur N° 82A-10. Su construcción consiste en 9 torres, apartamentos con áreas desde 43m2, parqueaderos y más de 10 zonas sociales tales como terrazas transitables, piscina, zonas deportivas y una sala de espera ubicada en la recepción.

En seguridad, el conjunto cuenta con una cerca eléctrica, un sistema de videovigilancia con 180 cámaras y un cuarto de monitoreo. Dispone de 2 puntos de acceso peatonal, uno ubicado en la recepción y otro en la entrada del parqueadero, con sistemas de acceso peatonal y vehicular por medio de tarjetas de proximidad.

Actualmente, se encuentra en curso la construcción de la Avenida Ciudad de Cali, Alimentadora de la Línea Metro, por lo cual hay cierres viales, vías en obras y levantamiento de estaciones de Transmilenio, esto ocasiona que acceso al conjunto se vea limitado y se deban tomar vías alternas.

1.7.3. MARCO LEGAL

El creciente auge de la construcción de conjuntos residenciales como estrategia de aprovechamiento del suelo en Colombia impulsó la necesidad de regular la convivencia de los ciudadanos en una forma especial de propiedad conocido como el Régimen de Propiedad Horizontal. Así nació la ley 675 de 2001. (Acevedo, & Piedrahita, 2016).

1.7.3.1. LEY 675 DEL 2001

El conjunto residencial se rige por la Ley 675 del 2001 que permite un tipo especial de dominio, conocido como propiedad horizontal, donde se dan derechos de propiedad exclusiva sobre los bienes privados y derechos de copropiedad sobre el terreno y los bienes comunes, con el objetivo de garantizar la seguridad y la convivencia pacífica en los inmuebles y la función social de la propiedad (Congreso de la República, 2001).

1.7.3.2. LEY 1582 DE 2012

En cumplimiento a la ley estatutaria 1581 de 2012, que dicta disposiciones para el tratamiento y protección de datos personales, el conjunto adopta la política para cumplir la regulación vigente y definir el marco para la atención de consultas y reclamos sobre el tratamiento de datos personales que recoja y maneje según la Ley. (Congreso de la República, 2012).

1.8.MARCO METODOLOGICO

1.8.1. SCRUM

SCRUM es un marco de gestión de proyectos ágil que ayuda a los equipos a estructurar y gestionar su trabajo a través de un conjunto de valores, principios y prácticas (Schwaber, Sutherland, 2020).

1.8.1.1. EQUIPO SCRUM

Idealmente, cada equipo scrum cuenta con: un Scrum Master, un Product Owner y un grupo de desarrolladores.

1.8.1.2. DESARROLLADORES

Son las personas del equipo que realzan el trabajo práctico para completar tareas en un sprint.

1.8.1.3. PRODUCT OWNER

Es responsable de que el equipo Scrum esté alineado con los objetivos generales del producto y comprende las necesidades y las expectativas de los interesados.

1.8.1.4. SCRUM MASTER

Es un guía que se asegura de que el equipo comprenda el marco scrum, sus principios, valores y prácticas.

1.8.1.5. ARTEFACTOS DE SCRUM

Definen el trabajo que se debe realizar y siempre agregan valor durante un sprint. Pueden verse como fragmentos de información vital para el equipo. Proporcionan estructura al proceso de Scrum porque actúan como pautas para el plan de desarrollo del producto.

1.8.1.6. PRODUCT BACKLOG

Es un listado de todo lo que se debe lograr en un proyecto, desglosado en elementos individuales. Aquí es donde el Product Owner prioriza los requisitos básicos de cada característica necesaria para el producto final.

1.8.1.7. SPRINT BACKLOG

Es la parte del Product Backlog en la que trabaja el equipo durante el sprint. Se puede ver como una lista de tareas pendientes del sprint. Se divide en tareas que el equipo debe ejecutar. Cada elemento del Sprint Backlog debe desarrollarse, probarse y documentarse.

1.8.1.8. INCREMENT

Son todos los elementos del Product Backlog que se han completado durante un sprint. Cada sprint genera incrementos de producto que pueden entregarse, por lo que el incremento del producto debe ajustarse a la definición de "terminado" del equipo y ser aceptable para el Product Owner.

1.8.2. DJANGO

Django es un framework web de back-end basado en Python, lo que significa que la mayor parte del código se escribe en Python. Pero Django viene con mucho código repetitivo y, por lo tanto, se vuelve bastante rápido en el desarrollo. (Mozilla, 2024) Tiene las siguientes características:

Fácil de usar: Django es un marco fácil de aprender y usar.

Documentación: Django cuenta con una excelente documentación, que es completa y detallada, lo que hace que sea aún más fácil aprender y usar.

Escalabilidad: Está diseñado para manejar aplicaciones web de alta demanda, Lo convierte en una opción ideal para proyectos que necesitan ser escalables.

Seguridad: En Django se encuentran una amplia gama de funciones de seguridad integradas, como autenticación de usuarios y protección contra ataques comunes como la inyección SQL.

1.8.3. API REST

Una API (Interfaz de programación de aplicaciones) es un conjunto de solicitudes que permiten la comunicación de datos entre aplicaciones. Utiliza peticiones HTTP para encargarse de las operaciones básicas necesarias para la manipulación de datos.

REST (Transferencia de estado representacional), es un conjunto de restricciones que garantizan que las solicitudes HTTP sigan las pautas definidas en la arquitectura.

API REST se refiere al uso de API para acceder a aplicaciones backend de modo que esta comunicación se realice de acuerdo con los estándares marcados por el estilo arquitectónico REST.

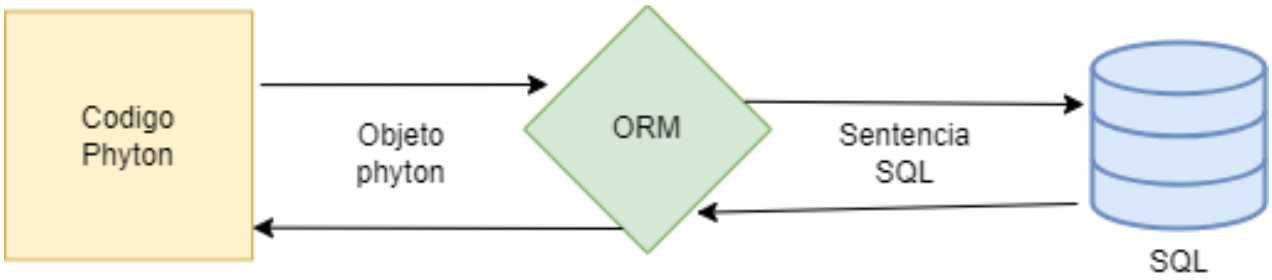
1.8.4. ORM

ORM (Object-Relational Mapping) es una de las características más importantes proporcionadas por Django. Un ORM se utiliza para interactuar con bases de datos relacionales mediante conjuntos de consultas.

Un conjunto de consultas representa una colección de objetos en una base de datos. Los conjuntos de consultas permiten leer, filtrar y ordenar datos de una base de datos. Se puede considerar ORM como un traductor que convierte el código Python en instrucciones SQL como se observa en la *Figura 1.1*.

FIGURA 1.1

Figura 1.1 Diagrama funcionamiento de ORM.



Fuente: Elaboración propia

1.8.5. MySQL

MySQL es un sistema de gestión de bases de datos relacionales de código abierto. Aunque MySQL saltó a la fama hace casi tres décadas, no muestra signos de decaimiento y se ubica como la segunda base de datos más popular en general, superada solo por Oracle Database, según DB-Engines. MySQL es lo suficientemente versátil como para respaldar una amplia variedad de aplicaciones, desde pequeños proyectos personales hasta sistemas empresariales críticos, y cuenta con el respaldo de una gran y entusiasta comunidad de código abierto (Erickson, 2024). Otros factores también ayudan a mantener la popularidad duradera de MySQL son:

Código abierto con un fuerte apoyo de la comunidad: Durante las casi tres décadas de MySQL como el RDBMS de código abierto líder, ha crecido una comunidad global vibrante a su alrededor. Eso es importante porque la comunidad proporciona una gran cantidad de experiencia y recursos, como tutoriales, consejos en foros y más. Al probar el software en múltiples escenarios de casos de uso, la comunidad también ha ayudado a descubrir y corregir errores, lo que hace que MySQL sea altamente confiable.

Alto rendimiento y confiabilidad: MySQL se siente cómodo en muchos entornos diferentes, incluidos proyectos de desarrolladores individuales y aplicaciones de misión crítica que exigen una estabilidad inquebrantable. El RDBMS de código abierto puede manejar grandes volúmenes de datos y conexiones simultáneas y proporcionar operaciones ininterrumpidas en circunstancias exigentes. Esto se debe en parte a los robustos mecanismos de replicación y conmutación por error de MySQL, que ayudan a minimizar el riesgo de pérdida de datos.

Facilidad de uso y compatibilidad: MySQL suele ser elogiado por ser fácil de usar y por ofrecer una amplia compatibilidad con plataformas tecnológicas y lenguajes de programación, incluidos Java, Python, PHP y JavaScript. MySQL también admite la replicación de una versión a la siguiente, por lo que una aplicación que ejecute MySQL 5.7 puede replicarse fácilmente a MySQL 8.0. Además, MySQL ofrece flexibilidad para desarrollar aplicaciones de base de datos sin esquemas SQL y NoSQL tradicionales.

Rentabilidad y escalabilidad: Debido a que MySQL es de código abierto, está disponible para su uso gratuito sin costo, más allá del hardware local en el que se ejecuta y la capacitación sobre cómo usarlo. Para esto último, una comunidad global de usuarios de MySQL brinda acceso rentable a recursos de aprendizaje y experiencia en resolución de problemas.

1.8.6. TWILIO

Es una plataforma para crear aplicaciones de voz, video y mensajería. Consiste en un conjunto de API que permiten a los desarrolladores realizar y recibir llamadas de voz y vídeo, enviar y recibir SMS, WhatsApp y correo electrónico, y realizar otras funciones de comunicación. Se basa casi en su totalidad en un modelo de precios de pago por uso, y la API de Twilio, facilita a los desarrolladores la integración de Twilio en sus aplicaciones (Twilio, s.f.).

1.8.7. VUE.JS

Es un framework de código abierto de JavaScript que los desarrolladores utilizan para crear interfaces de usuario web interactivas. Fue diseñado por Evan You teniendo en cuenta la simplicidad y el rendimiento. Vue.js opera en la capa de visualización de una aplicación, lo que significa que se ocupa principalmente de lo que el usuario ve y con lo que interactúa en una página web o en la pantalla de una aplicación (Vue.js s.f.).

1.9. ALCANCES, LIMITACIONES Y RESULTADOS ESPERADOS

1.9.1. ALCANCES

El alcance del proyecto es diseñar un prototipo de aplicación web para automatizar el sistema de gestión de correspondencia en conjuntos residenciales, analizando el sistema actual, la experiencia de los actores del proceso, diseñando la arquitectura de la aplicación web y la base de datos en la cual se almacenará la información de los residentes.

