

UNIVERSIDAD NACIONAL AUTÓNOMA DE NICARAGUA UNAN-León
ÁREA DE CONOCIMIENTO CIENCIAS Y TECNOLOGÍA
ÁREA ESPECÍFICA INGENIERÍA EN SISTEMA DE INFORMACIÓN
CARRERA INGENIERÍA EN SISTEMA DE INFORMACIÓN



TESIS MONOGRÁFICA PARA OPTAR POR EL TÍTULO DE
INGENIERÍA EN SISTEMA DE INFORMACIÓN

**Aplicación web de gestión y administración de servicios de alquiler de
cuartos o habitaciones a estudiantes universitarios de la ciudad de León
aplicando el lenguaje Python.**

Autores:

Br. Katerin Maribel Montes Pichardo.
Br. Nathaly Gissel Mendoza.
Br. Mynor Miguel Arauz Rugama.

Tutor:

M.Sc. William Noel Martínez Orozco

León, Nicaragua, diciembre del 2025

2025: 46/19 Siempre más allá! ¡avanzamos en la revolución!

UNIVERSIDAD NACIONAL AUTÓNOMA DE NICARAGUA UNAN-León

ÁREA DE CONOCIMIENTO CIENCIAS Y TECNOLOGÍA

ÁREA ESPECÍFICA INGENIERÍA EN SISTEMA DE INFORMACIÓN

CARRERA INGENIERÍA EN SISTEMAS DE INFORMACIÓN



**TESIS MONOGRÁFICA PARA OPTAR POR EL TÍTULO DE
INGENIERÍA EN SISTEMA DE INFORMACIÓN**

**Aplicación web de gestión y administración de servicios de alquiler de
cuartos o habitaciones a estudiantes universitarios de la ciudad de León
aplicando el lenguaje Python.**

Tutor M.Sc. William Noel Martínez Orozco

Br. Katerin Maribel Montes Pichardo

Br. Nathaly Gissel Mendoza

Br. Mynor Miguel Arauz Rugama

León, Nicaragua, diciembre del 2025

2025: 46/19 Siempre más allá! ¡avanzamos en la revolución!

Agradecimiento

Agradecemos primeramente a Dios, fuente de sabiduría, fortaleza y guía en cada paso de nuestro camino académico y personal. Gracias por permitirnos culminar con éxito esta importante etapa de nuestras vidas y por brindarnos la perseverancia necesaria para superar cada reto.

Expresamos nuestro profundo agradecimiento a nuestras familias, por su amor incondicional, comprensión y apoyo constante. Su confianza y palabras de aliento fueron el motor que nos impulsó a seguir adelante, incluso en los momentos más difíciles.

De igual manera, extendemos un sincero agradecimiento a nuestro tutor, M.Sc. William Noel Martínez Orozco, por su orientación, paciencia y valiosos aportes durante el desarrollo de este trabajo. Su acompañamiento fue esencial para alcanzar los objetivos propuestos y fortalecer nuestros conocimientos.

Agradecemos también a nuestros docentes y compañeros de la carrera de Ingeniería en Sistemas de Información de la Universidad Nacional Autónoma de Nicaragua, UNAN-León, por compartir experiencias, conocimientos y esfuerzos que enriquecieron nuestra formación profesional.

Finalmente, agradecemos a todas las personas que, de una u otra manera, contribuyeron al logro de este proyecto, brindándonos su apoyo, motivación y confianza a lo largo de este proceso académico.

Dedicatoria

Dedicamos este trabajo, en primer lugar, a Dios, por ser nuestra guía y fortaleza en cada momento de nuestra vida, por permitirnos alcanzar este logro y darnos la sabiduría necesaria para culminar con éxito este proyecto académico.

A nuestras familias, quienes con amor, paciencia y apoyo incondicional nos acompañaron en todo este proceso. Su comprensión, sacrificio y motivación fueron el pilar fundamental que nos impulsó a seguir adelante y no rendirnos ante las dificultades.

Dedicamos también este esfuerzo a nuestros docentes, quienes con su enseñanza, exigencia y orientación contribuyeron a nuestra formación profesional y personal, inculcándonos el valor del esfuerzo, la disciplina y la responsabilidad.

Finalmente, dedicamos este logro a todos aquellos que creyeron en nosotros, que nos brindaron su apoyo moral y nos animaron a alcanzar nuestras metas. Este trabajo representa no solo un resultado académico, sino también el fruto de la constancia, la unión y la dedicación compartida.

Indice

1	Introducción	1
2	Planteamiento del problema	2
3	Justificación	3
4	Antecedentes	4
4.1	Antecedentes nacionales	4
4.2	Antecedentes internacionales	5
5	Objetivos	6
5.1	Objetivo general	6
5.2	Objetivos específicos	6
6	Marco teórico	7
6.1	Contratos de alquiler	7
6.2	Tipos de contratos de alquiler	7
6.3	Alquiler	7
6.4	Tipo de alquiler	8
6.5	Periodo de alquiler	8
6.6	Pago de alquiler	8
6.7	Formas de pago permitidas	8
6.8	Tipos de habitaciones	9
6.9	Criterios de elección del estudiante	9
6.10	Servicios valorados por estudiantes	10
6.11	Factores sociales, psicológicos y de diseño	10

6.12	Condiciones para recibir el pago-----	11
6.13	¿Qué es reservación de habitación? -----	11
6.14	Forma de pago de reservación -----	11
6.15	Periodo de pre-reservación-----	12
6.16	Cómo ubican la habitación o cuarto-----	12
6.17	Fin del inquilinato-----	13
6.18	Tecnología para la administración del alquiler -----	13
6.19	Automatización de procesos administrativos -----	13
6.20	Herramientas para el desarrollo del sistema-----	13
6.20.1	Python-----	13
6.20.2	Flask-----	14
6.20.3	Sqlite -----	14
6.20.4	Db Browser For Sqlite -----	14
6.20.5	HTML5 Y CSS3-----	14
6.20.6	Javascript (Es6+)-----	15
6.20.7	Visual Studio Code -----	15
6.21	Sistemas de información y su importancia -----	15
6.22	Tecnologías y herramientas de desarrollo-----	15
6.23	Tecnologías del lado del servidor -----	16
6.24	Tecnologías del lado del cliente -----	16
6.25	Herramientas de desarrollo-----	16
7	Metodología -----	17
7.1	Tipo de estudio, descriptivo de corte transversal -----	17

7.2	Área de estudio	18
7.2.1	Descripción del sistema de información	18
7.3	Población de estudio	18
7.3.1	Definición y justificación focalizada en la gestión	19
7.3.2	Criterios de inclusión	19
7.4	Procedimiento y fuentes de la obtención de la información	20
7.4.1	Procedimiento de recolección de datos	20
7.4.2	Descripción detallada de la entrevista semi-estructurada	21
7.5	Metodología ágil Scrum	22
7.6	Personal Involucrado	23
7.7	Etapas del proceso Scrum aplicado al proyecto	24
7.7.1	Planificación inicial y creación del Product Backlog	24
7.7.2	Planificación del Sprint (Sprint Planning)	24
7.7.3	Ejecución del Sprint (Sprint Execution)	24
7.7.4	Revisión del Sprint (Sprint Review)	24
7.7.5	Retrospectiva del Sprint (Sprint Retrospective)	25
7.7.6	Entrega e implementación del producto final	25
7.8	Conclusión metodológica	25
8	Análisis	26
8.1	Especificación de requisitos de software (ERS)	26
8.1.1	Introducción	26
8.1.2	Propósito	26
8.1.3	Alcance	26

8.1.4	Definiciones y Acrónimos	27
8.1.5	Perspectiva del producto	28
8.1.6	Funcionalidad del producto	28
9	Requisitos específicos	28
9.1	Requisitos funcionales	28
9.2	Requisitos no funcionales (RNF)	34
9.2.1	Requisitos de rendimiento	34
9.2.2	Requisito de usabilidad	34
9.2.3	Confiabilidad y rentabilidad	35
9.2.4	Restricciones de implementación	35
10	Diagrama de casos de usos	36
11	Diagrama de clases	40
12	Diagrama de actividades	41
13	Conclusión	44
14	Recomendaciones	45
15	Referencias bibliográficas	46
Anexo		49

Tabla de ilustraciones

Ilustración 1. Ciclo de trabajo en la metodología Scrum-----	22
Ilustración 2 Diagrama de casos de uso general -----	36
Ilustración 3 Diagrama de caso de uso Inicio de sesión-----	37
Ilustración 4 Diagrama de caso de uso gestionar servicios-----	37
Ilustración 5 Diagrama de caso de uso gestionar inmuebles-----	37
Ilustración 6 Diagrama de caso de uso gestionar inmuebles-----	38
Ilustración 7 Diagrama de caso de uso gestionar contratos -----	38
Ilustración 8 Diagrama de caso de uso gestionar contratos -----	38
Ilustración 9 Diagrama de caso de uso gestionar pagos -----	39
Ilustración 10 Diagrama de clase gestionar pagos-----	39
Ilustración 11 Diagrama de caso de uso gestión de finanzas -----	39
Ilustración 12 Diagrama de clases-----	40
Ilustración 13 Diagrama de actividades registrar inmueble -----	41
Ilustración 14 Diagrama actividad registrar pago-----	41
Ilustración 15 Diagrama de actividad registrar contrato -----	42
Ilustración 16 Diagrama de actividad iniciar seccion -----	42
Ilustración 17 Diagrama de actividad generar reporte -----	43
Ilustración 18 Diagrama de actividad administración inmueble-----	43
Ilustración 19 Identificación y Clasificación de Funciones del Sistema-----	49
Ilustración 20 Cálculos de punto de función -----	49
Ilustración 21 Gasto y tiempo de desarrollo-----	50
Ilustración 22 Interfaz de Loguin -----	52
Ilustración 23 Interfaz de Inicio -----	52

Ilustración 24 Interfaz de Contrato-----	53
Ilustración 25 Interfaz de Pagos-----	53
Ilustración 26 Interfaz de pago de deuda -----	54
Ilustración 27 Interfaz de catálogo de Inmueble -----	54
Ilustración 28 Reporte en formato PDF -----	55
Ilustración 29 Reporte en formato Excel -----	55

1 Introducción

El presente trabajo de monografía tiene como propósito desarrollar una aplicación web para la gestión y administración de servicios de alquiler de cuartos o habitaciones dirigida a estudiantes universitarios en la ciudad de León, utilizando el lenguaje de programación Python y el framework Flask. Esta propuesta surge ante la necesidad de modernizar los procesos de administración que se realizan de forma manual, generando desorganización, pérdida de información y dificultades en el control de contratos, pagos y registros de inquilinos.

El alquiler de habitaciones representa una fuente de ingresos para muchas familias leonesas y, al mismo tiempo, una necesidad básica para los estudiantes que se trasladan desde otros departamentos. Sin embargo, la falta de herramientas tecnológicas limita la eficiencia administrativa de los arrendadores y la transparencia en el manejo de los datos.

Con la implementación de esta aplicación web se busca automatizar los procesos administrativos relacionados con los alquileres, mejorando la precisión, agilidad y seguridad en la gestión de contratos, pagos y reportes. Además, se pretende fomentar el uso de soluciones tecnológicas accesibles que contribuyan al desarrollo organizativo y tecnológico de los arrendadores en la ciudad universitaria de León.

2 Planteamiento del problema

La movilidad de estudiantes desde sus lugares de origen hacia la ciudad universitaria de León genera una creciente demanda de alquiler de cuartos o habitaciones para residir durante el período académico. Esta necesidad es atendida, en su mayoría, por pequeños arrendadores o familias que, a modo de micro y pequeñas empresas, aprovechan la oportunidad de obtener ingresos estables a lo largo del ciclo escolar.

Sin embargo, tanto los arrendadores como los estudiantes enfrentan dificultades debido a la ausencia de un sistema automatizado que facilite la gestión de contratos, pagos y registros. En la actualidad, la mayoría de estos procesos se realizan de forma manual, utilizando cuadernos o documentos físicos, lo que ocasiona desorganización, pérdida de información y errores en los registros.

