



UNIVERSIDAD
CATÓLICA
DE CUENCA

UNIVERSIDAD CATÓLICA DE CUENCA

Comunidad Educativa al Servicio del Pueblo

**UNIDAD ACADÉMICA INFORMÁTICA, CIENCIAS
DE LA COMPUTACIÓN E INNOVACIÓN
TECNOLÓGICA**

**CARRERA DE INGENIERÍA DE SISTEMAS DE
INFORMACIÓN**

**DESARROLLO DE UN SISTEMA DE INFORMACIÓN PARA LA
GESTIÓN INMOBILIARIA DE BURIS Y BIENES RAÍCES, BASADO
EN METODOLOGÍAS AGILES**

**PROYECTO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO EN SISTEMAS DE INFORMACIÓN**

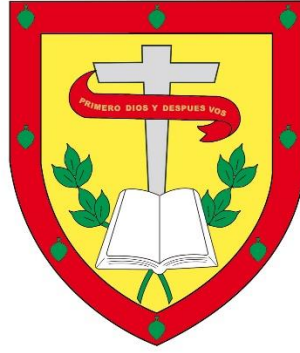
AUTORA: NARCISA ALEXANDRA GUAMAN DUCHI

DIRECTOR: ING. LUS FERMANDO PINOS CASTILLO.

CAÑAR - ECUADOR

2024

DIOS, PATRIA, CULTURA Y DESARROLLO



UNIVERSIDAD CATÓLICA DE CUENCA

Comunidad Educativa al Servicio del Pueblo

**UNIDAD ACADÉMICA INFORMÁTICA, CIENCIAS
DE LA COMPUTACIÓN E INNOVACIÓN
TECNOLÓGICA**

**CARRERA DE INGENIERÍA DE SISTEMAS DE
INFORMACIÓN**

DESARROLLO DE UN SISTEMA DE INFORMACIÓN PARA LA
GESTIÓN INMOBILIARIA DE BURIS Y BIENES RAÍCES, BASADO EN
METODOLOGÍAS AGILES

**PROYECTO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO EN SISTEMAS DE INFORMACIÓN**

AUTORA: NARCISA ALEXANDRA GUAMAN DUCHI

DIRECTOR: ING. LUS FERNANDO PINOS CASTILLO.

CAÑAR - ECUADOR

2024

DIOS, PATRIA, CULTURA Y DESARROLLO



Declaratoria de Autoría y Responsabilidad

Narcisa Alexandra Guamán Duchi, portador(a) de la cedula de ciudadanía N° **0302422894**. Declaro ser la autora de la obra: **Desarrollo de un sistema de Información para la gestión Inmobiliaria de Buris y bienes raíces, basado en metodologías ágiles**, sobre la cual me hago responsable sobre las opiniones, versiones e ideas expresadas. Declaro que la misma ha sido elaborada respetando los derechos de propiedad intelectual de terceros y eximo a la Universidad Católica de Cuenca sobre cualquier reclamación que pudiera existir al respecto. Declaro finalmente que mi obra ha sido realizada cumpliendo con todos los requisitos legales, éticos y bioéticos de investigación, que la misma no incumple con la normativa nacional e internacional en el área específica de investigación, sobre la que también me responsabilizo y eximo a la Universidad Católica de Cuenca de toda reclamación al respecto.

Cañar, 18 de noviembre de 2024

F:

Narcisa Alexandra Guamán Duchi

C.I. 0302422894

CERTIFICACIÓN DEL TUTOR

El trabajo de titulación denominado **Desarrollo de un sistema de Información para la gestión Inmobiliaria de Burís y bienes raíces, basado en metodologías ágiles**, elaborado por **Narcisa Alexandra Guamán Duchi**, previo a la obtención del título de **Ingeniera en Sistema de Información**, ha sido asesorado, revisado y supervisado durante su ejecución bajo mi tutoría, por lo que certifico que el presente documento fue desarrollado siguiendo los parámetros del método científico, se sujeta a las normas éticas de investigación, por lo que esta expedito para su presentación y sustentación ante el respectivo tribunal.

Cañar, 27 de noviembre de 2024



Ing. Luis Fernando Pinos Castillo Mgs.

CI: 0301829255

TUTOR

AGRADECIMIENTO

Antes de cualquier otra palabra, quiero dar gracias a Dios. En cada momento de incertidumbre, en cada duda y en cada paso dado hacia adelante, he sentido Su mano sosteniéndome. Su presencia ha sido mi brújula y mi aliento, y con humildad reconozco que todo logró es reflejo de su amor y guía constante.

A mis docentes, no solo los agradezco, sino los admirado. Cada uno ha dejado en mí un legado, mucho más allá del aula, en cada enseñanza que supera el simple contenido académico. Ellos han sido mentores y modelos, construyendo con paciencia y entrega un camino seguro sobre el cual avanzar.

Y, finalmente, mi más sincero reconocimiento y aprecio para el Ing. Luis Pinos, mi tutor de tesis. Su compromiso con mi crecimiento ha sido un verdadero ejemplo de dedicación. No fue solo un guía, sino un compañero en esta travesía, siempre dispuesto a brindar su apoyo y consejo con generosidad. Gracias por enseñarme que cada detalle cuenta y por ser un pilar en este proceso.

Resumen

Este trabajo presenta el desarrollo de un sistema de información diseñado para la gestión inmobiliaria de la empresa Buris y bienes raíces, utilizando metodologías ágiles, en particular, Programación Extrema (XP). El sistema propuesto busca solucionar las deficiencias en la gestión manual de propiedades, mejorando la organización, la eficiencia operativa y la satisfacción del cliente mediante la automatización de tareas críticas. Las funcionalidades principales incluyen la gestión centralizada de propiedades, la capacidad de búsqueda avanzada con filtros, la visualización interactiva mediante geolocalización en mapas y la integración con redes sociales para compartir propiedades. El uso de metodologías ágiles, como XP, permite un desarrollo flexible y adaptable a las necesidades cambiantes del mercado inmobiliario, asegurando entregas continuas de valor y una rápida implementación. Se espera que el sistema reduzca significativamente los tiempos de respuesta, mejore la toma de decisiones y optimice la experiencia tanto para los usuarios internos como para los clientes finales. Además, la centralización de la información en tiempo real permitirá a la empresa gestionar más eficientemente su inventario de propiedades y aumentar su competitividad en el sector. La implementación de este sistema también facilita la colaboración entre los miembros del equipo y mejora la visibilidad digital de las propiedades.

Palabras clave: gestión inmobiliaria, sistema de información, metodologías ágiles, XP, geolocalización.

Abstract

This paper presents the development of an information system designed for real estate management at the company Buris y bienes raíces, using agile methodologies, particularly Extreme Programming (XP). The proposed system aims to address the deficiencies in the manual management of properties, improving organization, operational efficiency, and customer satisfaction by automating critical tasks. Key features include centralized property management, advanced search functionality with filters, interactive visualization using map geolocation, and integration with social networks to share properties. The use of agile methodologies, such as XP, enables flexible development that adapts to the changing needs of the real estate market, ensuring continuous value delivery and rapid implementation. The system is expected to significantly reduce response times, improve decision-making, and optimize the experience for internal users and end customers. In addition, real-time information centralization will allow the company to manage its property inventory more efficiently and increase its competitiveness in the sector. Implementing this system also fosters collaboration among team members and enhances the digital visibility of properties.

Keywords: real estate management, information system, agile methodologies, XP, geolocation.

ÍNDICE

Contenido

<u>DECLARATORIA DE AUTORÍA Y RESPONSABILIDAD</u>	;	Error! Marcador no definido.	
<u>CERTIFICACIÓN DEL TUTOR</u>	;	Error! Marcador no definido.
<u>DEDICATORIA</u>	;	Error! Marcador no definido.
<u>AGRADECIMIENTO</u>		iii
<u>Resumen</u>	;	Error! Marcador no definido.
<u>Abstract</u>		vii
<u>ÍNDICE</u>		viii
<u>Tabla de ilustraciones</u>		xiv
<u>Tabla de tablas</u>		xv
<u>Introducción</u>		16
<u>CAPÍTULO I</u>		18
<u>1.1. Planteamiento del problema</u>		18
<u>1.1.1. Formulación del problema</u>		19
<u>1.2. Antecedentes de la Investigación</u>		19
<u>1.3. Justificación de la investigación</u>		21
<u>1.4. Objetivos</u>		22
<u>1.4.1. Objetivo General</u>		22
<u>1.4.2. Objetivos Específicos</u>		22
<u>1.5. Limitaciones</u>		23
<u>1.6. Delimitaciones</u>		23

<u>CAPÍTULO II</u>	24
<u>2. MARCO TEÓRICO</u>	24
<u>2.1. Gestión Inmobiliaria</u>	24
<u>2.1.1. Definición de la gestión inmobiliaria</u>	24
<u>2.1.2. Procesos de gestión de propiedades (ventas, alquileres, administración)</u>	24
<u>2.1.3. Problemáticas actuales en la gestión inmobiliaria sin automatización</u>	25
<u>2.1.4. Beneficios de la digitalización y automatización en el sector inmobiliario</u>	26
<u>2.2. Sistemas de Información</u>	27
<u>2.2.1. Definición de sistemas de información</u>	27
<u>2.2.2. Características de los sistemas de información</u>	28
<u>2.2.3. Componentes de un sistema de información (hardware, software, datos, procedimientos, personas)</u>	29
<u>2.2.4. Tipos de sistemas de información (TPS, MIS, DSS)</u>	29
<u>2.2.5. Aplicaciones de los sistemas de información en el sector inmobiliario</u>	31
<u>2.3 Aplicaciones WEB y su implementación</u>	33
<u>2.3.1. Desarrollo de Aplicaciones web</u>	33
<u>2.3.1.1. Características</u>	34
<u>2.3.2. Lenguaje de programación</u>	34
<u>2.3.2.1. Java</u>	34
<u>2.3.2.2. Python</u>	35
<u>2.3.2.3. C#</u>	35
<u>2.3.2.4. Comparación de los lenguajes de programación</u>	36

<u>2.3.3. Frameworks de desarrollo web</u>	38
<u>2.3.3.1. Visual Studio</u>	38
<u>2.3.3.2. Netbeans</u>	38
<u>2.3.3.3. Eclipse</u>	38
<u>2.3.3.4. Comparativa de los frameworks de desarrollo web</u>	39
<u>2.3.4 Base de datos</u>	40
<u>2.3.4.1. MySQL</u>	40
<u>2.3.4.2. PostgreSQL</u>	40
<u>2.3.4.3. Sql Server</u>	41
<u>2.3.4.4. Comparación de las bases de datos</u>	42
<u>2.4. Plataformas de desarrollo FROND-END</u>	44
<u>2.4.1. Bulma</u>	44
<u>2.4.2. Metro UI</u>	44
<u>2.4.3. Bootstrap</u>	44
<u>2.4.4. Matriz comparativa</u>	45
<u>2.5. Plataforma de desarrollo BACK-END</u>	47
<u>2.5.1. Django</u>	47
<u>2.5.2. Laravel</u>	47
<u>2.5.3. Flask</u>	47
<u>2.5.4. Matriz comparativa de los frameworks WEB Back end</u>	48
<u>2.6. Metodologías Ágiles</u>	51

<u>2.6.1. Definición de las metodologías ágiles</u>	51
<u>2.6.2. Principios de las metodologías ágiles</u>	51
<u>2.6.3. Tipos de metodologías</u>	52
<u>2.6.3.1. Scrum</u>	52
<u>2.6.3.2. Kanban</u>	53
<u>2.6.3.3. Programación Extrema (XP)</u>	53
<u>2.6.3.4. Comparación entre metodologías tradicionales y ágiles</u>	54
<u>CAPÍTULO III</u>	57
<u>3. MARCO METODOLÓGICO</u>	57
<u>3.1. Enfoque de la investigación</u>	57
<u>3.2. Nivel de la investigación</u>	57
<u>3.3. Población</u>	57
<u>3.4. Técnicas e instrumentos de recolección</u>	57
<u>3.5. Tratamiento de la información</u>	58
<u>3.6. Resultados</u>	58
<u>3.6.1. Análisis de los resultados</u>	58
<u>3.6.2. Análisis general de la entrevista</u>	61
<u>CAPÍTULO IV</u>	62
<u>4. PROPUESTA</u>	62
<u>4.1. Título de la Propuesta</u>	62
<u>4.2. Presentación</u>	62

4.3. <u>Justificación</u>	62
4.4. <u>Descripción de la propuesta</u>	63
4.4.1. <u>¿Qué se va a realizar?</u>	63
4.4.2. <u>¿Cómo se va a realizar?</u>	63
4.4.3. <u>¿Para qué se va a realizar?</u>	63
4.5. <u>Desarrollo del proyecto</u>	64
4.5.1. <u>Planificación</u>	64
4.5.1.1. <u>Establecimiento de las Historias de Usuario</u>	65
4.5.1.2. <u>Priorización de las historias de usuario</u>	66
4.5.1.3. <u>Estimación de tiempo y esfuerzo</u>	67
4.5.1.4. <u>Selección de Historias para la Iteración (Iteration Planning)</u>	68
4.5.2. <u>Diseño</u>	68
4.5.2.1. <u>Diseño simple</u>	69
4.5.2.2. <u>Uso de Tarjetas CRC (Class-Responsibility-Collaborator)</u>	69
4.5.3. <u>Desarrollo</u>	71
4.5.3.1. <u>Base de datos</u>	71
4.5.3.2. <u>Codificación</u>	74
4.5.3.2. <u>Interfaces del Sistema de Información</u>	79
4.5.4. <u>Pruebas</u>	86
4.5.4.1. <u>Pruebas Unitarias</u>	86
4.5.4.2. <u>Pruebas de Integración</u>	87

<u>4.5.4.3. Pruebas de Aceptación</u>	88
<u>4.5.4.4. Pruebas Automatizadas</u>	90
<u>4.5.5. Retroalimentación</u>	93
<u>4.5.5.1. Retroalimentación del cliente</u>	93
<u>4.5.5.2. Tabla de Ajustes Continuos</u>	1
<u>4.5.5.3. Calidad de Código</u>	2
<u>Conclusiones</u>	3
<u>Recomendaciones</u>	5
<u>Bibliografía</u>	6
<u>Anexos</u>	13

Tabla de ilustraciones

<u>Ilustración 1. Tipos de Sistemas de Información. Fuente: Autoría Propia.</u>	30
<u>Ilustración 2. Logo del lenguaje de programación JAVA. Fuente: (Rosa, 2019).</u>	34
<u>Ilustración 3. Logo del lenguaje de programación de python. (python, 2024)</u>	35
<u>Ilustración 4. Logo del lenguaje C#. Fuente: (Gutierrez Garay, 2020).</u>	35
<u>Ilustración 5. Diseño Simple. Fuente: Autoría Propia.</u>	69
<u>Ilustración 6. Diseño Físico de la base de datos. Fuente: Autoría propia.</u>	72
<u>Ilustración 7. Diseño lógico de la base de datos. Fuente: Autoría propia.</u>	73
<u>Ilustración 8. Clase Propiedad: Fuente: Autoría Propia.</u>	77
<u>Ilustración 9. Clase Usuario. Fuente: Autoría Propia.</u>	79
<u>Ilustración 10. Login del usuario. Fuente: Autoría propia.</u>	79
<u>Ilustración 11. Interfaz Principal. Fuente: Autoría Propia.</u>	80
<u>Ilustración 12. Pantalla de administración. Fuente: Autoría Propia.</u>	82
<u>Ilustración 13. Lista de usuarios. Fuente: Autoría Propia.</u>	83
<u>Ilustración 14. Pantalla de localidades. Fuente: Autoría Propia.</u>	83
<u>Ilustración 15. Pantalla de propiedades. Fuente: Autoría Propia.</u>	84
<u>Ilustración 16. Características de la propiedad. Fuente: Autoría propia.</u>	85
<u>Ilustración 17. Ubicación. Fuente: Autoría Propia.</u>	85
<u>Ilustración 18. Ajuste continuo. Fuente: Autoría Propia.</u>	1

Tabla de tablas

<u>Tabla 1. Beneficios de la digitalización y automatización en el sector inmobiliario. Fuente:</u>	
<u>Autoría propia.</u>	26
<u>Tabla 3 Comparación de los lenguajes de programación. Fuente: Autoría Propia.</u>	36
<u>Tabla 4. Comparativas de los frameworks de desarrollo web. Fuente: Autoría Propia.</u>	39
<u>Tabla 5. Comparativa de bases de datos. Fuente: Autoría Propia.</u>	42
<u>Tabla 6. Comparativa de plataformas de desarrollo FROND-END. Fuente: Autoría Propia.</u> ..	45
<u>Tabla 7. Comparativa de los frameworks backend. Fuente: Autoría Propia.</u>	48
<u>Tabla 8. Comparativas de las metodologías. Fuente: Autoría Propia.</u>	54
<u>Tabla 9. Establecimiento de historias de usuario. Fuente: Autoría Propia.</u>	65
<u>Tabla 10. Priorización de las historias de usuario. Fuente: Autoría Propia.</u>	66
<u>Tabla 11. Estimación de tiempo y esfuerzo. Fuente: Autoría propia.</u>	67
<u>Tabla 12. Iteration planning. Fuente: Autoría Propia.</u>	68
<u>Tabla 13. Tarjetas CRC. Fuente: Autoría Propia.</u>	70
<u>Tabla 14. Matriz de pruebas unitarias. Fuente: Autoría Propia.</u>	86
<u>Tabla 15. Pruebas de integración. Fuente: Autoría propia.</u>	87
<u>Tabla 16. Pruebas de Aceptación. Fuente: Autoría Propia.</u>	89
<u>Tabla 17. Pruebas Automatizadas. Fuente: Autoría Propia.</u>	91
<u>Tabla 18. Retroalimentación del cliente. Fuente: Autoría Propia.</u>	94
<u>Tabla 19. Calidad del código. Fuente: Autoría Propia.</u>	2

Introducción

La gestión eficiente de propiedades en el sector inmobiliario es un desafío constante, especialmente para empresas como Buris y bienes raíces. Actualmente, la empresa enfrenta problemas relacionados con la falta de un sistema de información centralizado, lo que provoca ineficiencias en la administración de propiedades. Los procesos manuales y descentralizados generan errores en la duplicación de datos y tiempos de respuesta lentos, afectando tanto la operatividad interna como la satisfacción de los clientes.

Ante este panorama, surge la necesidad de implementar un Sistema de Información (SI) que permita centralizar y automatizar la gestión de propiedades. Este sistema deberá mejorar la precisión en la recolección de datos, reducir los tiempos de respuesta y optimizar la toma de decisiones en la empresa. Además, se utilizarán metodologías ágiles, específicamente Extreme Programming (XP), para asegurar la flexibilidad y adaptación del sistema a las necesidades cambiantes del negocio.

El interés en este proyecto radica en la capacidad de las tecnologías de información para transformar los procesos operativos de las empresas inmobiliarias, permitiendo una mayor eficiencia en la administración de sus activos. Asimismo, el uso de metodologías ágiles ofrece una ventaja competitiva al permitir una rápida implementación y ajuste de las funcionalidades del sistema en función de los cambios en el entorno.

El objetivo general de este trabajo es desarrollar un Sistema de Información que centralice la gestión inmobiliaria en Buris y Bienes Raíces, promoviendo una mejora operativa y una mayor satisfacción del cliente. Los objetivos específicos incluyen analizar diferentes metodologías ágiles, desarrollar el sistema de información utilizando la metodología XP y evaluar el impacto de su implementación en la empresa.

Este trabajo se organiza en los siguientes capítulos:

Capítulo I: Planteamiento del Problema

En este capítulo se detalla la problemática actual de la empresa Buris y Bienes Raíces, los antecedentes de la investigación, la justificación del proyecto y los objetivos generales y específicos del mismo.

Capítulo II: Marco Teórico

Se ofrece una revisión de la literatura sobre la gestión inmobiliaria, los beneficios de la automatización y digitalización de procesos, y la implementación de sistemas de información basados en metodologías ágiles en el sector inmobiliario.

Capítulo III: Marco Metodológico

Se describe el enfoque de investigación, las técnicas de recolección de datos y el análisis de los resultados esperados. Además, se detalla el proceso de desarrollo del sistema de información utilizando la metodología XP.

Capítulo IV: Propuesta

En este capítulo se presenta el desarrollo del sistema de información, incluyendo su diseño, implementación y las pruebas realizadas para garantizar su efectividad. Se analizan los beneficios esperados y el impacto en la operatividad de Buris y Bienes Raíces.

CAPÍTULO I

1.1. Planteamiento del problema

La gestión eficiente de las propiedades en el sector inmobiliario es un desafío recurrente, especialmente para empresas como Buris y Bienes Raíces. La falta de un sistema de información centralizado ha provocado una serie de problemas en la administración y comercialización de propiedades. Actualmente, la empresa enfrenta la dificultad de gestionar la información de manera manual y descentralizada, lo que conlleva a errores, duplicación de datos y tiempos de respuesta lentos. Estas ineficiencias afectan directamente el flujo de trabajo interno y la capacidad de la empresa para brindar un servicio rápido y eficiente a los clientes, lo cual es esencial en un mercado inmobiliario cada vez más competitivo.

La ausencia de un sistema adecuado para la gestión de propiedades disponibles para la venta o alquiler también repercute en la satisfacción del cliente, ya que los tiempos de respuesta para la consulta de información actualizada sobre propiedades son prolongados. Además, los métodos manuales de seguimiento de los proyectos inmobiliarios en curso generan retrasos en la entrega de propiedades y dificultan la gestión de recursos, afectando la relación con los clientes y la eficiencia operativa general.

Frente a este escenario, surge la necesidad de implementar un sistema de información que centralice y automatice los procesos de gestión inmobiliaria. Este sistema debe permitir a la empresa mejorar la precisión en la recolección de datos, reducir el tiempo de respuesta y optimizar la toma de decisiones. Asimismo, se debe garantizar que el sistema sea desarrollado utilizando metodologías ágiles para asegurar su flexibilidad y adaptación a las necesidades cambiantes del negocio. La implementación de este tipo de tecnologías

es crucial para que Buris y Bienes Raíces mejore su competitividad en el mercado y ofrezca una mejor experiencia al cliente.

1.1.1. Formulación del problema

A pesar de los avances en el uso de tecnologías para la gestión empresarial, la administración inmobiliaria de Buris y Bienes Raíces aún se realiza mediante procesos manuales y descentralizados, lo que genera ineficiencias operativas y afecta la satisfacción del cliente. La falta de un sistema de información centralizado impide una gestión eficiente de las propiedades y dificulta el seguimiento de proyectos inmobiliarios en curso. Además, los métodos tradicionales no permiten una rápida adaptación a las demandas del mercado, lo que afecta la competitividad de la empresa.

En este contexto, surge la necesidad de investigar cómo un sistema de información basado en metodologías ágiles puede optimizar la gestión de propiedades en Buris y Bienes Raíces, mejorando tanto la eficiencia operativa como la satisfacción del cliente. La pregunta clave que guía esta investigación es:

¿Cómo puede la implementación de un sistema de información basado en metodologías ágiles mejorar la eficiencia operativa en la gestión de propiedades y aumentar la satisfacción del cliente en Buris y Bienes Raíces?