1.9.2. LIMITACIONES

- El proyecto será para el conjunto residencial parque primaveral ubicado en Bogotá.
- Se debe tener coherencia de los datos de propietarios y arrendatarios para un buen funcionamiento del sistema.
- El prototipo se trabajará en un ambiente local.
- Los datos de los residentes deben estar migrados a la base de datos.

1.9.3. RESULTADOS ESPERADOS

Se espera tener un prototipo del sistema de gestión de correspondencia automatizado, optimo, que permitirá notificar a los residentes al recibir correspondencia con una base de datos centralizada la cual almacenará la información del residente, contando con la participación de los actores del proceso los cuales proporcionaran cometarios y mejora continua.

PARTE II DESARROLLO

2. CAPÍTULO II INVESTIGACIÓN

2.1. RECOLECCIÓN DE INFORMACIÓN

Al iniciar el proyecto se entendió cómo funcionaba el proceso de gestión de correspondencia por medio de observación directa y experiencia propia, pero, para poder automatizar el proceso fue necesario realizar una investigación para captar las conductas y preferencias de los actores del proceso, lo que permitió crear un prototipo fácil de usar. En este contexto se plantearon 2 metodologías de recolección de información: encuestas y entrevistas.

Los actores del proceso son los residentes, el administrador del conjunto y el personal de seguridad por lo que se decidió entrevistar al administrador del conjunto debido a que es una sola persona y luego encuestar a los residentes y personal de seguridad con preguntas referentes a la información recopilada en la entrevista con el administrador y lo que se pretende entregar.

2.1.1. ENTREVISTA

La entrevista con el administrador se segmento en los siguientes temas:

- El estado actual del sistema de gestión de correspondencia.
- Problemas comunes.

- El impacto de tener un sistema de gestión de correspondencia automatizado.

De los cuales se concluyó que:

- La correspondencia de los residentes puede durar un tiempo prolongado en la recepción del conjunto ya que en algunos casos los residentes no se acercan a recogerla o porque son apartamentos sin habitar, por medio de las notificaciones se puede agilizar la entrega.
- Se han presentado problemas con las correspondencias debido a desconocimiento del guarda o del residente, un guarda de seguridad puede entregar la correspondencia a un residente de un apartamento que no le corresponde o un residente puede reclamar la correspondencia de un apartamento que no es el suyo.
- De forma mensual, la empresa de seguridad recoge las bitácoras de correspondencia, sin embargo, no se conoce que hace la empresa de seguridad con estas bitácoras.
- No se tiene centralizada la información de los residentes en una base de datos.

2.1.2. ENCUESTAS

Con base en la entrevista realizada al administrador del conjunto se crearon las encuestas orientadas a conocer:

De los residentes:

- Frecuencia en la que recibe correspondencia.
- Su medio de notificación de preferencia.
- Su experiencia con el sistema de gestión de correspondencia actual.

Del personal de seguridad:

- Su conocimiento y dominio en tecnologías de información.
- Su experiencia con el sistema de gestión de correspondencia actual.

2.2. ORDENAMIENTO DE LA INFORMACIÓN

La información obtenida de la entrevista con el administrador ayuda a orientar las encuestas que son la fuente real de información para los módulos a desarrollar en el proyecto, los resultados de las encuestas se pueden observar en los Anexo I y Anexo II.

2.3.ANÁLISIS DE LA INFORMACIÓN

La información recopilada da luz verde al desarrollo del proyecto, ya que soluciona un problema común para la administración del conjunto residencial que es la gestión de la correspondencia, bajo el análisis de los resultados se generan las siguientes conclusiones:

- El 70% de los residentes son propietarios.
- El 90% de los residentes no es notificado cuando tiene correspondencia en la recepción del conjunto.
- Al 100% de los residentes le gustaría ser notificado cuando recibe correspondencia.
- Al 60% de los residentes encuestados le gustaría ser notificado vía WhatsApp.
- El 80% de los residentes encuestados no ha presentado problemas, novedades o inconvenientes con su correspondencia.
- El personal de seguridad cuenta con conocimientos en TIC, sus conocimientos abarcan: Acceder a internet, navegar en internet manejar ofimática utilizar un computador.
- El 100% de los guardas indica que ha presentado problemas, inconvenientes o novedades con la correspondencia de los residentes.

2.4. MODULOS A IMPLEMENTAR

Se plantean los siguientes módulos para el desarrollo del proyecto que son módulos funcionales del sistema de gestión de correspondencia:

2.4.1. MODULO RESIDENTES

El módulo “gestión de resientes” permite crear, consultar, modificar y eliminar los datos del residente, esta información se aloja en la base de datos.

2.4.2. MODULO CORRESPONDENCIA

El módulo “correspondencia” gestiona el proceso de recepción y entrega de correspondencia, la información de la correspondencia se asocia al residente y se almacena en la base de datos, es decir, toda correspondencia debe tener un residente que la reciba.

La correspondencia maneja 2 estados: PENDIENTE y ENTREGADA, en cada estado, se envía una notificación al número de WhatsApp que el residente tiene registrado en la base de datos.

Adicional, el guarda de seguridad podrá crear filtros de la correspondencia, ya sea por el estado en el que se encuentra o el residente al que pertenece.

3. CAPÍTULO III ARQUITECTURA

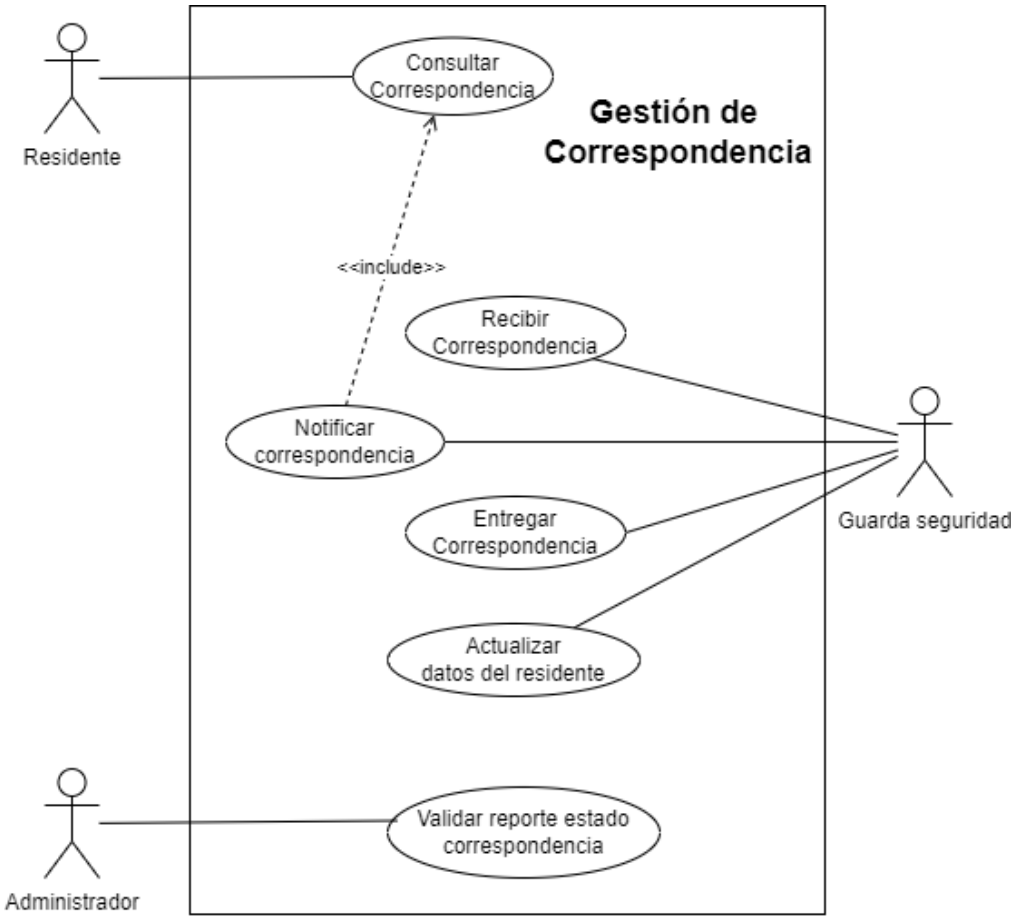
El objetico principal de la arquitectura de un sistema de software es proporcionar una comprensión común del sistema para guiar su diseño y construcción. Una arquitectura también puede actuar como un concepto inicial un sistema que proporciona una base para analizar y evaluar alternativas. Un tercer uso es permitir la ingeniería inversa (o arquitectura inversa) al ayudar a quienes trabajan con él a comprender un sistema de software existente antes de emprender tareas de mantenimiento, mejora o modificación. Crear una arquitectura bien planificada y documentada es una forma de mejorar la aplicabilidad y reutilización de los diseños y componentes de software. (Washizaki. 2024).

Las vistas ayudan a representar las diferentes perspectivas como se compone el sistema; resuelven diferentes problemas: por ejemplo, la vista lógica (describe cómo el sistema satisfará los requisitos funcionales); una vista de proceso (describe cómo el sistema utilizará la concurrencia); una vista física (describe cómo se implementará y distribuirá el sistema) y una vista de desarrollo (describe cómo se divide el diseño de nivel superior en unidades de implementación, las dependencias entre esas unidades y cómo se construirá la implementación). Separar las cuestiones por vista permite centrarse en unas pocas cosas a la vez y ofrece un medio para gestionar la comprensibilidad de la arquitectura y la complejidad general (Washizaki. 2024).

3.1.DIAGRAMA DE CASOS DE USO

FIGURA 3.1

Diagrama casos de uso del sistema de gestión de correspondencia.



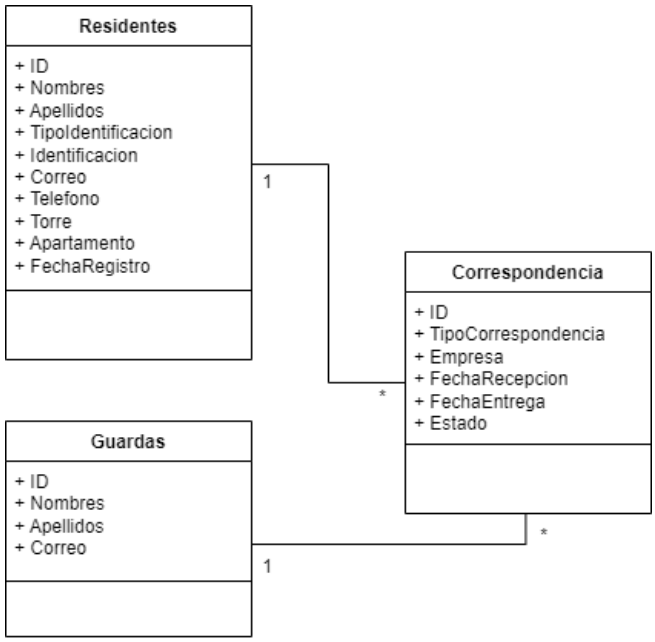
Fuente: Elaboración propia

Se definen 3 actores principales y las interacciones que realizan dentro del sistema.