Esta situación provoca problemas frecuentes, como la falta de claridad en las fechas de inicio y finalización de los contratos, errores en los cobros mensuales y dificultad para mantener un control actualizado de los alquileres vigentes. Todo ello afecta la transparencia en la relación entre arrendador e inquilino y limita la eficiencia administrativa, evidenciando la necesidad de implementar una herramienta tecnológica que optimice dichos procesos y brinde un sistema más ágil, preciso y confiable.

¿Cómo mejorar la eficiencia y precisión en la gestión de alquileres de habitaciones para asegurar un adecuado registro de pagos y una administración confiable?

3 Justificación

Este proyecto es fundamental porque busca ofrecer una solución a la necesidad de muchos estudiantes universitarios de acceder a cuartos o habitaciones temporales cerca de sus centros de estudio en León. Muchos provienen de distintas localidades y necesitan un espacio seguro y cercano que facilite su movilidad y optimice su tiempo dedicado a actividades académicas. Contar con un lugar accesible durante este proceso contribuye a un ambiente adecuado que favorece su rendimiento académico y bienestar personal.

La propuesta de desarrollar una aplicación web que automatice la gestión de contratos, pagos y disponibilidad de cuartos permitirá mejorar la precisión, agilidad y transparencia en la administración de alquileres. Esta herramienta facilita el trabajo de los arrendadores y ofrece a los estudiantes una experiencia más organizada, confiable y segura.

Desde el enfoque teórico, el proyecto se sustenta en los principios de los sistemas de información y la gestión administrativa, aplicando la automatización como medio para reducir errores y optimizar los procesos de alquiler. Metodológicamente, se empleará la metodología ágil Scrum, garantizando un desarrollo iterativo, adaptable y centrado en las necesidades de los estudiantes universitarios. A nivel práctico y social, la aplicación favorecerá la organización económica de los arrendadores, impulsará el uso de herramientas tecnológicas en la comunidad y fortalecerá la confianza mutua entre arrendadores e inquilinos.

4 Antecedentes

4.1 Antecedentes nacionales

El servicio de alquiler de habitaciones es una fuente de ingresos para muchas familias en la ciudad de León, lo cual aporta recursos a la económica familiar. Los estudiantes universitarios, por la modalidad de estudios, algunos se ven obligados a solicitar servicio de alquilar para poder permanecer durante el ciclo académico. Según el plan de estudio que esté cursando, el porcentaje de estudiantes que alquilan habitaciones en la ciudad de León es significativo, lo que incrementa la demanda de servicios de habitación y la necesidad de una mejor organización administrativa para gestionar dichos procesos.

En investigaciones previas se ha trabajado en el desarrollo de aplicaciones web orientadas a la gestión de servicios con el fin de optimizar procesos internos y mejorar la administración. Tal es el caso de la propuesta de (Berrios y otros, 2016) , quienes diseñaron un sitio web y una aplicación móvil para la publicidad de bares, restaurantes y discotecas en la ciudad de León. Su objetivo era centralizar información, administrar registros y facilitar la interacción entre negocios y clientes, contribuyendo así a una mejor organización de los servicios ofrecidos.

Este antecedente resulta útil para el presente proyecto, ya que evidencia cómo las aplicaciones web pueden ser adaptadas a diferentes contextos para fortalecer la administración de servicios. En el caso de la gestión de alquileres para estudiantes universitarios en León, la propuesta se orienta específicamente a los arrendadores, quienes tendrán acceso exclusivo a la aplicación como una herramienta local que les permitirá controlar contratos, pagos y registros.

4.2 Antecedentes internacionales

El alquiler de habitaciones es una práctica común en distintos países, especialmente en zonas turísticas y ciudades universitarias donde la demanda de hospedaje es elevada. A nivel internacional, diversos estudios han abordado el desarrollo de aplicaciones web y sistemas automatizados para gestionar la reserva y administración de habitaciones, buscando mejorar la organización interna de los arrendadores y optimizar la experiencia de los inquilinos. Estos antecedentes permiten evidenciar la importancia de incorporar soluciones tecnológicas en la gestión de alquileres, lo cual resulta aplicable al contexto de León, donde las familias que ofrecen habitaciones a estudiantes universitarios enfrentan retos similares en la administración de contratos, pagos y registros.

Por ejemplo, (Méndez Reyes, 2020) desarrolló una aplicación web para la gestión de reservas en los hostales de Montañita, Ecuador, evidenciando la necesidad de herramientas tecnológicas que optimicen estos procesos. Su propuesta permitió a los clientes consultar en tiempo real la disponibilidad de habitaciones, mejorando la experiencia del usuario y reduciendo el trabajo manual.

Asimismo, (Martínez Quispe, 2020) diseñaron una aplicación web para el Hotel Thiana, ubicada en Perú, utilizando metodologías ágiles como Scrum para optimizar la gestión administrativa. Este enfoque permitió reducir errores y facilitar la gestión de reservas y habitaciones.

De igual forma, (Gallo, 2010) desarrolló un sistema digital orientado a sustituir los métodos manuales de reservas y planificación en el Hotel *El Tucano*, en Ecuador. Su propuesta demostró cómo la digitalización de procesos contribuye a mejorar la eficiencia, reducir errores y facilitar la organización de la información en el sector hotelero, constituyendo un antecedente importante para la gestión tecnológica de alquileres.

5 Objetivos

5.1 Objetivo general

- Desarrollar una aplicación web que automatice la gestión de alquileres de habitaciones, con módulos de pagos, contratos y reportes.

5.2 Objetivos específicos

- Gestionar la creación de contratos de alquiler de habitaciones, el registro preciso y la actualización oportuna de la información.
- Registrar los pagos y deudas de los inquilinos de forma exacta y en tiempo real.
- Elaborar reportes automáticos en formato PDF y Excel con información clara y estructurada sobre contratos y pagos realizados.

6 Marco teórico

6.1 Contratos de alquiler

Un contrato de alquiler es un acuerdo legal entre dos partes: el arrendador, quien posee el bien inmueble, y el arrendatario, quien lo utiliza temporalmente a cambio de una renta. Este contrato establece las condiciones bajo las cuales se cede el uso del bien, incluyendo duración, monto de la renta, derechos y obligaciones de ambas partes, y las consecuencias legales en caso de incumplimiento. En Nicaragua, la Ley de Inquilinato (Nicaragua, Ley de Inquilinato, Decreto Legislativo No. 92 MG., 1948) regula los arrendamientos de bienes inmuebles destinados a vivienda, estableciendo derechos y deberes para arrendadores e inquilinos.

6.2 Tipos de contratos de alquiler

Existen diversos tipos de contratos de alquiler, entre los cuales se destacan (Idealista/news, 2024):

- **Contrato de arrendamiento de vivienda habitual:** destinado a la residencia permanente del inquilino.
- Contrato de arrendamiento de temporada: para uso temporal, común en zonas turísticas o para estudiantes.
- Contrato con opción a compra: permite al inquilino adquirir la propiedad en el futuro.
- Contrato de habitación: se alquila una parte específica de una vivienda, como un cuarto individual.

6.3 Alquiler

El alquiler es un contrato mediante el cual una parte (arrendador) cede a otra (arrendatario) el uso temporal de un bien a cambio de una contraprestación económica. Este acuerdo es común en el ámbito inmobiliario y permite a los inquilinos acceder a viviendas sin necesidad de adquirirlas. (iAhorro, (s. f.))

6.4 Tipo de alquiler

Los tipos de alquiler se clasifican según la duración y el uso del bien:

- Alquiler temporal: de corta duración, ideal para estudiantes o turistas.
- Alquiler mensual: pago periódico mensual, común en arrendamientos residenciales.
- Alquiler anual: contrato de larga duración, generalmente con condiciones más estables.
- Alquiler por habitación: se alquila una parte específica de una vivienda, como un cuarto individual.

6.5 Periodo de alquiler

El periodo de alquiler es el tiempo durante el cual el arrendatario tiene derecho a utilizar el bien arrendado. Este periodo debe estar claramente especificado en el contrato, incluyendo la fecha de inicio y finalización, y puede renovarse según lo acordado por ambas partes.

6.6 Pago de alquiler

El pago de alquiler es la contraprestación económica que el inquilino debe abonar al arrendador por el uso del bien inmueble. Este pago puede ser mensual, semanal o según lo estipulado en el contrato. Es fundamental que las condiciones de pago estén claramente detalladas para evitar malentendidos. (Homming, 2023)

6.7 Formas de pago permitidas

Las formas de pago permitidas deben ser acordadas entre arrendador e inquilino y especificadas en el contrato. Las opciones comunes incluyen:

- Efectivo: pago directo en persona.
- Transferencia bancaria: depósito en cuenta del arrendador.
- Pago en línea: a través de plataformas digitales.
- Cheque: menos común, pero aún utilizado en algunos casos.

6.8 Tipos de habitaciones

Las habitaciones disponibles para alquiler pueden clasificarse según diversas características (Ciullini, 2023):

- Individuales: para una sola persona.
- Dobles: para dos personas.
- Amuebladas: incluyen muebles básicos como cama, escritorio y silla.
- Sin amueblar: sin mobiliario, permitiendo al inquilino personalizar el espacio.
- Con baño privado: incluyen un baño exclusivo para el inquilino.

6.9 Criterios de elección del estudiante

Los estudiantes al momento de elegir una habitación o alojamiento consideran múltiples criterios. Un estudio (Students, 2019) señala que los estudiantes valoran especialmente seguridad, limpieza, protección, libertad y comportamientos de convivencia.

Además, datos estadísticos muestran que factores como ubicación, calidad del alojamiento, precio (valor por dinero), estado de los servicios e instalaciones figuran entre los más influyentes. Por ejemplo, un informe indica que el 63 % de los estudiantes en Reino Unido (Imed Bouchrika, 2025) consideran que la calidad del alojamiento es “muy importante”, y otros criterios como ubicación, costo global, limpieza y servicios también tienen gran peso.

Entonces, algunos criterios clave son:

- Proximidad / ubicación (cercanía a universidad, transporte público)
- Costo total (renta + servicios incluidos)
- Seguridad (física, vigilancia, controles de acceso)
- Limpieza y mantenimiento
- Libertad y flexibilidad (horarios, visitas, normas)
- Privacidad / tipo de habitación (individual, compartida)
- Condiciones del entorno (ambiente, ruido, iluminación)
- Posibilidad de convivencia con amigos o compañeros compatibles.

6.10 Servicios valorados por estudiantes

Los servicios son un diferenciador importante en la decisión de los estudiantes. En el estudio “Student Housing: Trends, Preferences and Needs” (Claire Reeves La Roche, 2010) se señala que cosas como conectividad (internet / wifi), baños privados, espacios sociales / comunes, lavandería, aire acondicionado y mobiliario moderno ya no son lujos sino demandas esperadas.

El sitio “*Evolving Amenities at Purpose-Built Student Housing*” (Apartamentos, 2024) también menciona que los desarrollos modernos están ofreciendo espacios más flexibles que fomentan estudio, socialización, bienestar e incluso para mascotas.

Algunos servicios:

- Internet de alta velocidad / WiFi confiable
- Baños privados (o semi-privados)
- Mobiliario moderno (cama, escritorio, armario)
- Lavandería (lavadora / secadora)
- Áreas comunes de estudio / coworking
- Sala de socialización, salón de ocio
- Espacios verdes, patios, jardines
- Zonas de cocina compartida
- Seguridad (cámaras, control de acceso)
- Servicio de mantenimiento rápido

6.11 Factores sociales, psicológicos y de diseño

El alojamiento estudiantil no solo implica un espacio físico, sino también un entorno social y psicológico que influye en la convivencia, el bienestar y la integración. Aspectos como sentido de comunidad, privacidad, diseño flexible y apoyo al bienestar mental son fundamentales para la satisfacción del estudiante (Mendler, 2025).