1.2. Antecedentes de la Investigación

En los últimos años, varios estudios han abordado la implementación de sistemas de información y el uso de metodologías ágiles en diferentes sectores, incluyendo el inmobiliario. Estos estudios proporcionan una base sólida para desarrollar un sistema de información para Buris y Bienes Raíces, mejorando la eficiencia y adaptabilidad en la gestión de propiedades.

Criollo Chanchicocha, Vilmer David (2020)– **"Desarrollo del Sistema de Información que Soporta la Gestión de Procesos en la Dirección de Posgrados de la Universidad Técnica de Cotopaxi"**

Este trabajo se centró en la creación de un sistema de información para la gestión de procesos administrativos en una universidad. Aunque su enfoque está en el ámbito académico, los resultados demostraron que un sistema de información bien diseñado puede optimizar significativamente los procesos, reduciendo el tiempo de respuesta y mejorando la precisión de los datos. Este estudio servirá como referencia para estructurar el sistema de información que se implementará en Buris y Bienes Raíces, garantizando que se minimicen los errores en la gestión de propiedades y se mejore la eficiencia operativa.

Arthur Pethermar (2020) – "El Desarrollo de un Sistema de Información para Automatizar Procesos"

Este estudio analizó el uso de metodologías ágiles, como Scrum, Kanban y XP, en el desarrollo de sistemas de información para automatizar procesos. Los hallazgos mostraron que las metodologías ágiles permiten una mayor flexibilidad y adaptabilidad durante el desarrollo, lo que es esencial para proyectos en sectores cambiantes como el inmobiliario. Este trabajo será clave para seleccionar las metodologías ágiles más adecuadas para el desarrollo del sistema propuesto, permitiendo que se ajuste a las necesidades dinámicas de Buris y Bienes Raíces.

Mario Segovia y Yuritha Monte (2021) – **"Un Sistema de Información para la Transformación Digital en Empresas del Sector Inmobiliario"**

Este estudio se enfocó en la digitalización de los procesos de gestión en empresas inmobiliarias. Los autores demostraron que la implementación de un sistema de información centralizado mejora la comunicación entre departamentos, facilita el acceso a información en tiempo real y optimiza la toma de decisiones estratégicas. Este trabajo será útil para comprender

cómo la digitalización de los procesos inmobiliarios puede impactar positivamente en Buris y Bienes Raíces, especialmente en lo que respecta a la gestión de propiedades y la satisfacción del cliente.

Carlos Ruiz (2021) – **"Impacto de los Sistemas de Información en la Gestión de Proyectos Inmobiliarios"**

Ruiz exploró cómo los sistemas de información pueden mejorar la gestión de proyectos inmobiliarios al facilitar el seguimiento de las propiedades y optimizar la asignación de recursos. Además, destacó los desafíos de la implementación, como la resistencia al cambio organizacional. Este estudio proporcionará una base para anticipar posibles obstáculos durante la implementación del sistema de información en Buris y Bienes Raíces y plantear estrategias para superarlos.

José López (2022) – **"Sistemas de Información y Competitividad en el Sector Inmobiliario"**

López investigó cómo la adopción de sistemas de información basados en metodologías ágiles mejora la competitividad en empresas del sector inmobiliario. Su trabajo mostró que estas herramientas permiten a las empresas responder más rápidamente a las demandas del mercado, mejorar la eficiencia interna y aumentar la satisfacción del cliente. Este estudio será crucial para fundamentar el impacto positivo que tendrá el sistema de información propuesto en Buris y Bienes Raíces, tanto en la optimización de sus procesos como en su capacidad para competir en el mercado.

1.3. Justificación de la investigación

El desarrollo del presente trabajo se sustenta en una serie de consideraciones fundamentales que abordan tanto las necesidades del mercado como los desafíos operativos actuales dentro del sector inmobiliario. En primer lugar, el mercado inmobiliario es altamente

competitivo y dinámico, donde la eficiencia y la capacidad de adaptación son factores críticos para el éxito. Los profesionales del sector enfrentan constantemente la presión de ofrecer servicios de alta calidad de manera oportuna y precisa, lo que requiere una gestión eficiente de la información, los recursos y las transacciones. La adopción de un aplicativo web específicamente diseñado para este propósito permitirá a los profesionales inmobiliarios optimizar sus operaciones diarias, minimizando el tiempo y los recursos necesarios para realizar tareas administrativas y operativas.

Además, la naturaleza descentralizada y dispersa de la información en el mercado inmobiliario puede dificultar la colaboración y la comunicación efectiva entre los miembros del equipo y con los clientes. El sistema permitirá automatizar y centralizar procesos clave de la gestión inmobiliaria, reduciendo la dependencia de métodos manuales propensos a errores y tiempos de respuesta lentos. Esto se traducirá en una mayor eficiencia operativa, liberando tiempo y recursos que pueden ser reinvertidos en actividades estratégicas y de valor agregado

1.4. Objetivos

1.4.1. Objetivo General

Desarrollar un sistema de información para Buris y bienes raíces, **usando metodologías ágiles** para optimizar la gestión de propiedades y mejorar la administración de clientes y procesos internos

1.4.2. Objetivos Específicos

- Analizar fuentes bibliográficas que permitirá recopilar información adecuada para la investigación relacionadas al desarrollo de un sistema de información para la inmobiliaria Buris y bienes raíces basadas en metodologías ágiles.
- Analizar las diferentes metodologías eficientes mediante la información recopilada.

- Desarrollar un sistema de información que permita visualizar datos de diferentes propiedades, mediante la ubicación geográfica, fotografías y videos.

1.5. Limitaciones

- Dificultades al buscar información sobre el tema propuesto.
- El plazo de ejecución del proyecto es muy corto.

1.6. Delimitaciones

- Dado que el tiempo de desarrollo es de tres meses, el sistema de información se enfocará únicamente en las características esenciales.
- Solo se enfocará en estudios realizados entre 2020 y 2024.
- Este sistema de información estará enfocado en la gestión inmobiliaria, por lo que atenderá exclusivamente las necesidades de la inmobiliaria Buris y bienes raíces.

CAPÍTULO II

MARCO TEÓRICO

2.1. Gestión Inmobiliaria

2.1.1. Definición de la gestión inmobiliaria

La gestión inmobiliaria es el conjunto de actividades y procesos orientados a la administración eficiente de bienes inmuebles, como propiedades residenciales, comerciales o terrenos, que incluye su compra, venta, alquiler y mantenimiento. Su objetivo es optimizar el valor de los inmuebles, garantizar su adecuado funcionamiento y maximizar su rentabilidad mediante la adecuada gestión de contratos, inquilinos y recursos relacionados (Carvajal Camacho, 2021).

2.1.2. Procesos de gestión de propiedades (ventas, alquileres, administración)

La gestión de propiedades en el sector inmobiliario implica una serie de procesos clave que permiten la eficiente administración de activos inmobiliarios, ya sea para la venta, el alquiler o la administración general. A continuación, se describen los principales procesos involucrados:

Ventas: Implica la captación, valoración, promoción y negociación de propiedades. Un manejo adecuado permite aumentar el valor de la propiedad y asegurar una venta exitosa (Muevecela Bermeo, 2020).

Alquileres: Consiste en la selección de inquilinos, elaboración de contratos y gestión del cumplimiento de las obligaciones, tales como pagos y mantenimiento (Barahona Blanco, 2020).

Administración: Abarca la supervisión del mantenimiento, la planificación financiera y la gestión de servicios. Una administración eficiente maximiza el valor de las propiedades y garantiza el bienestar de sus ocupantes (Esquivel Oviedo, 2020).

2.1.3. Problemáticas actuales en la gestión inmobiliaria sin automatización

- **Errores y duplicidad de datos:** La gestión manual aumenta el riesgo de cometer errores humanos, como duplicación de información o registro incorrecto de datos, lo que puede derivar en confusión y retrasos en la toma de decisiones (CEPAL, 2020).
- **Procesos y falta de eficiencia:** La falta de automatización provoca que las tareas como la actualización de inventarios, la gestión de contratos o la comunicación con clientes e inquilinos se realicen de manera lenta, afectando la productividad general de la empresa y la capacidad de respuesta ante el mercado (Nicolau Medina & Gozalbes Ballester).
- **Dificultad para acceder a información en tiempo real:** Sin un sistema automatizado, acceder a datos actualizados sobre las propiedades disponibles, estados de pagos o gestiones pendientes puede ser complicado, lo que limita la capacidad de tomar decisiones informadas ya tiempo.
- **Falta de seguimiento eficiente de las propiedades:** La gestión manual de propiedades dificulta el control sobre los contratos de alquileres, las fechas de vencimiento, el estado de mantenimiento de cada una de las propiedades y otras responsabilidades operativas, lo que puede ocasionar retrasos y problemas con los clientes (Ibáñez Chuquín & Ulcuango Ulcuango , 2020).
- **Costos operativos más altos:** El tiempo y los recursos que requiere la gestión manual aumentan los costos operativos. Las tareas repetitivas y el exceso de papeleo no solo son ineficientes, sino que también incrementan el gasto en mano de obra y materiales

(Arguello Pazmiño, Llumiguano Poma, Gavilánez Cárdenas, & Torres Ordoñez, 2020).

- **Poca adaptabilidad a cambios del mercado:** La gestión manual es más rígida y no permite una respuesta rápida a los cambios del mercado, como fluctuaciones en la oferta y demanda de propiedades, afectando la competitividad de la empresa (Izurieta Castelo, Miño Cascante, & Vallejo Sánchez, 2021, pág. 101).
- **Experiencia del cliente deficiente:** Al no contar con sistemas automatizados que mejoren la interacción con los clientes (como la disponibilidad de información en línea), los clientes enfrentan una experiencia más lenta y menos accesible, lo que puede afectar su satisfacción y fidelidad (managementsolutions, 2020, pág. 9).

2.1.4. Beneficios de la digitalización y automatización en el sector inmobiliario

La digitalización y automatización han transformado significativamente el sector inmobiliario, permitiendo que las empresas optimicen sus procesos y mejoren su competitividad. A continuación, se presentan los principales beneficios que estas tecnologías aportan al sector:

Tabla 1. Beneficios de la digitalización y automatización en el sector inmobiliario. Fuente: Autoría propia.

Beneficios de la digitalización	Beneficios de la automatización
La digitalización ha facilitado el acceso a información sobre propiedades en cualquier momento y lugar, mejorando la conveniencia y acelerando los procesos de compra y venta inmobiliaria.	Automatiza tareas repetitivas, reduciendo tiempos y errores.

La digitalización ha mejorado la eficiencia en la gestión de propiedades mediante herramientas en línea que facilitan la comunicación, el seguimiento de pagos y la programación de mantenimiento, reduciendo errores administrativos.

Optimiza procesos, disminuyendo costos operativos y aumentando la rentabilidad.

Las tecnologías digitales permiten el análisis de datos para personalizar estrategias de marketing y ofrecer experiencias más relevantes a los clientes en el mercado inmobiliario.

Facilita el mantenimiento y monitoreo de pagos, mejorando la administración de inmuebles.

La digitalización ha mejorado la transparencia y seguridad en las transacciones inmobiliarias mediante el acceso a información detallada y el uso de tecnologías como blockchain para garantizar la integridad.

Facilita el análisis en tiempo real para decisiones más informadas.

2.2. Sistemas de Información

2.2.1. Definición de sistemas de información

Un **sistema de información** es un conjunto interrelacionado de componentes que recopilan, procesan, almacenan y distribuyen información para apoyar la toma de decisiones y el control dentro de una organización. Estos sistemas también facilitan la coordinación y comunicación entre los diversos departamentos de una empresa. Los sistemas de información

ayudan a las organizaciones a operar de manera más eficiente y eficaz, brindando acceso a información relevante en tiempo real, lo que es crucial para la toma de decisiones estratégicas (Arévalo-Cordovilla, Ordoñez-Sigcho, Peñaherrera-Larenas , & Suárez-Matamoros, 2020)

2.2.2. Características de los sistemas de información

Los sistemas de información son fundamentales en las organizaciones modernas, permitiendo la gestión eficiente de datos y optimización de procesos. A continuación, según (Gómez Vieites & Suárez Rey , 2020) detallan las principales características.

- **Exactitud:** La información que genera el sistema de información debe ser precisa y libre de errores para ser útil en la toma de decisiones.
- **Completitud:** La información debe incluir todos los hechos necesarios para que sea útil para la persona que la utilizará.
- **Economía:** El costo para obtener la información debe ser menor que el beneficio que proporciona a la organización.
- **Confianza:** Es necesario que la información sea confiable. Esto se logra asegurando la calidad tanto de los datos como de las fuentes de información.
- **Relevancia:** La información proporcionada por el sistema debe ser relevante para la toma de decisiones, evitando datos innecesarios.
- **Nivel de detalle:** La información debe tener el nivel de detalle adecuado para el uso previsto, ni demasiado general ni excesivamente detallado.
- **Oportunidad:** La información debe ser entregada en el momento adecuado, cuando la persona la necesita para tomar decisiones.
- **Verificabilidad:** La información debe poder ser contrastada y verificada en todo momento.

2.2.3. Componentes de un sistema de información (hardware, software, datos, procedimientos, personas)

Un sistema de información se compone de varios elementos esenciales que interactúan para procesar y manejar la información dentro de una organización. Estos componentes incluyen:

- **Hardware:** Los dispositivos físicos como computadoras, servidores y dispositivos de almacenamiento que permiten la ejecución de tareas relacionadas con el sistema.
- **Software:** Las aplicaciones y programas que procesan la información y permiten realizar tareas específicas, como la gestión de bases de datos o la visualización de información.
- **Datos:** Información bruta que se convierte en un recurso valioso cuando se organiza y procesa correctamente. Los datos son el corazón de cualquier sistema de información.
- **Procedimientos:** Las políticas y normas que rigen el uso del sistema de información dentro de la organización.
- **Personas:** Los usuarios del sistema, quienes lo operan y gestionan (Bernardi & Dranca, 2020)

2.2.4. Tipos de sistemas de información (TPS, MIS, DSS)

Los sistemas de información se clasifican en diferentes tipos según su función dentro de una organización. Algunos de los más relevantes son:

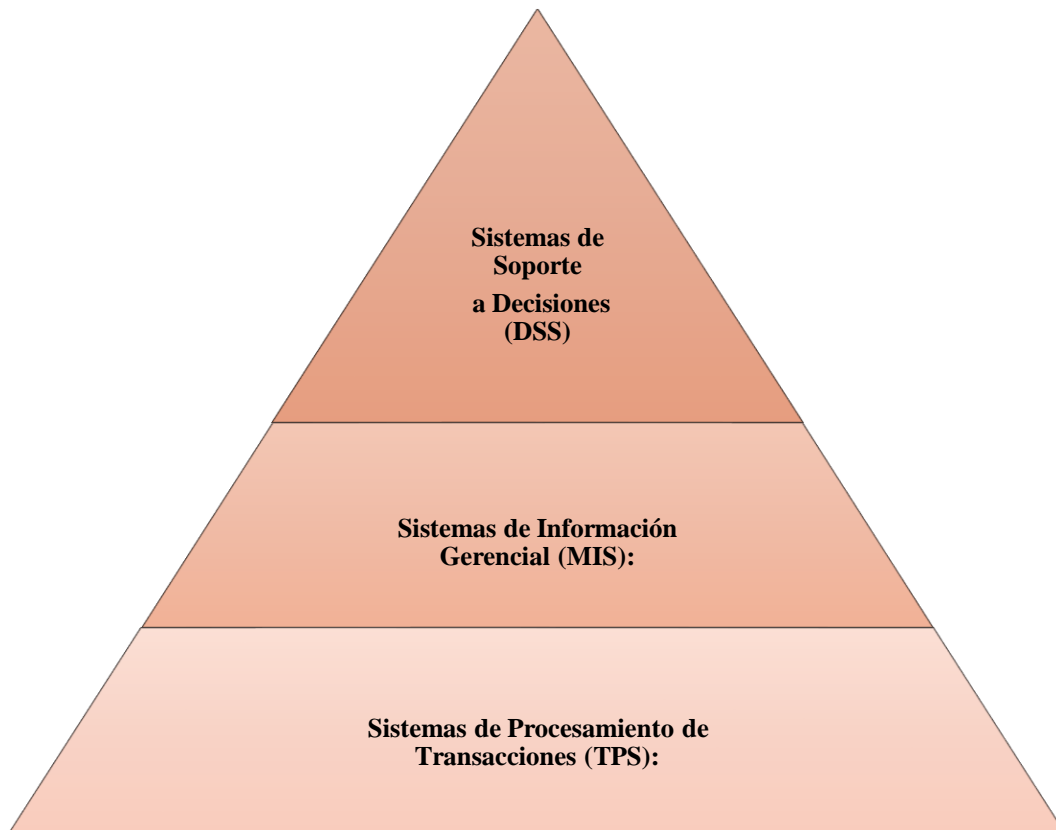


Ilustración 1. Tipos de Sistemas de Información. Fuente: Autoría Propia.

- **Sistemas de Procesamiento de Transacciones (TPS):** Un sistema de procesamiento de transacciones (TPS, por sus siglas en inglés) es un tipo de software de procesamiento de información de gestión de datos que se emplea durante una transacción comercial para gestionar la recopilación y recuperación de datos tanto del cliente como de la empresa (IBM, 2024).
- **Sistemas de Información Gerencial (MIS):** Los sistemas MIS dan soporte a la toma de decisiones normalmente mediante la realización, con cierta periodicidad, de informes. Se relacionan con las decisiones regulares y estructuradas. El tipo de información que proporcionan suele tener relación con el desempeño actual de la organización y se nutren de los datos resumidos que les proporcionan los sistemas TPS de la organización. Algunos ejemplos de decisiones a las que pueden dar apoyo los sistemas MIS son la administración de ventas, el control de inventarios, la elaboración del presupuesto anual (Bernardi & Dranca, 2020).

- **Sistemas de Soporte a Decisiones (DSS):** Es una herramienta informática diseñada para ayudar a los responsables en la toma de decisiones, proporcionando análisis y evaluaciones de datos complejos. Estos sistemas facilitan la toma de decisiones estratégicas mediante la recopilación y análisis de grandes volúmenes de datos, presentando resultados que permiten evaluar diferentes escenarios y alternativas (Morales Lugo, Figueroa Millán, Farías Mendoza, & Chávez Valdez, 2020).

2.2.5. Aplicaciones de los sistemas de información en el sector inmobiliario

Los sistemas de información han transformado de manera significativa la gestión inmobiliaria, proporcionando herramientas tecnológicas que optimizan los procesos de administración, venta y alquiler de propiedades. Estas aplicaciones permiten a las empresas inmobiliarias gestionar grandes volúmenes de datos relacionados con propiedades, clientes, transacciones y contratos, facilitando la toma de decisiones y mejorando la eficiencia operativa.

Entre las principales aplicaciones están:

Sistemas de Gestión de Documentos (DMS): Estos sistemas permiten almacenar, gestionar y organizar documentos digitalmente, facilitando el acceso y el control de versiones. En el sector inmobiliario, los DMS se utilizan para gestionar contratos, facturas, planos y otros documentos críticos, eliminando el uso de papel y mejorando la seguridad en la gestión de la información (docuSign, 2023).

Ejemplos:

- **DocuWare:** Utilizado por empresas inmobiliarias para gestionar contratos, facturas y otros documentos de manera digital, permitiendo su almacenamiento, organización y acceso seguro en la nube (start.docuware, 2024).
- **M-Files:** Facilita la organización y control de versiones de documentos importantes, como contratos de arrendamiento, planos de propiedades y reportes de mantenimiento (m-files, 2024).

- **Sistemas de Planificación de Recursos Empresariales (ERP):** Los ERP integran todas las áreas operativas de una empresa inmobiliaria, como contabilidad, ventas, marketing y recursos humanos. En el sector, estos sistemas permiten gestionar propiedades, realizar un seguimiento de ingresos y gastos, coordinar proyectos de construcción y mejorar la eficiencia operativa a través de un único sistema centralizado (Jaime Adrián & Mishell Mellisa, 2023).
- **Ejemplo:**
- **SAP ERP:** Permite a grandes empresas inmobiliarias integrar áreas como contabilidad, gestión de propiedades, marketing y ventas en una sola plataforma, optimizando la eficiencia operativa (sapbusinessone.company, 2024).
- **Sistemas de Gestión de Mantenimiento Asistido por Computadora (CMMS):** Según: (ibm, 2024) afirma que estos sistemas son fundamentales para la gestión del mantenimiento de edificios y propiedades. Permiten programar, realizar seguimiento y gestionar el mantenimiento preventivo y correctivo, asegurando que los inmuebles estén en condiciones óptimas, lo que incrementa su valor y satisface las necesidades de los inquilinos o propietarios.
- **Ejemplo:**
- **Hippo CMMS:** Es utilizado por empresas inmobiliarias para programar y realizar el seguimiento de tareas de mantenimiento preventivo y correctivo en edificios y propiedades (hoteltechreport, 2024)
- **Sistemas de Geolocalización:** Los sistemas de geolocalización ayudan a las empresas inmobiliarias a identificar las mejores ubicaciones para desarrollos futuros, analizar tendencias demográficas y acceder a información sobre infraestructura local, transporte y servicios públicos. Además, permiten a los compradores visualizar la ubicación de las propiedades y su cercanía a puntos de interés (Avilés, 2023).

Ejemplo:

- **GIS (Geographic Information Systems):** Empresas como Esri proporcionan herramientas GIS que ayudan a identificar áreas de desarrollo, analizar tendencias demográficas y evaluar la infraestructura de una ubicación (Lock, 2022).
- **Sistemas de Realidad Virtual y Aumentada (VR/AR):** La realidad virtual y aumentada permite a los compradores y arrendadores explorar propiedades de manera inmersiva. Estos sistemas se utilizan para realizar recorridos virtuales por inmuebles, visualizaciones en 3D de proyectos de construcción o remodelación, y para mejorar la experiencia del cliente, eliminando la necesidad de visitas físicas a las propiedades (Ballesteros Reyes & Borrero Escalante, 2023).

Ejemplo:

- **Matterport:** Utilizado para crear recorridos virtuales en 3D de propiedades, permitiendo a los compradores explorar los inmuebles de manera interactiva y detallada sin estar físicamente presentes (matterport, 2024).

2.3 Aplicaciones WEB y su implementación

2.3.1. Desarrollo de Aplicaciones web

Una aplicación web es un tipo de sistema cliente/servidor en el que tanto el cliente (navegador), el servidor web, como el protocolo de comunicación (HTTP) están estandarizados, lo que facilita su desarrollo ya que el programador no necesita construir estos elementos desde cero. Su popularidad se debe a que el navegador funciona como un cliente ligero, permitiendo que las aplicaciones sean fácilmente actualizables sin necesidad de distribuir software a múltiples usuarios. Además, las páginas web ofrecen interactividad, permitiendo a los usuarios interactuar con los datos, como completar formularios o acceder a bases de datos, de forma dinámica y eficiente (Guerra Poalasin , 2021).

2.3.1.1. Características

- No requieren la instalación de software adicional, ya que todo se ejecuta en el navegador.
- Las actualizaciones se realizan directamente en el servidor, siendo instantáneas para todos los usuarios.
- Funcionan en cualquier sistema operativo que tenga un navegador compatible.
- Permiten una interacción dinámica sin necesidad de recargar la página completa.
- La seguridad se gestiona en el servidor, protegiendo los datos mediante HTTPS (Bonilla Adriano, 2021).