3.2.DIAGRAMA DE CLASES

FIGURA 3.2

Diagrama de clases sistema de gestión de correspondencia.



Fuente: Elaboración propia

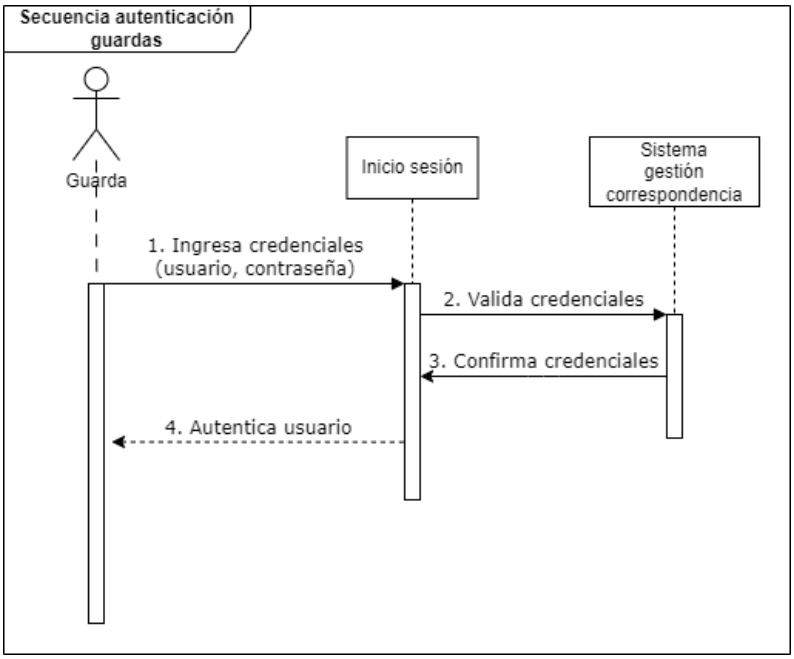
Se definen 3 clases que son “Residentes”, “Correspondencia” y “Guarda” con sus atributos y relaciones que son:

- Un residente puede tener muchas correspondencias (relación 1:N).
- Una correspondencia pertenece a un residente (relación N:1).
- Un guarda puede entregar muchas correspondencias (relación 1:N).
- Una correspondencia es entregada por un guarda (relación N:1).

3.3.DIAGRAMAS DE SECUENCIA

FIGURA 3.3.

Diagrama de secuencia autenticación del sistema.

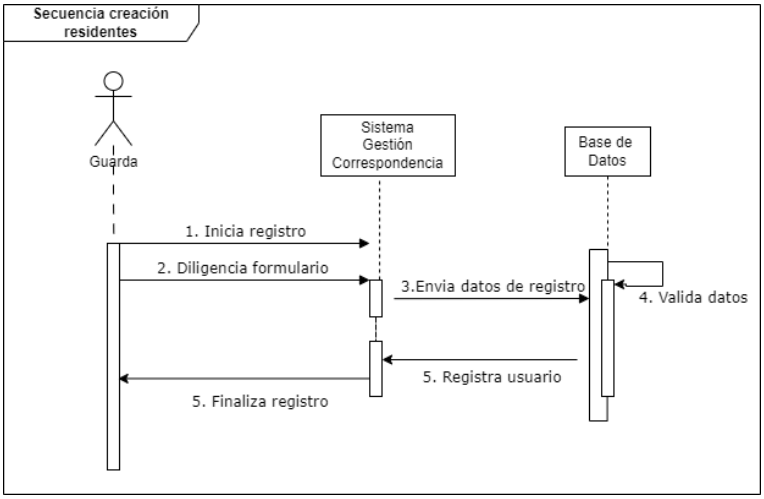


Fuente: Elaboración propia

El módulo de autenticación nos da una capa de seguridad en el sistema, ya que para poder acceder los guardas tendrán un usuario asignado.

FIGURA 3.4

Diagrama de secuencia creación de residentes.

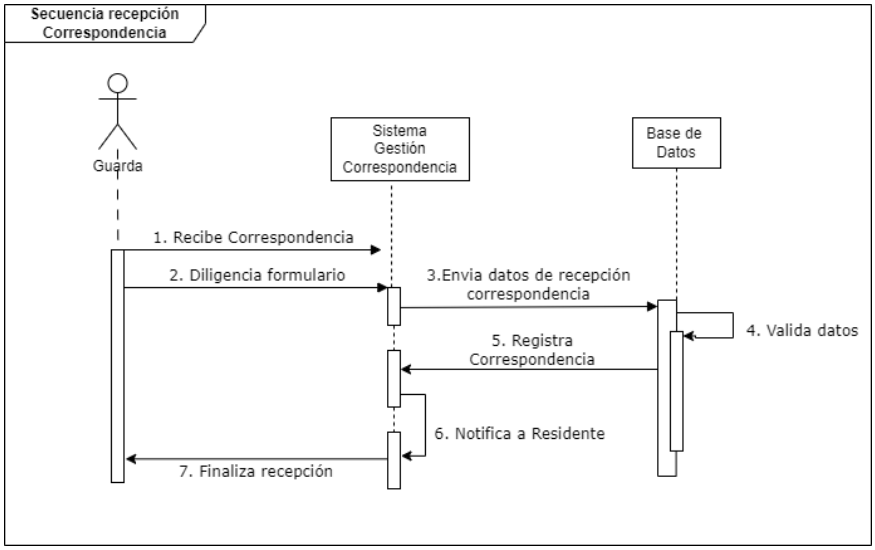


Fuente: Elaboración propia

El módulo de creación de residentes guarda la información en la base de datos.

FIGURA 3.5

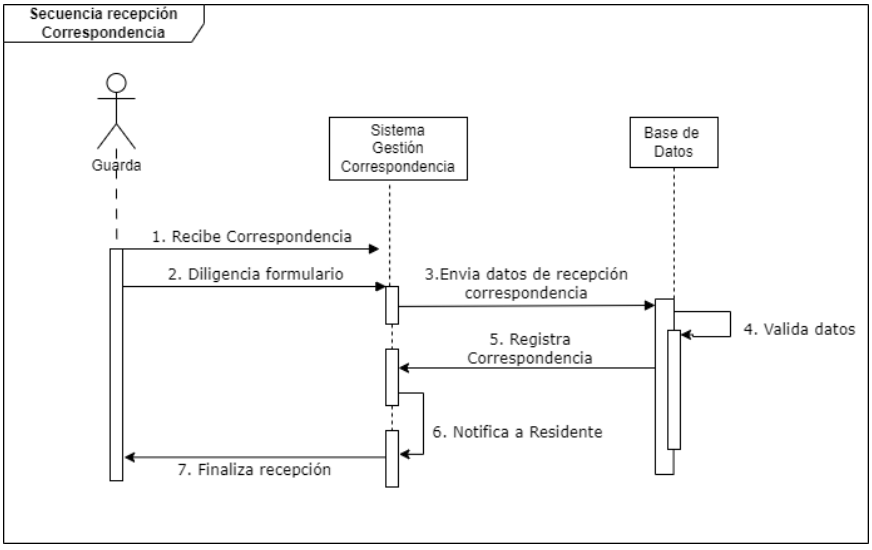
Diagrama de secuencia recepción de correspondencia.



Fuente: Elaboración propia

FIGURA 3.6

Diagrama de secuencia entrega de correspondencia.



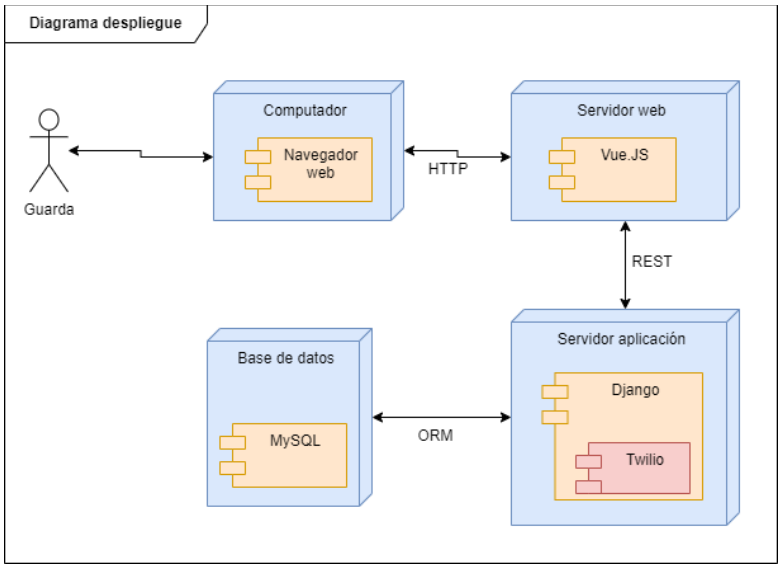
Fuente: Elaboración propia

El módulo de recepción y entrega de correspondencia muestra las iteraciones del guarda con el sistema, cada vez que se reciba o entregue correspondencia al residente se le envía un mensaje vía WhatsApp a su número de teléfono registrado en la base de datos.

3.4.DIAGRAMAS DE DESPLIEGUE

FIGURA 3.7

Diagrama de despliegue del sistema.



Fuente: Elaboración propia

El tipo de arquitectura del prototipo es de 3 capas:

- Presentación: Maneja la interfaz gráfica y la experiencia de usuario.
- Lógica de Negocio: Maneja la lógica del sistema de gestión de correspondencia.
- Datos: Se guarda la información en una base de datos. El backend se comunica con la base de datos para consultar, agregar o actualizar la información.

4. CAPÍTULO IV METODOLOGIA

Para desarrollar este proyecto se decidió utilizar el marco ligero SCRUM utilizando algunos de sus conceptos y filosofía, pero no en su totalidad.

El equipo consta de una persona, con conocimientos y experiencia en el marco metodológico (esta es una de las razones para elegir SCRUM), desde esta premisa, se omite el rol de Scrum Master y los roles se plantean de la siguiente forma:

- Sergio Andres Nova Aguilar (Product Owner y Desarrollador)

Esta persona es responsable de gestionar el Product Backlog y el Sprint Backlog para maximizar el valor del producto entregable.

Se definieron un total de 9 sprints en iteraciones de 2 semanas, en el siguiente cuadro se puede ver el plan de los sprints realizados.

TABLA 4.1

Planeación de las historias de usuario por sprint.