6.12 Condiciones para recibir el pago

Las condiciones para recibir el pago deben estar claramente especificadas en el contrato e incluir:

- Fecha de vencimiento: día del mes en que se debe realizar el pago.
- Método de pago: forma en que se realizará el abono.
- Consecuencias por retraso: posibles cargos por pagos atrasados.
- Confirmación de recepción: comprobante que el arrendador debe proporcionar al inquilino.

De acuerdo con la Ley de Inquilinato, Ley N°. 118 (Nacional, 1991), el inquilino es responsable del pago de los servicios básicos, y el arrendador no podrá suspender dichos servicios en caso de incumplimiento, salvo que la normativa disponga lo contrario.

6.13 ¿Qué es reservación de habitación?

La reservación es un acuerdo, verbal o escrito, que se establece entre el arrendador y una persona física o jurídica, por lo cual el primero se compromete a realizar una prestación de alojamiento y demás servicios en una fecha determinada y por un periodo definido; y el segundo, a recibir tales prestaciones en esa fecha, con su consecuencia inmediata de pago. (Márquez, (s.f.))

6.14 Forma de pago de reservación

El pago de la reservación puede realizarse de diversas maneras, dependiendo de lo acordado entre las partes:

- Depósito en efectivo: pago directo en persona.
- Transferencia bancaria: depósito en cuenta del arrendador.
- Pago en línea: a través de plataformas digitales.

Es esencial que el método de pago y las condiciones estén claramente especificados en el contrato de reservación.

6.15 Periodo de pre-reservación

El periodo de pre-reservación es el tiempo durante el cual una habitación se mantiene reservada para un inquilino antes de la fecha de inicio del alquiler. Este periodo debe estar claramente definido en el contrato, incluyendo la duración máxima de la pre-reservación y las condiciones para su cancelación.

6.16 Cómo ubican la habitación o cuarto

La asignación de una habitación o cuarto puede realizarse mediante dos métodos principales:

1. Asignación interna: Se basa en criterios internos del inmueble o del sistema de gestión. Esto incluye la disponibilidad de habitaciones en el momento de la solicitud, las características de cada cuarto (como tamaño, equipamiento o ubicación dentro del inmueble) y las preferencias del inquilino. Este método garantiza una distribución organizada y controlada, permitiendo al arrendador mantener un registro claro de las habitaciones ocupadas y disponibles.
2. Búsqueda externa mediante redes o recomendaciones: Los estudiantes o inquilinos pueden ubicar habitaciones a través de redes sociales, grupos de comunicación, aplicaciones de alojamiento o recomendaciones de conocidos. Este método es especialmente útil para personas que llegan a una localidad por primera vez, ya que les permite conocer opciones confiables y adecuadas según la experiencia de otros. Las recomendaciones y redes facilitan la selección de un alojamiento cercano a universidades o centros de interés, aunque requieren verificación adicional para asegurar la disponibilidad y la legalidad del alquiler.

En conjunto, ambos métodos permiten a los inquilinos acceder a información sobre habitaciones de manera eficiente, mientras que los arrendadores pueden garantizar un proceso de asignación ordenado y confiable.

6.17 Fin del inquilinato

Según la Ley de Inquilinato de Nicaragua (Nicaragua, Ley de Inquilinato, Ley N.º 118, 1990) el inquilino puede finalizar el contrato de arrendamiento avisando por escrito al arrendador con al menos 30 días de anticipación, entregando la vivienda en la fecha acordada y haciéndose responsable de los daños o pagos pendientes de servicios públicos.

6.18 Tecnología para la administración del alquiler

Una aplicación para administración de alquileres permite centralizar y automatizar procesos como pagos en línea, gestión de mantenimiento, comunicación entre arrendador e inquilino, reservas de habitaciones y firmas de contratos digitales. Esto mejora la eficiencia, la transparencia y la experiencia del estudiante, facilitando el control de ocupación, recordatorios de pago, solicitudes de mantenimiento, seguridad mediante accesos inteligentes y analítica de datos para optimizar la gestión del inmueble (ManageCasa, 2024).

6.19 Automatización de procesos administrativos

La automatización de tareas como la emisión de facturas, el seguimiento de pagos, la renovación de contratos y la gestión de incidencias reduce la carga administrativa y mejora la eficiencia operativa (Housing.Cloud, 2025).

6.20 Herramientas para el desarrollo del sistema

6.20.1 Python

Python es un lenguaje de programación de alto nivel, interpretado y multiparadigma, ampliamente utilizado en desarrollo web, ciencia de datos y automatización de procesos. Su sintaxis clara y legible facilita el desarrollo de aplicaciones robustas y escalables, permitiendo integrar diversas bibliotecas y frameworks que simplifican la programación de funciones complejas (Foundation, 2024)

6.20.2 Flask

Flask es un microframework para Python diseñado para la creación de aplicaciones web ligeras y escalables. Proporciona herramientas básicas para gestionar rutas, solicitudes y respuestas, permitiendo al desarrollador construir aplicaciones web dinámicas con flexibilidad para integrar bases de datos, autenticación de usuarios y plantillas HTML (Grinberg, 2024).

6.20.3 Sqlite

SQLite es un sistema de gestión de bases de datos relacional, liviano y autónomo, ideal para aplicaciones que requieren almacenamiento local sin necesidad de un servidor complejo. Permite gestionar datos de manera eficiente y segura, siendo ampliamente utilizado en aplicaciones web y móviles que requieren portabilidad y rapidez en consultas (Hipp, 2024).

6.20.4 Db Browser For Sqlite

DB Browser for SQLite es una herramienta visual de código abierto diseñada para crear, diseñar y editar bases de datos compatibles con SQLite. Permite a los desarrolladores gestionar de manera gráfica las tablas, registros y consultas de la base de datos, facilitando la administración de la información sin necesidad de escribir comandos SQL complejos (SQLite, 2024).

6.20.5 HTML5 Y CSS3

HTML5 es la última versión del lenguaje de marcado estándar utilizado para estructurar el contenido en páginas web, incluyendo nuevas etiquetas y funcionalidades como APIs de geolocalización, almacenamiento local y soporte multimedia (W3C, HTML5, 2024). CSS3, por su parte, es el lenguaje de hojas de estilo que permite definir la presentación visual de los documentos HTML, controlando colores, tipografías, disposiciones y animaciones. Juntos, HTML5 y CSS3 permiten desarrollar interfaces web modernas, interactivas y accesibles, facilitando la creación de aplicaciones con diseños consistentes, responsivos y atractivos para el usuario (W3C, CSS3, 2024).

6.20.6 Javascript (Es6+)

JavaScript es un lenguaje de programación orientado a objetos y eventos que permite implementar funcionalidades dinámicas en páginas web. Con la versión ES6 y posteriores, ofrece características modernas como clases, módulos y funciones flecha, facilitando la creación de interfaces interactivas y aplicaciones web complejas (Network, 2024)

6.20.7 Visual Studio Code

Visual Studio Code es un editor de código fuente ligero y multiplataforma que soporta depuración, control de versiones y extensiones para diversos lenguajes y frameworks. Su integración con herramientas de desarrollo web lo hace ideal para gestionar proyectos completos de manera eficiente (Microsoft, 2024).

6.21 Sistemas de información y su importancia

Los sistemas de información son conjuntos organizados de recursos humanos, tecnológicos y de datos que permiten recopilar, procesar, almacenar y distribuir información para apoyar la toma de decisiones y el control dentro de una organización (Laudon, 2023). Estos sistemas integran componentes como el hardware, software, redes y bases de datos, los cuales trabajan de manera conjunta para transformar los datos en información útil.

6.22 Tecnologías y herramientas de desarrollo

Las tecnologías, lenguajes de programación, frameworks, bases de datos y demás herramientas utilizadas en el desarrollo del sistema son esenciales para garantizar una aplicación funcional, segura y eficiente. La elección adecuada de cada recurso tecnológico permite optimizar el proceso de implementación, asegurar la calidad del producto y facilitar su mantenimiento y escalabilidad.

6.23 Tecnologías del lado del servidor

Tecnología	Versión	Costo
Python	3.1.1	Gratis
Flask	2.3	Gratis
SQLite	3.41	Gratis
DB Browser for SQLite	3.12	Gratis
Visual Studio Code	1.91	Gratis

6.24 Tecnologías del lado del cliente

Tecnología	Versión	Costo
HTML5	5	Gratis
CSS3	3	Gratis
JavaScript (ES6+)	ES6	Gratis

6.25 Herramientas de desarrollo

Tecnología	versión	Costo
Visual Studio Code	1.91	Gratis
Git	2.37.1	Gratis
Navegadores Web (Chrome, Edge, Firefox)	Últimas versiones	Gratis
DB Browser for SQLite	3.12	Gratis

7 Metodología

7.1 Tipo de estudio, descriptivo de corte transversal

El presente proyecto se enmarca en una investigación empírica o práctica (Explorable.com, Sep 21, 2009), ya que se basa en la observación, análisis y aplicación directa de conocimientos teóricos en un contexto real. Además, se tomó como referencia metodológica el trabajo (Silva Bertha, 2024) quienes desarrollaron un sistema web aplicado a la gestión de ventas en comercios locales, sirviendo como un ejemplo práctico de investigación aplicada y de implementación tecnológica en un entorno real. Su propósito fue desarrollar una aplicación web funcional que responda a las necesidades detectadas en la gestión de alquileres de habitaciones para estudiantes universitarios en la ciudad de León.

Este tipo de investigación se caracteriza por la experimentación y la comprobación de resultados a través de la implementación de un producto tecnológico concreto.

Durante el proceso, se recolectaron datos reales de usuarios (arrendadores y estudiantes), se diseñaron soluciones informáticas y se evaluó su efectividad en la mejora de los procesos administrativos. De esta manera, la investigación no se limita a la teoría, sino que aplica el conocimiento en la práctica para validar su utilidad y eficiencia.

7.2 Área de estudio

El área de estudio se define como el Sistema de Información de Administración Local de Arrendadores de Habitaciones, Cuartos o Casas (SIALA), centrado en el contexto operativo interno de los arrendadores en la ciudad de León.

Este sistema no se orienta a la publicación pública en Internet, sino que abarca la estructura interna de datos y los procesos de control de negocio que el arrendador maneja para gestionar eficientemente sus propiedades y sus clientes.

7.2.1 Descripción del sistema de información

El foco de la investigación se sitúa en la transformación del proceso de gestión interna del arrendador. Actualmente, el SIAH se caracteriza por:

- **Administración Manual y Dispersa:** Los arrendadores suelen manejar la información de sus propiedades (inventario, tarifas, pagos, contratos, historial de inquilinos) mediante métodos manuales, como hojas de cálculo, cuadernos o documentos físicos.
- **Ineficiencia Operacional:** La falta de un sistema centralizado local genera problemas de trazabilidad, errores en el registro de pagos, dificultades en la identificación de vacantes y demoras en la emisión de reportes de ocupación.

7.3 Población de estudio

La Población de Estudio está constituida por la totalidad de los Arrendadores (Propietarios o Administradores de Inmuebles) que ofrecen habitaciones, cuartos o casas en alquiler a la comunidad universitaria dentro de la ciudad de León.

7.3.1 Definición y justificación focalizada en la gestión

La población se centra específicamente en los arrendadores, ya que son los usuarios directos y los principales beneficiarios del Sistema de Información de Administración Local de Arrendadores (SIALA). La elección se justifica por las siguientes razones:

1. Unidad de Gestión Administrativa: El sistema a desarrollar tiene como objetivo primario optimizar la gestión administrativa interna (control de pagos, contratos, inventario de cuartos). Por lo tanto, los arrendadores son quienes poseen el conocimiento experto de los procesos actuales y la experiencia directa de las ineficiencias que el sistema busca resolver.
2. Fuente de Datos Operacionales: Son la única fuente para obtener los datos primarios esenciales que requiere el sistema local: el inventario de habitaciones, las tarifas, los periodos de pago y los formatos de contratos.
3. Validación del Impacto: El éxito del proyecto se medirá por la mejora en la eficiencia de la gestión interna. Es esta población la que validará si la aplicación reduce la carga manual de trabajo y proporciona un control más riguroso de su negocio de arrendamiento.