2.3.2. Lenguaje de programación

2.3.2.1. Java



Ilustración 2. Logo del lenguaje de programación JAVA. Fuente: (Rosa, 2019)

Java es un lenguaje de programación y una plataforma informática lanzado por primera vez por Sun Microsystems en 1995. Ha evolucionado desde sus humildes comienzos hasta impulsar una gran parte del mundo digital actual, proporcionando una plataforma confiable sobre la cual se construyen muchos servicios y aplicaciones. Los productos nuevos e innovadores y los servicios digitales diseñados para el futuro también siguen confiando en Java (java, 2024).

2.3.2.2. Python



Ilustración 3. Logo del lenguaje de programación de python. (python, 2024)

Según Infante (2023) afirma que Python es un lenguaje de programación ampliamente utilizado en diversas ramas de la ingeniería para tareas como el análisis de datos, el desarrollo de algoritmos y la creación de modelos. Su versatilidad y potente ecosistema de librerías lo convierten en una herramienta clave para desarrolladores e ingenieros. Entre sus librerías más destacadas se encuentran **NumPy**, especializada en cálculos numéricos con vectores multidimensionales; **Matplotlib**, que permite la creación de gráficos personalizables de alta calidad; **Tkinter**, una interfaz que facilita el desarrollo de interfaces gráficas; y **Pandas**, orientado al manejo y análisis de datos en campos como la estadística, economía e ingeniería.

2.3.2.3. C#



Ilustración 4. Logo del lenguaje C#. Fuente: (Gutierrez Garay, 2020).

El lenguaje C# es el lenguaje más popular para la plataforma .NET, un entorno de desarrollo gratuito, multiplataforma y de código abierto. Los programas C# pueden ejecutarse en muchos dispositivos diferentes, desde dispositivos de Internet de las cosas (IoT) hasta la nube y cualquier otro lugar intermedio. Puede escribir aplicaciones para teléfonos, computadoras de escritorio, portátiles y servidores.

C# es un lenguaje multiplataforma de propósito general que permite a los desarrolladores trabajar de manera productiva mientras escriben código de alto rendimiento. Con millones de desarrolladores, C# es el lenguaje .NET más popular. C# tiene un amplio respaldo en el ecosistema y en todas las cargas de trabajo .NET. Basado en principios orientados a objetos, incorpora muchas características de otros paradigmas, en particular la programación funcional. Las características de bajo nivel admiten escenarios de alta eficiencia sin escribir código inseguro. La mayor parte del entorno de ejecución y las bibliotecas de .NET están escritas en C#, y los avances en C# a menudo benefician a todos los desarrolladores de .NET (learn.microsoft.com, 2024).

2.3.2.4. Comparación de los lenguajes de programación

Tabla 2 Comparación de los lenguajes de programación. Fuente: Autoría Propia.

Criterio	Java	Python	C#
Recopilación	Compilado un código de bytes para la máquina virtual Java (JVM)	Interpretado, ejecutado línea por línea	Compilado en lenguaje intermedio (CIL) para .NET
Paradigma	Orientado a objetos	Multiparadigma (orientado a objetos,	Orientado a objetos

		funcional, imperativo)	
Propósito	Desarrollo de aplicaciones empresariales, móviles y web.	Desarrollo de aplicaciones científicas, web, automatización, IA	Desarrollo de aplicaciones empresariales, de escritorio, web.
Nivel	Lenguaje de alto nivel	Lenguaje de alto nivel	Lenguaje de alto nivel
Tipado	Tipado estático	Tipado dinámico	Tipado estático
SopORTE	Sí	Sí	Sí
multiplataforma			

Python se destaca frente a Java y C# debido a su flexibilidad y versatilidad. Al ser un lenguaje multiparadigma, admite diferentes enfoques de programación funcional como el orientado a objetos, e imperativo, mientras que Java y C# se limitan principalmente al paradigma orientado a objetos. Esto le otorga a Python una gran ventaja para adaptarse a diversas necesidades de desarrollo. Además, su tipado dinámico facilita un flujo de trabajo más ágil, ya que no requiere declarar explícitamente el tipo de las variables, lo que acelera el proceso de codificación y simplifica el manejo de datos. Esta característica lo hace especialmente útil en entornos científicos y educativos, donde la simplicidad es clave.

En términos de propósito, Python se posiciona como líder en áreas emergentes como la inteligencia artificial, automatización y ciencia de datos, con marcos y librerías potentes como TensorFlow, Pandas y SciPy. Si bien Java y C# son sólidos en el desarrollo empresarial y web, Python ofrece un espectro más amplio de aplicaciones, desde la automatización de tareas hasta el desarrollo web ágil con marcos como Django. Con su sintaxis sencilla y su naturaleza interpreta

2.3.3. Frameworks de desarrollo web

2.3.3.1. Visual Studio

IDE de Visual Studio es una plataforma de lanzamiento creativa que puede utilizar para editar, depurar y compilar código y, finalmente, publicar una aplicación. Además del editor y depurador estándar que ofrecen la mayoría de IDE, Visual Studio incluye compiladores, herramientas de completado de código, diseñadores gráficos y muchas más funciones para mejorar el proceso de desarrollo de software (Microsoft, 2024).

2.3.3.2. Netbeans

NetBeans IDE es un entorno de desarrollo integrado, gratuito y de código abierto para el desarrollo de aplicaciones en los sistemas operativos Windows, Mac, Linux y Solaris.

IDE simplifica el desarrollo de aplicaciones web, corporativas, de escritorio y móviles que utilizan plataformas Java y HTML5. El IDE también ofrece soporte para el desarrollo de aplicaciones PHP y C/C++ (Oracle, 2024).

2.3.3.3. Eclipse

El IDE Eclipse fue desarrollado por la Eclipse Foundation bajo una licencia de software libre y es compatible con múltiples plataformas. Aunque está diseñado principalmente para trabajar con Java, también admite otros lenguajes de programación, como C++, JavaScript, Python o PHP, mediante la instalación de plugins. Algunas de sus principales características incluyen la autocompletación de código, un acceso rápido a los archivos del proyecto, la facilidad para realizar pruebas de código y la corrección automática de errores de sintaxis (Picie-Alcaraz, Olivares-Zepahua, López-Martínez, Romero-Torres, & Reyes-Hernández, 2020).

2.3.3.4. Comparativa de los frameworks de desarrollo web

Tabla 3. Comparativas de los frameworks de desarrollo web. Fuente: Autoría Propia.

Parámetros	Visual Studio	NetBeans	Eclipse
Lenguaje de programación	C#, VB.NET, C++, Python, TypeScript	Java, PHP, HTML, CSS, C++	Java, C, C++, Python, PHP, Ruby,
Licencia	Gratuita y comercial	Gratuita	Gratuita
Plataformas de soporte	Windows, macOS	Windows, macOS, Linux	Windows, macOS, Linux
Código abierto	No es completamente de código abierto	Completamente de código Abierto	Completamente de código abierto
Desarrollo WEB	Sí	Limitado	A través de plugins (Android Development Tools)
Herramientas de diseño de UI/UX	A través de plugins (Android Development Tools)	A través de plugins (Android Development Tools)	A través de plugins (Android Development Tools)
Rendimiento	Alto rendimiento	Moderado rendimiento	Moderado rendimiento
Documentación y recursos	Documentación extensa y recursos oficiales	Documentación completa y comunidad activa	Documentación extensa y soporte comunitario

Visual Studio ofrece un entorno de desarrollo integrado con soporte para múltiples lenguajes como C#, VB.NET, C++ y Python, lo que lo convierte en una opción versátil. Aunque

no es completamente de código abierto, su versión gratuita incluye herramientas poderosas para desarrollo web y aplicaciones móviles, y su rendimiento es significativamente superior al de NetBeans y Eclipse. Además, Visual Studio se destaca por su extensa documentación oficial, su integración con servicios como Azure y su depurador avanzado. Si bien NetBeans y Eclipse son opciones de código abierto y ofrecen compatibilidad con varios lenguajes, el amplio soporte y rendimiento de Visual Studio lo convierten en la mejor elección para proyectos más exigentes.

2.3.4 Base de datos

2.3.4.1. MySQL

MySQL es una de las bases de datos relacionales de software libre más populares del mundo. Debe esta popularidad a su uso generalizado en sitios web de comercio electrónico, redes sociales y aplicaciones, entre los que se incluyen Drupal, Joomla, Magento y WordPress. También es una parte esencial de la pila de aplicaciones web Linux-Apache-MySQL-PHP/Perl/Python (LAMP), de uso generalizado, que pone en marcha muchas aplicaciones, sitios web y servicios populares. Está clasificada como la segunda base de datos más popular del mundo por DB-Engines (Google Cloud, 2024).

2.3.4.2. PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional conocido por su confiabilidad, integridad de datos y compatibilidad multiplataforma (Windows, Linux, UNIX). Soporta una amplia variedad de tipos de datos y características avanzadas como vistas, claves foráneas, disparadores y procedimientos almacenados. Su desarrollo comenzó en la Universidad de Control de Concurrencia Multiversión (MVCC), que permite una eficiente gestión de múltiples operaciones de lectura y escritura. Es ampliamente utilizado en aplicaciones críticas por su estabilidad y flexibilidad (Tuapanta Daquilema & Montenegro Garrido, 2021).

2.3.4.3. Sql Server

SQL Server es un sistema de gestión de bases de datos relacionales desarrollado por Microsoft. Está diseñado para almacenar, gestionar y acceder a grandes volúmenes de datos de manera eficiente. SQL Server es conocido por su rendimiento, seguridad y compatibilidad con aplicaciones empresariales, siendo una opción popular para empresas que utilizan tecnologías de Microsoft. Ofrece soporte para procedimientos almacenados, transacciones, activadores y permite realizar análisis avanzados de datos, todo mientras asegura la integridad de la información y un fácil manejo mediante su interfaz gráfica y herramientas integradas (ROMERO ARMAS, 2022).

2.3.4.4. Comparación de las bases de datos

Tabla 4. Comparativa de bases de datos. Fuente: Autoría Propia.

CRITERIO	MySQL Server	PostgreSQL	SQL Server
Tipo de base de datos	Sistema de gestión de bases de datos relacional	Sistema de gestión de bases de datos objeto-relacional	Sistema de gestión de bases de datos relacional
Soporte	Windows, Linux, macOS	Windows, Linux, UNIX, macOS	Windows, Linux
Multiplataforma			
Licencia	Gratuita y comercial	Gratuita	Gratuita y comercial
Uso principal	Principalmente utilizado en el desarrollo de aplicaciones web y plataformas en línea, especialmente en entornos LAMP (Linux, Apache, MySQL, PHP), gracias a su	Ideal para aplicaciones empresariales que requieren un manejo avanzado de datos, análisis complejos y una alta integridad de datos, además de ser popular en entornos científicos y académicos.	Es ampliamente utilizado en entornos empresariales para el manejo de grandes volúmenes de datos y el desarrollo de aplicaciones críticas, con un fuerte enfoque en la Integración con el ecosistema de Microsoft y análisis avanzado.

facilidad de uso y rápida

implementación.

Escalabilidad

Adecuado para aplicaciones pequeñas y medianas, aunque con configuraciones avanzadas puede soportar grandes volúmenes de datos.

Altamente escalable, ideal para aplicaciones de gran tamaño y manejo de grandes conjuntos de datos.

Muy escalable, con soporte para grandes aplicaciones empresariales y entornos de alta demanda.

MySQL Server es ideal para aplicaciones web en entornos LAMP, pero tiene limitaciones en escalabilidad para grandes proyectos. SQL Server se destaca en entornos empresariales de alto rendimiento, pero su integración está ligada al ecosistema de Microsoft y requiere licencias comerciales.

En comparación, PostgreSQL es la mejor opción por su escalabilidad, soporte multiplataforma y manejo avanzado de datos. Es ideal para aplicaciones empresariales y científicas, con la ventaja de ser gratuito y de código abierto, ofreciendo flexibilidad y rendimiento sin los costos de licencias comerciales.

2.4. Plataformas de desarrollo FROND-END

2.4.1. Bulma

Bulma es un marco gratuito y de código abierto que proporciona componentes front end listos para usar que puedes combinar fácilmente para crear interfaces web responsivas (bulma, 2024).

2.4.2. Metro UI

Metro UI es un conjunto de herramientas gratuito, de código abierto y basado en HTML para desarrollar sitios web con HTML, CSS y JS. Con Metro UI, se puede crear un sitio de manera fácil y rápida, desde el prototipo hasta la producción. Metro UI incluye estilos generales, cuadrícula adaptable, diseños, tipografía, más de 100 componentes y rutinas de JavaScript, y más de 800 íconos integrados. (metroui, 2024)

2.4.3. Bootstrap

Es un framework de código abierto que facilita el desarrollo rápido de sitios web responsivos mediante el uso de componentes prediseñados y un sistema de grid adaptable a diferentes dispositivos (Hernández Berrones, 2020).

2.4.4. Matriz comparativa

Tabla 5. Comparativa de plataformas de desarrollo FROND-END. Fuente: Autoría Propia.

Criterio	Bulma	Interfaz de usuario de Metro	Bootstrap
Lenguaje	CSS (utiliza Flexbox)	CSS (inspirado en la interfaz de usuario de Windows)	HTML, CSS, JavaScript (CSS con Flexbox en las últimas versiones)
Ventajas	<ul style="list-style-type: none"> - Ligero y fácil de usar - Flexbox integrado - Totalmente modular (se puede usar solo lo necesario) 	<ul style="list-style-type: none"> - Diseño limpio y moderno - Inspirado en Windows UI - Soporte para dispositivos táctiles 	<ul style="list-style-type: none"> - Gran comunidad de soporte - Muchos componentes prediseñados - Alta personalización - Diseño responsivo
Desventajas	<ul style="list-style-type: none"> - Carece de soporte oficial para JavaScript - Menos populares, por lo que tiene menos recursos externos 	<ul style="list-style-type: none"> - Menos conocido, con una comunidad más pequeña - Limitado en cuanto a flexibilidad fuera del estilo Metro 	<ul style="list-style-type: none"> - Pesado comparado con otros frameworks - Necesita aprendizaje adicional para personalización profunda

Plataforma de soporte	Totalmente multiplataforma (compatible con la mayoría de navegadores modernos)	Multiplataforma (con buen soporte para dispositivos táctiles y navegadores modernos)	Multiplataforma, con soporte extenso para navegadores y dispositivos móviles.
Características	<ul style="list-style-type: none"> - Basado en Flexbox - Diseño limpio y minimalista - Modularidad - No incluye JavaScript nativo 	<ul style="list-style-type: none"> - Inspirado en la interfaz de Windows 10 - Incluye componentes como mosaicos, cuadrículas y notificaciones - Ligero 	<ul style="list-style-type: none"> - Sistema de grillas responsive - Amplia colección de componentes UI - Soporte para JavaScript y plugins adicionales - Extensible y personalizable

Bulma y Metro UI son frameworks ligeros y fáciles de usar, cada uno con características específicas. Bulma , basado en CSS y Flexbox, ofrece un diseño minimalista y modularidad, ideal para proyectos pequeños o medianos que no requieren interacción dinámica, ya que carece de soporte oficial para JavaScript. Metro UI, inspirado en la interfaz de usuario de Windows, destaca por su diseño limpio y moderno, con buen soporte para dispositivos táctiles, pero su flexibilidad es

Por otro lado, Bootstrap es un framework más robusto y flexible, lo que lo convierte en una opción popular para proyectos de cualquier tamaño. Aunque es más pesado que Bulma y Metro UI, su gran comunidad de soporte, amplia colección de componentes prediseñados y

capacidad de personalización lo hacen ideal para aplicaciones web complejas. Además, el soporte para JavaScript y complementos adicionales, junto con su sistema de grillas responsivo, permite crear interfaces profesionales y adaptables. A pesar de su curva de aprendizaje, Bootstrap es la opción más sólida para proyectos que requieren una solución completa y adaptable.

2.5. Plataforma de desarrollo BACK-END

2.5.1. Django

Django es un marco web Python de alto nivel que fomenta un desarrollo rápido y un diseño limpio y pragmático. Creado por desarrolladores experimentados, se encarga de gran parte de las molestias del desarrollo web, para que puedas concentrarte en escribir tu aplicación sin necesidad de reinventar la rueda. Es gratis y de código abierto (Viejo Pomata, 2020).

2.5.2. Laravel

Laravel es un **framework de desarrollo web** basado en **PHP** que ayuda a los desarrolladores a crear aplicaciones web de forma más rápida y organizada. Se destaca por ser un framework de código abierto que proporciona una gran cantidad de herramientas y características predefinidas, como la gestión de bases de datos, autenticación de usuarios, enrutamiento y plantillas (Avalos Rojas, Díaz Contreras, Valdez Ramos, & Javier Baeza, 2023).

2.5.3. Flask

Flask es un **framework web minimalista** para el lenguaje de programación **Python**, diseñado para facilitar el desarrollo de aplicaciones web ligeras y sencillas. A diferencia de otros frameworks más completos, Flask proporciona solo las herramientas básicas necesarias para construir una aplicación web, lo que le otorga mucha flexibilidad y libertad a los desarrolladores para añadir las características que necesiten (Arias Fisteus, 2023).

2.5.4. Matriz comparativa de los frameworks WEB Back end

Tabla 6. Comparativa de los frameworks back end. Fuente: Autoría Propia.

Criterio	Django	Laravel	Flask
Alcance	Ideal para aplicaciones web grandes y complejas.	Ideal para aplicaciones de tamaño medio a grande.	Adecuado para aplicaciones pequeñas o APIs.
Seguridad	Altamente seguro	Proporciona características de seguridad como autenticación, protección CSRF y cifrado.	Proporciona lo básico, pero requiere agregar extensiones para mayor seguridad.
Flexibilidad	Moderada.	Moderada.	Muy alta.
Rendimiento	Rápido, pero debido a sus numerosas características, puede ser más lento en comparación con frameworks más ligeros.	Buen rendimiento, pero la cantidad de funcionalidades integradas puede afectar la	Muy alto. Al ser minimalista, es más liviano y rápido que otros frameworks completos.

		velocidad si no se optimiza bien.	
Curva de Aprendizaje	Moderada.	Moderada.	Baja.
Características	ORM integrado, sistema de plantillas, autenticación, administración automática, gestión de sesiones.	Sistema de ORM (Eloquent), enrutamiento sencillo, migraciones, autenticación, plantillas Blade.	No incluye tantas herramientas. Los desarrolladores deben agregar extensiones para funcionalidades adicionales.
Ventajas	Desarrollo rápido con muchas herramientas integradas. Excelente seguridad y administración automática.	Gran comunidad, documentación extensa. Muchas características útiles para proyectos medianos a grandes.	Ligero y flexible. Fácil de usar para pequeños proyectos y APIs. Permite un control total del código.

Desventajas	Puede ser demasiado pesado para proyectos pequeños. Menos flexible en comparación con frameworks más ligeros.	Al ser muy completo, puede ser innecesariamente complejo para proyectos pequeños o sencillos.	Necesita muchas extensiones para agregar características avanzadas, lo que aumenta el trabajo manual.
-------------	---	---	---

Al comparar Flask, Laravel y Django, cada uno tiene fortalezas para diferentes tipos de proyectos. Flask es ideal para aplicaciones pequeñas o APIs debido a su flexibilidad y alto rendimiento, pero requiere más trabajo manual. Laravel ofrece muchas herramientas útiles para proyectos medianos a grandes, aunque puede ser innecesariamente complejo para aplicaciones pequeñas.

Finalmente, Django es la mejor opción para aplicaciones grandes y complejas, gracias a su robustez, seguridad avanzada y herramientas integradas que facilitan el desarrollo rápido y escalable. Por estas razones, Django es el framework elegido para proyectos de gran alcance.

2.6. Metodologías Ágiles

2.6.1. Definición de las metodologías ágiles

Estas metodologías son un marco de trabajo para gestionar el desarrollo de proyectos tecnológicos y, por lo tanto, de ingeniería de software. Las metodologías ágiles promueven procesos iterativos en las distintas fases del ciclo de vida del proyecto. Una característica clave es la comunicación continua y eficiente con el cliente, considerada una parte fundamental del proceso (Abuchar Porras, 2023).

2.6.2. Principios de las metodologías ágiles

Según (Nhuel Gómez, 2024) la metodología Agile se basa en 12 principios:

- **Conversaciones cara a cara:** El método más eficiente y efectivo para comunicar información al equipo.
- **Entrega de valor temprana:** Se da prioridad a satisfacer la necesidad del cliente mediante la entrega continua de valor.
- **Aceptamos el cambio:** Se aprovecha el cambio para proporcionar ventaja competitiva al cliente, incluso en etapas tardías del desarrollo.
- **Entregamos valor frecuentemente:** Preferentemente en el periodo más corto posible.

- **Trabajo en conjunto:** Responsables y colaboradores trabajan en conjunto durante todo el proyecto.
- **Individuos motivados:** Se les da el entorno y apoyo necesario para que confíen en la ejecución del trabajo.
- **Entrega de valor como unidad de medida:** El progreso se mide en términos de valor y funcionalidad.
- **Equipos auto-organizados:** Los mejores productos emergen de estos equipos.
- **Aporte de valor sostenible:** Mantener un ritmo constante de aportes de forma indefinida.
- **Atención continua a la excelencia y buen diseño:** Se debe reflexionar continuamente para mantener la calidad.
- **Ajustes regulares de rendimiento:** Reflexión regular del equipo para mejorar.
- **Simplicidad:** Maximizar el trabajo no realizado.

2.6.3. Tipos de metodologías

2.6.3.1. Scrum

Es un framework o marco de trabajo desarrollado bajo el manifiesto ágil, que ayuda a las personas, equipos y organizaciones a generar valor a través de soluciones adaptativas para problemas complejos. Scrum se basa en la teoría de control de procesos empíricos o empirismo y el pensamiento Lean. El empirismo asegura que el conocimiento proceda de la experiencia y de tomar decisiones basándose en evidencias. Mientras que el pensamiento Lean reduce el desperdicio y se enfoca en lo esencial. Scrum emplea un enfoque iterativo e incremental para optimizar la predictibilidad y el control del riesgo. Involucra a grupos de personas que colectivamente tienen todas las habilidades y experiencia para hacer el trabajo y compartir o adquirir dichas habilidades según sea necesario (Zalimben, 2022).

2.6.3.2. Kanban

Según Yápez & Armijos (2020) afirma que es un sistema de gestión visual derivado del método de producción Just in Time (JIT), creado por Toyota en la década de 1940. Se utiliza para optimizar el control de inventarios y la reducción de desperdicios. El sistema funciona a través de tarjetas conectadas a materiales, comunicando cuándo reponer o recargar suministros. Además, permite identificar cuellos de botella en la producción, mejorar la calidad del proceso y gestionar el flujo de materiales de acuerdo con la demanda del cliente. Las tarjetas Kanban también pueden integrarse en sistemas electrónicos y tableros, brindando flexibilidad, mejora continua y reducción de tareas innecesarias.