ID	Historia de usuario	Sprint	Puntos	Estado
PG-1	Requerimientos - Instrumentos de recolección	1	3	Finalizada
PG-2	Requerimientos - Elección de las herramientas a utilizar para desarrollar la aplicación	2	5	Finalizada
PG-3	Arquitectura - Creación de diagramas de la aplicación	3	3	Finalizada
PG-4	Arquitectura - Definición de la arquitectura de la aplicación	3	1	Finalizada
PG-5	DB - Diseño de la Base de Datos	4	5	Finalizada
PG-6	Backend - Creación modulo Residentes	5	5	Finalizada
PG-7	Backend - Creación modulo Correspondencia	5	5	Finalizada
PG-8	Backend - Integración con Twilio	6	8	Finalizada
PG-9	Backend - Creación modulo autenticación	7	5	Finalizada
PG-10	Frontend - Creación de la vista Residentes	8	5	Finalizada
PG-11	Frontend - Creación de la vista Correspondencia	8	5	Finalizada
PG-12	Frontend - Creación de la vista de Autenticación	9	5	Finalizada

Fuente: Elaboración Propia

TABLA 4.2

Historia PG-1

Requerimientos **[PG-1] Requerimientos - Instrumentos de recolección**			
Estado:	Finalizada		
Proyecto:	PG-GCorrespondencia		
Principal:	Requerimientos		
Tipo:	Historia	**Prioridad:**	Medium
Informador:	Sergio Andres Nova Aguilar	**Persona asignada:**	Sergio Andres Nova Aguilar
Resolución:	Listo	**Votos:**	0
Sprint:	Sprint 1		
Story point:	3		

Descripción

Yo como PO quiero diseñar los instrumentos de recolección, compartir con los residentes, tabular, analizar y concluir los datos que se levantaron.

Criterios de Aceptación

1. Diseñar los instrumentos de recolección, implementar, tabular, analizar y concluir los datos que se levantaron

Solución

1. Se diseñan los instrumentos de recolección Entrevista y Encuestas
2. Se comparten con los residentes
3. Se recopila la información
4. Se tabula la información
5. Se analiza la información
6. Se generan conclusiones de los datos obtenidos

Fuente: Elaboración Propia

TABLA 4.3

Historia PG-2

Requerimientos **[PG-2] Requerimientos - Elección de las herramientas a utilizar para desarrollar la aplicación**			
Estado:	Finalizada		
Proyecto:	PG-GCorrespondencia		
Principal:	Requerimientos		
Tipo:	Historia	**Prioridad:**	Medium
Informador:	Sergio Andres Nova Aguilar	**Persona asignada:**	Sergio Andres Nova Aguilar
Resolución:	Listo	**Votos:**	0
Sprint:	Sprint 2		
Story point estimate:	5		

Descripción

Yo como PO quiero que se revise las herramientas que se utilizarán para desarrollar la aplicación

Criterios de Aceptación

1. Validar las herramientas para desarrollar
2. Probar las herramientas validadas

Solución

El Backend se desarrollará en Django que ofrece muchas herramientas y funcionalidades como: autenticación de usuarios, panel de administración, manejo de formularios, protección CSRF, ORM (Object-Relational Mapping). Esto acelera mucho el desarrollo, especialmente en proyectos que requieren muchas funcionalidades.

La integración del Backend con el API de WhatsApp se realizará por medio de Twilio, una plataforma de interacción con los clientes.

El Frontend se desarrollará en VUE.JS

Fuente: Elaboración Propia

TABLA 4.4

Historia PG-3

Arquitectura **[PG-3] Arquitectura - Creación de diagramas de la aplicación**			
Estado:	Finalizada		
Proyecto:	PG-GCorrespondencia		
Principal:	Arquitectura		
Tipo:	Historia	**Prioridad:**	Medium
Informador:	Sergio Andres Nova Aguilar	**Persona asignada:**	Sergio Andres Nova Aguilar
Resolución:	Listo	**Votos:**	0
Sprint:	Sprint 3		
Story point estimate:	3		

Descripción

Yo como PO quiero que se diseñen los diagramas de comportamiento y estructurales necesarios para tener un concepto de la arquitectura de la aplicación.

Criterios de Aceptación

1. Diseñar los diagramas de arquitectura.

Solución

1. Se realizan los diagramas descritos.

Fuente: Elaboración Propia

TABLA 4.5

Historia PG-4

Arquitectura **[PG-4] Arquitectura - Definición de la arquitectura de la aplicación**			
Estado:	Finalizada		
Proyecto:	PG-GCorrespondencia		
Principal:	Arquitectura		
Tipo:	Historia	**Prioridad:**	Medium
Informador:	Sergio Andres Nova Aguilar	**Persona asignada:**	Sergio Andres Nova Aguilar
Resolución:	Listo	**Votos:**	0
Sprint:	Sprint 3		
Story point estimate:	1		

Descripción

Yo como PO quiero que se valide la arquitectura que se implementará para la aplicación.

Criterios de Aceptación

1. Validar la arquitectura para la aplicación.

Solución

Presentación: Maneja la interfaz gráfica y la experiencia de usuario.

Lógica de Negocio: Maneja la lógica del sistema de gestión de correspondencia.

Datos: Se guarda la información en una base de datos. El backend se comunica con la base de datos para consultar, agregar o actualizar la información.

Fuente: Elaboración Propia

TABLA 4.6

Historia PG-5

Base de datos **[PG-5] DB - Diseño de la Base de Datos**			
Estado:	Finalizada		
Proyecto:	PG-GCorrespondencia		
Principal:	Base de datos		
Tipo:	Historia	**Prioridad:**	Medium
Informador:	Sergio Andres Nova Aguilar	**Persona asignada:**	Sergio Andres Nova Aguilar
Resolución:	Listo	**Votos:**	0
Sprint:	Sprint 4		
Story point estimate:	5		

Descripción

Yo como PO quiero que se planifique y diseñe la base de datos que almacenará la información de los residentes y de la correspondencia.

Criterios de Aceptación

1. Diseñar con MySQL
2. Realizar pruebas de almacenamiento e integridad de datos.

Solución

Se diseña la base de datos para guardar la información de los resientes y de su correspondencia, se realizan pruebas de almacenamiento e integridad de datos.

Fuente: Elaboración Propia

TABLA 4.7

Historia PG-6

Backend **[PG-6] Backend - Creación modulo Residentes**			
Estado:	Finalizada		
Proyecto:	PG-GCorrespondencia		
Principal:	Backend		
Tipo:	Historia	**Prioridad:**	Medium
Informador:	Sergio Andres Nova Aguilar	**Persona asignada:**	Sergio Andres Nova Aguilar
Resolución:	Listo	**Votos:**	0
Sprint:	Sprint 5		
Story point estimate:	5		

Descripción

Yo como PO quiero que se desarrolle la lógica del módulo de resientes:

Criterios de Aceptación

1. Desarrollar el módulo de residentes que permita la creación, consulta, modificación y eliminación de residentes.

Solución

Se crea el módulo de resientes con las siguientes características:

1. Se puede crear, consultar modificar y eliminar la información del residente.

Fuente: Elaboración Propia

TABLA 4.8

Historia PG-7

Backend **[PG-7] Backend - Creación modulo Correspondencia**			
Estado:	Finalizada		
Proyecto:	PG-GCorrespondencia		
Principal:	Backend		
Tipo:	Historia	**Prioridad:**	Medium
Informador:	Sergio Andres Nova Aguilar	**Persona asignada:**	Sergio Andres Nova Aguilar
Resolución:	Listo	**Votos:**	0
Sprint:	Sprint 5		
Story point estimate:	5		

Descripción

Yo como PO quiero que se desarrolle la lógica del módulo de correspondencia:

Criterios de Aceptación

1. Desarrollar el módulo de que permita gestionar el proceso de recepción y entrega de correspondencia, la información de la correspondencia se asocia al residente y se almacena en la base de datos, es decir, toda correspondencia debe tener un residente que la reciba.
2. La correspondencia debe manejar 2 estados: PENDIENTE y ENTREGADA, en cada estado, se envía una notificación al número de WhatsApp que el residente tiene registrado en la base de datos.

Solución

1. Se desarrolla el módulo de correspondencia cumpliendo con los criterios de aceptación descritos.

Fuente: Elaboración Propia

TABLA 4.9

Historia PG-8

Backend **[PG-8] Backend - Integración con Twilio**			
Estado:	Finalizada		
Proyecto:	PG-GCorrespondencia		
Principal:	Backend		
Tipo:	Historia	**Prioridad:**	Medium
Informador:	Sergio Andres Nova Aguilar	**Persona asignada:**	Sergio Andres Nova Aguilar
Resolución:	Listo	**Votos:**	0
Sprint:	Sprint 6		
Story point estimate:	8		

Descripción

Yo como PO quiero que se integre el módulo de correspondencia con el API de WhatsApp por medio de la herramienta Twilio para que al momento de recibir o entregar una correspondencia está sea notificada al residente:

Criterios de Aceptación

1. Consultar información sobre la herramienta Twilio
2. Registrarse en la plataforma
3. Configurar Twilio en ambiente sandbox para integrar el backend con el API de WhatsApp
4. Verificar el funcionamiento del envió de notificaciones
5. Realizar pruebas

Solución

1. Se realiza la integración de backend con el API de WhatsApp por medio de Twilio.

Fuente: Elaboración Propia

TABLA 4.10

Historia PG-9

Backend **[PG-9] Backend - Creación modulo autenticación**			
Estado:	Finalizada		
Proyecto:	PG-GCorrespondencia		
Principal:	Backend		
Tipo:	Historia	**Prioridad:**	Medium
Informador:	Sergio Andres Nova Aguilar	**Persona asignada:**	Sergio Andres Nova Aguilar
Resolución:	Listo	**Votos:**	0
Sprint:	Sprint 7		
Story point estimate:	5		

Descripción

Yo como PO quiero que se desarrolle un módulo de autenticación para reforzar la seguridad de la aplicación, todos los guardas de seguridad deben tener un usuario para poder acceder al sistema

Criterios de Aceptación

1. Consultar información sobre el módulo de autenticación
2. Implementar el módulo
3. Realizar pruebas

Solución

1. Se realiza la integración del módulo de autenticación

Fuente: Elaboración Propia

TABLA 4.11

Historia PG-10

FrontEnd **[PG-10] Frontend - Creación de la vista Residentes**			
Estado:	Finalizada		
Proyecto:	PG-GCorrespondencia		
Principal:	FrontEnd		
Tipo:	Historia	**Prioridad:**	Medium
Informador:	Sergio Andres Nova Aguilar	**Persona asignada:**	Sergio Andres Nova Aguilar
Resolución:	Listo	**Votos:**	0
Sprint:	Sprint 8		
Story point estimate:	5		

Descripción

Yo como PO quiero diseñar la vista del módulo de resientes:

Criterios de Aceptación

1. Diseñar la vista del módulo de residentes que permita al guarda de seguridad la creación, consulta, modificación y eliminación de residentes.
2. Integrar la vista con el backend.

Solución

1. Se diseña la vista del módulo de creación de usuarios y se integra con el backend.

Fuente: Elaboración Propia

TABLA 4.12

Historia PG-11

FrontEnd **[PG-11] Frontend - Creación de la vista Correspondencia**			
Estado:	Finalizada		
Proyecto:	PG-GCorrespondencia		
Principal:	FrontEnd		
Tipo:	Historia	**Prioridad:**	Medium
Informador:	Sergio Andres Nova Aguilar	**Persona asignada:**	Sergio Andres Nova Aguilar
Resolución:	Listo	**Votos:**	0
Sprint:	Sprint 8		
Story point estimate:	5		

Descripción

Yo como PO quiero diseñar la vista del módulo de correspondencia:

Criterios de Aceptación

1. Diseñar la vista del módulo de correspondencia que permita al guarda de seguridad la recepción y entrega de la correspondencia de los residentes.
2. Integrar la vista con el backend.