7.3.2 Criterios de inclusión

Para delimitar la población que es relevante para el estudio y la posterior aplicación del muestreo, se establecen los siguientes criterios:

- Rol Administrativo: Arrendadores que actualmente gestionan la administración, contratos y cobranza de sus propiedades.
- Ubicación: Propietarios de inmuebles (habitaciones, cuartos o casas) ubicados en la ciudad de León.
- Segmento Objetivo: Aquellos que dirigen (o tienen potencial de dirigir) su oferta a estudiantes universitarios, ya que el diseño del sistema está especializado en las dinámicas de este segmento.

7.4 Procedimiento y fuentes de la obtención de la información

Para asegurar el desarrollo exitoso del Sistema de Información de Administración Local de Arrendadores (SIALA), se implementó un procedimiento riguroso de recolección de datos, combinando información de fuentes primarias y secundarias.

7.4.1 Procedimiento de recolección de datos

El proceso se llevó a cabo en la fase de análisis y diseño del proyecto, siguiendo el siguiente esquema:

- I. Identificación de la Muestra: Se contactó a los arrendadores seleccionados mediante el Muestreo No Probabilístico por Conveniencia (detallado en la sección 4) priorizando aquellos con procesos de gestión manual o dispersa.
- II. Preparación de Instrumentos: Se diseñó un guion de entrevista semi-estructurada específico para arrendadores.
- III. Aplicación de la Técnica: Se realizaron entrevistas a profundidad a la muestra seleccionada.

7.4.2 Descripción detallada de la entrevista semi-estructurada

La técnica principal de recolección de datos primarios fue la Entrevista Semi-estructurada, aplicada a los arrendadores seleccionados. Esta técnica permitió la recopilación de dos tipos de datos esenciales para el desarrollo del SIALA:

- **Datos Operacionales (Estructurados):** Necesarios para la construcción de la base de datos. Se recogieron detalles precisos sobre:
 - Inventario de inmuebles: Número de cuartos, características, servicios incluidos y tarifas.
 - Control Financiero: Métodos de pago aceptados, fechas de corte, y procedimientos para el registro de morosidad.
 - Contratos: Campos y cláusulas esenciales que deben ser digitalizados.
- **Datos de Procesos y Requisitos (Cualitativos):** Necesarios para el diseño de la lógica de negocio y la Interfaz de Usuario (UI) administrativa. Se documentó:
 - Flujo de Trabajo Actual: Pasos exactos que realiza el arrendador para gestionar un nuevo contrato, registrar un pago y dar de baja a un inquilino.
 - Puntos de Dolor: Identificación de las tareas administrativas más lentas o propensas a errores en el manejo manual.

Entrevistas a Inquilinos (Fuente Complementaria):

Adicionalmente, se realizaron entrevistas a un grupo de estudiantes inquilinos. Aunque el sistema es de administración interna, esta información se utilizó para validar los campos de datos de las propiedades y los registros (ej. qué datos son más relevantes para ellos), asegurando que el sistema capture información de calidad que, aunque administrada localmente, es relevante para el servicio final.

7.5 Metodología ágil Scrum

Scrum es una metodología ágil utilizada para la gestión y desarrollo de proyectos complejos, especialmente en el ámbito del software. El proceso se basa en iteraciones llamadas Sprints, que suelen durar entre 1 y 4 semanas. En cada Sprint, el equipo selecciona un conjunto de tareas del Product Backlog (lista priorizada de requisitos o funcionalidades del producto) y las traslada al Sprint Backlog, donde se planifican y ejecutan.

Durante el Sprint, se realizan reuniones diarias llamadas Daily Meetings o “Scrum Daily”, en las que el equipo revisa el progreso y resuelve obstáculos. Al finalizar el Sprint, se obtiene un Incremento de producto terminado, es decir, una versión funcional del producto que puede ser entregada o mejorada en el siguiente ciclo.

El objetivo principal de Scrum es fomentar la colaboración, la adaptación constante y la entrega continua de valor al cliente.

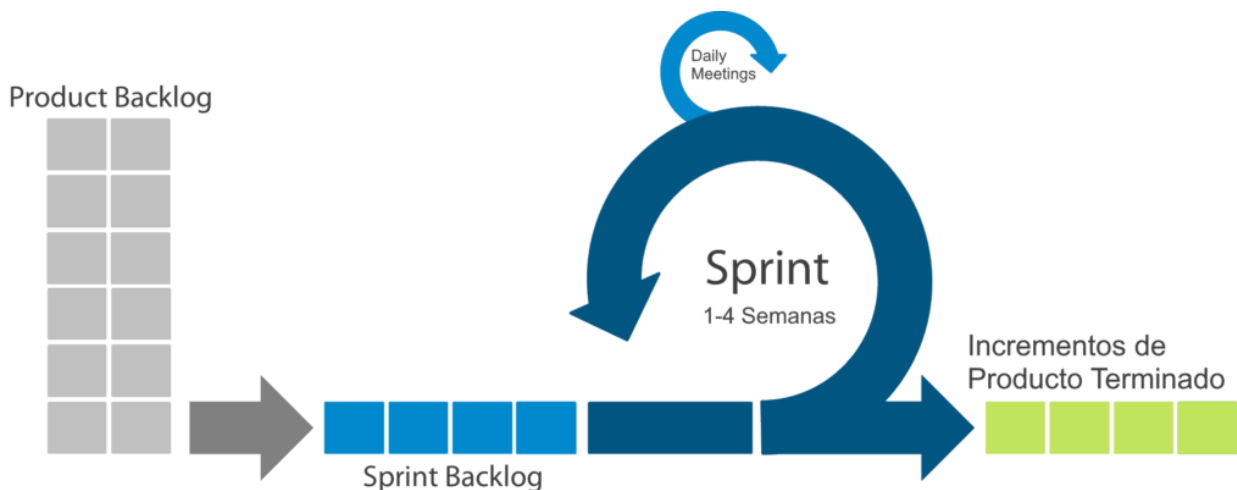


Ilustración 1. Ciclo de trabajo en la metodología Scrum

7.6 Personal Involucrado

El equipo de desarrollo estuvo conformado por tres miembros, cada uno con un rol claramente definido según el marco de Scrum:

Rol	Integrante	Responsabilidad principal
Scrum master	Br. Katerin Maribel Montes Pichardo	Facilitar el cumplimiento de la metodología Scrum, coordinar reuniones, promover la comunicación efectiva y eliminar impedimentos durante el desarrollo.
Product owner	Br. Nathaly Gissel Mendoza	Representar las necesidades de los usuarios, priorizar las funcionalidades en el Product Backlog y validar los incrementos del producto al finalizar cada Sprint.
Development team	Br. Mynor Miguel Arauz Rugama	Desarrollar los módulos de la aplicación, implementar las funcionalidades en Python, Flask y SQLite, realizar pruebas y documentar los avances técnicos.

7.7 Etapas del proceso Scrum aplicado al proyecto

7.7.1 Planificación inicial y creación del Product Backlog

En esta etapa se definió la visión general del sistema, los objetivos específicos y los requerimientos iniciales. La Product Owner elaboró el Product Backlog, que incluyó las funcionalidades priorizadas según las necesidades de los arrendadores, tales como: gestión de habitaciones, registro de contratos, control de pagos y generación de reportes. La Scrum Master organizó las primeras reuniones de planificación (Sprint Planning), estableciendo los tiempos de trabajo, las metas y las tareas asignadas a cada integrante.

7.7.2 Planificación del Sprint (Sprint Planning)

En esta fase, el equipo seleccionó las funcionalidades del Product Backlog que serían desarrolladas durante el primer Sprint. Se estableció la duración de dos semanas por iteración y se definieron los criterios de aceptación de cada funcionalidad. El equipo acordó la estructura del sistema, el diseño de la base de datos y el desarrollo del módulo de autenticación e inicio de sesión como punto de partida.

7.7.3 Ejecución del Sprint (Sprint Execution)

Durante cada Sprint, el equipo trabajó en el desarrollo de las funcionalidades priorizadas. Se realizaron reuniones diarias (Daily Scrum), dirigidas por la Scrum Master, donde se discutía el progreso, los retos encontrados y las tareas por ejecutar. En esta fase se construyeron los módulos principales del sistema, entre ellos: gestión de habitaciones e inmuebles, registro y administración de contratos, módulo de pagos y generación de reportes en PDF y Excel, y control de usuarios con copias de seguridad.

7.7.4 Revisión del Sprint (Sprint Review)

Al finalizar cada Sprint, el equipo presentó el incremento del producto al tutor y a usuarios de prueba (arrendadores). La Product Owner validó las funcionalidades implementadas y registró observaciones para su mejora en el siguiente ciclo. Este proceso permitió obtener retroalimentación constante, mejorar la interfaz del sistema y optimizar el rendimiento general de la aplicación.

7.7.5 Retrospectiva del Sprint (Sprint Retrospective)

Después de cada entrega, el equipo realizó una sesión de reflexión sobre el proceso de trabajo. Se identificaron fortalezas, como la buena comunicación y la eficiencia en el cumplimiento de plazos, así como áreas de mejora relacionadas con la documentación técnica y la integración de pruebas automatizadas. Estas retrospectivas fomentaron el aprendizaje continuo y la mejora de los métodos de colaboración del equipo.

7.7.6 Entrega e implementación del producto final

En la fase final, se integraron todas las funcionalidades validadas en los Sprints previos, obteniendo una versión completa y estable de la aplicación web. El sistema fue evaluado por su funcionamiento, usabilidad y confiabilidad, comprobando que cumplía con los objetivos propuestos y resolvía las problemáticas detectadas en la gestión manual de los alquileres. Finalmente, se documentó el impacto del sistema en la eficiencia administrativa, evidenciando una mejora significativa en el control y registro de los procesos de alquiler.

7.8 Conclusión metodológica

La aplicación del marco ágil Scrum permitió al equipo mantener una estructura flexible, dinámica y orientada a resultados. Gracias a la colaboración constante entre los roles y a la validación continua de los avances, se logró desarrollar una herramienta funcional, eficiente y adaptada al contexto de los arrendadores y estudiantes de la ciudad de León. Esta metodología demostró ser efectiva para proyectos académicos de desarrollo de software, al promover la organización, la comunicación y la entrega continua de valor.

8 Análisis

8.1 Especificación de requisitos de software (ERS)

8.1.1 Introducción

Esta sección establece el alcance, los requisitos funcionales y no funcionales de la Aplicación Web de Gestión y Administración de Servicios de Alquiler de Cuartos, adaptada para su uso local en los equipos de los pequeños emprendedores de León, Nicaragua.

8.1.2 Propósito

El propósito de este documento es establecer de manera clara y no ambigua los requisitos funcionales y no funcionales del sistema de gestión de alquileres, sirviendo como:

- Guía principal para el desarrollo del software en el lenguaje Python.
- Documento de validación que asegura que el producto final cumpla con las necesidades operativas de los pequeños emprendedores.

8.1.3 Alcance

El alcance del proyecto se limita al diseño e implementación de una Aplicación Web local utilizando Python y el framework Flask.

- El sistema será instalado y ejecutado en una máquina local (computadora de escritorio o laptop) del emprendedor, accediéndose a través del navegador web local (ej. <http://127.0.0.1:5000>).
- No requiere conexión a Internet para su operación diaria y almacenamiento de datos.
- El sistema no incluirá funcionalidades que dependan directamente de Internet (ej. envío automático de correos electrónicos a inquilinos).