2.6.3.3. Programación Extrema (XP)

Según (JACOME CARRASCO, 2022) afirma que:

La metodología ágil **Programación Extrema (XP)** está orientada a ser más práctica y accesible para la gestión de aplicaciones en desarrollo. Facilita el trabajo al enfocarse en tareas que no requieren componentes complicados. Además, es ideal para que las empresas gestionen sus proyectos de manera eficiente, garantizando tanto calidad como simplicidad. XP también ayuda a realizar diversas actividades sin sobrecargar al equipo de desarrollo, lo que resulta en excelentes resultados y en la satisfacción del cliente.

2.6.3.4. Comparación entre metodologías tradicionales y ágiles

Tabla 7. Comparativas de las metodologías. Fuente: Autoría Propia.

Criterio	Scrum	Kanban	XP (Programación extrema)
Enfoque	Iterativo e incremental	Flujo continuo	Iterativo e incremental
Fases	<ol style="list-style-type: none"> 1. Planificación del Sprint 2. Ejecución del Sprint 3. Reuniones Diarias (Stand-up diario) 4. Revisión del Sprint 5. Retrospectiva del Sprint 	<ol style="list-style-type: none"> 1. Visualización del flujo de trabajo 2. Lím 3. Gestión del flujo 4. Revisión continua 	<ol style="list-style-type: none"> 1. Planificación 2. Diseño 3. Codificación 4. Pruebas 5. Retroalimentación

Adaptabilidad	Adaptabilidad moderada: Carreras de velocidad.	Alta Adaptabilidad	Alta Adaptabilidad
Estructura	Sprints con una duración fija (generalmente 2-4 semanas)	ablero Kanban para visualizar el flujo de trabajo	Iteraciones cortas (generalmente de 1-2 semanas)
Características	<ul style="list-style-type: none"> -Equipos autogestionados -Ciclos cortos y repetitivos (Sprints) - Feedback continuo -Reuniones periódicas y revisiones constantes 	<ul style="list-style-type: none"> - Flujo continuo - Visualización clara del trabajo mediante un tablero -No hay roles específicos en el equipo 	<ul style="list-style-type: none"> - Feedback rápido y constante - Foco en la simplicidad y calidad - Pruebas automatizadas constantes durante el desarrollo
Ventajas	<ul style="list-style-type: none"> - Gestión clara de proyectos - Transparencia y visibilidad del progreso - Mejora continua 	<ul style="list-style-type: none"> - Alta flexibilidad -Fácil de implementar y ajustar - Visualización clara de bloqueos y cuellos de botella 	<ul style="list-style-type: none"> - Mejora la calidad del código -Detecta problemas de manera temprana -Equipos altamente colaborativos

Desventajas	- Puede ser rígido si los Sprints no están bien definidos	- Sin estructura definida de tiempo	- No adecuado para equipos o proyectos con poca experiencia
	- La adaptación a cambios durante el Sprint puede ser limitada	- Puede ser menos eficaz si no se gestionan adecuadamente los límites del WIP	- Requiere alta disciplina en la implementación de prácticas XP

La metodología XP (Programación Extrema) es ideal para proyectos que requieren alta calidad y adaptabilidad. Su enfoque en iteraciones cortas de 1 a 2 semanas permite una retroalimentación constante y la detección temprana de problemas. Las prácticas clave como las pruebas automatizadas y la refactorización continua garantizan un código limpio y de alta calidad.

Además, XP fomenta la colaboración estrecha a través de la programación en pareja y prioriza la simplicidad, lo que permite entregar soluciones funcionales rápidamente. Esto la hace adecuada para proyectos que necesitan ajustarse constantemente a los cambios del cliente y del mercado.

CAPÍTULO III

3. MARCO METODOLÓGICO

3.1. Enfoque de la investigación

El enfoque de investigación que se utilizará es mixto, combinando tanto lo cuantitativo como lo cualitativo. El objetivo es evaluar el impacto de la implementación de un sistema de información en la eficiencia operativa, medido a través de indicadores cuantitativos, y analizar la satisfacción del cliente y las mejoras en los procesos mediante una entrevista y análisis. Este estudio empleará ambos métodos para obtener una visión integral del problema y sus posibles soluciones.

3.2. Nivel de la investigación

El nivel de investigación será descriptivo y explicativo, ya que se busca detallar la situación actual de la gestión inmobiliaria en Buris y Bienes Raíces, identificando las deficiencias del sistema manual y los problemas derivados de la falta de automatización, mientras que también se analizarán las causas de estas deficiencias y explicará cómo la implementación de un sistema de información basado en metodologías ágiles puede mejorar tanto la eficiencia operativa como la satisfacción del cliente.

3.3. Población

Este estudio se aplicará al dueño de Buris y Bienes Raíces como la población, no existe muestra.

3.4. Técnicas e instrumentos de recolección

Para la recolección de datos, se utilizará la técnica de entrevista semiestructurada. Este tipo de entrevista permite obtener información detallada y profunda, adaptándose a las

respuestas del entrevistado para explorar aspectos específicos. El instrumento principal será un cuestionario diseñado para el dueño de Buris y Bienes Raíces, el cual abordará temas relacionados con la gestión actual de las propiedades, los problemas operativos identificados, las expectativas de mejora y su visión sobre la implementación de un sistema de información. Esta entrevista permitirá captar tanto datos cualitativos como percepciones personales sobre la situación actual y el impacto esperado del nuevo sistema.

3.5. Tratamiento de la información

Una vez realizada la entrevista, se obtendrán los datos más relevantes, los cuales serán analizados de manera **rigurosa** y **minuciosa** para asegurar que todos los resultados obtenidos sean comprendidos y utilizados adecuadamente.

3.6. Resultados

Para determinar los requerimientos de la inmobiliaria, se llevará a cabo una entrevista, en la cual se realizará un análisis exhaustivo de las respuestas obtenidas, permitiendo una identificación precisa de las áreas de mejora.

3.6.1. Análisis de los resultados

Preguntas	Respuesta
1. ¿Utiliza actualmente algún tipo de software para la gestión de las propiedades?	Actualmente, Buris y Bienes Raíces no utiliza ningún software dedicado para la gestión de propiedades. Todas las operaciones se realizan de forma manual, lo que conlleva desafíos relacionados con la organización y eficiencia en la gestión de datos y propiedades.
2. ¿Considera que la implementación de un sistema de información es	Sí, un sistema de información es esencial para optimizar el manejo de las propiedades, automatizar procesos, mejorar la

- necesaria para mejorar la gestión inmobiliaria en su empresa?
- organización de los datos y agilizar el acceso a la información de las propiedades, lo cual impactará positivamente en la eficiencia operativa.
3. ¿Cuáles cree que serían los principales beneficios que un sistema de información podría aportar a la gestión de Buris y Bienes Raíces?
- Los principales beneficios incluyen la automatización de tareas repetitivas como la gestión de propiedades (agregar, editar o eliminar), la generación automática de reportes y la capacidad de acceder a la información en tiempo real. Además, el sistema reduciría el tiempo que se dedica a la búsqueda manual de propiedades, aumentando la eficiencia del equipo.
4. ¿Qué funcionalidades cree que podrían agilizar el proceso de gestión de propiedades (por ejemplo, agregar, editar, o eliminar) y cómo mejoraría la eficiencia general del equipo?
- Funcionalidades como la gestión centralizada de propiedades, la posibilidad de editar y agregar detalles de cada propiedad (precio, características, estado), la carga de imágenes, y la geolocalización en mapas interactivos serían clave para agilizar el trabajo del equipo y mejorar la precisión de la información gestionada
5. ¿Cómo espera que un sistema de información mejore la colaboración y la comunicación entre los diferentes miembros de su equipo?
- Un sistema de información centralizado permitirá a todos los miembros del equipo acceder a la misma base de datos actualizada en tiempo real, lo que mejorará la colaboración. Además, al tener un panel de control claro y accesible, será más fácil asignar y supervisar tareas, lo que mejorará la comunicación interna

- | | |
|---|---|
| 6. ¿Qué tipo de filtros u opciones de búsqueda le gustaría tener en el sistema para encontrar propiedades rápidamente? | El sistema debe permitir la búsqueda por filtros como ubicación (provincia, cantón, parroquia), precio, tipo de propiedad (casa, apartamento, terreno), número de dormitorios y estado de la propiedad (venta, arriendo). Estos filtros facilitarían la búsqueda rápida de propiedades específicas. |
| 7. ¿Le gustaría que el sistema incluyera un mapa interactivo para mostrar la ubicación exacta de las propiedades? ¿Qué beneficios espera de integrar la geolocalización de las propiedades? | Sí, la integración de un mapa interactivo es fundamental. Permitiría a los clientes visualizar la ubicación exacta de las propiedades, lo que mejoraría la experiencia de usuario. Además, el equipo podría destacar propiedades cerca de puntos de interés, como escuelas o transporte público, lo que agilizaría la toma de decisiones. |
| 8. ¿Cómo debería ser la interfaz del sistema de información para facilitar su uso por parte del equipo de trabajo? | La interfaz debe ser intuitiva, con menús claros y accesibles. El diseño debe facilitar la navegación y permitir el acceso rápido a la información de las propiedades. Además, debe ser adaptable a dispositivos móviles y escritorio, lo que mejorará la accesibilidad y uso diario. |
| 9. ¿Qué tan importante es para usted que el sistema permita la carga y gestión de múltiples imágenes de las propiedades, organizadas de manera sencilla? | Es muy importante que el sistema permita la carga de múltiples imágenes y videos, organizadas de manera sencilla, para que los clientes puedan visualizar las propiedades de manera clara. Además, la función de arrastrar y soltar imágenes facilitaría la gestión multimedia por parte del equipo. |
-

3.6.2. Análisis general de la entrevista

Durante la entrevista se identificó que **Buris y Bienes Raíces** no utiliza un software especializado para la gestión de propiedades, lo que ha generado dificultades en la organización y eficiencia operativa. Ante esta situación, el entrevistado resaltó la necesidad de implementar un sistema de información que permita automatizar procesos clave, como la actualización y gestión de propiedades, la carga de imágenes y la visualización de información en tiempo real.

Se destacó la importancia de contar con una interfaz intuitiva que facilite el trabajo diario del equipo y la posibilidad de integrar un mapa interactivo que ofrezca la ubicación exacta de las propiedades, mejorando tanto la experiencia del equipo como la de los clientes. Asimismo, la capacidad de filtrar propiedades por criterios como ubicación y precio permitiría una búsqueda rápida y eficiente, optimizando el tiempo de gestión.

La implementación de un sistema web es fundamental, ya que ofrece una accesibilidad mayor al permitir que tanto el equipo como los clientes accedan a la información desde cualquier lugar y dispositivo. Un sistema web no solo facilita la colaboración entre los miembros del equipo, quienes podrán gestionar propiedades de forma centralizada, sino que también mejora la visibilidad de las propiedades en línea, permitiendo que los clientes accedan fácilmente a la información y compartan propiedades a través de redes sociales. Esta accesibilidad y flexibilidad que brinda un sistema web asegura que la inmobiliaria pueda operar de manera más eficiente y competitiva en el mercado digital actual.

CAPÍTULO IV

4. PROPUESTA

4.1. Título de la Propuesta

“DESARROLLO DE UN SISTEMA DE INFORMACIÓN PARA LA GESTIÓN INMOBILIARIA DE BURIS Y BIENES RAÍCES, BASADO EN METODOLOGÍAS ÁGILES.”

4.2. Presentación

En este capítulo se presenta una propuesta para el desarrollo de un sistema de información destinado a optimizar la gestión de las propiedades de Buris y Bienes Raíces. La propuesta se fundamenta en el uso de metodologías ágiles, específicamente XP (Programación Extrema), para garantizar un desarrollo altamente flexible y adaptable a las necesidades cambiantes del negocio inmobiliario. XP se caracteriza por su enfoque en la calidad del código, la simplicidad en las soluciones y la retroalimentación continua, lo que permitirá automatizar procesos clave, mejorar la colaboración interna y optimizar la gestión y visualización de propiedades, tanto para el equipo de trabajo como para los clientes finales.

4.3. Justificación

La implementación de un sistema de información es crucial para mejorar la eficiencia y competitividad de Buris y Bienes Raíces. Actualmente, la falta de un software especializado conlleva dificultades en la gestión manual de las propiedades, lo que resulta en tiempos de respuesta lentos y una menor capacidad de respuesta ante los clientes. Este sistema, basado en metodologías ágiles, permitirá automatizar procesos clave como la gestión de propiedades, la

carga de imágenes, la visualización en mapas interactivos y la búsqueda avanzada de propiedades, ofreciendo una solución tecnológica moderna que optimice tanto la operativa interna como la experiencia del cliente.

Además, el desarrollo ágil garantiza que el sistema pueda adaptarse rápidamente a los cambios y mejoras necesarias, proporcionando flexibilidad durante su implementación. Un sistema web como el propuesto no solo facilitará la accesibilidad y gestión remota, sino que también permitirá una mayor visibilidad de las propiedades, mejorando la presencia digital de la empresa.

4.4. Descripción de la propuesta

4.4.1. ¿Qué se va realizar?

Se desarrollará un sistema de información web para Buris y Bienes Raíces, con el objetivo de automatizar y centralizar la gestión de propiedades.

4.4.2. ¿Cómo se va a realizar?

El sistema será desarrollado utilizando metodologías ágiles, principalmente **XP (Programación Extrema)**, para garantizar un proceso de desarrollo iterativo, con un fuerte enfoque en la calidad del código y la retroalimentación constante. La implementación técnica estará basada en el framework Django, junto con HTML, CSS y JavaScript para el frontend, y Google Maps para la integración del mapa interactivo. El equipo de desarrollo trabajará en iteraciones cortas, permitiendo ajustes continuos basados en la retroalimentación del cliente y asegurando que las funcionalidades críticas estén disponibles rápidamente

4.4.3. ¿Para qué se va a realizar?

El sistema se va a desarrollar para resolver los problemas de gestión manual que actualmente enfrenta Buris y Bienes Raíces, como la falta de organización centralizada y la

dificultad en la actualización y visualización de propiedades. Este sistema mejorará la eficiencia operativa al reducir los tiempos de gestión, aumentar la capacidad de respuesta ante los clientes y facilitar la colaboración interna. Además, permitirá a la empresa optimizar su presencia digital, atrayendo más clientes a través de la búsqueda online de propiedades y su visibilidad en redes sociales, mejorando así su competitividad en el mercado inmobiliario.

4.5. Desarrollo del proyecto

En este apartado se presentará una propuesta basada en la metodología ágil XP (Programación Extrema), cuyo objetivo principal es asegurar un desarrollo iterativo y colaborativo del sistema de información para la gestión inmobiliaria de Buris y Bienes Raíces. A través de iteraciones cortas y retroalimentación continua, se busca maximizar la calidad del código y asegurar una rápida adaptabilidad a los cambios y necesidades del negocio. Esta propuesta técnica detalla las fases clave de la implementación, desde la planificación hasta la codificación y pruebas, priorizando la simplicidad y la eficiencia operativa para lograr un sistema robusto y flexible. Las fases de la metodología XP son las siguientes:

- I. Planificación
- II. Diseño
- III. Desarrollo
- IV. Pruebas
- V. Retroalimentación

4.5.1. Planificación

De acuerdo con la metodología XP, se llevaron a cabo varias reuniones con el dueño, lo que permitió recopilar información clave para el desarrollo de la aplicación web. A través de estas reuniones, se definieron múltiples historias de usuario, junto con sus tareas

correspondientes, lo que facilitó la priorización de las funcionalidades más relevantes y permitió al equipo trabajar de manera efectiva en su implementación.

4.5.1.1. Establecimiento de las Historias de Usuario

Después de varias reuniones con el dueño, se definieron los requerimientos para el sistema informático. A continuación, se detallarán las historias de usuario en la siguiente tabla:

Tabla 8. establecimiento de historias de usuario. Fuente: Autoría Propia.

ID Historia	Funcionalidad	Descripción
HU01	Autenticación y Autorización	"Como administrador, quiero iniciar sesión de manera segura para gestionar propiedades en el panel de administración."
HU02	Gestión de Propiedades	"Como administrador, quiero agregar, editar y eliminar propiedades para tener un control completo del inventario de inmuebles."
HU03	Visualización de Propiedades	"Como usuario, quiero buscar propiedades mediante un mapa interactivo para encontrar

		fácilmente la ubicación que me interesa."
HU04	Búsqueda y Filtros	"Como usuario, quiero filtrar propiedades por ubicación y características para ver solo las opciones que se ajusten a mis necesidades."
HU06	Integración con Redes Sociales	"Como usuario, quiero compartir propiedades en redes sociales para mostrar inmuebles interesantes a mis contactos."

4.5.1.2. Priorización de las historias de usuario

Con base en los requerimientos funcionales y las necesidades del cliente, las funcionalidades se priorizan de la siguiente manera:

Tabla 9. Priorización de las historias de usuario. Fuente: Autoría Propia.

ID Historia	Funcionalidad	Prioridad
HU01	Autenticación y Autorización	Alta
HU02	Gestión de Propiedades	Alta
HU03	Visualización de Propiedades	Media

HU04	Búsqueda y Filtros	Alta
HU05	Integración con Redes Sociales	Baja

4.5.1.3. Estimación de tiempo y esfuerzo

La siguiente tabla presenta un desglose de historias de usuario para un sistema de gestión de propiedades, detallando las funcionalidades clave, los puntos de historia que miden la complejidad de cada tarea y las iteraciones necesarias para su implementación. Este formato permite organizar el desarrollo y priorizar las actividades en función de su esfuerzo y complejidad.

Tabla 10. Estimación de tiempo y esfuerzo. Fuente: Autoría propia.

ID Historia	Funcionalidad	Punto de historia	Iteraciones
HU01	Autenticación y Autorización	5	1
HU02	Gestión de propiedades: Agregar	8	2
	Gestión de Propiedades: Editar y eliminar	8	1
HU03	Visualización de Propiedades	6	1
HU04	Búsqueda y Filtros	4	1
HU05	Integración con Redes Sociales	3	1

4.5.1.4. Selección de Historias para la Iteración (Iteration Planning)

La subfase de Selección de Historias para la Iteración en la metodología XP permite planificar la implementación de las funcionalidades en ciclos de desarrollo cortos. En este proyecto, se distribuyeron las historias de usuario clave en seis iteraciones de dos semanas, priorizando las funcionalidades más críticas para garantizar la entrega en tres meses.

Tabla 11. Iteration planning. Fuente: Autoría Propia.

Historias de Usuario	Iteración	Duración
Seleccionadas		
HU01: Autenticación y Autorización	Iteración 1	Semana 1 - 2
HU02: Gestión de Propiedades (Agregar)	Iteración 2	Semana 3 - 4
HU02: Gestión de Propiedades (Editar y Eliminar)	Iteración 3	Semana 5 - 6
HU03: Visualización de Propiedades	Iteración 4	Semana 7 - 8
HU04: Búsqueda y Filtros	Iteración 5	Semana 9 - 10
HU05: Integración con Redes Sociales	Iteración 6	Semana 11 - 12

4.5.2. Diseño

La fase de diseño en la metodología XP busca desarrollar una arquitectura simple y adaptable, enfocada en resolver los requerimientos clave sin agregar complejidades innecesarias. En este proyecto, se llevaron a cabo varias actividades de diseño que permitieron construir un sistema de gestión inmobiliaria eficiente, escalable y fácil de mantener.

4.5.2.1. Diseño simple

El diseño del sistema se mantuvo simple para asegurar que sea flexible y fácil de modificar en el futuro. Esto incluyó la definición de estructuras de datos claras para las entidades principales, como propiedades, usuarios, y localidades. Las funcionalidades se diseñaron para resolver los problemas actuales de manera directa, evitando la sobre ingeniería.

Ilustración 5. Diseño Simple. Fuente: Autoría Propia.

Entidades	Descripción
Propiedad	Se modeló como una entidad con atributos clave como título, descripción, tipo, estado (en venta, arriendo), precio, características (dormitorios, baños, plantas), y ubicación (provincia, cantón, parroquia).
Usuario	Se definieron los roles de administrador y usuario, con niveles de acceso diferenciados para la gestión de propiedades y búsqueda de inmuebles.

4.5.2.2. Uso de Tarjetas CRC (Class-Responsibility-Collaborator)

Para garantizar un diseño claro y organizado, se utilizaron **tarjetas CRC** para definir las clases principales y sus responsabilidades. Estas tarjetas ayudaron a identificar las colaboraciones entre las clases y facilitaron el diseño orientado a objetos.

Tabla 12. Tarjetas CRC. Fuente: Autoría Propia.

Clases	Responsabilidad	Colaboradores
Propiedad	Gestionar los datos de las propiedades, incluyendo agregar, editar y eliminar propiedades.	Colabora con la clase Usuario para permitir que los administradores gestionen propiedades y con la clase Localidades para obtener la ubicación de las propiedades.
Usuario	Manejar la autenticación y autorización de los usuarios del sistema.	Interactúa con la clase Propiedad para gestionar el inventario de inmuebles y con la clase Autenticación para controlar el acceso.
Localidades	Gestionar provincias, cantones y parroquias para especificar la ubicación de las propiedades.	Colabora con la clase Propiedad para vincular cada propiedad con una localidad específica.

4.5.3. Desarrollo

4.5.3.1. Base de datos

En el diseño lógico, se definen las tablas, campos y relaciones necesarias para estructurar los datos, como las propiedades, usuarios y transacciones. Por otro lado, el diseño físico describe cómo se implementará esta estructura en el sistema de gestión de bases de datos, asegurando un buen rendimiento y almacenamiento eficiente. Este modelo es fundamental para que el sistema funcione correctamente y permita a la inmobiliaria Buris y Bienes Raíces gestionar sus procesos de manera óptima y segura.