Solución

Se diseña la vista del módulo correspondencia y se integra con el backend.

Fuente: Elaboración Propia

TABLA 4.13

Historia PG-12

FrontEnd **[PG-12] Frontend - Creación de la vista de Autenticación**			
Estado:	Finalizada		
Proyecto:	PG-GCorrespondencia		
Principal:	FrontEnd		
Tipo:	Historia	**Prioridad:**	Medium
Informador:	Sergio Andres Nova Aguilar	**Persona asignada:**	Sergio Andres Nova Aguilar
Resolución:	Listo	**Votos:**	0
Sprint:	Sprint 9		
Story point estimate:	5		

Descripción

Yo como PO quiero diseñar la vista del módulo de autenticación:

Criterios de Aceptación

1. Diseñar la vista del módulo de autenticación que solicite usuario y contraseña al momento de ingresar a la aplicación
2. Integrar con el backend.

Solución

Se diseña la vista del módulo autenticación y se integra con el backend.

Fuente: Elaboración Propia

5. CAPÍTULO V MODELO DE DATOS

El almacenamiento de los datos de los residentes y personal de seguridad se realiza con MySQL, y se consultan con MySQL Workbench.

La comunicación del Backend con la base de datos se realiza utilizando ORM (Object-Relational Mapping) de Django, que proporciona una capa de abstracción que permite interactuar con bases de datos relacionales utilizando objetos de Python en lugar de escribir consultas SQL.

El modelado de datos se enfoca en la relación entre los residentes y la correspondencia que reciben:

5.1.RELACIÓN ENTRE ENTIDADES

5.1.1. TABLA RESIDENTES

- Es el punto de partida para identificar un residente en el sistema.
- Se relaciona con la tabla en donde se almacenan las correspondencias.

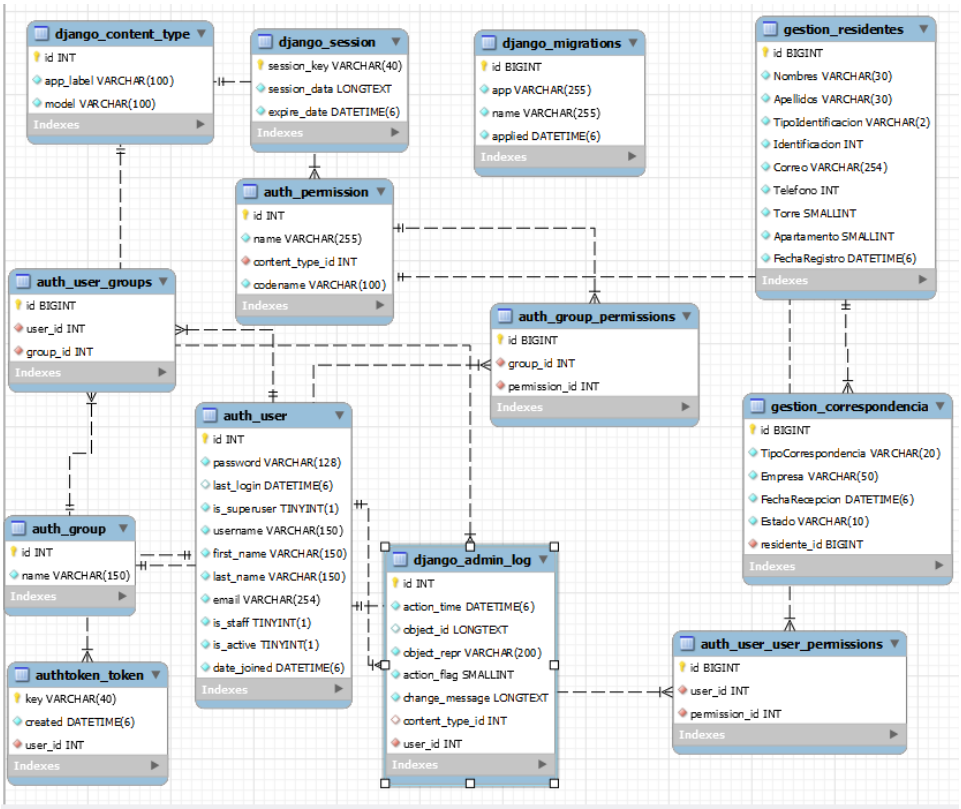
5.1.2. TABLA CORRESPONDENCIA

- Se relaciona con la tabla Residentes mediante una clave foránea, es decir, un residente puede recibir muchas correspondencias, pero cada correspondencia pertenece a un único residente.

A continuación, se presenta el diagrama entidad/relación del prototipo, se pueden observar las tablas y los atributos de cada una de ellas:

FIGURA 5.1

Diagrama entidad/relación modelo de datos.



Fuente: Elaboración propia.

6. CAPÍTULO VI PROTOTIPO GESTIÓN DE CORRESPONDENCIA

De forma local se instala MySQL para la base de datos, Django para el desarrollo de la lógica de la aplicación y para las vistas de cada módulo se trabajó con VUE.JS, la arquitectura del prototipo es de 3 capas, al separarse las responsabilidades por capas facilita el crecimiento del sistema y la adaptación a nuevos requerimientos.

6.1.HERRAMIENTAS

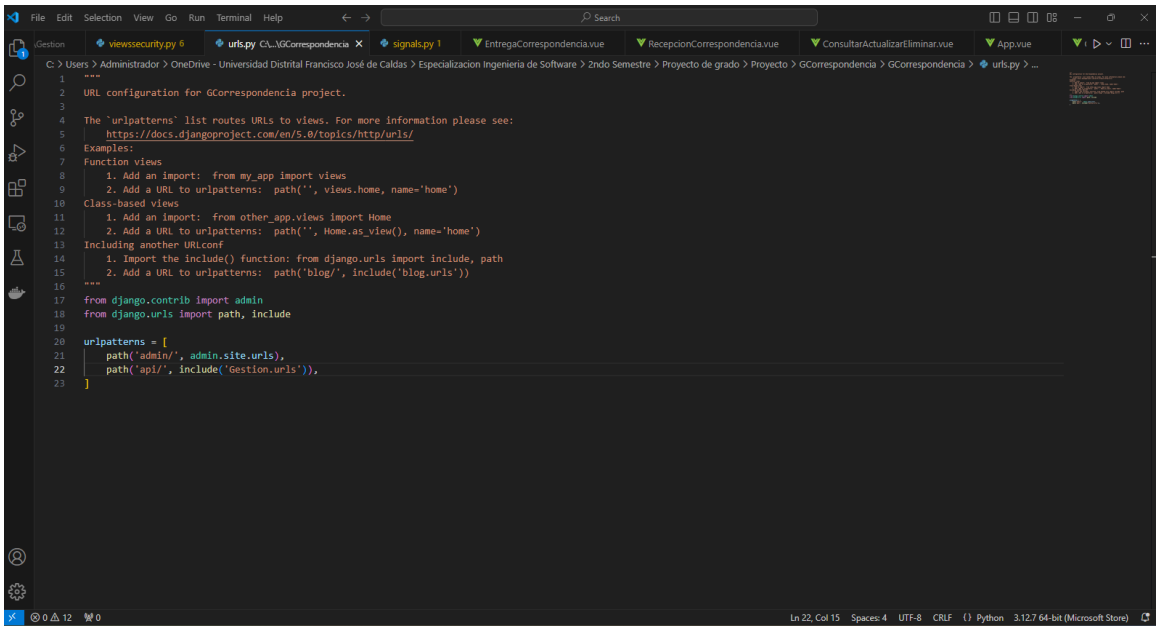
Las herramientas descritas a continuación permitieron diseñar, desarrollar y probar el prototipo.

6.1.1. VISUAL STUDIO CODE

VS Code es un editor de código gratuito para los sistemas operativos macOS, Linux y Windows. Es liviano y compatible con la mayoría de las versiones de hardware y plataformas disponibles. Toda la codificación del prototipo se realizó con este editor ligero, fácil de administrar que admite casi todos los principales lenguajes de programación (VS Code, 2024).

FIGURA 6.1

Interfaz de usuario VS Code.



Fuente: Elaboración propia.

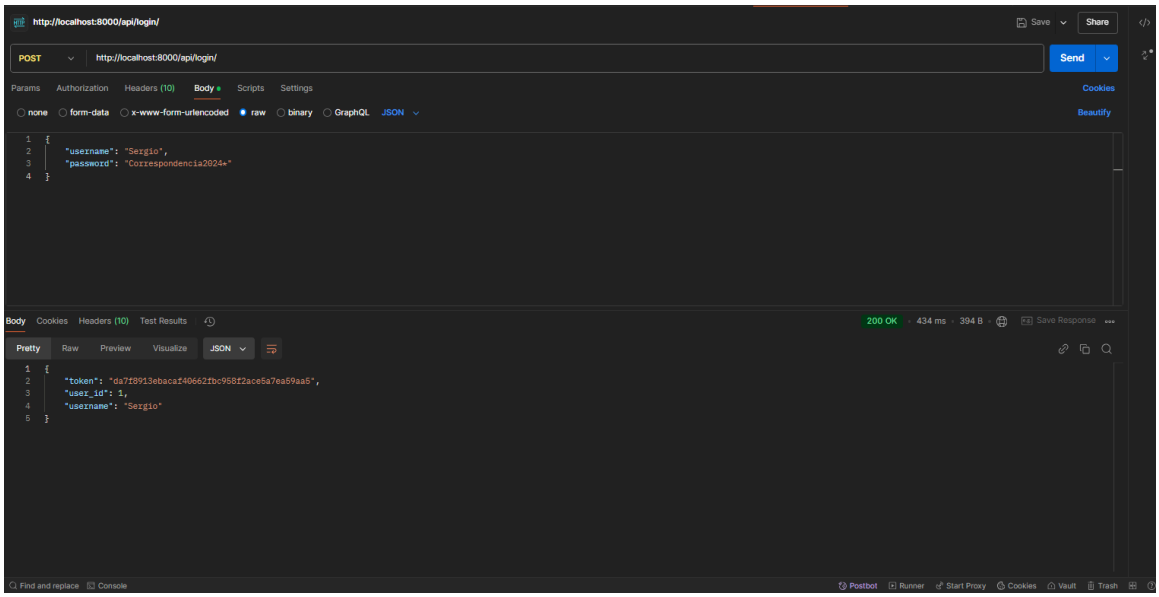
6.1.2. POSTMAN

Es una plataforma de desarrollo y prueba de API que ofrece una variedad de funciones. Permite crear y compartir fácilmente solicitudes y colecciones de API, automatizar pruebas, simular API y monitorear el rendimiento (Postman, 2024).