8.1.4 Definiciones y Acrónimos

- **ERS:** Especificación de Requisitos de Software. Documento formal que define los requerimientos de la aplicación a desarrollar.
- **RF:** Requisito Funcional. Define una acción o tarea específica que el sistema debe realizar para el usuario (ej. Registrar un pago).
- **RNF:** Requisito No Funcional. Define una característica de calidad del software (ej. Usabilidad, Seguridad, Rendimiento).
- **App / Sistema Web Local:** La aplicación de gestión de alquileres desarrollada. Es un conjunto organizado de elementos que interactúan entre sí para lograr un objetivo, ejecutándose exclusivamente en la máquina del Administrador.
- **Python:** Lenguaje de programación de alto nivel utilizado para el desarrollo del *backend* de la aplicación.
- **Flask:** Micro-framework de Python, liviano y modular, seleccionado para construir la estructura de la aplicación web local.
- **SQLite:** Base de datos relacional sencilla y compacta utilizada para almacenar todos los datos del sistema en un único archivo local.
- **Administrador:** Usuario principal del sistema (el emprendedor) que tiene privilegios totales para gestionar habitaciones, contratos y pagos.
- **Inquilino / Estudiante:** El usuario final que arrienda el cuarto. Su información personal y estado de pago es gestionado por el sistema.
- **Contrato:** Registro formal dentro del sistema que asocia un Inquilino a una Habitación, estableciendo las condiciones de alquiler.
- **Control:** Supervisión o examen detallado para verificar o comprobar algo. En este caso, la verificación del estado de los pagos y la disponibilidad de cuartos.
- **UNAN-LEÓN:** Universidad Nacional Autónoma de Nicaragua, León.

8.1.5 Perspectiva del producto

El sistema de Aplicación Web de Gestión y Administración de Servicios de Alquiler de Cuartos es una herramienta informática de naturaleza stand-alone (autónoma) diseñada para reemplazar el sistema manual tradicional de control de arriendos.

8.1.6 Funcionalidad del producto

La funcionalidad del producto se define por las capacidades operativas de alto nivel que el sistema proporciona al Administrador (emprendedor) para automatizar y optimizar la gestión tradicional de alquileres de cuartos.

9 Requisitos específicos

9.1 Requisitos funcionales

Los Requisitos Funcionales (RF) describen las acciones concretas que el sistema debe poder ejecutar para cumplir con los objetivos del proyecto. Se presentan con codificación para su trazabilidad:

Requisito funcional 1	
Numero de requisito	RF-1
Nombre del requisito	Acceso al sistema
Descripción	Los usuarios en la pantalla de login deberán introducir sus credenciales para poder acceder al sistema
Prioridad del requisito	Alta/Esencial

Requisito funcional 2	
Numero de requisito	RF-2
Nombre del requisito	Registrar al usuario
Descripción	El administrador podrá registrar nuevos usuarios en el sistema
Prioridad del requisito	Alta/Esencial

Requisito funcional 3	
Numero de requisito	RF-3
Nombre del requisito	Registrar cuartos
Descripción	El administrador podrá registrar nuevos cuartos en el sistema
Prioridad del requisito	Alta/Esencial

Requisito funcional 4	
Numero de requisito	RF-4
Nombre del requisito	Editar cuartos
Descripción	El administrador podrá editar la información (precio, descripción, imagen, servicios) de cualquier cuarto que no esté ocupado.
Prioridad del requisito	Alta/Esencial

Requisito funcional 5	
Numero de requisito	RF-5
Nombre del requisito	Consultar estados
Descripción	Mostrar el estado actual de los cuartos disponibles ya sea disponible o ocupado
Prioridad del requisito	Alta/Esencial

Requisito funcional 6	
Numero de requisito	RF-6
Nombre del requisito	Actualizar estados
Descripción	Actualizar automáticamente el estado de un cuarto a 'Ocupado' al crearse un contrato y a 'Disponible' al finalizarse dicho contrato.
Prioridad del requisito	Alta/Esencial

Requisito funcional 7	
Numero de requisito	RF-7
Nombre del requisito	Registro de información
Descripción	Registrar la información personal y de contacto del inquilino (nombre, ID, teléfono, Documento de identidad).
Prioridad del requisito	Alta/Esencial

Requisito funcional 8	
Numero de requisito	RF-8
Nombre del requisito	Historial de contrato
Descripción	Consultar el historial de arriendo completo de un Inquilino (contratos activos).
Prioridad del requisito	Alta/Esencial

Requisito funcional 9	
Numero de requisito	RF-9
Nombre del requisito	Búsqueda de inquilino
Descripción	Buscar inquilinos utilizando filtros como su nombre o su ID y así obtener información en tiempo real
Prioridad del requisito	Alta/Esencial

Requisito funcional 10	
Numero de requisito	RF-10
Nombre del requisito	Crear contrato
Descripción	Crear un nuevo contrato asociando un Inquilino con un cuarto Disponible, registrando la fecha de inicio y el monto de la fianza.
Prioridad del requisito	Alta/Esencial

Requisito funcional 11	
Numero de requisito	RF-11
Nombre del requisito	Finalizar contrato
Descripción	Finalizar un contrato activo, solicitando la fecha de término y liberando el cuarto asociado
Prioridad del requisito	Alta/Esencial

Requisito funcional 12	
Numero de requisito	RF-12
Nombre del requisito	Resumir contratos
Descripción	Mostrar un resumen en tiempo real del estado actual del contrato (días restantes, fecha límite de pago)
Prioridad del requisito	Alta/Esencial

Requisito funcional 13	
Numero de requisito	RF-13
Nombre del requisito	Registrar pago
Descripción	Registrar un pago parcial o total a un contrato activo, indicando la fecha y el monto recibido.
Prioridad del requisito	Alta/Esencial

Requisito funcional 14	
Numero de requisito	RF-14
Nombre del requisito	Calcular saldo
Descripción	Calcular automáticamente el saldo pendiente o el excedente después de cada registro de pago.
Prioridad del requisito	Alta/Esencial

Requisito funcional 15	
Numero de requisito	RF-15
Nombre del requisito	Historial de pagos
Descripción	Generar un historial de pagos detallado por cada contrato.
Prioridad del requisito	Alta/Esencial

Requisito funcional 16	
Numero de requisito	RF-16
Nombre del requisito	Costos
Descripción	Mostrar los costos operativos de cada inmueble registrado (Ingresos, costos y ganancias).
Prioridad del requisito	Alta/Esencial

Requisito funcional 17	
Numero de requisito	RF-17
Nombre del requisito	Respaldo
Descripción	Función para generar un Respaldo (backup) de la base de datos local (SQLite) a un archivo en el disco duro.
Prioridad del requisito	Alta/Esencial

9.2 Requisitos no funcionales (RNF)

Los RNF aseguran que el sistema sea usable, seguro, eficiente y que cumpla con las restricciones tecnológicas definidas por el proyecto (Python/Flask/SQLite).

9.2.1 Requisitos de rendimiento

El tiempo de respuesta para cualquier consulta o generación de reporte no debe exceder los 2 segundos. Este requisito es crucial dado que el sistema se ejecuta localmente, el tiempo de carga de cualquiera de las interfaces una vez se haya registrado no será inferior a un segundo y se podrá operar continuamente durante un mínimo de 8 horas sin presentar fallos o bloqueos inesperados.

9.2.2 Requisito de usabilidad

Las interfaces presentan un diseño limpio y simple usando una terminología clara asegurándose que el administrador pueda usar el sistema sin complicaciones. Los colores, la tipografía, ubicación de menús y los botones son uniforme en todas las ventanas de la aplicación web. Se muestran mensajes de errores claros y amigables que guían al usuario a corregir entradas invalidas.

9.2.3 Confiabilidad y rentabilidad

El sistema asegura la integridad de los datos financieros (pagos y saldos) mediante restricciones a la base de datos para evitar incoherencias. Además la función del backup o respaldo de la base de datos se hace sin interrumpir la operación normal del sistema.

9.2.4 Restricciones de implementación

El desarrollo del backend del sistema debe hacerse exclusivamente en Python (versión 3.x), el framework de desarrollo debe de ser flask y se deben utilizar sus librerías recomendadas para la interacción con la base de datos, esta misma debe de ser SQLite debido a su facilidad de instalación y manejo de entorno local.

El Administrador solo podrá interactuar con la base de datos a través de la interfaz de la aplicación web; el acceso directo al archivo **.sqlite** se evitará para proteger la integridad.

10 Diagrama de casos de usos

Se muestran las funciones generales a las cuales tiene acceso el usuario administrador y el usuario común del sistema.

Casos de uso:

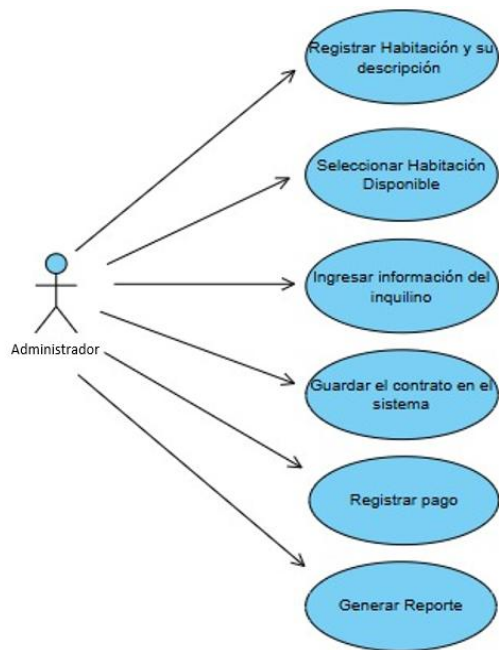


Ilustración 2 Diagrama de casos de uso general

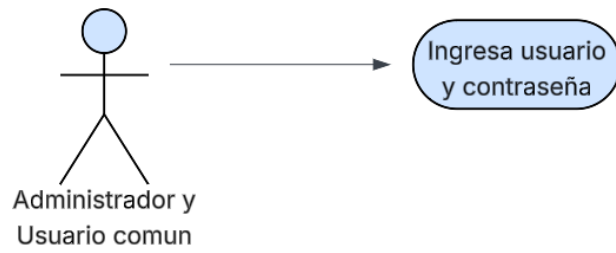


Ilustración 3 Diagrama de caso de uso Inicio de sesión

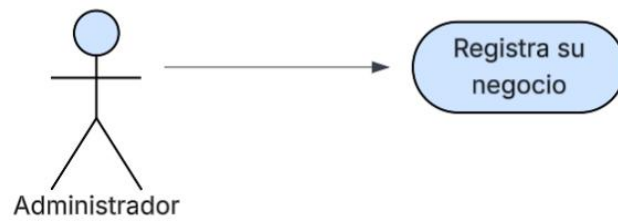


Ilustración 4 Diagrama de caso de uso gestionar servicios

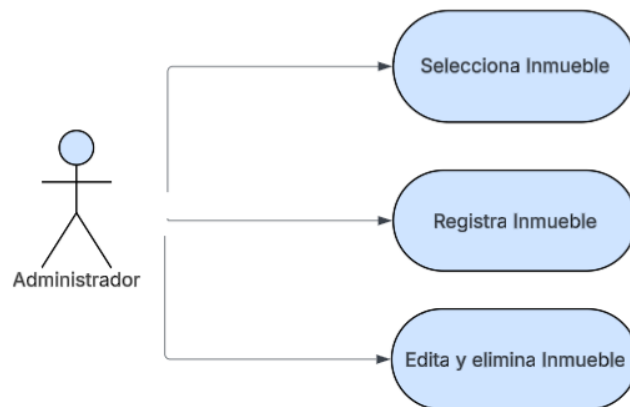


Ilustración 5 Diagrama de caso de uso gestionar inmuebles

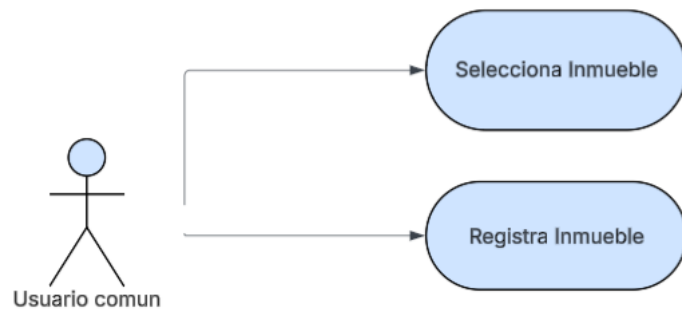


Ilustración 6 Diagrama de caso de uso gestionar inmuebles

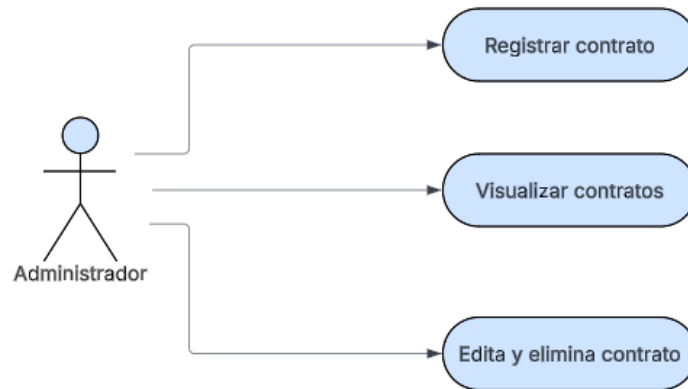


Ilustración 7 Diagrama de caso de uso gestionar contratos

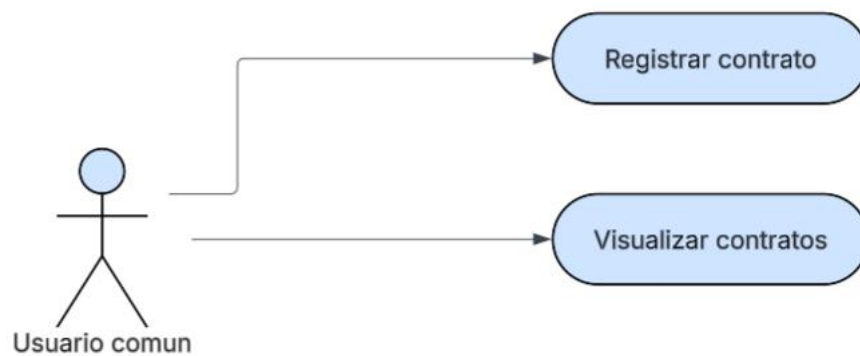


Ilustración 8 Diagrama de caso de uso gestionar contratos

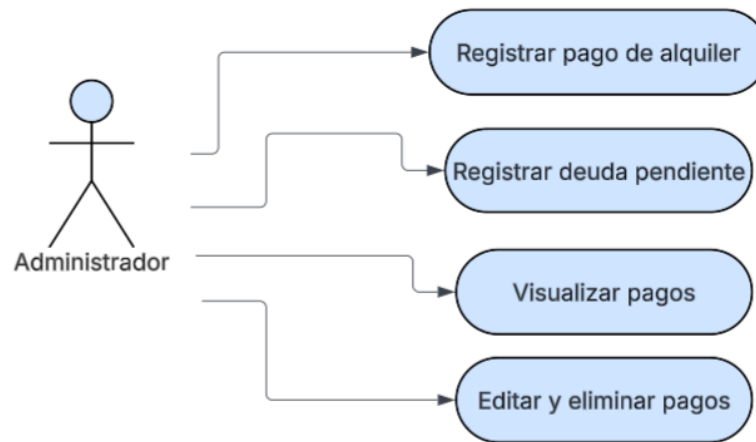


Ilustración 9 Diagrama de caso de uso gestionar pagos

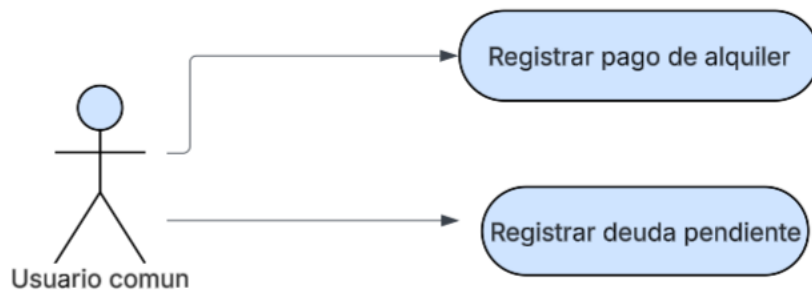


Ilustración 10 Diagrama de clase gestionar pagos

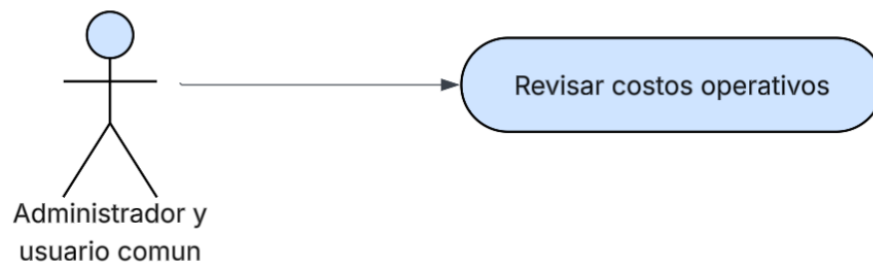


Ilustración 11 Diagrama de caso de uso gestión de finanzas

11 Diagrama de clases

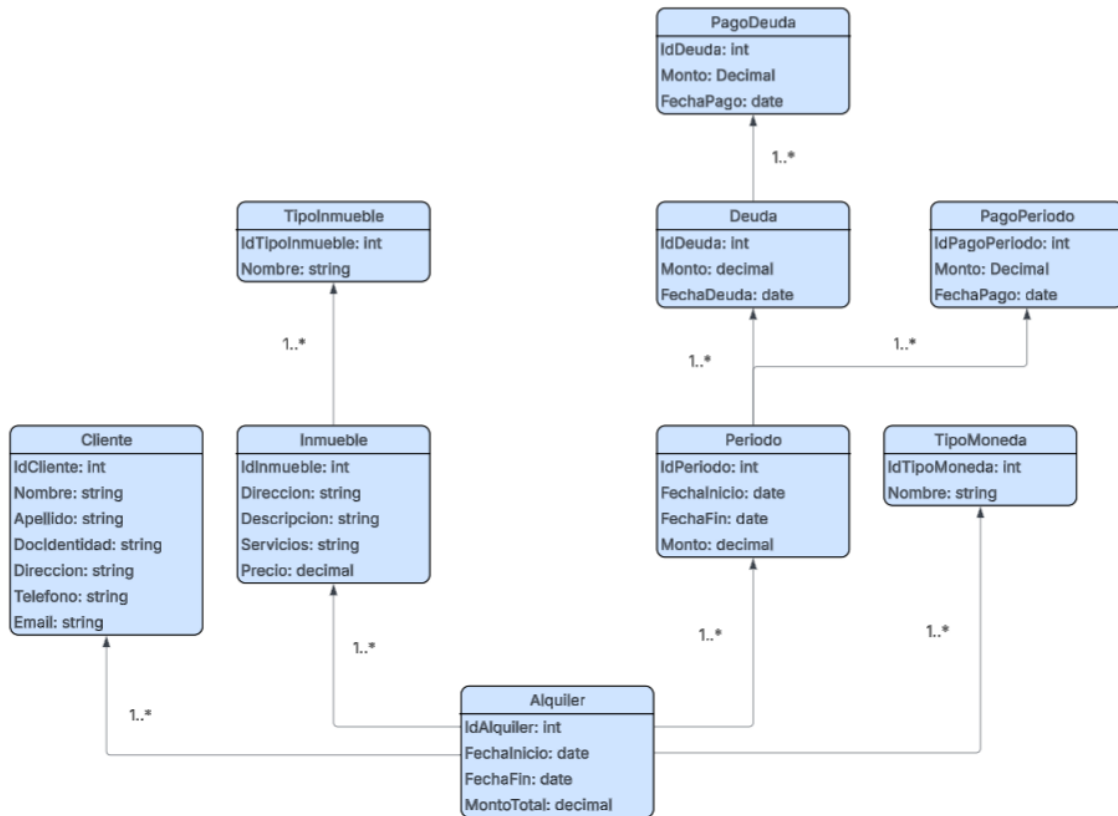


Ilustración 12 Diagrama de clases

12 Diagrama de actividades

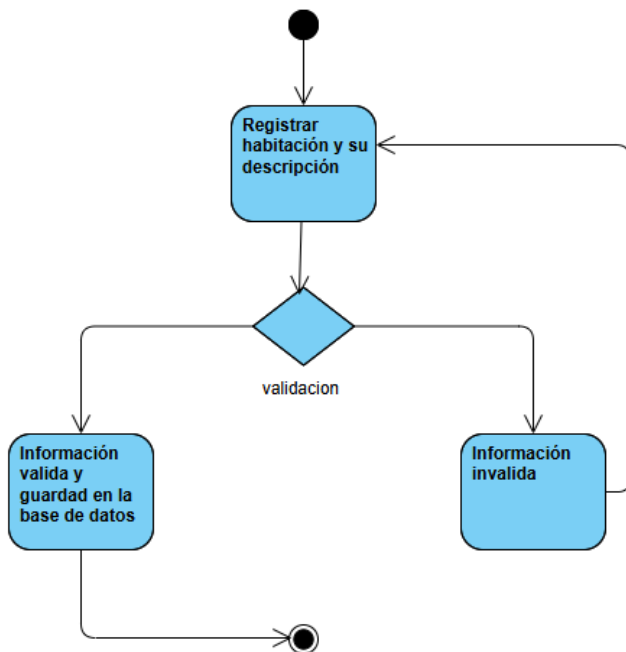


Ilustración 13 Diagrama de actividades registrar inmueble

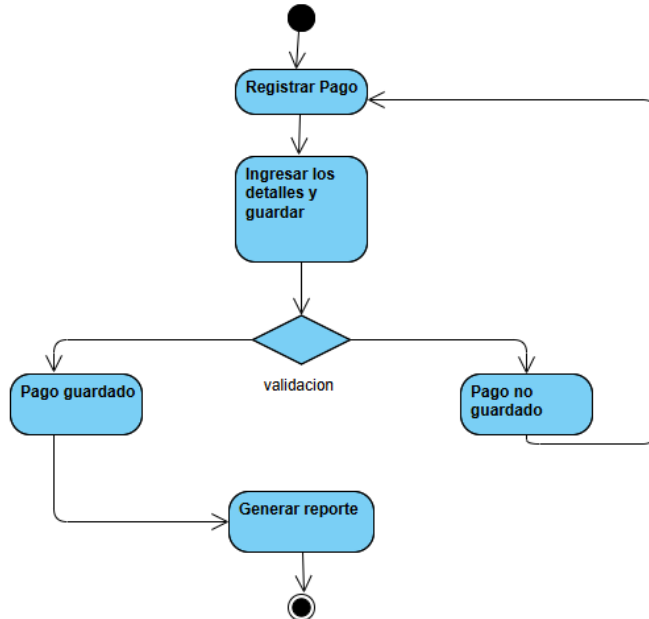


Ilustración 14 Diagrama actividad registrar pago

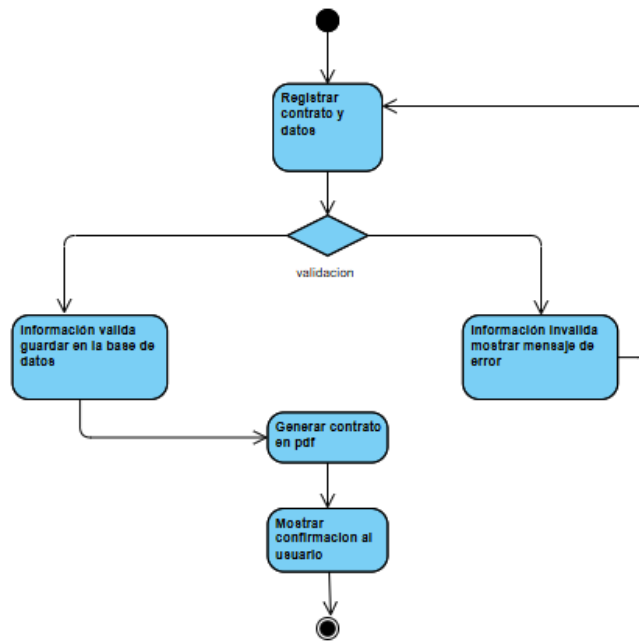


Ilustración 15 Diagrama de actividad registrar contrato

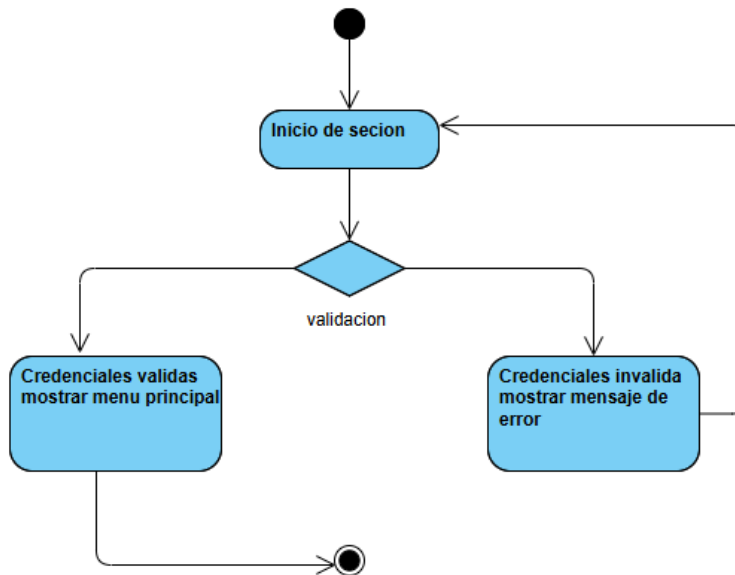


Ilustración 16 Diagrama de actividad iniciar sesion

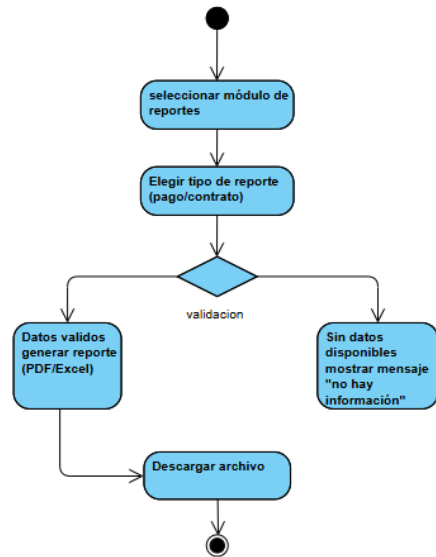


Ilustración 17 Diagrama de actividad generar reporte

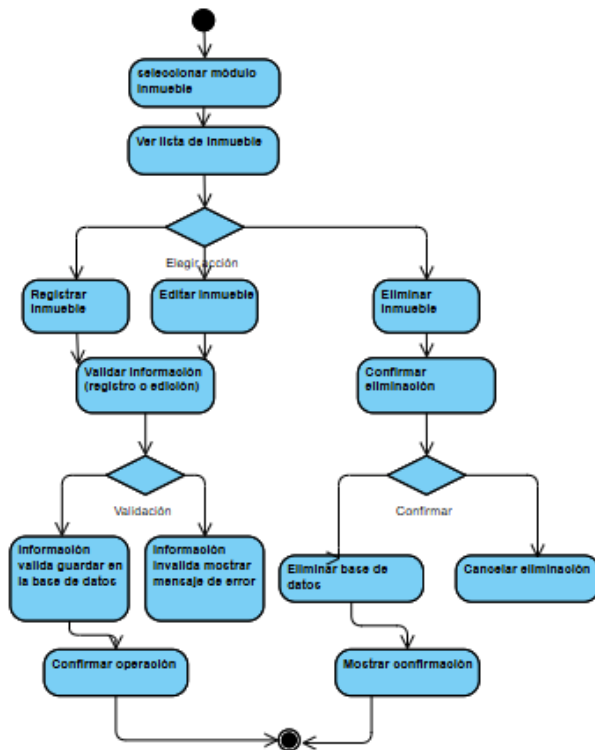


Ilustración 18 Diagrama de actividad administración inmueble

13 Conclusión

El desarrollo de la aplicación web para la gestión y administración de servicios de alquiler de cuartos o habitaciones dirigida a estudiantes universitarios de la ciudad de León permitió demostrar cómo el uso de herramientas tecnológicas, específicamente el lenguaje de programación Python con el framework Flask, contribuye a optimizar los procesos administrativos de los arrendadores.

La investigación evidenció que la mayoría de los arrendadores realizan la gestión de sus contratos, pagos y registros de forma manual, lo que ocasiona desorganización, pérdida de información y errores frecuentes. La aplicación desarrollada ofrece una solución práctica y eficiente a esta problemática, al automatizar las tareas más críticas de la administración de alquileres, garantizando precisión, seguridad y control en el manejo de datos.

Asimismo, la implementación del sistema permitió comprobar que la digitalización de procesos favorece la transparencia en las relaciones entre arrendadores e inquilinos, reduce los tiempos de gestión y mejora la confiabilidad de la información. De esta manera, el proyecto no solo aporta beneficios tecnológicos, sino también sociales y económicos, al fortalecer la organización de los arrendadores y brindar a los estudiantes un servicio más eficiente y moderno.

En conclusión, el proyecto cumplió con los objetivos planteados al desarrollar una herramienta funcional, accesible y adaptable a las necesidades del contexto local, contribuyendo al avance del uso de tecnologías de la información en la ciudad universitaria de León y sirviendo como base para futuras mejoras e innovaciones en la gestión de alquileres.

14 Recomendaciones

Ampliar las funcionalidades del sistema: incorporar módulos adicionales como notificaciones automáticas de pago, sistema de mensajería entre arrendador e inquilino, y control de mantenimiento de habitaciones.

Implementar medidas de seguridad avanzadas: integrar autenticación de usuarios mediante encriptación y protocolos HTTPS para proteger la información sensible de los usuarios.

Capacitar a los arrendadores en el uso del sistema: realizar talleres o guías prácticas que faciliten la adopción de la herramienta tecnológica y promuevan su uso efectivo.

Migrar a una base de datos más robusta: en futuras versiones, considerar el uso de motores como MySQL o PostgreSQL para gestionar mayor volumen de datos y usuarios concurrentes.

Escalar la aplicación a nivel regional o nacional: adaptar el sistema para que pueda ser utilizado en otras ciudades universitarias de Nicaragua, contribuyendo a la modernización del sector de alquiler estudiantil.

Mantener la actualización tecnológica: incorporar periódicamente mejoras en diseño, rendimiento y compatibilidad con nuevas versiones de Python, Flask y navegadores web.

15 Referencias bibliográficas

- Apartamentos, A. N. (2024). *Evolución de las comodidades en las comunidades de viviendas para estudiantes especialmente diseñadas*.
- Berrios , J., Cerda , G., & Altamirano, E. (2016). Sitio Web y Aplicación Móvil publicitario para los Bares, Restaurantes y discotecas de la ciudad de León. *UNAN-León*.
- Ciullini, G. (2023). *Tipos de habitaciones en hotel: ¿cómo se clasifican los tipos de habitaciones?* Octorate.
- Claire Reeves La Roche, L. U. (2010). “*Student Housing: Trends, Preferences and Needs*”.
- Creswell, J. W. (2018). *Research design: Qualitative, quantitative, and mixed methods approaches (5th ed.)*. SAGE Publications.