Diseño Físico

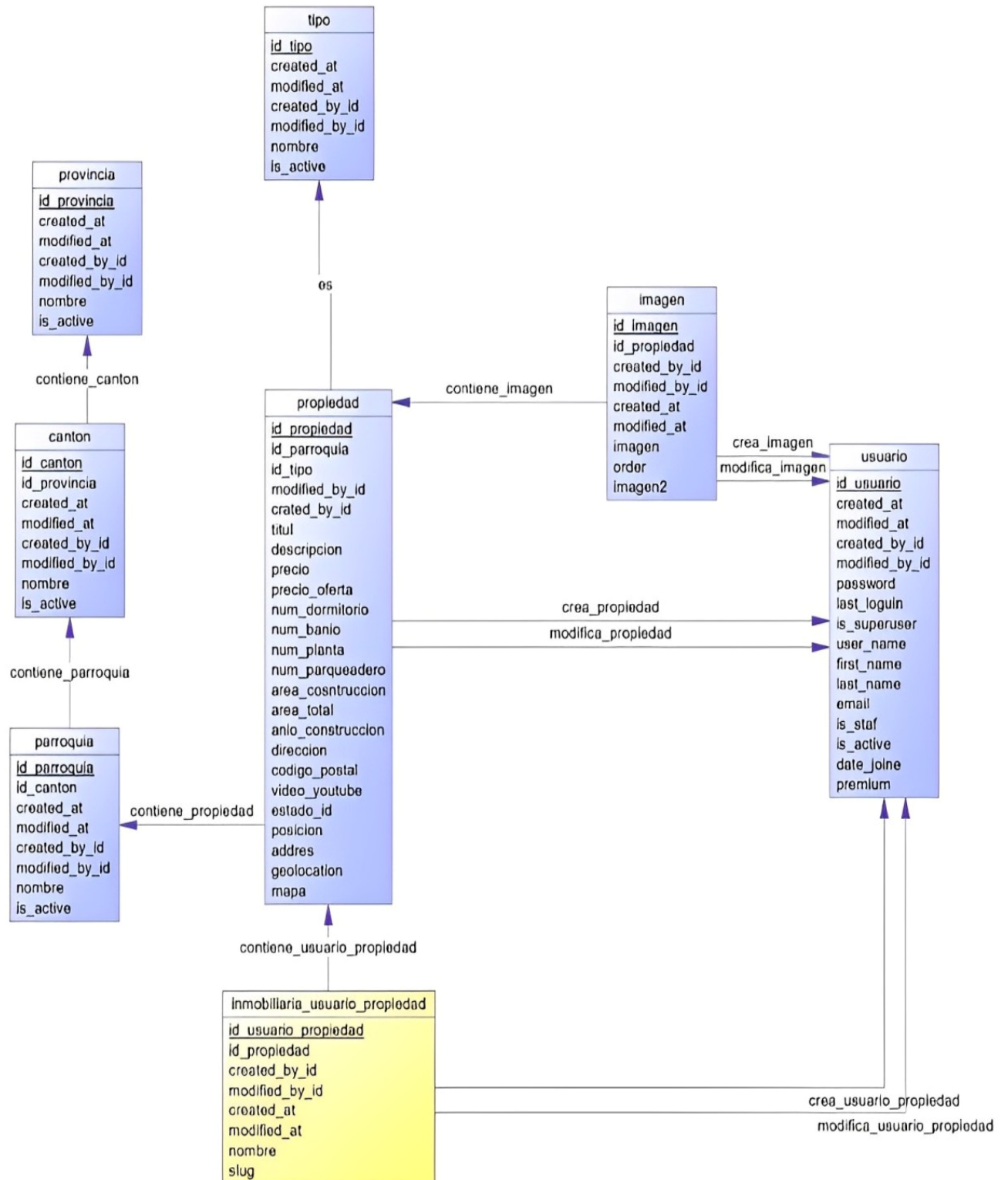


Ilustración 6. Diseño Físico de la base de datos. Fuente: Autoría propia.

Diseño Lógico

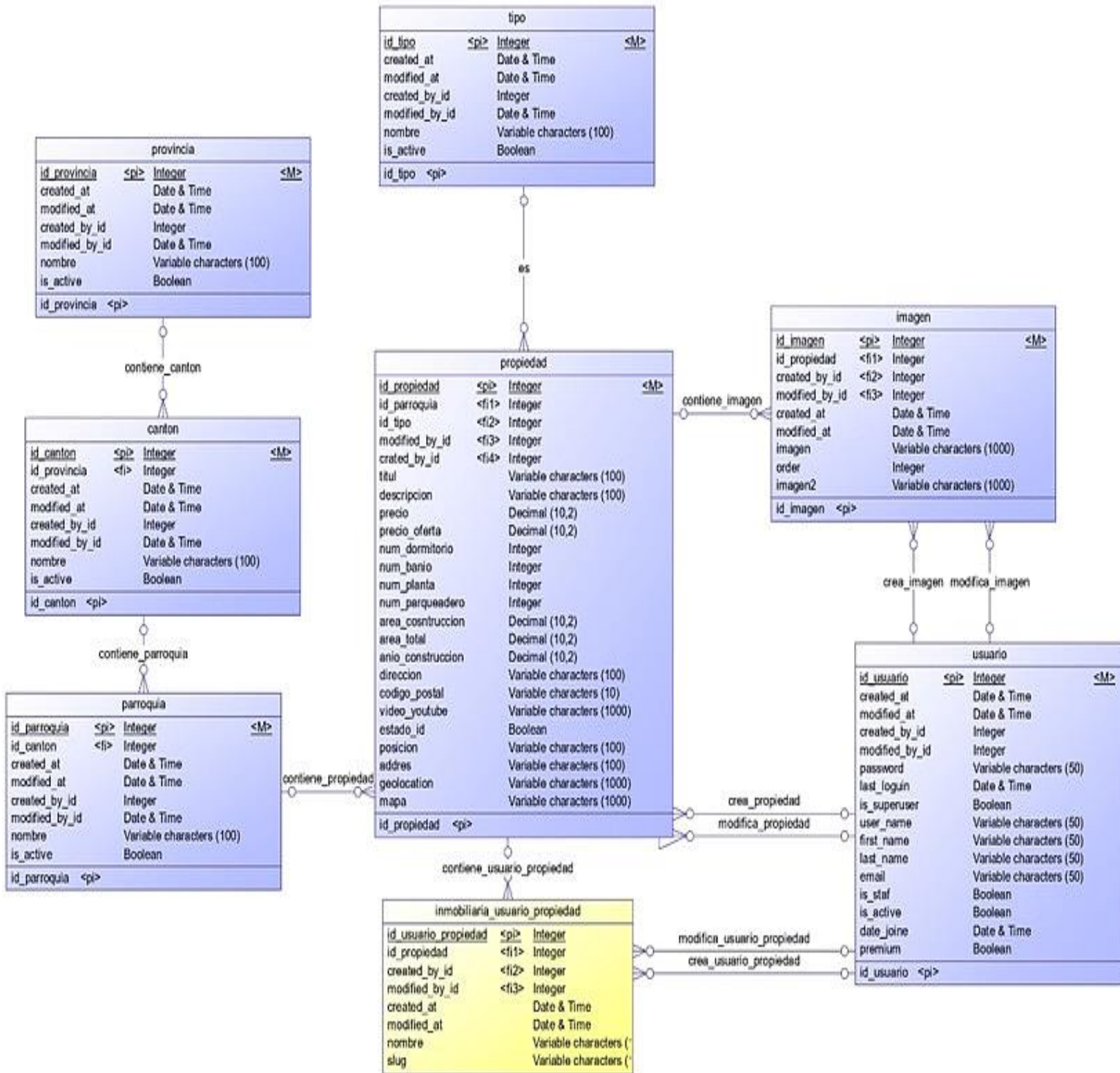


Ilustración 7. Diseño lógico de la base de datos. Fuente: Autoría propia.

4.5.3.2. Codificación

Una de las fases clave del ciclo de desarrollo en XP es la codificación, en la cual el diseño previamente establecido se convierte en implementaciones funcionales. Durante este proceso, se han creado las clases que forman la base del sistema, cada una con sus atributos definidos y sus respectivos tipos de datos. Asimismo, se han implementado los constructores necesarios para inicializar dichos atributos de manera eficiente. Esta organización no solo permite la creación y manipulación de objetos, sino que también establece una base sólida para el desarrollo de las funcionalidades del sistema, siguiendo los principios de simplicidad y flexibilidad característicos de XP.

Clase propiedad.

```
class Propiedad(ModeloBase):
    asesor = models.ForeignKey(Asesor, on_delete=models.PROTECT)
    titulo = models.CharField('Título', max_length=150)
    descripcion = tinymce_models.HTMLField('Descripción')
    tipo = models.ForeignKey(TipoPropiedad, on_delete=models.PROTECT)
    estado = models.ForeignKey(EstadoPropiedad, on_delete=models.PROTECT)
    precio = models.DecimalField('Precio', max_digits=10, decimal_places=2)
    precio_oferta = models.DecimalField('Precio de Oferta', max_digits=10, decimal_places=2, null=True, blank=True)

    # Características
    num_dormitorios = models.PositiveIntegerField('Número de Dormitorios')
    num_banios = models.PositiveIntegerField('Número de Baños')
    num_plantas = models.PositiveIntegerField('Número de Plantas')
    num_parqueaderos = models.PositiveIntegerField('Número de Parqueaderos')
    area_construccion = models.DecimalField('Área de Construcción', max_digits=10, decimal_places=2)
    area_total = models.DecimalField('Área Total', max_digits=10, decimal_places=2)
    anio_construccion = models.PositiveIntegerField('Año de Construcción')

    # Ubicación
    direccion = models.CharField('Dirección', max_length=150)
    codigo_postal = models.CharField('Código Postal', max_length=10)
    provincia = models.ForeignKey(Provincia, on_delete=models.PROTECT)
    canton = models.ForeignKey(Canton, on_delete=models.PROTECT)
    parroquia = models.ForeignKey(Parroquia, on_delete=models.PROTECT)
    mapa = GoogleMapsField('Mapa', max_length=300)

    # Otros
    video_youtube = models.URLField('Video YouTube', null=True, blank=True)
    destacado = models.BooleanField('Destacado', default=False)

    objects = PropiedadManager()

    def __str__(self):
        return self.titulo
```

```

# Otros
video_youtube = models.URLField('Video YouTube', null=True, blank=True)
destacado = models.BooleanField('Destacado', default=False)

objects = PropiedadManager()

def __str__(self):
    return self.titulo

class Meta:
    verbose_name = 'Propiedad'
    verbose_name_plural = 'Propiedades'
    ordering = ['-created_at']

@cached_property
def mi_precio_final(self):
    if self.precio_oferta:
        return self.precio_oferta
    return self.precio

@cached_property
def mis_cuotas(self):
    return round(self.mi_precio_final / 12, 0)

@cached_property
def mi_entrada(self):
    return round(self.mi_precio_final /12, 0)

```

```

@cached_property
def mi_imagen_principal(self):
    imagenes = PropiedadImagen.objects.filter(propiedad=self)
    imagen_principal = imagenes.filter(principal=True).first()
    if imagen_principal:
        return imagen_principal.imagen.url
    else:
        imagen = imagenes.first()
        if imagen:
            return imagen.imagen.url
    return None

@cached_property
def mis_coordenadas(self):
    return str(self.mapa).split(',')

@cached_property
def mi_latitud(self):
    return self.mis_coordenadas[0].replace(',', '.')

@cached_property
def mi_longitud(self):
    return self.mis_coordenadas[1].replace(',', '.')

def mis_imagenes(self):
    return PropiedadImagen.objects.filter(propiedad=self).order_by('orden')

def mis_propiedades_similares(self):
    return Propiedad.objects.filter(tipo=self.tipo).exclude(id=self.id).order_by('?')

```

Ilustración 8. Clase Propiedad: Fuente: Autoría Propia.

Clase Usuario

```

▼ class CustomUser(AbstractUser):
    premium = models.BooleanField(default=False)

▼ def get_photo_user(self):
    if self.socialaccount_set.exists():
        social_account = self.socialaccount_set.first()
        return social_account.get_avatar_url()
    else:
        return None

▼ def get_nombre_completo(self):
    if self.first_name and self.last_name:
        return self.first_name + ' ' + self.last_name
    else:
        return self.username

def mis_correos(self):
    return EmailAddress.objects.filter(user=self)

def mis_social_accounts(self):
    return SocialAccount.objects.filter(user=self)

▼ def mi_email(self):
    correos = EmailAddress.objects.filter(user=self, verified=True)
    if correos.exists():
        return correos.first().email
    else:
        if self.email:
            return self.email
        else:
            correos = EmailAddress.objects.filter(user=self)
            if correos.exists():
                return correos.first().email
            else:
                return None

```

```

@staticmethod
def flexbox_query(query):
    return CustomUser.objects.filter(Q(first_name__search=query) | Q(first_name__icontains=query) |
                                     Q(last_name__icontains=query) | Q(email__icontains=query) |
                                     Q(username__icontains=query))

```

Ilustración 9. Clase Usuario. Fuente: Autoría Propia.

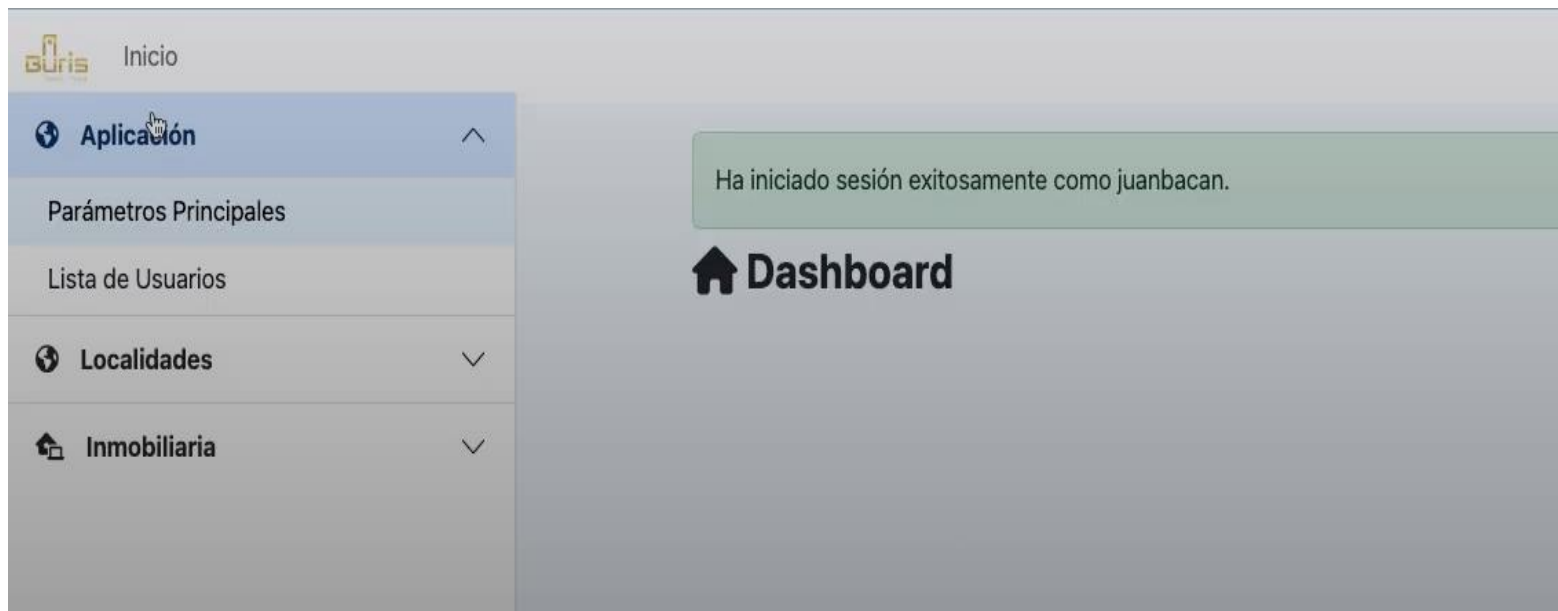
4.5.3.2. Interfaces del Sistema de Información

- **Login**

La interfaz de el login del sistema de información incluye el logotipo de Buris y bienes Raíces, y un cuadro con la información que se debe de ingresar: El usuario y la contraseña, si no posee una cuenta existe la opción de registrarse como se muestra en la siguiente ilustración.

Ilustración 10. Login del usuario. Fuente: Autoría propia.

- Interfaz Principal



- Una vez ingresado al sistema de información se muestra la siguiente pantalla con las opciones que posee:

Ilustración 11. Interfaz Principal. Fuente: Autoría Propia.

Se tiene opciones de Aplicación, Localización e inmobiliaria.

- **Pantalla de aplicación**

Parámetros principales

La interfaz de administración de una aplicación web con un panel lateral que contiene opciones como "Parámetros Principales", "Localidades" e "Inmobiliaria". En la sección principal, se pueden ver campos para configurar la URL del sitio, el título corto, favicon, SEO Meta Tags, notificaciones Webpush y enlaces de contacto como email, Facebook e Instagram.

Parámetros Principales de la Aplicación

Información Principal

URL del Sitio

Favicon

Actualmente: [aplicacion_web/ba710452-6ac7-4458-bb00-e0c1fb0cb95f.png](#)

Limpiar

Modificar:

Título Corto del Sitio

Logo

Actualmente: [aplicacion_web/ba710452-6ac7-4458-bb00-e0c1fb0cb95f_Xh81uDA.png](#)

Limpiar

Modificar:

SEO Meta Tags

Title

Meta Title

Meta Description

Imagen Redes Sociales

Imagen Redes Sociales

Seleccionar archivo Sin archivos seleccionados

Notificaciones Webpush

Logo webpush
Actualmente: [aplicacion_web/ba710452-6ac7-4458-bb00-e0c1fb0cb95f_scNmZHm.png](#) Limpiar
Modificar:
Seleccionar archivo Sin archivos seleccionados

Group name webpush
buris bienes raices

Contacto

Email de Contacto
buris@gmail.com

Instagram
<https://www.instagram.com/buris/>

Facebook
<https://www.facebook.com/buris>

Tiktok
<https://www.tiktok.com/@buris.com>

Ilustración 12. Pantalla de administración. Fuente: Autoría Propia.

Lista de usuarios

En la siguiente ilustración muestra el listado de los usuarios que ya existen en el sistema de información mostrando los detalles como los nombres, apellidos, su e-mail. Incluye botones de acción que permite realizar tareas a cada uno de ellos como editar la información

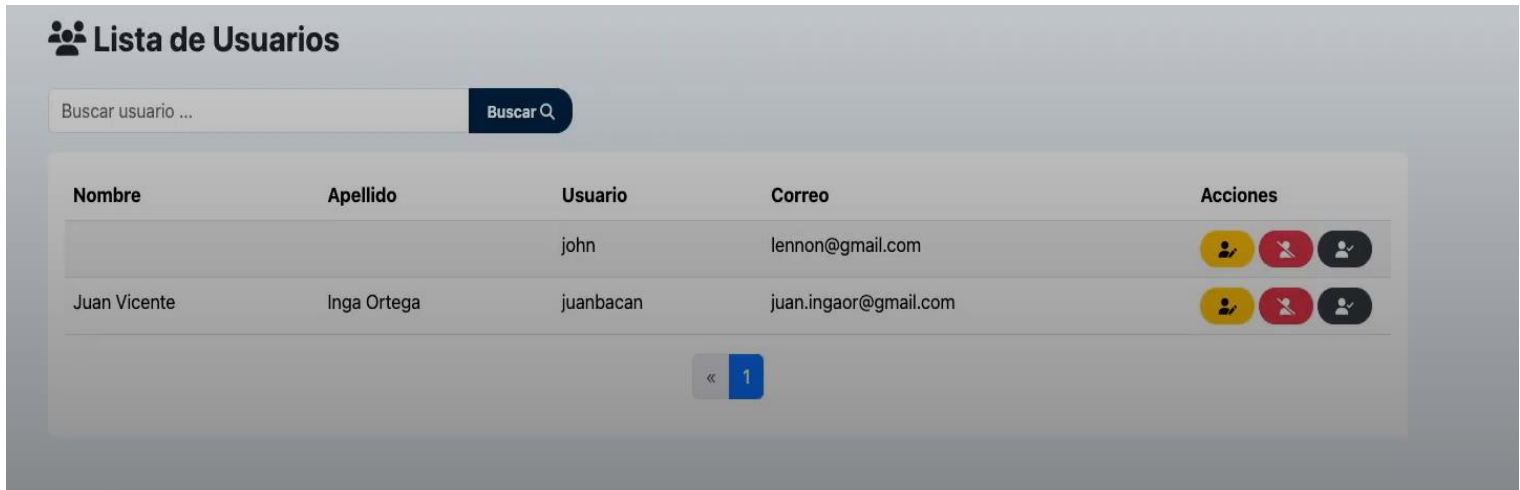


Ilustración 13. Lista de usuarios. Fuente: Autoría Propia.

- Pantalla de localidades

Se cargaron todas las provincias, cantones, parroquias

Código Provincia	Código Cantón	Código Parroquia	Provincia	Cantón	Parroquia
3	2	50	CAÑAR	BIBLIAN	BIBLIAN
3	2	54	CAÑAR	BIBLIAN	JERUSALEN
3	2	51	CAÑAR	BIBLIAN	NAZON (CAB. EN PAMPA DE DOMINGUEZ)
3	2	52	CAÑAR	BIBLIAN	SAN FRANCISCO DE SAGEO
3	2	53	CAÑAR	BIBLIAN	TURUPAMBA
3	6	50	CAÑAR	DELEG	DELEG
3	6	51	CAÑAR	DELEG	SOLANO
3	4	50	CAÑAR	LA TRONCAL	LA TRONCAL
3	4	51	CAÑAR	LA TRONCAL	MANUEL J. CALLE
3	4	52	CAÑAR	LA TRONCAL	PANCHO NEGRO
3	1	50	CAÑAR	AZOGUES	AZOGUES
3	1	51	CAÑAR	AZOGUES	COJITAMBO

Ilustración 14. Pantalla de localidades. Fuente: Autoría Propia.

- Pantalla Inmobiliaria

En la sección inmobiliaria, el usuario puede gestionar las propiedades que se encuentran en el sistema. Se presenta una vista rápida de cada propiedad, mostrando la imagen, estado (en venta, construcción o arriendo) la opción de marcar como destacada y también agregar el precio, se puede realizar acciones como editar o eliminar, así como acciones de agregar propiedad lo que facilita la actualización del inventario inmobiliario.

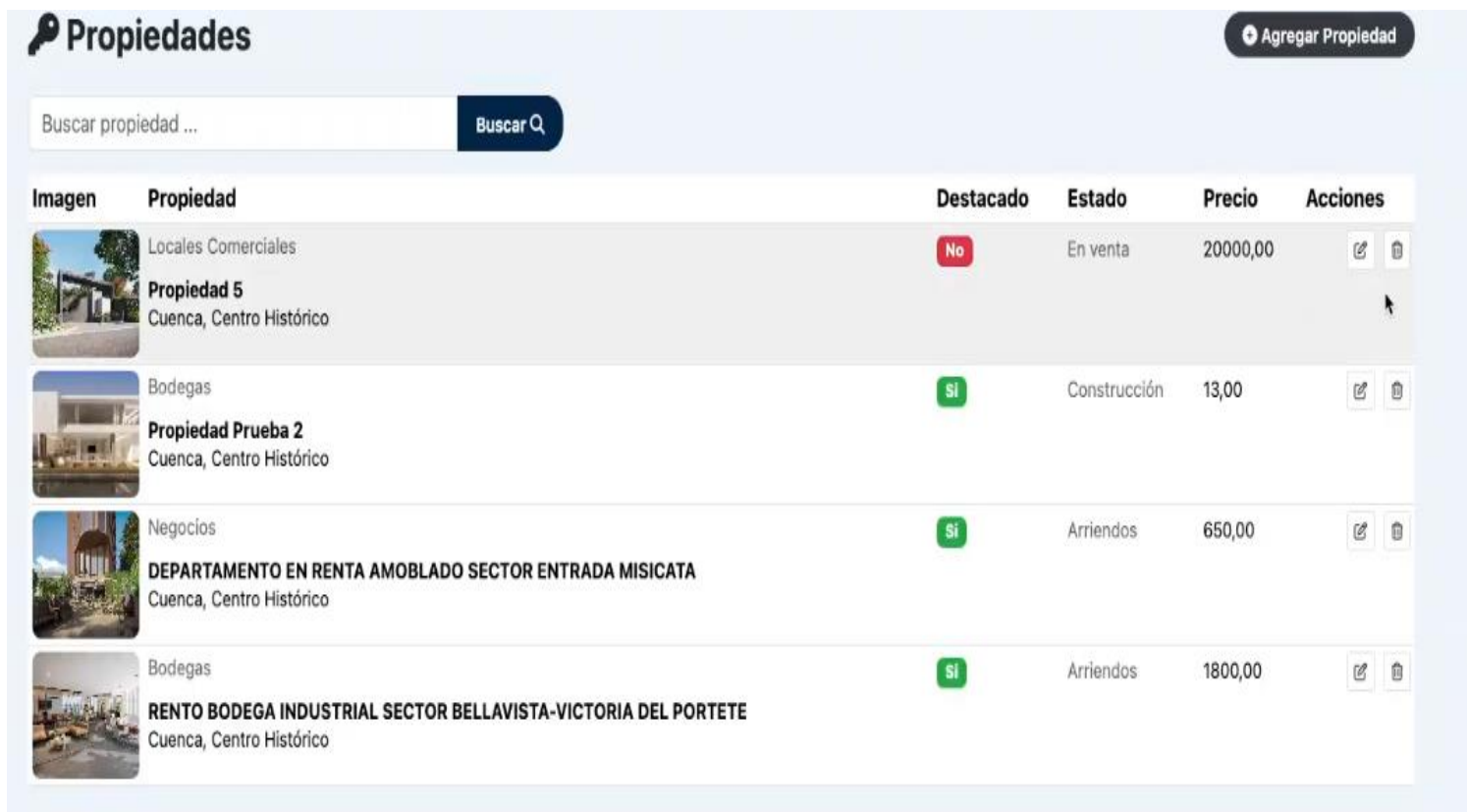














Imagen	Propiedad	Destacado	Estado	Precio	Acciones
	Locales Comerciales Propiedad 5 Cuenca, Centro Histórico	No	En venta	20000,00	 
	Bodegas Propiedad Prueba 2 Cuenca, Centro Histórico	Si	Construcción	13,00	 
	Negocios DEPARTAMENTO EN RENTA AMOBLADO SECTOR ENTRADA MISICATA Cuenca, Centro Histórico	Si	Arriendos	650,00	 
	Bodegas RENTO BODEGA INDUSTRIAL SECTOR BELLAVISTA-VICTORIA DEL PORTETE Cuenca, Centro Histórico	Si	Arriendos	1800,00	 

Ilustración 15. Pantalla de propiedades. Fuente: Autoría Propia.