Las validaciones de los métodos implementados en el backend como la gestión de correspondencia se probaron con Postman para después ser integradas con el frontend.

FIGURA 6.2.

Interfaz de usuario Postman.



Fuente: Elaboración propia.

6.1.3. BOOTSTRAP

Bootstrap es una potente biblioteca de desarrollo front-end creada por Mark Otto y Jacob Thornton, que incluye componentes HTML, CSS y JS prediseñados para crear sitios web.

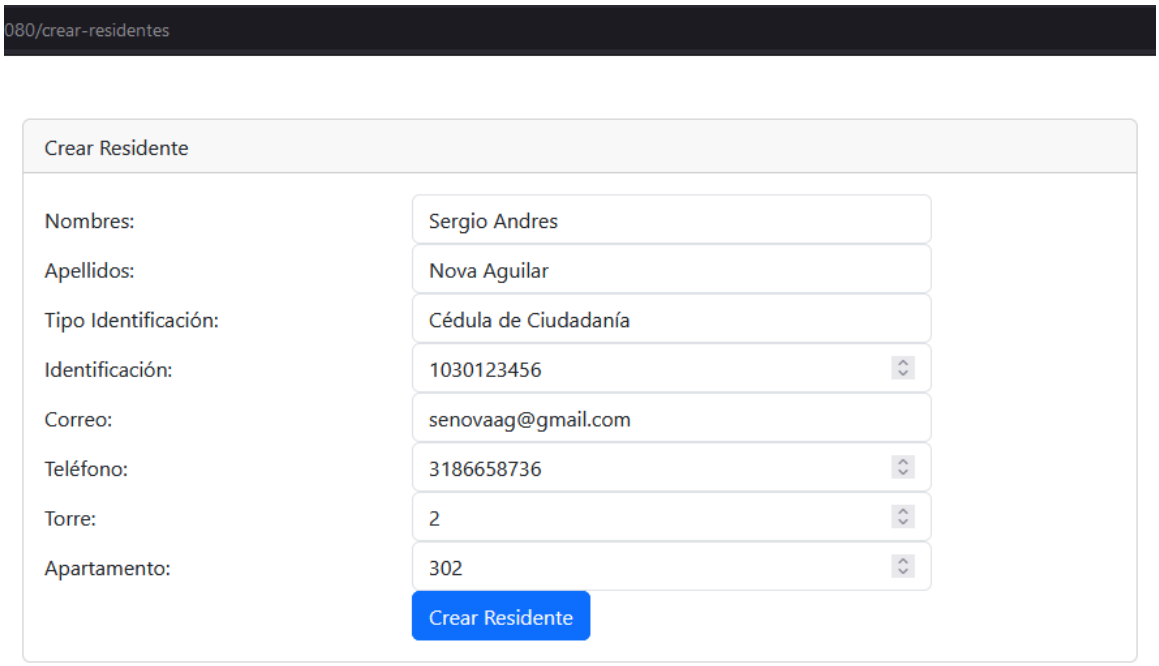
Bootstrap básicamente ayuda a los desarrolladores a ahorrar mucho tiempo escribiendo CSS y HTML al proporcionar plantillas gratuitas como punto de partida para su proyecto web. (Londoño, 2024).

6.2.MODULOS

6.2.1. MODULO REGISTRO RESIDENTES

FIGURA 6.3

Interfaz de usuario módulo creación de residentes.



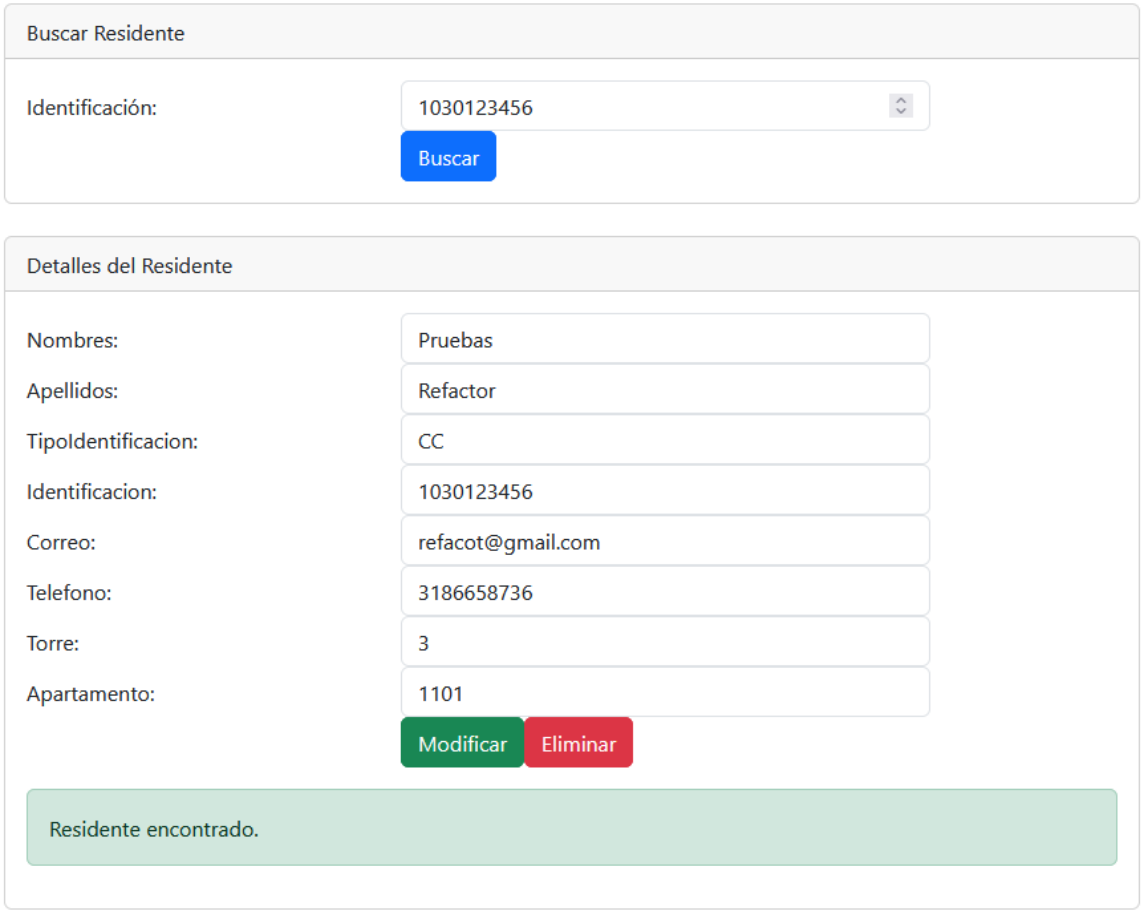
Fuente: Elaboración propia.

Este módulo permite a los guardas de seguridad crear residentes que no estén registrados en el sistema, diligencia el formulario, se validan los datos insertados y si son correctos, se guardan en la base de datos.

6.2.2. MODULO CONSULTA RESIDENTES

FIGURA 6.4

Interfaz de usuario módulo consulta de residentes.



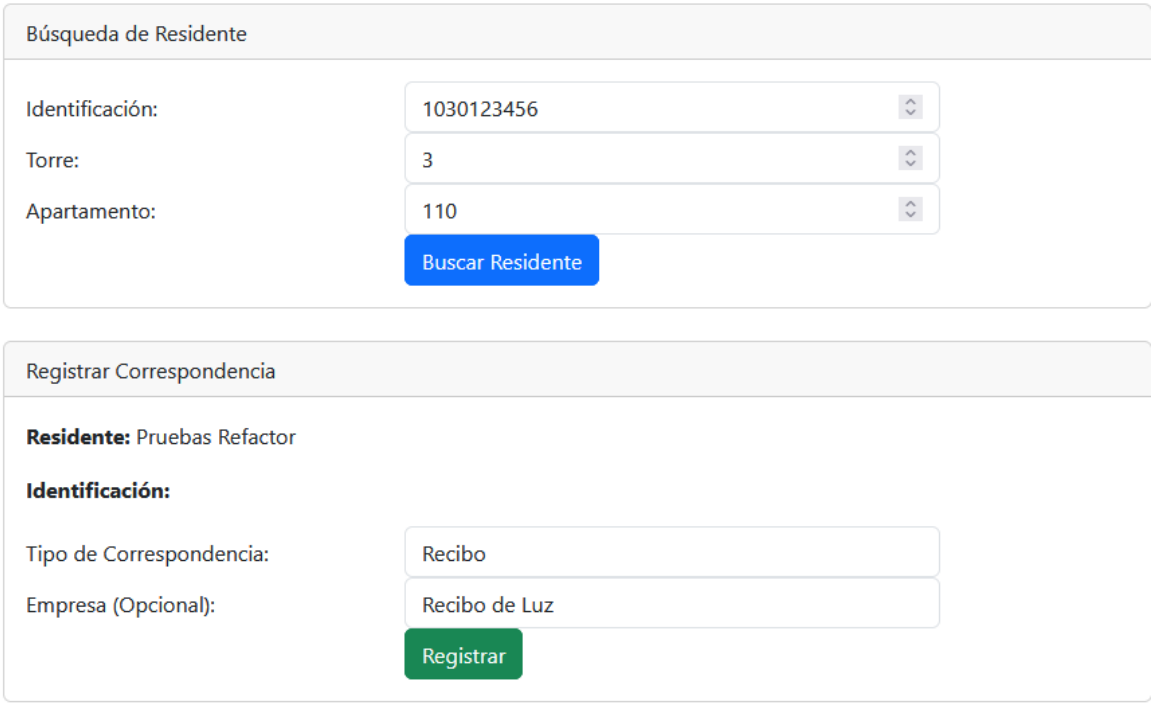
Fuente: Elaboración propia.

La consulta de los residentes se realiza por medio de su número de documento, si el registro existe, presenta la información en una nueva vista. Además, en este módulo se puede modificar o eliminar la información del residente.

6.2.3. MODULO RECEPCIÓN CORRESPONDENCIA

FIGURA 6.5

Interfaz de usuario módulo recepción de correspondencia.