- Explorable.com. (Sep 21, 2009). *Investigación Empírica*. Nov 13, 2025 Obtenido de *Explorable.com*. Explorable.com .
- Foundation, P. S. (2024). *Python 3.11.4 documentation*.
- Gallo, S. (2010). *Diseño de un sistema digital para optimizar los métodos manuales de reservas y planificación de habitaciones en el hotel El Tucano*. Repositorio Institucional UPSE.
- Grinberg, M. (2024). *The Flask Mega-Tutorial, Part I: Hello, World!* .
- Hevner, A. R. (2021). *Design science in information systems research: Theory and practice*. . Springer. .
- Hipp, D. R. (2024). *SQLite Home Page*.
- Homming. (2023). *Pago del alquiler, ¿cuándo y cómo hacerlo?* . Homming.
- Housing.Cloud. (2025). *5 ventajas de usar un software moderno de alojamiento universitario*.

- iAhorro. ((s. f.)). *¿Qué es el alquiler y cómo funciona?* iAhorro.
- Idealista/news. (2024). *Tipos de contratos de alquiler ¿Cuáles son y en que se diferencian?* idealista .
- Imed Bouchrika, P. (2025). *82 Estadísticas de alojamiento para estudiantes: Datos, perspectivas y predicciones de 2025*. Research.com.
- In, F. A. (2019). *Investigating important factors affecting students' decision making process accommodation*. International Journal of Engineering and Advanced Technology (IJEAT) .
- Laudon, K. C. (2023). *Sistemas de información gerencial (17.^a ed.)*. Pearson Educación.
- ManageCasa, S. &. (2024). *Detener el seguimiento manual: actualice al software de administración de propiedades para viviendas para estudiantes*.
- Márquez, L. ((s.f.)). *Definición de reservación*. Scribd.
- Martínez Quispe, J. A. (2020). *Diseño de una aplicación web para la gestion de reservas en el hotel Thiana*. UTP.
- Méndez Reyes, W. G. (2020). *Aplicación web para la gestión de reservas de habitaciones, en los hostales de la comuna Montañita*. UPSE.
- Mendler, M. &. (2025). *Las empresas inmobiliarias privadas pueden aprobar el examen de vivienda para estudiantes de EE. UU*.
- Microsoft. (2024). *Visual Studio Code*.
- Nacional, A. (1991). *Ley de Inquilinato, Ley N°. 118 (aprobada el 6 de diciembre de 1990)*. La Gaceta, Diario Oficial, N°. 11.
- Network, M. D. (2024). *JavaScripts*.

Nicaragua, A. N. (1948). *Ley de Inquilinato, Decreto Legislativo No. 92 MG*. La Gaceta No. 192.

Nicaragua, A. N. (1990). *Ley de Inquilinato, Ley N.º 118*. La Gaceta, Diario Oficial N.º 11.

Pressman, R. S. (2020). *Ingeniería del Software: Un Enfoque Práctico (9na ed.)*. McGraw-Hill Education.

Schwaber, K. &. (2020). *The Scrum Guide: The definitive guide to Scrum: The rules of the game*. Scrum.org.

Silva Bertha, G. F. (2024). *DESARROLLO DE UN SISTEMA WEB EN NEXT.JS Y NODE.JS CON INTELIGENCIA* . Repositorio unan león .

Silva, B. G. (2024). *Desarrollo de un sistema web en NEXT.JS y node.js con inteligencia artificial para gestionar ventas en comercios locales de Nicaragua*. Repositorio unan León.

SpaceBasic. (2024). *Detener el seguimiento manual: actualice al software de administración de propiedades para viviendas para estudiantes*.

SQLite, D. B. (2024). *DB Browser for SQLite*.

Students, F. A. (2019). *Investigating important factors affecting students' decision making process accommodation*. International Journal of Engineering and Advanced Technology (IJEAT).

W3C. (2024). *CSS3*.

W3C. (2024). *HTML5*.

Anexo

Estimación de Esfuerzo y Costos para el Proyecto de Alquiler de Habitaciones

Función	Tipo	Complejidad	Puntos
Almacenamiento de habitaciones, contratos y pagos	Archivos lógicos internos (ALI)	Alta	15
Iniciar sesión	Entrada externa (EE)	Baja	3
Registrar contratos de alquiler	Entrada externa (EE)	Alta	6
Registrar pagos	Entrada externa (EE)	Alta	6
Generación de reportes financieros en PDF	Salida externa (SE)	Alta	7
Visualización del historial de contratos	Salida externa (SE)	Alta	7
Búsqueda de habitaciones disponibles	Consulta externa (CE)	Media	4
Consulta de pagos pendientes	Consulta externa (CE)	Media	4
Vista personalizada según rol	Consulta externa (CE)	Media	4

Ilustración 19 Identificación y Clasificación de Funciones del Sistema

Cálculos de punto de función

Tipo de Función	Baja	Media	Alta	Total Puntos
Archivos lógicos internos (ALIs)	0	0	1 * 15	15
Entradas externas (EEs)	1 * 3	0	2 * 6	15
Salidas externas (SEs)	0	0	2 * 7	14
Consultas externas (CEs)	0	3 * 4	0	12
Total Puntos de Función No Ajustados				56

Ilustración 20 Cálculos de punto de función

Cálculo del Factor de Ajuste

$$\text{Factor de Ajuste} = (40 \times 0.01) + 0.65 = 1.05$$

Cálculo de los Puntos de Función Ajustados

$$\text{Puntos de Función Ajustados} = 56 \times 1.05 = 58.8 \approx 59$$

Cálculo del Esfuerzo en Horas Hombre

Promedio: 8 horas por punto de función $59 \times 8 = 472$ horas

Tiempo total para 1 desarrollador:

$472 \div 40$ horas / semana = 11.8 semanas $\approx$ 12semanas

(Aunque el esfuerzo técnico equivale a aproximadamente 12 semanas de trabajo efectivo, la duración real del proyecto se extendió a 6 meses (26 semanas) debido a las actividades adicionales de documentación, pruebas, correcciones y tiempos administrativos propios del ciclo de desarrollo.)