Al momento que se agrega o se edita la propiedad se puede cambiar las características como el precio, el número de dormitorios que existe, la dirección en donde se encuentra la propiedad ya que cuenta con el acceso a un mapa interactivo que permite arrastrar el marcador para seleccionar con precisión la ubicación exacta de la propiedad.

4.5.4. Pruebas

Se presenta la fase de pruebas en formato de matrices, detallando las diferentes pruebas realizadas en el proyecto de Desarrollo de un Sistema de Información para la gestión inmobiliaria de Buris y Bienes Raíces.

4.5.4.1. Pruebas Unitarias

Las pruebas unitarias validan el comportamiento de los componentes individuales del sistema, como clases o métodos, asegurando que cada funcionalidad clave opere correctamente de forma aislada y sin errores.

Tabla 13. Matriz de pruebas unitarias. Fuente: Autoría Propia.

ID Historia	Funcionalidad	Descripción de la Prueba Unitaria	Resultado Esperado	Estado de la Prueba
HU01	Autenticación y Autorización	Verificar que el administrador pueda iniciar sesión con credenciales correctas.	Inicio de sesión exitoso	Aprobada
HU02	Gestión de Propiedades (Agregar)	Validar que una nueva propiedad pueda ser agregada con todos los datos necesarios.	Propiedad agregada correctamente	Aprobada

HU02	Gestión de Propiedades (Editar)	Verificar que se pueda editar una propiedad existente y que los cambios se reflejen correctamente.	Propiedad editada correctamente	Aprobada
HU02	Gestión de Propiedades (Eliminar)	Asegurar que una propiedad puede ser eliminada y que ya no aparece en el inventario.	Propiedad eliminada	Aprobada
HU03	Visualización de Propiedades	Comprobar que las propiedades se visualicen en el mapa según su ubicación.	Propiedades visualizadas correctamente en el mapa	Aprobada

4.5.4.2. Pruebas de Integración

Las pruebas de integración verifican que los diferentes módulos del sistema trabajen juntos sin problemas. Evalúan la correcta interacción entre las funcionalidades para garantizar que el sistema funcione como un todo.

Tabla 14. Pruebas de integración. Fuente: Autoría propia.

ID Historia	Funcionalidad	Descripción de la Prueba de Integración	Resultado Esperado	Estado de la Prueba
-------------	---------------	---	--------------------	---------------------

HU01 y HU02	Autenticación y Gestión de Propiedades	Verificar que solo los usuarios autenticados puedan agregar, editar o eliminar propiedades.	Acceso a funcionalidades restringido correctamente	Aprobada
HU03	Visualización de Propiedades	Probar la interacción entre la clase Propiedad y Localidades para asegurar que las propiedades se muestren en el mapa.	Propiedades ubicadas correctamente	Aprobada
HU04	Búsqueda y Filtros	Validar que los filtros de búsqueda (ubicación, precio, características) interactúan correctamente con la visualización.	Filtros aplicados correctamente	Aprobada

4.5.4.3. Pruebas de Aceptación

Las pruebas de aceptación están diseñadas para verificar que el sistema cumpla con los requisitos del cliente. Estas pruebas se basan en escenarios reales, evaluando si el sistema satisface las expectativas del usuario final.

Tabla 15. Pruebas de Aceptación. Fuente: Autoría Propia.

ID Historia	Funcionalidad	Descripción de la Prueba de Aceptación	Criterios de Aceptación	Estado de la Prueba
HU01	Autenticación y Autorización	Validar que el sistema permita a los administradores gestionar propiedades tras la autenticación.	Administrador autenticado puede gestionar propiedades	Aprobada
HU05	Integración con Redes Sociales	Comprobar que los usuarios puedan compartir propiedades en redes sociales (Facebook, Twitter).	Propiedad compartida correctamente en redes	Aprobada
HU04	Búsqueda y Filtros	Asegurar que los usuarios puedan aplicar filtros y buscar propiedades por características específicas.	Búsqueda funcional con filtros aplicados	Aprobada

4.5.4.4. Pruebas Automatizadas

Las pruebas automatizadas permiten ejecutar pruebas de manera continua durante el desarrollo. Ayudan a detectar errores de forma temprana y aseguran que las nuevas funcionalidades no rompan el código existente.

Tabla 16. Pruebas Automatizadas. Fuente: Autoría Propia.

Tipo de Prueba	Descripción de la Prueba	Frecuencia	Resultado Esperado	Estado de la Prueba
Pruebas Unitarias	Se ejecutan pruebas unitarias automáticas después de cada cambio en el código.	Continuamente	Todas las pruebas pasan correctamente	Aprobada
Pruebas de Integración	Se ejecutan pruebas de integración después de cada integración de código.	Después de cada commit	Interacciones entre módulos funcionan correctamente	Aprobada

Pruebas de Aceptación	Pruebas automáticas para verificar la funcionalidad desde la perspectiva del usuario.	Semanalmente	El sistema cumple con los requisitos del cliente	Aprobada
-----------------------	---	--------------	--	----------

4.5.5. Retroalimentación

La retroalimentación es un componente clave que se integra de manera continua a lo largo del desarrollo del proyecto. Para el Sistema de Información para la gestión inmobiliaria de Buris y Bienes Raíces, la fase de retroalimentación y ajustes fue esencial para asegurar que el software cumpliera con las expectativas del cliente y mejorara con cada iteración.

4.5.5.1. Retroalimentación del cliente

Esta fase consiste en reuniones periódicas con el cliente para revisar las funcionalidades entregadas. El cliente proporciona comentarios y sugerencias, ayudando a mejorar y ajustar el sistema según sus necesidades.

Tabla 17. Retroalimentación del cliente. Fuente: Autoría Propia.

Iteración	Funcionalidad Evaluada	Comentarios del Cliente	Acciones a Tomar	Prioridad
Iteración 2	Gestión de Propiedades	El cliente solicitó mejorar la interfaz de administración para agregar más detalles de cada propiedad.	Añadir campos adicionales para descripción, imágenes, etc.	Alta
Iteración 4	Visualización de Propiedades en el Mapa	Mejorar la claridad del mapa interactivo, haciendo que la información de las propiedades sea más visible.	Ajustar el diseño del mapa y mejorar los marcadores.	Media

Iteración 5	Filtros de Búsqueda	El cliente pidió que los filtros de búsqueda sean más rápidos y eficientes.	Optimizar los filtros de búsqueda por características y ubicación.	Alta
Iteración 6	Integración con Redes Sociales	Solicitud de incluir más opciones de redes sociales para compartir propiedades.	Incluir más plataformas de redes sociales (LinkedIn, WhatsApp).	Baja

4.5.5.2. Tabla de Ajustes Continuos

Basándose en la retroalimentación del cliente, se realizan modificaciones o mejoras en las funcionalidades del sistema. Este proceso asegura que el software esté alineado con las expectativas y requisitos del cliente.

Ilustración 18. Ajuste continuo. Fuente: Autoría Propia.

Iteración	Funcionalidad Ajustada	Descripción del Ajuste	Resultado del Ajuste	Estado
Iteración 2	Gestión de Propiedades	Añadir campos adicionales (descripción, características, imágenes) en la interfaz de administración.	Mejor visualización y gestión de propiedades.	Completado
Iteración 4	Visualización de Propiedades en el Mapa	Mejorar el diseño del mapa interactivo y los marcadores de propiedades.	Propiedades más visibles y fáciles de identificar.	Completado
Iteración 5	Filtros de Búsqueda	Optimización del rendimiento de los filtros para mejorar la velocidad de respuesta.	Filtros más rápidos y efectivos.	Completado

Iteración 6	Integración con Redes Sociales	Añadir la opción de compartir propiedades en LinkedIn y WhatsApp.	Mayor visibilidad de las propiedades en redes sociales.	En Proceso
-------------	--------------------------------	---	---	------------

4.5.5.3. Calidad de Código

Durante la fase de retroalimentación, se aprovecha para refactorizar y optimizar el código. El objetivo es mantener el código limpio, eficiente y fácil de modificar en el futuro.

Tabla 18. Calidad del código. Fuente: Autoría Propia.

Iteración	Funcionalidad Refactorizada	Descripción de la Refactorización	Resultado de la Refactorización	Estado
Iteración 3	Gestión de Propiedades	Reducción de duplicaciones de código en la gestión de propiedades (agregar/editar/eliminar).	Código más limpio y rendimiento optimizado.	Completado
Iteración 4	Visualización de Propiedades	Optimización del código de carga de propiedades en el mapa interactivo.	Carga más rápida y eficiente de las propiedades.	Completado

Conclusiones

El análisis de fuentes bibliográficas resultó crucial para establecer una base sólida en el desarrollo del sistema de información para Buris y Bienes Raíces. La investigación permitió identificar las mejores prácticas en la implementación de sistemas de gestión inmobiliaria, contribuyendo significativamente a la estructuración del sistema y su adaptación a las necesidades específicas del cliente. Las metodologías ágiles, en particular, se demostraron altamente efectivas para gestionar el proyecto de manera dinámica y flexible.

Entre las metodologías ágiles evaluadas, Extreme Programming (XP) fue seleccionada como la más adecuada para el desarrollo de este sistema. XP proporcionó una estructura que facilitó la entrega rápida y frecuente de funcionalidades, permitiendo recibir retroalimentación constante del cliente y hacer ajustes de manera eficiente. Esta metodología aseguró que el sistema evolucionara de acuerdo con las expectativas del cliente, manteniendo un enfoque en la calidad del código y la mejora continua.

El sistema de información desarrollado cumplió con éxito los objetivos planteados, proporcionando una herramienta robusta para la gestión inmobiliaria. Se logró integrar eficazmente la visualización de propiedades mediante mapas interactivos, imágenes y videos, optimizando la experiencia del usuario en la búsqueda de inmuebles. Además, la capacidad de aplicar filtros avanzados y la integración con redes sociales añadieron un valor significativo al sistema, mejorando tanto su funcionalidad como su usabilidad.

El proyecto alcanzó todos los objetivos de manera exitosa. El uso de metodologías ágiles, específicamente XP, permitió un desarrollo flexible y adaptativo que garantizó la entrega de un sistema funcional, eficiente y alineado con los requerimientos del cliente.

La metodología ágil y la retroalimentación continua fueron claves para asegurar un producto final de alta calidad y satisfacción del cliente.

Recomendaciones

- Establecer un plan de mantenimiento continuo para el sistema de información, asegurando que se mantenga actualizado y optimizado según las necesidades cambiantes del mercado inmobiliario.
- Capacitar a todo el personal en el uso adecuado del sistema de gestión de propiedades, lo que permitirá aprovechar al máximo todas las funcionalidades del software y mejorar la eficiencia en los procesos.
- Explorar la implementación de nuevas funcionalidades, como la integración de un sistema de mensajería interna para facilitar la comunicación entre compradores y vendedores, así como el desarrollo de un módulo de pagos en línea.
- Optimizar el sistema para su uso en dispositivos móviles, mejorando el diseño responsivo y el tiempo de carga, lo cual garantizará una mejor experiencia de usuario desde smartphones y tabletas.
- Ampliar las opciones de integración con más redes sociales y plataformas de mensajería, permitiendo que las propiedades puedan compartirse en más canales como LinkedIn, WhatsApp y otras redes relevantes.
- Implementar un sistema de retroalimentación constante para medir la satisfacción de los usuarios finales, lo que permitirá identificar áreas de mejora y ajustar las funcionalidades de acuerdo con las necesidades del mercado.

Bibliografía

- Arguello Pazmiño, A. M., Llumiguano Poma, M. E., Gavilánez Cárdenas, C. V., & Torres Ordoñez, L. H. (2020). *ADMINISTRACIÓN DE EMPRESAS ELEMENTOS BÁSICOS*. Guayaquil: ASOCIACIÓN LATINOAMERICANA DE. Obtenido de <https://fs.unm.edu/Administracion-de-Empresas.pdf>
- Bernardi, S., & Dranca, L. (2020). *Sistemas de informacion para la direccion: Un enfoque guiado por un caso de estudio*. Catalunya: Centro Universitario de la Defensa.
- Bonilla Adriano, A. D. (01 de 01 de 2021). *dspace.unach.edu.ec*. Obtenido de [dspace.unach.edu.ec: http://dspace.unach.edu.ec/bitstream/51000/7636/1/8.-%20documento%20de%20tesis.pdf](http://dspace.unach.edu.ec/bitstream/51000/7636/1/8.-%20documento%20de%20tesis.pdf)
- Carvajal Camacho, E. E. (01 de 01 de 2021). *epositorio.uasb.edu.ec*. Obtenido de [epositorio.uasb.edu.ec: https://repositorio.uasb.edu.ec/bitstream/10644/8189/1/T3574-MAE-Carvajal-Modelo.pdf](https://repositorio.uasb.edu.ec/bitstream/10644/8189/1/T3574-MAE-Carvajal-Modelo.pdf)
- HERNÁNDEZ BERRONES, E. A. (2020). *dspace.esPOCH.edu.ec*. Obtenido de [dspace.esPOCH.edu.ec: http://dspace.esPOCH.edu.ec/bitstream/123456789/13654/1/18T00807.pdf](http://dspace.esPOCH.edu.ec/bitstream/123456789/13654/1/18T00807.pdf)
- Picie-Alcaraz, I., Olivares-Zepahua, B. A., López-Martínez, I., Romero-Torres, C., & Reyes-Hernández, L. Á. (2020). Herramienta para la Enseñanza de la Programación usando Elementos Gráficos. *Risti*, 50-62.

Abuchar Porras, A. (2023). *Metodologías ágiles para el desarrollo de software*. Bogotá: UD Editorial.

Arévalo-Cordovilla, F. E., Ordoñez-Sigcho, I. B., Peñaherrera-Larenas, M. F., & Suárez-Matamoros, V. J. (2020). Importancia de la seguridad de los sistemas de información frente al abuso, error. *Ciencias de la computación*, 2477-8818.

Arias Fisteus, J. (01 de 01 de 2023). *ocw.uc3m.es*. Obtenido de *ocw.uc3m.es*:
https://ocw.uc3m.es/pluginfile.php/7176/mod_page/content/6/flask.pdf

Avalos Rojas, R., Díaz Contreras, S., Valdez Ramos, M., & Javier Baeza, T. (2023). Revisión de Framework Laravel y su aplicación en sistema web de mesa de ayuda. *INNOVACIÓN Y DESARROLLO TECNOLÓGICO REVISTA DIGITAL*, 12.

Avilés, C. A. (13 de 06 de 2023). *uvadoc.uva.es*. Obtenido de *uvadoc.uva.es*:
<https://uvadoc.uva.es/bitstream/handle/10324/61315/TFG-E-1739.pdf?sequence=1&isAllowed=y>

Ballesteros Reyes, M. P., & Borrero Escalante, P. M. (01 de 01 de 2023). *repository.unilibre.edu.co*. Obtenido de *repository.unilibre.edu.co*:
<https://repository.unilibre.edu.co/bitstream/handle/10901/25155/ARTICULO%20PAPER.pdf?sequence=4&isAllowed=y>

Barahona Blanco, C. (01 de 01 de 2020). *oa.upm.es*. Obtenido de *oa.upm.es*:
https://oa.upm.es/63315/1/TFG_CELIA_BARAHONA_BLANCO.pdf

bulma. (01 de 01 de 2024). *bulma.io*. Obtenido de *bulma.io*: <https://bulma.io/>

CEPAL. (18 de 11 de 2020). *repositorio.cepal.org*. Obtenido de repositorio.cepal.org:

<https://repositorio.cepal.org/server/api/core/bitstreams/44b29e87-7423-4b28-976f-ed78face527e/content>

Criollo Chanchicocha, V. D. (01 de 01 de 2020). *repositorio.utc.edu.ec*. Obtenido de

repositorio.utc.edu.ec: <https://repositorio.utc.edu.ec/items/fed6ed8f-38a0-4db7-b9f0-ff021f03b2b5>

docusign. (25 de 07 de 2023). *www.docusign.com*. Obtenido de www.docusign.com:

<https://www.docusign.com/es-mx/blog/sistema-de-gestion-documental-dms>

Esquivel Oviedo, J. C. (01 de 01 de 2020). *tesis.pucp.edu.pe*. Obtenido de

tesis.pucp.edu.pe:

https://tesis.pucp.edu.pe/repositorio/bitstream/handle/20.500.12404/19149/ESQ_UIVEL_OVIEDO_JUAN_CARLOS%20%281%29.pdf?sequence=1&isAllowed=y

Gómez Vieites, Á., & Suárez Rey, C. (2020). *Sistema de información. Herramientas*

Prácticas para la gestión. México: RA-MA Editorial.

Google Cloud. (12 de 08 de 2024). *cloud.google.com*. Obtenido de cloud.google.com:

<https://cloud.google.com/mysql?hl=es>

Guerra Poalasin, D. O. (01 de 01 de 2021). *repositorio.uta.edu.ec*. Obtenido de

repositorio.uta.edu.ec:

<https://repositorio.uta.edu.ec/bitstream/123456789/32673/1/t1791si.pdf>

Gutierrez Garay, J. A. (19 de 11 de 2020). *niixer.com*. Obtenido de niixer.com:

<https://niixer.com/index.php/2020/11/19/c-sharp-uno-de-los-lenguajes-de-programacion-mas-utilizados-actualmente/>

hoteltechreport. (01 de 01 de 2024). *hoteltechreport.com*. Obtenido de

hoteltechreport.com:

<https://hoteltechreport.com/es/news/cmms#:~:text=Hippo%20CMMS%20es%20una%20de,de%20la%20orden%20de%20trabajo>.

Ibáñez Chuquín , M. B., & Ulcuango Ulcuango , G. M. (01 de 01 de 2020).

repositorio.utn.edu.ec/bitstream. Obtenido de *repositorio.utn.edu.ec/bitstream*:

<https://repositorio.utn.edu.ec/bitstream/123456789/13339/2/02%20ICA%201870%20TRABAJO%20DE%20GRADO.pdf>

ibm. (01 de 04 de 2024). *www.ibm.com*. Obtenido de *www.ibm.com*:

<https://www.ibm.com/mx-es/topics/what-is-a-cmms>

IBM. (01 de 05 de 2024). *www.ibm.com*. Obtenido de *www.ibm.com*:

<https://www.ibm.com/mx-es/topics/transaction-processing-system>

Infante Pilco, B. I. (01 de 01 de 2023). *dspace.esPOCH.edu.ec*. Obtenido de

dspace.esPOCH.edu.ec:

<http://dspace.esPOCH.edu.ec/bitstream/123456789/19201/1/15T00863.pdf>

Izurieta Castelo, M. I., Miño Cascante, G. E., & Vallejo Sánchez, D. P. (01 de 01 de

2021). *cimogsys.esPOCH.edu.ec*. Obtenido de *cimogsys.esPOCH.edu.ec*:

http://cimogsys.esPOCH.edu.ec/direccion-publicaciones/public/docs/books/2023-11-17-214604-economia_basica_para_organizaciones.pdf

JACOME CARRASCO, H. (01 de 01 de 2022). *dspace.utb.edu.ec*. Obtenido de

dspace.utb.edu.ec: <http://dspace.utb.edu.ec/bitstream/handle/49000/13044/E-UTB-FAFI-SIST-000386.pdf?sequence=1&isAllowed=y>

Jaime Adrián, R. R., & Mishell Mellisa, M. S. (01 de 01 de 2023).

repositorio.upao.edu.pe. Obtenido de *repositorio.upao.edu.pe*:

https://repositorio.upao.edu.pe/bitstream/handle/20.500.12759/32051/REP_JAIME.ROJAS_MISHELL.MEDINA_USO.DEL.SISTEMA.ERP.pdf?sequence=1&isAllowed=y

java. (17 de 08 de 2024). *www.java.com*. Obtenido de *www.java.com*:

https://www.java.com/en/download/help/whatis_java.html

learn.microsoft.com. (29 de 01 de 2024). Obtenido de *learn.microsoft.com*:

<https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/overview>

managementsolutions.com. (01 de 01 de 2020). *www.managementsolutions.com*. Obtenido de *www.managementsolutions.com*:

<https://www.managementsolutions.com/sites/default/files/publicaciones/esp/auto-machine-learning-sp.pdf>

matterport.com. (07 de 05 de 2024). *matterport.com*. Obtenido de *matterport.com*:

<https://matterport.com/es/about-us>

metroui.org.ua. (01 de 01 de 2024). *metroui.org.ua*. Obtenido de *metroui.org.ua*:

<https://metroui.org.ua/index.html>

m-files.com. (16 de 05 de 2024). *www.m-files.com*. Obtenido de *www.m-files.com*:

<https://www.m-files.com/es/>

Microsoft. (11 de 06 de 2024). *visualstudio.microsoft.com*. Obtenido de

visualstudio.microsoft.com: <https://visualstudio.microsoft.com/es/>

Morales Lugo, H. A., Figueroa Millán, P. E., Farías Mendoza, N., & Chávez Valdez, R.

E. (2020). SISTEMA DE INTELIGENCIA DE NEGOCIOS PARA SOPORTE

DE DECISIONES EN LA COMERCIALIZACIÓN DE PLANTAS
ORNAMENTALES. *3C Tecnología*, 2254 – 4143.