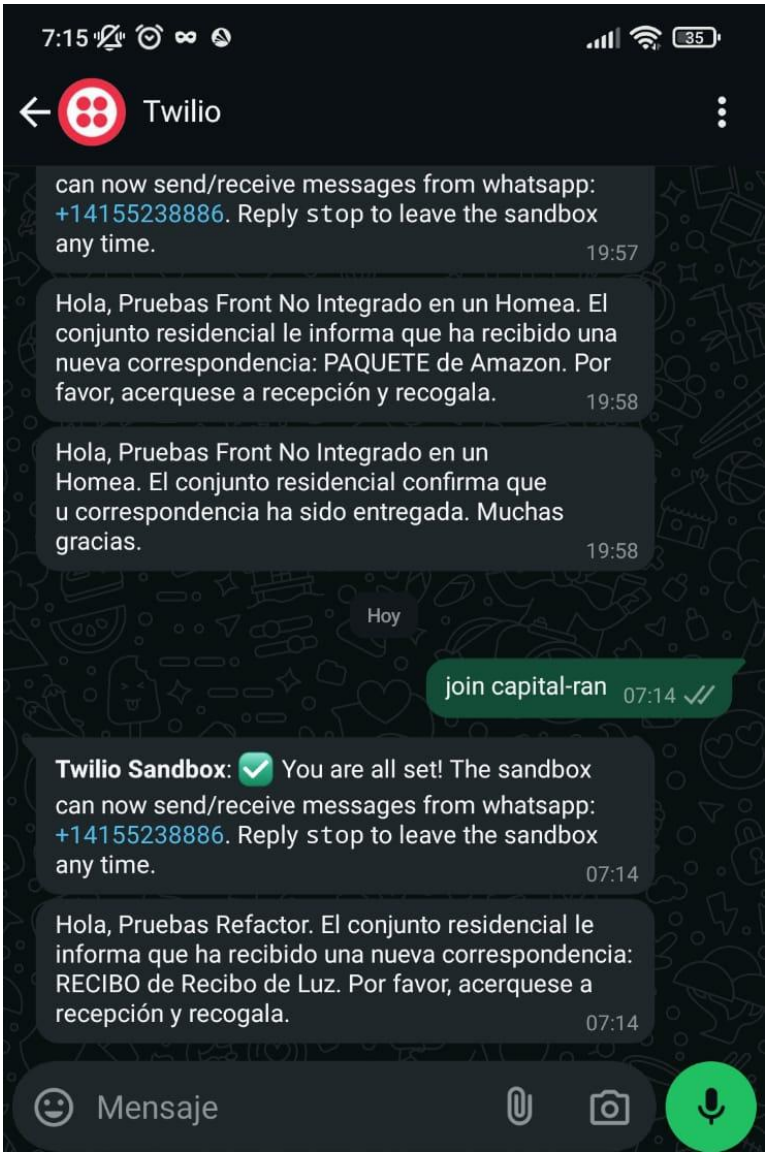


Fuente: Elaboración propia.

Se consulta el número de identificación, la torre y el apartamento, en caso de que coincida con los datos del residente se despliega un nuevo listado que permite registrar la correspondencia del residente, posterior a ser registrada, llega una notificación al número de WhatsApp que el residente tiene registrado como se ve en la *Figura 6.5*

FIGURA 6.6

Notificación enviada al residente vía WhatsApp.

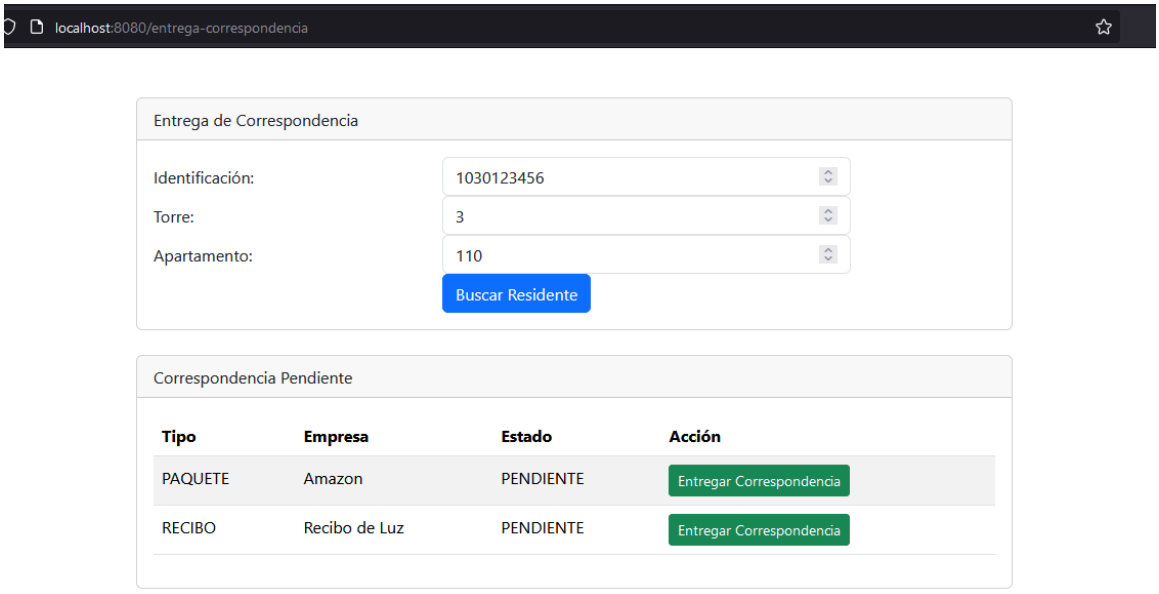


Fuente: Elaboración propia.

6.2.4. MODULO ENTREGA CORRESPONDENCIA

FIGURA 6.7

Interfaz de usuario módulo entrega de correspondencia.

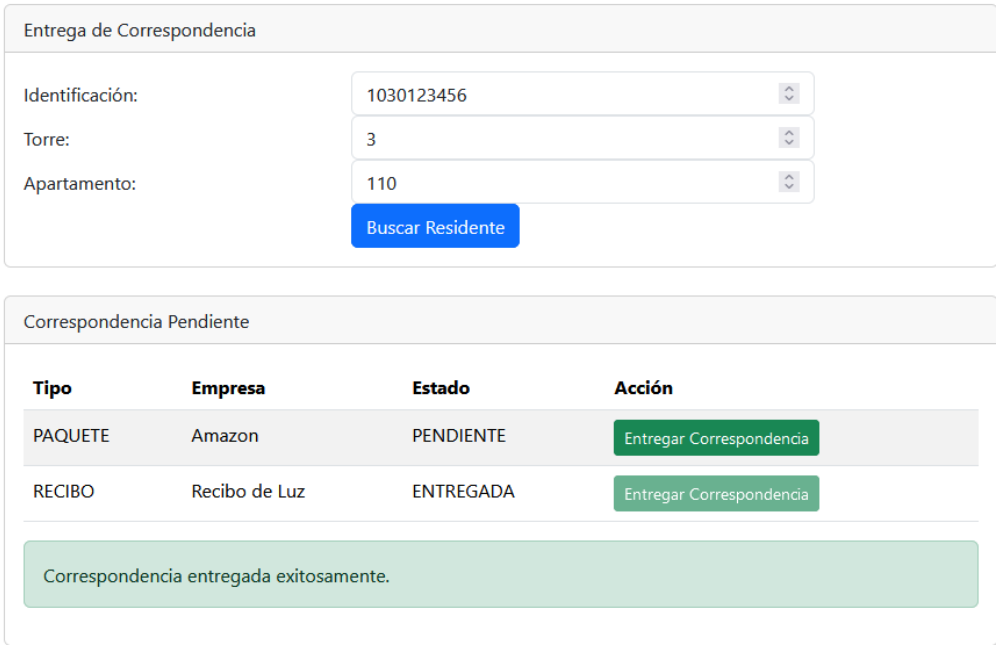


Fuente: Elaboración propia.

Para entregar la correspondencia se debe consultar el número de identificación, la torre y el apartamento, en caso de que coincida con los datos almacenados se desplegara un listado con la correspondencia que está pendiente por entregar, se entrega la correspondencia y al residente llega una notificación al número de WhatsApp que tiene registrado indicando que la correspondencia ha sido entregada

FIGURA 6.8

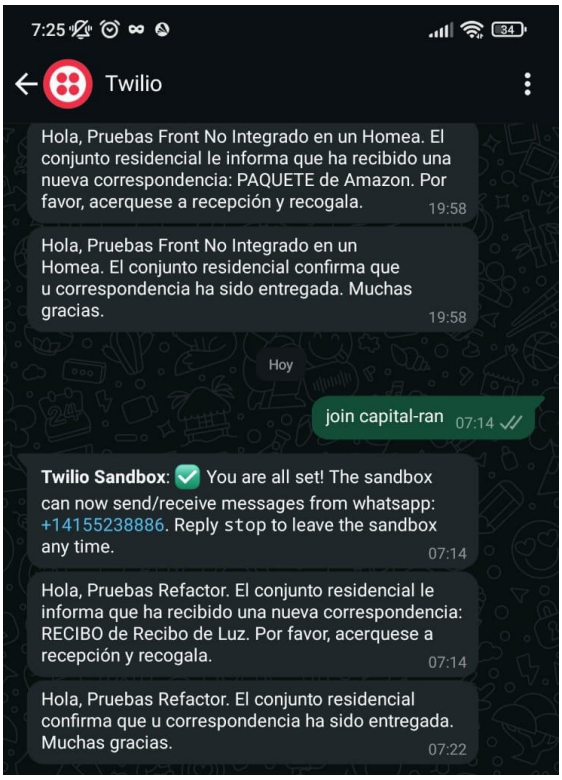
Interfaz de usuario módulo entrega de correspondencia.



Fuente: Elaboración propia.

FIGURA 6.9

Notificación de entrega de correspondencia enviada al residente vía WhatsApp.



Fuente: Elaboración propia.

PARTE III CIERRE

7. CAPÍTULO VII RESULTADOS Y DISCUSIÓN

Los resultados de la investigación y el desarrollo del prototipo nos dan una aplicación web con las principales funcionalidades de un sistema de gestión de correspondencia que se propuso al inicio del proyecto, que se desarrolló en un tiempo determinado y que fue guiada bajo la metodología SCRUM. La aplicación puede ser accesible desde varios dispositivos como computadores, portátiles, smartphones y tablets y también desde cualquier navegador.

La arquitectura implementada permite una interacción limpia, organizada facilitando el crecimiento del sistema y la adaptación a nuevos requerimientos.

8. CAPÍTULO VIII TRABAJO FUTURO

Aunque el proyecto se enfoca en un proceso particular dentro de la administración del conjunto residencia puede expandirse y ser parte de una aplicación más completa como una red social o una plataforma de administración de conjuntos residenciales en la cual existan módulos tipo: pagos, asambleas, zonas comunes etc.

Adicional, podría expandirse en conjuntos residenciales que manejan su correspondencia en casilleros para hacerlos “inteligentes” por medio de IoT, en el cual, al momento de que se reciba una correspondencia en el casillero, se notifique al residente.

9. CAPÍTULO IX CONCLUSIONES

En este tipo de proyectos es frecuente encontrar procesos manuales los cuales son propensos a errores y generar insatisfacción entre los residentes y la administración del conjunto, la automatización de procesos juega un papel importante a la hora de agilizar procesos, aumentar la eficiencia, reducir los errores y optimizar los costos.

Comprendiendo las necesidades, comportamiento y preferencias de los actores del proceso se desarrolló un prototipo simple de manejar y a la vez completo que optimizó tiempos de notificación, centralizó los datos garantizando confidencialidad, integridad y disponibilidad de la información.

La arquitectura del sistema nos permite entender las entre los participantes del proceso y guiar su diseño, desarrollo, construcción e implementación, separando responsabilidades, facilitando el crecimiento del sistema y la adaptación a nuevas necesidades.