Costo de desarrollo (6 meses)

Suponiendo un salario mensual de C\$18,000 por desarrollador

Tiempo de desarrollo: 6 meses

$6 \text{ meses} \times \text{C\$}18,000 = \text{C\$}324,000$

Rubro	Detalle	Costo mensual (C\$)	Total (6 meses) C\$
Salario programadores	3 personas $\times$ C\$18,000 por mes	C\$ 54,000.00	C\$ 324,000.00
Licencia de software	IDE, extensiones, editores, herramientas específicas	C\$ 500.00	C\$ 3,000.00
Energía eléctrica	Consumo estimado por computadoras y equipos	C\$ 750.00	C\$ 4,500.00
Internet	Servicio mensual necesario para desarrollo	C\$ 1,200.00	C\$ 7,200.00
Mantenimiento de equipos	Limpieza, soporte, piezas menores	C\$ 950.00	C\$ 5,700.00
Comida y bebida	Alimentación del equipo durante jornadas	C\$ 2,500.00	C\$ 15,000.00
Seguridad informática	Antivirus, firewall, respaldo	C\$ 720.00	C\$ 4,320.00
Transporte	Movilización para reuniones	C\$ 300.00	C\$ 1,800.00
Total			C\$ 365,520.00

Ilustración 21 Gasto y tiempo de desarrollo

Valor comercial de la aplicación

Cálculo	Resultado
<i>PF No Ajustados</i>	56
<i>Factor de Ajuste</i>	1.05
<i>PF Ajustados (PFA)</i>	59

Costo por punto de función

US\$ 100 por punto de función (estándar académico y empresarial a nivel centroamericano)

Calculo final de la aplicación

<i>Dato</i>	<i>Valor</i>
<i>Puntos de función</i>	59
<i>Precio por punto</i>	U\$100
<i>Precio Total</i>	59*100 = US\$5,900

Precio sugerido por venta = US\$ 5,000 Y US\$ 6,000

Equivalente en córdobas (C\$) =

$5,900 * 36.5 \approx \text{C}\$215,350$

Interfaz de la Aplicación



The login interface is divided into two main sections. On the left, a blue vertical panel contains a red square with a black circle and a white dollar sign, and the text 'Sistema de alquiler' below it. On the right, a white panel contains the following elements: a label 'Número Documento:' followed by a light blue input field containing '0000'; a label 'Contraseña:' followed by a light blue input field with three dots; two buttons, 'Ingresar' (blue) and 'Salir' (red); and a footer line 'Todos los derechos reservados ©'.

Ilustración 22 Interfaz de Login



Ilustración 23 Interfaz de Inicio

Sistema de alquiler

Usuario: Administrador

Fecha: 29/09/2025

Salir

Inicio

Perfil

Servicios

Configuración

Alquiler

Pagos

Finanzas

Respaldo

Acerca de

CONTRATO DE ALQUILER

DATOS DEL CONTRATO

Código Contrato: 00003

Fecha de Registro: dd/mm/aaaa

DATOS DEL ARRENDADOR

Nombre Completo:

Teléfono:

Correo Electrónico:

DATOS DEL ARRENDATARIO

Nombre Completo:

Tipo Documento: -- Seleccione --

Número Documento:

Teléfono:

Correo Electrónico:

Nacionalidad:

INFORMACIÓN DEL INMUEBLE

Código Inmueble:

Tipo Inmueble:

Dirección del Inmueble:

Precio de Alquiler por día:

Tipo Moneda:

Ubicación Geográfica:

SERVICIOS INCLUIDOS

Cama:

Closet:

Mesa:

Abanico:

Sillón:

Lavadora:

Microondas:

Agua:

Luz:

Internet:

TV:

Cable:

Refrigerador:

CONDICIONES DEL ALQUILER

Fecha Inicio: dd/mm/aaaa

Días de estadía: 0

Fecha Fin: dd/mm/aaaa

Total a pagar:

DEPÓSITO DE GARANTÍA

Monto del Depósito:

Moneda:

Ilustración 24 Interfaz de Contrato

Sistema de alquiler

Usuario: Administrador

Fecha: 29/09/2025

Salir

Inicio

Perfil

Servicios

Configuración

Alquiler

Pagos

Finanzas

Respaldo

Acerca de

PAGO DE ALQUILER

Buscar por Código o por Nombre: 00001

Buscar

INFORMACIÓN DEL INQUILINO

Nombre Completo: Mynor Arauz

Tipo Documento: Cédula

Número Documento: 281-260202-2005M

Teléfono: 57135174

Correo Electrónico: Miguel@gmail.com

Nacionalidad: Nicaraguense

INFORMACIÓN DEL INMUEBLE

Código Inmueble: 00001

Tipo Inmueble:

Precio del inmueble por día: 3600.0

Tipo Moneda: Cordobas

INFORMACIÓN DE PAGO

Fecha límite del pago: 29/08/2025

Días de estadía: 2

Total a pagar: 7200.0

Deuda pendiente: 7000.0

Monto a pagar ahora:

Método de Pago: Efectivo

Registrar Pago

Pagos Registrados

Contrato	Periodo	Fecha Limite	Monto	Estado	Acción
00001	1	2025-08-29	200.0	Pagado	

Ilustración 25 Interfaz de Pagos

53



Ilustración 26 Interfaz de pago de deuda

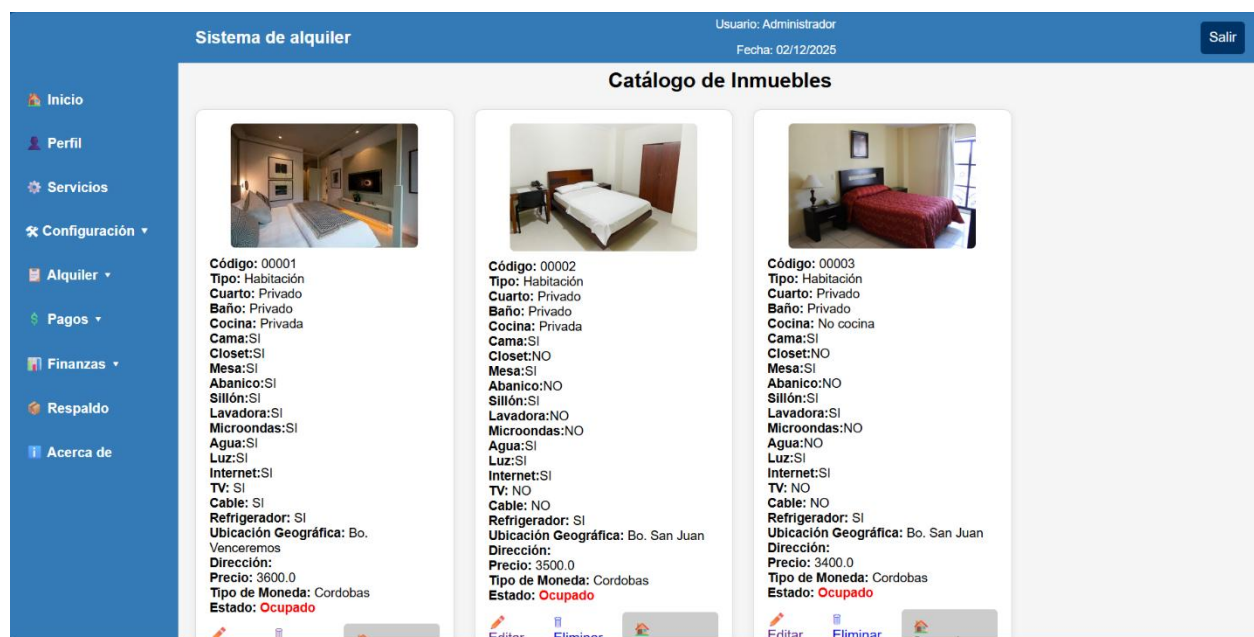


Ilustración 27 Interfaz de catálogo de Inmueble



Habitaciones del Este
 Dirección: Bo. San Juan, León, Nicaragua
 Teléfono: +505 8888-8888

CONTRATO DE ALQUILER

DATOS DEL CONTRATO
 Código Contrato: 00001

Fecha de Registro: 2025-09-20

DATOS DEL ARRENDADOR
 Nombre Completo: carlos fonseca
 Correo Electrónico: carlos@gmail.com

Teléfono: 58762378

DATOS DEL ARRENDATARIO
 Nombre Completo: Mynor Arauz
 Número Documento: 281-260202-2005M
 Correo Electrónico: Miguel@gmail.com

Tipo Documento: Cédula
 Teléfono: 57135174
 Nacionalidad: Nicaraguense

INFORMACIÓN DEL INMUEBLE
 Código Inmueble: 00001
 Dirección: pasillo 1, habitación 3
 Precio de Alquiler por día: 3600.0 Cordobas

Tipo Inmueble: Habitación
 Ubicación: Bo. Venceremos

SERVICIOS INCLUIDOS

Cama: SI	Closet: SI	Mesa: SI
Abanico: SI	Sillón: SI	Lavadora: SI
Microondas: SI	Agua: SI	Luz: SI
Internet: SI	TV: SI	Cable: SI
Refrigerador: SI		

CONDICIONES DEL ALQUILER

Fecha Inicio: 2025-08-27	Fecha Fin: 2025-08-29
Días de estadía: 2	Total a pagar: 7200.00 Cordobas

DEPÓSITO DE GARANTÍA

Monto del Depósito: 20.0	Moneda: Dolares
--------------------------	-----------------

OBSERVACIONES ADICIONALES
 Ejemplo de observaciones adicionales

Generado el 02/12/2025 08:57:52
 Firma: _____

Ilustración 28 Reporte en formato PDF

A		B	C
		Habitaciones del Este Dirección: Bo. San Juan, León, Nicaragua Teléfono: +505 8888-8888	
DATOS DEL CONTRATO			
Código Contrato	00001		
Fecha Registro	2025-09-20		
DATOS DEL ARRENDADOR			
Nombre Completo	carlos fonseca		
Teléfono	58762378		
Correo Electrónico	carlos@gmail.com		
DATOS DEL ARRENDATARIO			
Nombre Completo	Mynor Arauz		
Tipo Documento	Cédula		
Número Documento	281-260202-2005M		
Teléfono	57135174		
Correo Electrónico	Miguel@gmail.com		
Nacionalidad	Nicaraguense		
INFORMACIÓN DEL INMUEBLE			
Código Inmueble	00001		
Tipo Inmueble	Habitación		
Dirección	pasillo 1, habitación 3		
Ubicación	Bo. Venceremos		
Precio Alquiler por Día	3600.0 Cordobas		
SERVICIOS INCLUIDOS			
Cama: SI	Closet: SI	Mesa: SI	
Abanico: SI	Sillón: SI	Lavadora: SI	
Microondas: SI	Agua: SI	Luz: SI	
Internet: SI	TV: SI	Cable: SI	
Refrigerador: SI			
CONDICIONES DEL ALQUILER			
Fecha Inicio	2025-08-27		
Fecha Fin	2025-08-29		

<

>

Contrato

+

Ilustración 29 Reporte en formato Excel