Muevecela Bermeo, R. C. (01 de 01 de 2020). *dspace.ups.edu.ec*. Obtenido de

dspace.ups.edu.ec:

<https://dspace.ups.edu.ec/bitstream/123456789/19635/1/UPS-CT008919.pdf>

Nhuel Gómez, J. (16 de 07 de 2024). *fcefyn.unc.edu.ar*. Obtenido de *fcefyn.unc.edu.ar*:

https://fcefyn.unc.edu.ar/documents/2998/2._Introducci%C3%B3n_a_las_Metodolog%C3%ADas_%C3%81giles.pdf

Nicolau Medina, J., & Gozalbes Ballester, M. (s.f.). *Calidad, Productividad y*

competividad. La salida de la crisis. Madrid: Cambridge University Press.

Oracle. (29 de 05 de 2024). *www.oracle.com*. Obtenido de *www.oracle.com*:

<https://www.oracle.com/es/tools/technologies/netbeans-ide.html>

python. (9 de 05 de 2024). *www.python.org*. Obtenido de *www.python.org*:

<https://www.python.org/>

ROMERO ARMAS, M. (23 de 07 de 2022). *bibdigital.epn.edu.ec*. Obtenido de

bibdigital.epn.edu.ec:

<https://bibdigital.epn.edu.ec/bitstream/15000/23429/1/CD%2012858.pdf>

Rosa. (08 de 2019). *rosamarfil.es*. Obtenido de *rosamarfil.es*:

<https://rosamarfil.es/tutoriales/otras-entradas/que-es-java/>

sapbusinessone.company. (30 de 07 de 2024). Obtenido de

sapbusinessone.company2024:

https://sapbusinessone.company/ecuador/?utm_source=google&utm_medium=cpc&utm_campaign=landing&gad_source=1&gclid=CjwKCAjwooq3BhB3EiwA

YqYoEsV6sPQsgYzF_LGwEIxtMIDtGFPL7SIAjZl6Z5Q8DLAGeyZOfvG2ER
oCvdMQAvD_BwE

start.docuware. (01 de 01 de 2024). *start.docuware.com*. Obtenido de

start.docuware.com: <https://start.docuware.com/es/acerca-de-docuware>

Tuapanta Daquilema, A. D., & Montenegro Garrido, J. I. (01 de 01 de 2021).

dspace.unach.edu.ec. Obtenido de *dspace.unach.edu.ec*:

<http://dspace.unach.edu.ec/bitstream/51000/7540/2/7.%20Proyecto%20de%20Investigaci%3%b3n%20An%3%a1lisis%20de%20Bases%20de%20datos%20relacionales%20y%20no%20relacionales%20aplicado%20al%20problema%20de%20la%20ruta%20m%3%a1s%20corta..pdf>

Viejo Pomata, D. (01 de 01 de 2020). *core.ac.uk*. Obtenido de *core.ac.uk*:

<https://core.ac.uk/download/pdf/286369586.pdf>

Yépez Llerena , E. D., & Armijos Guillen, K. F. (01 de 01 de 2020).

dspace.unach.edu.ec. Obtenido de *dspace.unach.edu.ec*:

<http://dspace.unach.edu.ec/bitstream/51000/6457/1/APLICACI%3%93N%20DE%20LA%20METODOLOG%3%8DA%20KANBAN%20EN%20EL%20DESARROLLO%20DEL%20SOFTWARE%20PARA%20GENERACI%3%93N.pdf>

Zalimben, S. (2022). *Una pequeña Guía de Scrum*. Creative Commons Share-Alike

Attribution Commons.

Anexos



Manual de usuario

Desarrollo de un Sistema de información para la gestión inmobiliaria de Buris y Bienes Raíces, basado en metodologías ágiles.

UNIVERSIDAD CATÓLICA DE CUENCA EXTENSIÓN CAÑAR

2024

Manual de usuario

Desarrollo de un Sistema de información para la gestión inmobiliaria de Buris y Bienes Raíces, basado en metodologías ágiles.

Si en el futuro se implementa el sistema de información, será necesario adquirir un servicio de alojamiento web para acceder a la página e iniciar sesión de forma directa.

Pantalla de Inicio de Sesión

Al momento de ingresar al sistema de información se muestra la siguiente interfaz, en la cual se va a iniciar la sesión de usuario:

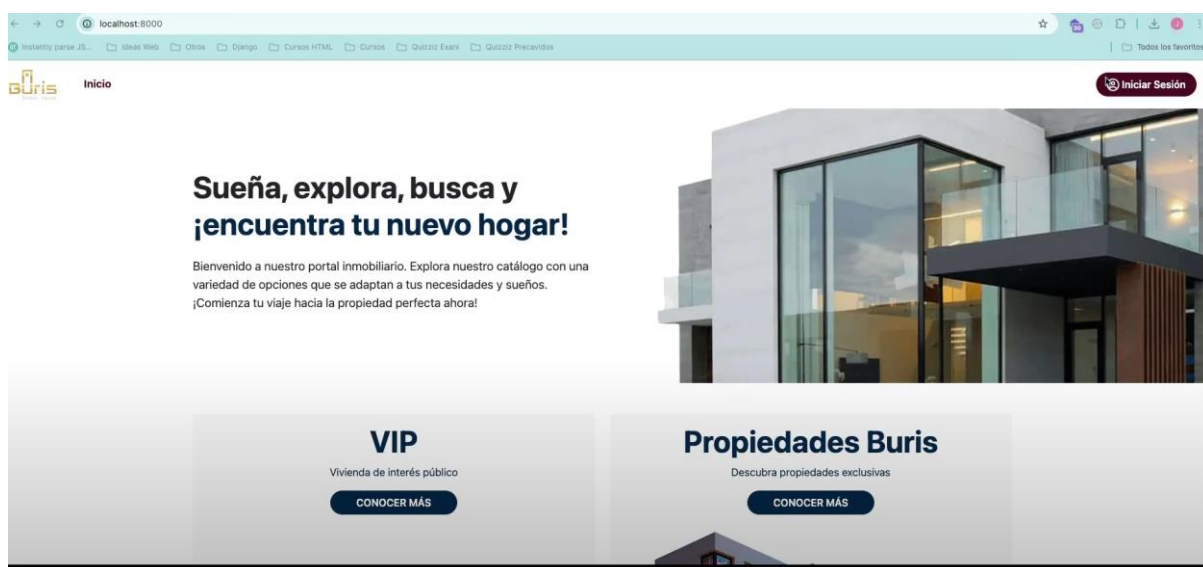
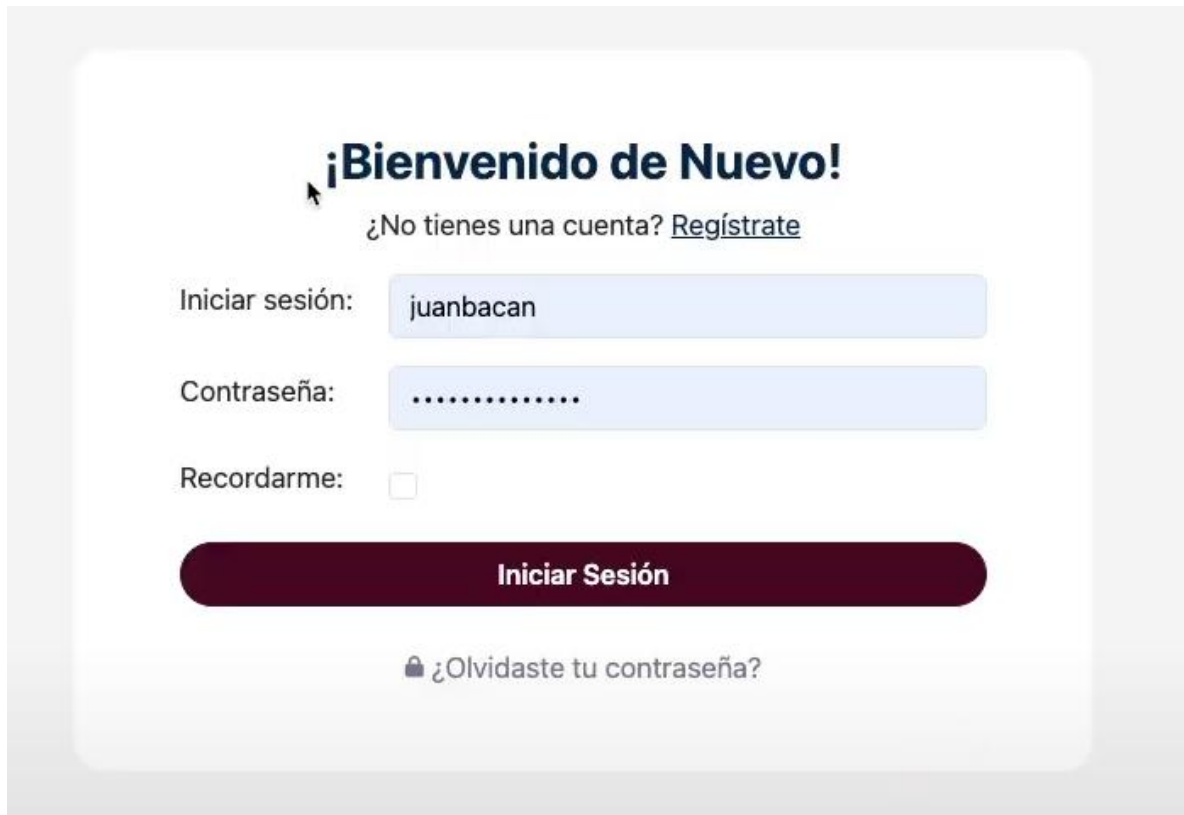


Ilustración 19. Interfaz de inicio de sesión. Fuente: Autoría Propia.

Se puede seleccionar la opción de recordarme el cual guardará los datos para iniciar sesión, sin tener que escribir nuevamente los datos. En el caso de que se olvide la contraseña, se puede recuperar de manera fácil mediante la opción de "Olvidaste tu contraseña".

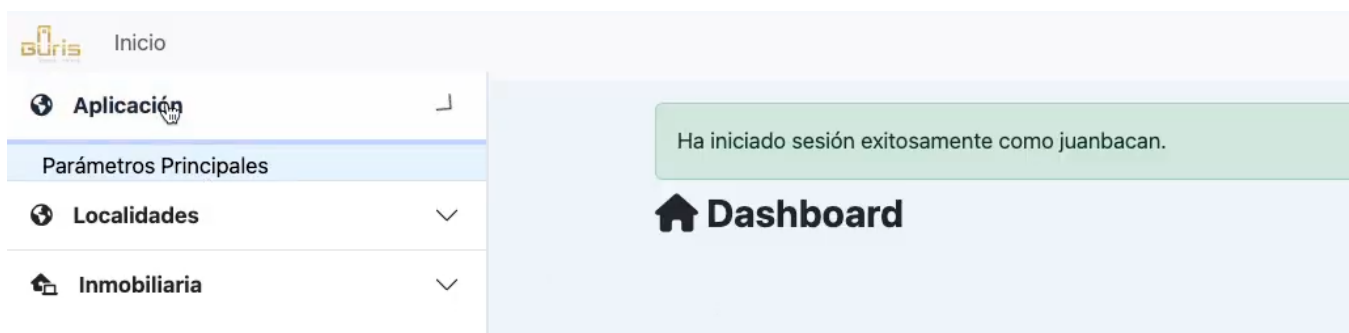


The screenshot shows a login form with the following elements:

- Header: **¡Bienvenido de Nuevo!**
- Text: ¿No tienes una cuenta? [Regístrate](#)
- Input field: Iniciar sesión:
- Input field: Contraseña:
- Form element: Recordarme:
- Button: **Iniciar Sesión**
- Text: ¿Olvidaste tu contraseña?

Ilustración 20. Interfaz de iniciar sesión. Fuente: Autoría Propia.

Una vez iniciado sesión se muestra la interfaz de las opciones que tiene el sistema de información en este caso se tiene:



The screenshot shows a dashboard interface with the following elements:

- Header: Inicio
- Menu items: **Aplicación**, **Parámetros Principales**, **Localidades**, **Inmobiliaria**
- Message: Ha iniciado sesión exitosamente como juanbacan.
- Section: **Dashboard**

Ilustración 21. Pantalla principal. Fuente: Autoría Propia.

Aplicación el cual se encuentran los parámetros principales consta de información de la URL de la página web, el nombre, el logo, contactos de la inmobiliaria, redes

Parámetros Principales de la Aplicación

Información Principal

URL del Sitio <input type="text" value="https://buris.com/"/>	Título Corto del Sitio <input type="text" value="Buris"/>
Favicon Actualmente: aplicacion_web/ba710452-6ac7-4458-bb00-e0c1fb0cb95f.png <input type="checkbox"/> Limpiar Modificar: <input type="button" value="Seleccionar archivo"/> Sin archivos seleccionados	Logo Actualmente: aplicacion_web/ba710452-6ac7-4458-bb00-e0c1fb0cb95f_Xh81uDA.png <input type="checkbox"/> Limpiar Modificar: <input type="button" value="Seleccionar archivo"/> Sin archivos seleccionados

SEO Meta Tags

Title <input type="text" value="Titulo del Sitio"/>	Meta Title <input type="text" value="¿Cómo obtener un certificado de nacimiento?"/>
Meta Description <input type="text" value="dasdasdsa"/>	







Imagen Redes Sociales
sociales etc.

Ilustración 22. Aplicación Parámetros principales y lista de usuarios

Se visualiza el listado de los usuarios

Lista de Usuarios

Buscar usuario ...

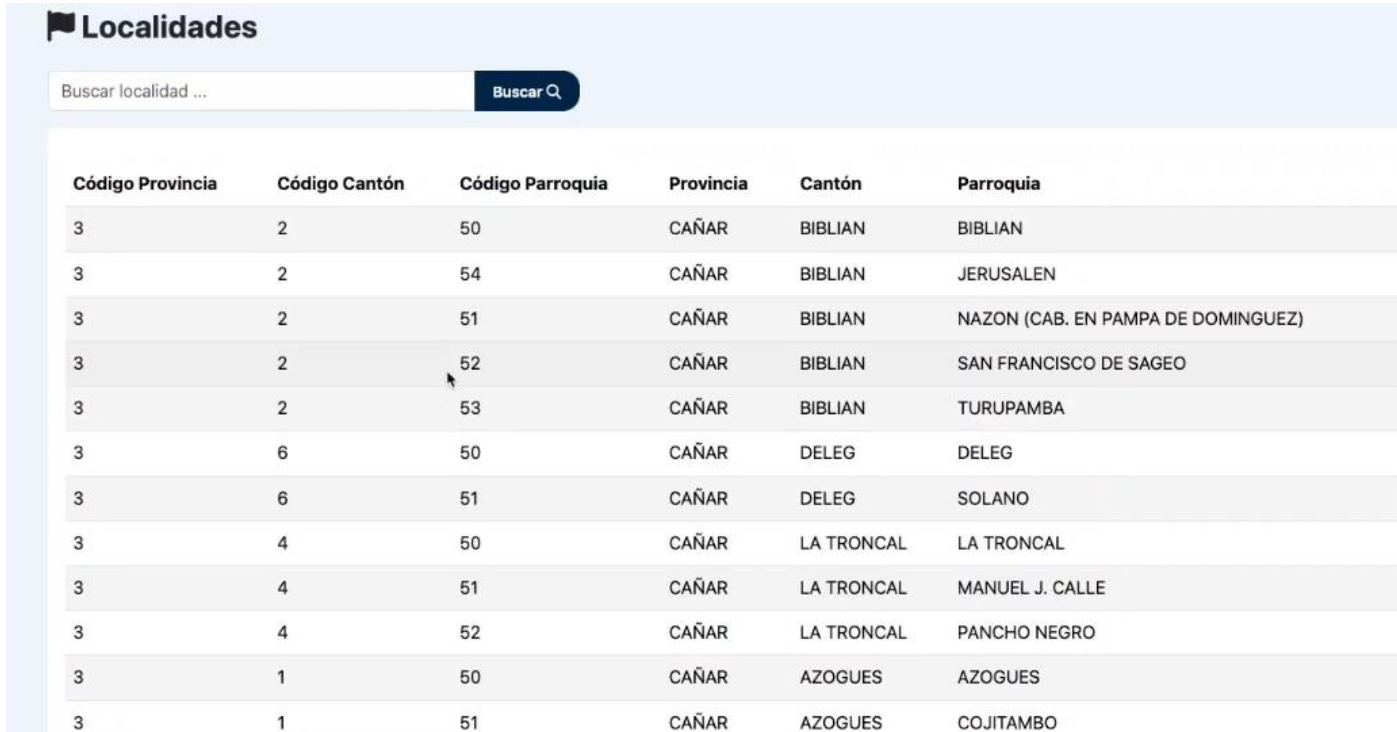
Nombre	Apellido	Usuario	Correo	Acciones
		john	lennon@gmail.com	  
Juan Vicente	Inga Ortega	juanbacan	juan.ingaor@gmail.com	  

« 1

Ilustración 23. Listado de usuarios. Fuente: Autoría propia.

Interfaz de localidades

En esta opción se procede a tener toda las provincias, catones y parroquias del Ecuador, con sus códigos.



The screenshot shows a web interface titled "Localidades" with a search bar and a table of locality data. The search bar contains the text "Buscar localidad ..." and a "Buscar" button with a magnifying glass icon. The table has six columns: "Código Provincia", "Código Cantón", "Código Parroquia", "Provincia", "Cantón", and "Parroquia". The table lists 14 rows of data for the province of Cañar.

Código Provincia	Código Cantón	Código Parroquia	Provincia	Cantón	Parroquia
3	2	50	CAÑAR	BIBLIAN	BIBLIAN
3	2	54	CAÑAR	BIBLIAN	JERUSALEN
3	2	51	CAÑAR	BIBLIAN	NAZON (CAB. EN PAMPA DE DOMINGUEZ)
3	2	52	CAÑAR	BIBLIAN	SAN FRANCISCO DE SAGEO
3	2	53	CAÑAR	BIBLIAN	TURUPAMBA
3	6	50	CAÑAR	DELEG	DELEG
3	6	51	CAÑAR	DELEG	SOLANO
3	4	50	CAÑAR	LA TRONCAL	LA TRONCAL
3	4	51	CAÑAR	LA TRONCAL	MANUEL J. CALLE
3	4	52	CAÑAR	LA TRONCAL	PANCHO NEGRO
3	1	50	CAÑAR	AZOGUES	AZOGUES
3	1	51	CAÑAR	AZOGUES	COJITAMBO

Ilustración 24. Localidades. Fuente: Autoría Propia.

Interfaz de Inmobiliaria

En esta interfaz se encuentra las propiedades en el cual se tiene las opciones de buscar la propiedad por nombre, de agregar nuevas propiedades. Las propiedades que están en venta se pueden editar o borrar.

Permiso uso industrial tipo: A, B, C y D.
NOTA: cuenta con acceso y parqueo para camiones – vehículos livianos.

p > em > strong Press ⌘0 for help 369 palabras

Tipo de Propiedad* **Estado de Propiedad***
 Locales Comerciales En venta

¿Destacada?

Precio

Precio de venta o alquiler* **Precio de Oferta (Opcional)**
 USD 20000.00 USD 0.00

Galería de Imágenes

Arrastra y suelta las imágenes de la galería aquí
[Seleccionar y Cargar](#)

Ilustración 26. Editar Propiedad. Fuente: Autoría Propia.

Cargar de imágenes

Precio de venta o alquiler* **Precio de Oferta (Opcional)**
 USD 20000.00 USD 18000

Galería de Imágenes

Arrastra y suelta las imágenes de la galería aquí
[Seleccionar y Cargar](#)

x imagen3.webp
104 KB

Powered by PGINA

Características

Ilustración 27. Cargar imágenes. Fuente: Autoría Propia.

Características

Se puede definir el número de dormitorios, baños, parqueaderos, de cuantos pisos, área total y en qué año se construyó.



The screenshot shows a web form titled "Características" with the following fields:

- Número de Dormitorios***: A numeric input field with a value of 6, flanked by minus and plus buttons.
- Número de Baños***: A numeric input field with a value of 3, flanked by minus and plus buttons.
- Número de Plantas***: A numeric input field with a value of 5, flanked by minus and plus buttons.
- Número de Parqueaderos***: A numeric input field with a value of 3, flanked by minus and plus buttons.
- Área de Construcción***: A text input field with the value "m2 2000.00".
- Área Total***: A text input field with the value "m2 2000.00".
- Año de Construcción***: A text input field with the value "1995" and a dropdown arrow on the right.

Ilustración 28. Pantalla de las características de la propiedad. Fuente: Autoría Propia.

Dirección

Se gestiona los datos de una propiedad, se puede ingresar la dirección, el código postal, y seleccionar la provincia, cantón, y parroquias correspondientes. Además, tiene acceso a un mapa interactivo que permite arrastrar el marcador para seleccionar con precisión la ubicación exacta de la propiedad. Esta función facilita la georreferenciación directa dentro del sistema, asegurando que la propiedad quede

Dirección*
Cuenca, Centro Histórico

Código Postal*
010107

Provincia*
AZUAY

Cantón*
CUENCA

Parroquia*
CHIQUINTAD

***Arrastra el marcador en el mapa para seleccionar la ubicación de la propiedad**




Ilustración 29. Mapa interactivo. Fuente: Autoría Propia.