La administración y los residentes que hicieron parte del proyecto ven con buenos ojos la notificación de su correspondencia. De cada iteración con ellos surgían nuevas ideas para ajustar el prototipo ofreciendo mejora continua.

ANEXOS

A ANEXO I: ENCUESTAS

1. Tipo de residente
   i) Propietario
   ii) Arrendatario

2. ¿Con qué frecuencia recibe correspondencia? (0 punto)
   i) Diariamente
   ii) Semanalmente
   iii) Mensualmente

3. ¿Es notificado por el personal de seguridad del conjunto cuando tiene correspondencia en la recepción?
   i) SI
   ii) Algunas veces
   iii) NO

4. ¿Le gustaría ser notificado cuando recibe correspondencia?
   i) SI
   ii) NO

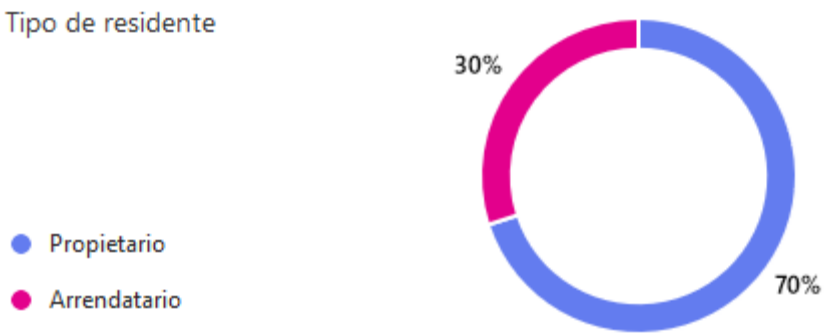
5. ¿Cuál es su medio de notificación de preferencia?
   i) Correo
   ii) SMS
   iii) Llamada
   iv) WhatsApp
   v) Otro

6. ¿Ha presentado problemas, inconvenientes o novedades con la entrega y recepción de su correspondencia? (0 punto)
   i) SI
   ii) NO

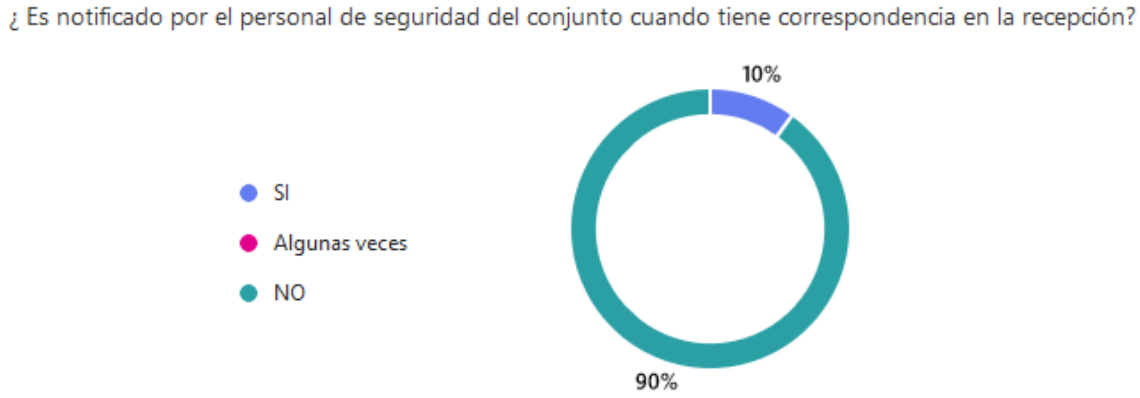
7. ¿Ha tenido experiencia manejando aplicaciones web?
   i) SI
   ii) NO

8. ¿Ha presentado problemas, inconvenientes o novedades con la entrega y recepción de la correspondencia?
   i) SI
   ii) NO

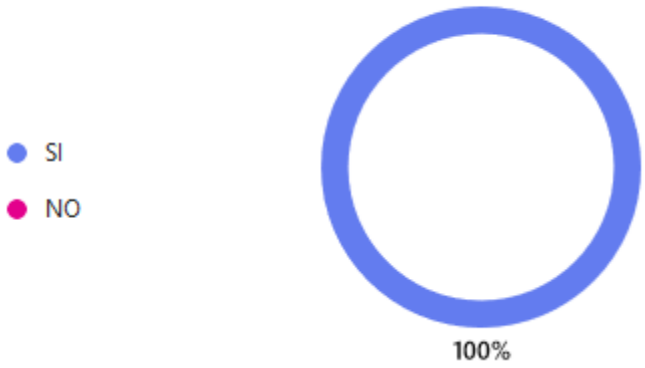
B ANEXO II: TABULACIÓN DE LA INFORMACIÓN



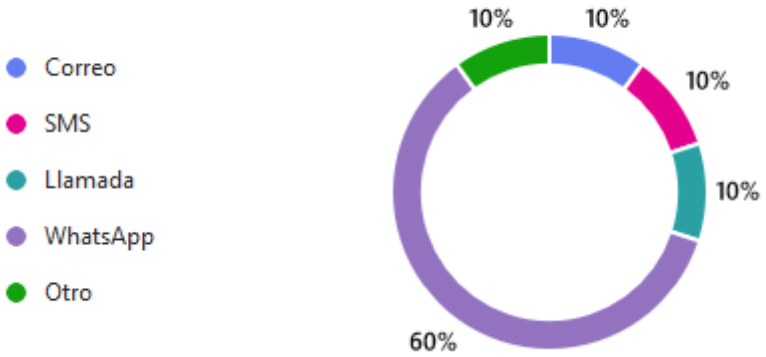




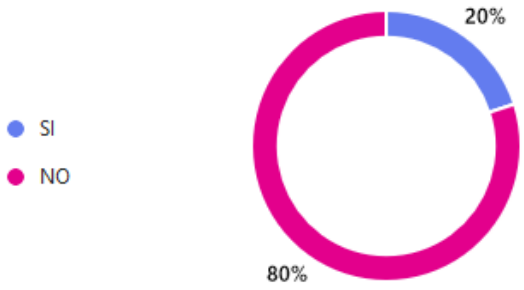
¿ Le gustaría ser notificado cuando recibe correspondencia?



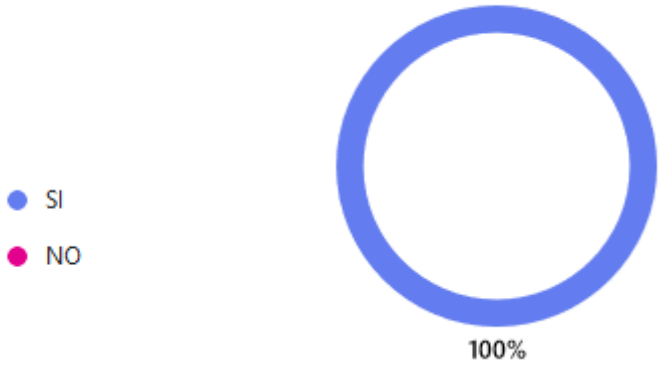
¿Cuál es su medio de notificación de preferencia?



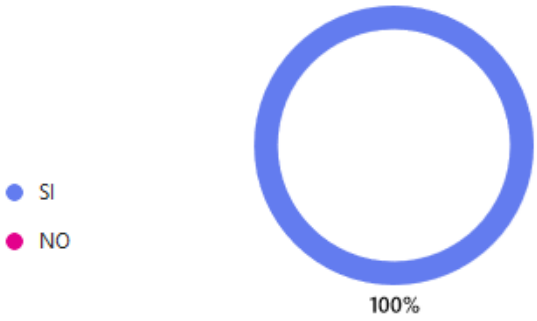
¿Ha presentado problemas, inconvenientes o novedades con la entrega y recepción de su correspondencia?



¿ Ha tenido experiencia manejando aplicaciones web?



¿Ha presentado problemas, inconvenientes o novedades con la entrega y recepción de la correspondencia?



10. REFERENCIAS

Pérez, M. (17 de noviembre de 2022). Resuelve dudas y conoce más sobre la Propiedad Horizontal. Pérez Lara - Expertos en Asesoría de Seguros. https://perezlara.com/propiedad-horizontal-que-es/

Portafolio. (18 de junio de 2022). Expertos de la propiedad horizontal se reúnen en Colombia. https://www.portafolio.co/tendencias/sociales/expertos-de-la-propiedad-horizontal-se-reunen-en-colombia-567127

García, M. (22 de marzo de 2023). Innovación tecnológica: qué es, sus tipos y sus beneficios. Tecnológico de Monterrey. https://blog.maestriasydiplomados.tec.mx/innovacion-tecnologica-que-es-sus-tipos-y-sus-beneficios

Ayerdi, A. (6 de noviembre de 2024). Automatización de procesos: La clave para la eficiencia. DocuWare. https://start.docuware.com/es/blog/automatizacion-de-procesos

Acevedo, U. C., & Piedrahita, N. (2016). Régimen de Propiedad Horizontal en Colombia: Estudio del alcance de la Ley 675 de 2001 y su eficacia frente a la relación entre copropietarios y administrador a 2016 (Artículo como trabajo de grado). Universidad de San Buenaventura Medellín, Facultad de Derecho.

Colombia. (3 de agosto de 2001). Ley 675 de 2001 Congreso de la República de Colombia. Alcaldía Mayor de Bogotá. https://www.alcaldiabogota.gov.co/sisjur/normas/Norma1.jsp?i=4162

Colombia. (30 de octubre de 2012). Ley 1582 De 2012 del Congreso de la República de Colombia. Secretaria del Senado. http://www.secretariasenado.gov.co/senado/basedoc/ley_1582_2012.html

Ken Schwaber & Jeff Sutherland (2020). La Guía Scrum. Scrum Guides.https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-European.pdf.

H. Washizaki, eds., Guide to the Software Engineering Body of Knowledge (SWEBOK Guide), Version 4.0, IEEE Computer Society, 2024; www.swebok.org.

Mozilla. (22 de noviembre de 2024). *Introduction to Django*. Mozilla Developer Network. https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Introduction

Erickson. Jeffrey, (29 de agosto de 2024). MySQL: Understanding What It Is and How It's Used. Oracle. https://www.oracle.com/mysql/what-is-mysql/

Twilio. (s.f.). ¿Qué es Twilio? Una introducción a la plataforma líder de interacción con el cliente. Twilio. https://www.twilio.com/es-mx/resource-center/what-is-twilio-an-introduction-to-the-leading-customer-engagement-platform

Vue.js (s.f.). ¿Qué es Vue.js? Vue.js. https://es.vuejs.org/v2/guide/

VS Code (10 de octubre de 2024). Setting up Visual Studio Code. Visual Studio Code. https://code.visualstudio.com/docs/setup/setup-overview

Postman (s.f.). What is Postman?. Postman. https://www.postman.com/product/what-is-postman/

Londoño. P, (30 de mayo de 2024). Qué es Bootstrap, para qué sirve y cómo funciona. Hubspot. https://blog.hubspot.es/website/que-es-bootstrap#que-es