Manual del programador

**Desarrollo de un Sistema de información para la
gestión inmobiliaria de Buris y Bienes Raíces,
basado en metodologías ágiles.**

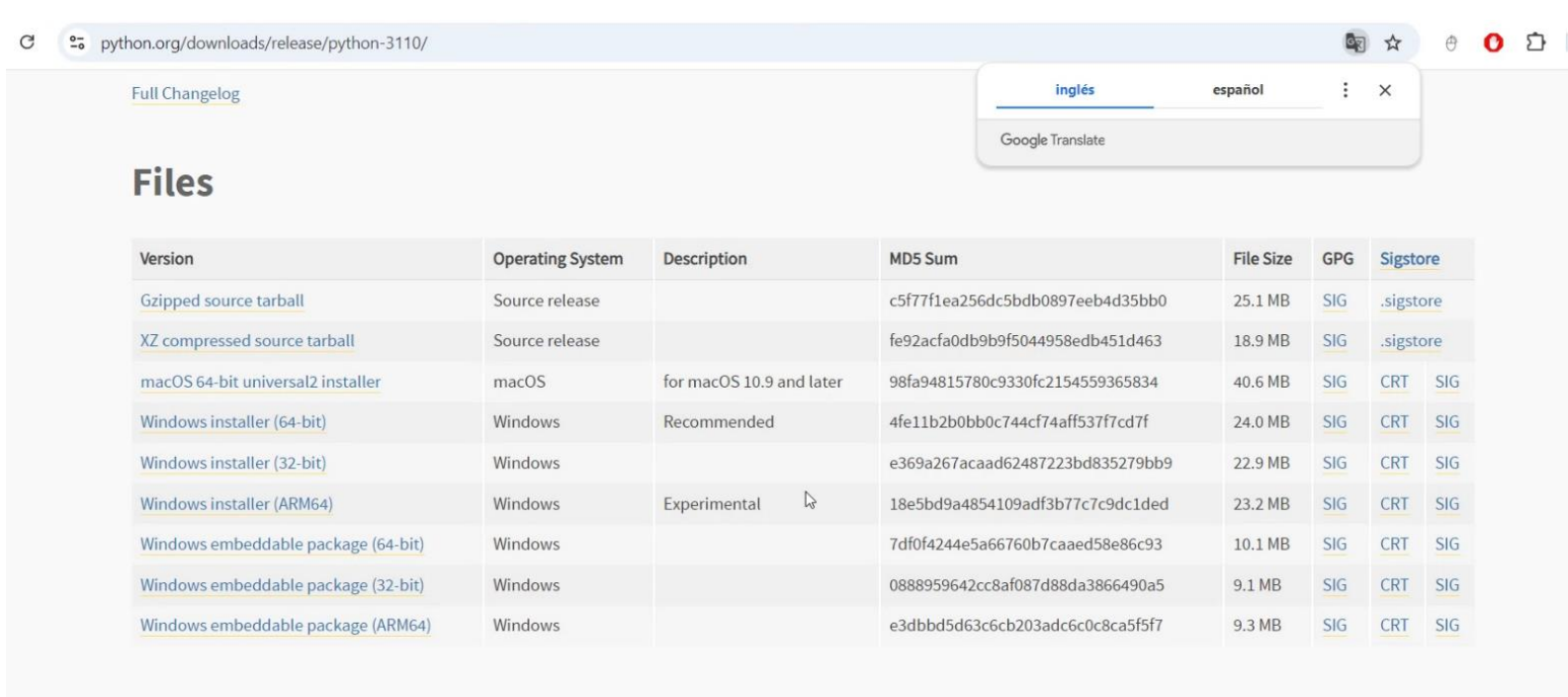
UNIVERSIDAD CATÓLICA DE CUENCA EXTENSIÓN CAÑAR

2024

Manual del Programador

El objetivo del manual del programador es describir la estructura y el proceso de desarrollo del sistema de información, proporcionando detalles sobre las herramientas empleadas durante su construcción. Se presenta información técnica redactada de tal manera que cualquier individuo con conocimientos en el área pueda entender e interpretar claramente el contenido expuesto.

Primero se descarga y se instala python desde la pagina oficial

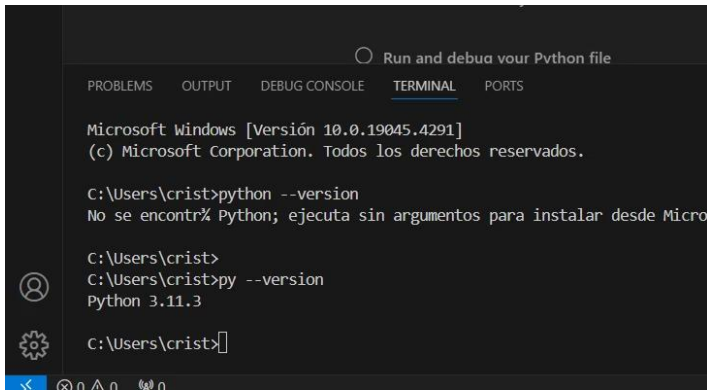


The screenshot shows the Python download page for version 3.11.0. It features a navigation bar with 'Full Changelog' and a language selector for 'inglés' and 'español'. Below the navigation is a 'Files' section containing a table with columns for Version, Operating System, Description, MD5 Sum, File Size, GPG, and Sigstore. The table lists various download options including source tarballs, macOS installers, and Windows installers for different architectures (64-bit, 32-bit, ARM64).

Version	Operating System	Description	MD5 Sum	File Size	GPG	Sigstore
Gzipped source tarball	Source release		c5f77f1ea256dc5bdb0897eeb4d35bb0	25.1 MB	SIG	.sigstore
XZ compressed source tarball	Source release		fe92acfa0db9b9f5044958edb451d463	18.9 MB	SIG	.sigstore
macOS 64-bit universal2 installer	macOS	for macOS 10.9 and later	98fa94815780c9330fc2154559365834	40.6 MB	SIG	CRT SIG
Windows installer (64-bit)	Windows	Recommended	4fe11b2b0bb0c744cf74aff537f7cd7f	24.0 MB	SIG	CRT SIG
Windows installer (32-bit)	Windows		e369a267acaad62487223bd835279bb9	22.9 MB	SIG	CRT SIG
Windows installer (ARM64)	Windows	Experimental	18e5bd9a4854109adf3b77c7c9dc1ded	23.2 MB	SIG	CRT SIG
Windows embeddable package (64-bit)	Windows		7df0f4244e5a66760b7caaed58e86c93	10.1 MB	SIG	CRT SIG
Windows embeddable package (32-bit)	Windows		0888959642cc8af087d88da3866490a5	9.1 MB	SIG	CRT SIG
Windows embeddable package (ARM64)	Windows		e3d5bd5d63c6cb203adc6c0c8ca5f5f7	9.3 MB	SIG	CRT SIG

Ilustración 30. Python descargas. Fuente: Autoría Propia.

Luego al abrir la terminal se comprueba la respectiva instalacion se haya realizado correctamente `py --version`



```
Run and debug your Python file
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Microsoft Windows [Versión 10.0.19045.4291]
(c) Microsoft Corporation. Todos los derechos reservados.

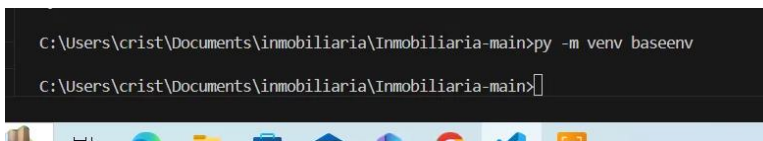
C:\Users\cris>python --version
No se encontró Python; ejecuta sin argumentos para instalar desde Microsoft Store.

C:\Users\cris>
C:\Users\cris>py --version
Python 3.11.3

C:\Users\cris>
```

Ilustración 31. Terminal de python. Fuente: Autoría Propia.

Se crea un entorno virtual `py -m venv baseenv`

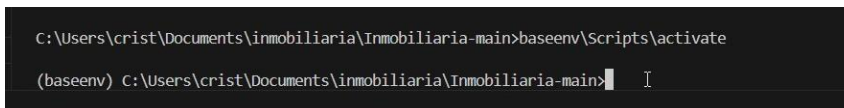


```
C:\Users\cris\Documents\inmobiliaria\Inmobiliaria-main>py -m venv baseenv
C:\Users\cris\Documents\inmobiliaria\Inmobiliaria-main>
```

Ilustración 32. Entorno virtual. Fuente: Autoría Propia.

Se activa el entorno, si está en windows es:

`baseenv/Scripts/activate`

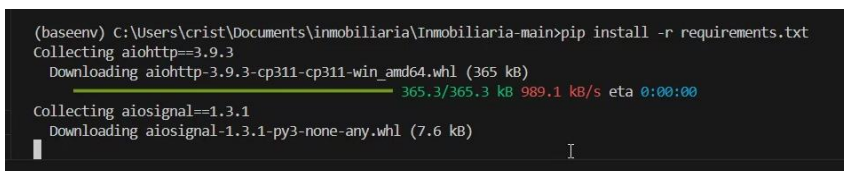


```
C:\Users\cris\Documents\inmobiliaria\Inmobiliaria-main>baseenv\Scripts\activate
(baseenv) C:\Users\cris\Documents\inmobiliaria\Inmobiliaria-main>
```

Ilustración 33. Activación del entorno. Fuente: Autoría Propia.

Procede la respectiva instalación de los paquetes

`py -m pip install -r requirements`



```
(baseenv) C:\Users\cris\Documents\inmobiliaria\Inmobiliaria-main>pip install -r requirements.txt
Collecting aiohttp==3.9.3
  Downloading aiohttp-3.9.3-cp311-cp311-win_amd64.whl (365 kB)
    365.3/365.3 kB 989.1 kB/s eta 0:00:00
Collecting aiosignal==1.3.1
  Downloading aiosignal-1.3.1-py3-none-any.whl (7.6 kB)
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Collecting bson==0.5.10
  Downloading bson-0.5.10.tar.gz (10 kB)
  Preparing metadata (setup.py) ... done
Collecting certifi==2024.2.2
  Downloading certifi-2024.2.2-py3-none-any.whl (163 kB)
  163.8/163.8 kB ? eta 0:00:00
Collecting cffi==1.16.0
  Downloading cffi-1.16.0-cp311-cp311-win_amd64.whl (181 kB)
  181.5/181.5 kB 10.7 MB/s eta 0:00:00
Collecting charset-normalizer==3.3.2
  Downloading charset_normalizer-3.3.2-cp311-cp311-win_amd64.whl (99 kB)
  99.9/99.9 kB 5.6 MB/s eta 0:00:00
Collecting cryptography==42.0.5
  Downloading cryptography-42.0.5-cp39-abi3-win_amd64.whl (2.9 MB)
  2.9/2.9 MB 7.1 MB/s eta 0:00:00
Collecting cssselect2==0.7.0
  Downloading cssselect2-0.7.0-py3-none-any.whl (15 kB)
Collecting defusedxml==0.7.1
  Using cached defusedxml-0.7.1-py2.py3-none-any.whl (25 kB)
Collecting Django==5.0.4
  Downloading Django-5.0.4-py3-none-any.whl (8.2 MB)
  8.2/8.2 MB 8.9 MB/s eta 0:00:00
Collecting django-allauth==0.61.1
  Downloading django-allauth-0.61.1.tar.gz (951 kB)
  951.5/951.5 kB 10.0 MB/s eta 0:00:00
Installing build dependencies ... /
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

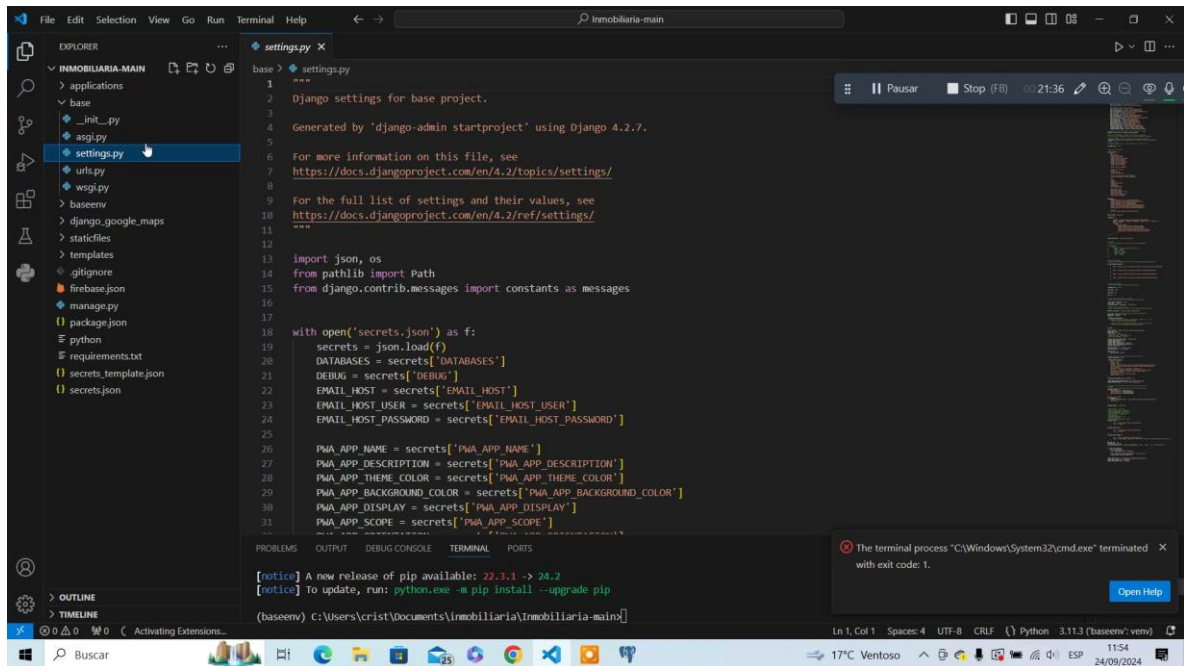
nd at https://github.com/pypa/pip/issues/8559
Running setup.py install for bson ... done
DEPRECATION: py-vapid is being installed using the legacy 'setup.py install' method, because it does not have a 'pyproject.toml' and the 'wheel' package is not installed. pip 23.1 will enforce this behaviour change. A possible replacement is to enable the '--use-pep517' option. Discussion can be found at https://github.com/pypa/pip/issues/8559
Running setup.py install for py-vapid ... done
DEPRECATION: http_ece is being installed using the legacy 'setup.py install' method, because it does not have a 'pyproject.toml' and the 'wheel' package is not installed. pip 23.1 will enforce this behaviour change. A possible replacement is to enable the '--use-pep517' option. Discussion can be found at https://github.com/pypa/pip/issues/8559
Running setup.py install for http_ece ... done
DEPRECATION: pywebpush is being installed using the legacy 'setup.py install' method, because it does not have a 'pyproject.toml' and the 'wheel' package is not installed. pip 23.1 will enforce this behaviour change. A possible replacement is to enable the '--use-pep517' option. Discussion can be found at https://github.com/pypa/pip/issues/8559
Running setup.py install for pywebpush ... done
Successfully installed Brotli-1.1.0 Django-5.0.4 PyJWT-2.8.0 aiohttp-3.9.3 aiosignal-1.3.1 asgiref-3.8.1 attrs-23.2.0 beautifulsoup4-4.12.3 bson-0.5.10 certifi-2024.2.2 cffi-1.16.0 charset-normalizer-3.3.2 cryptography-42.0.5 cssselect2-0.7.0 defusedxml-0.7.1 django-allauth-0.61.1 django-autocompletemenu-1.1.0 django-cas-2.7.0 django-csp-3.10.0 django-csp-headers-1.7.0 django-extensions-3.2.3 django-filter-23.2 django-js-asset-2.2.0 django-mptt-0.16.0 django-nested-admin-4.0.2 django-pwa-1.1.0 django-resized-1.0.2 django-social-share-2.3.0 django-tinymce-4.0.0 django-webpush-0.3.4 fonttools-4.51.0 frozenlist-1.4.1 html5lib-1.1 http_ece-1.2.0 idna-3.6 multidict-6.0.5 numpy-1.26.4 oauthlib-3.2.2 pandas-2.2.1 pillow-10.3.0 psychopg2-binary-2.9.9 py-vapid-1.9.0 pycparser-2.22 pydyf-0.9.0 pyphen-0.14.0 python-dateutil-2.9.0.post0 python-mock-4.0.3 python-openid-3.2.0 pytz-2024.1 pywebpush-1.9.4 requests-2.31.0 requests-oauthlib-2.0.0 six-1.16.0 soupsieve-2.5 sqlparse-0.4.4 tinycss2-1.2.1 tzdata-2024.1 urllib3-2.2.1 weasyprint-61.2 webencodings-0.5.1 yarl-1.9.4 zopfli-0.2.3

[notice] A new release of pip available: 22.3.1 -> 24.2
[notice] To update, run: python.exe -m pip install --upgrade pip

(baseenv) C:\Users\cris\Documents\inmobiliaria\Inmobiliaria-main>
```

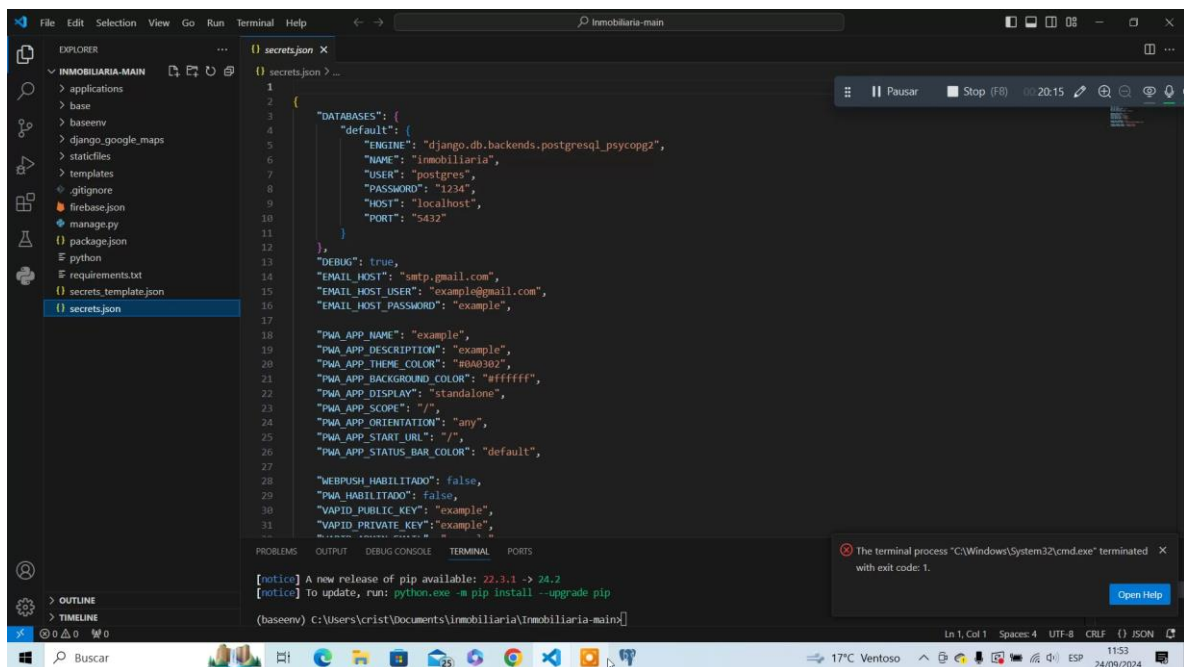
Ilustración 34. Instalación de paquetes. Fuente: Autoría Propia.

Seguidamente, se crea el archivo secrets.json de acuerdo a la configuración de la base de datos. En este caso, se utilizó PostgreSQL.



The screenshot shows the Visual Studio Code editor with the 'settings.py' file open. The file content is as follows:

```
1 Django settings for base project.
2
3
4 Generated by 'django-admin startproject' using Django 4.2.7.
5
6 For more information on this file, see
7 https://docs.djangoproject.com/en/4.2/topics/settings/
8
9 For the full list of settings and their values, see
10 https://docs.djangoproject.com/en/4.2/ref/settings/
11
12
13 import json, os
14 from pathlib import Path
15 from django.contrib.messages import constants as messages
16
17
18 with open('secrets.json') as f:
19     secrets = json.load(f)
20     DATABASES = secrets['DATABASES']
21     DEBUG = secrets['DEBUG']
22     EMAIL_HOST = secrets['EMAIL_HOST']
23     EMAIL_HOST_USER = secrets['EMAIL_HOST_USER']
24     EMAIL_HOST_PASSWORD = secrets['EMAIL_HOST_PASSWORD']
25
26     PMA_APP_NAME = secrets['PMA_APP_NAME']
27     PMA_APP_DESCRIPTION = secrets['PMA_APP_DESCRIPTION']
28     PMA_APP_THEME_COLOR = secrets['PMA_APP_THEME_COLOR']
29     PMA_APP_BACKGROUND_COLOR = secrets['PMA_APP_BACKGROUND_COLOR']
30     PMA_APP_DISPLAY = secrets['PMA_APP_DISPLAY']
31     PMA_APP_SCOPE = secrets['PMA_APP_SCOPE']
```



The screenshot shows the Visual Studio Code editor with the 'secrets.json' file open. The file content is as follows:

```
1 {
2   "DATABASES": {
3     "default": {
4       "ENGINE": "django.db.backends.postgresql_psycopg2",
5       "NAME": "inmobiliaria",
6       "USER": "postgres",
7       "PASSWORD": "1234",
8       "HOST": "localhost",
9       "PORT": "5432"
10    }
11  },
12  "DEBUG": true,
13  "EMAIL_HOST": "smtp.gmail.com",
14  "EMAIL_HOST_USER": "example@gmail.com",
15  "EMAIL_HOST_PASSWORD": "example",
16
17  "PMA_APP_NAME": "example",
18  "PMA_APP_DESCRIPTION": "example",
19  "PMA_APP_THEME_COLOR": "#808080",
20  "PMA_APP_BACKGROUND_COLOR": "#ffffff",
21  "PMA_APP_DISPLAY": "standalone",
22  "PMA_APP_SCOPE": "/",
23  "PMA_APP_ORIENTATION": "any",
24  "PMA_APP_START_URL": "/",
25  "PMA_APP_STATUS_BAR_COLOR": "default",
26
27  "WEBPUSH_HABILITADO": false,
28  "PMA_HABILITADO": false,
29  "VAPID_PUBLIC_KEY": "example",
30  "VAPID_PRIVATE_KEY": "example",
31 }
```

Ilustración 35. Creación del archivo. Fuente: Autoría Propia.

Se crea la base de datos a la cual se realizarán las migraciones

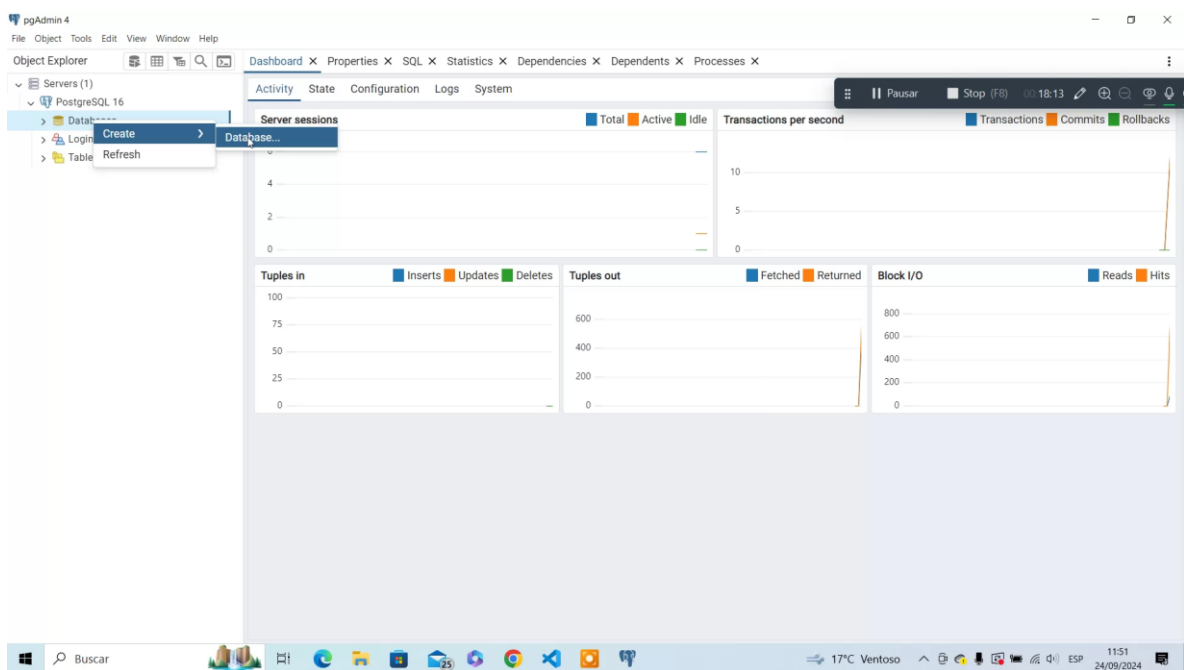
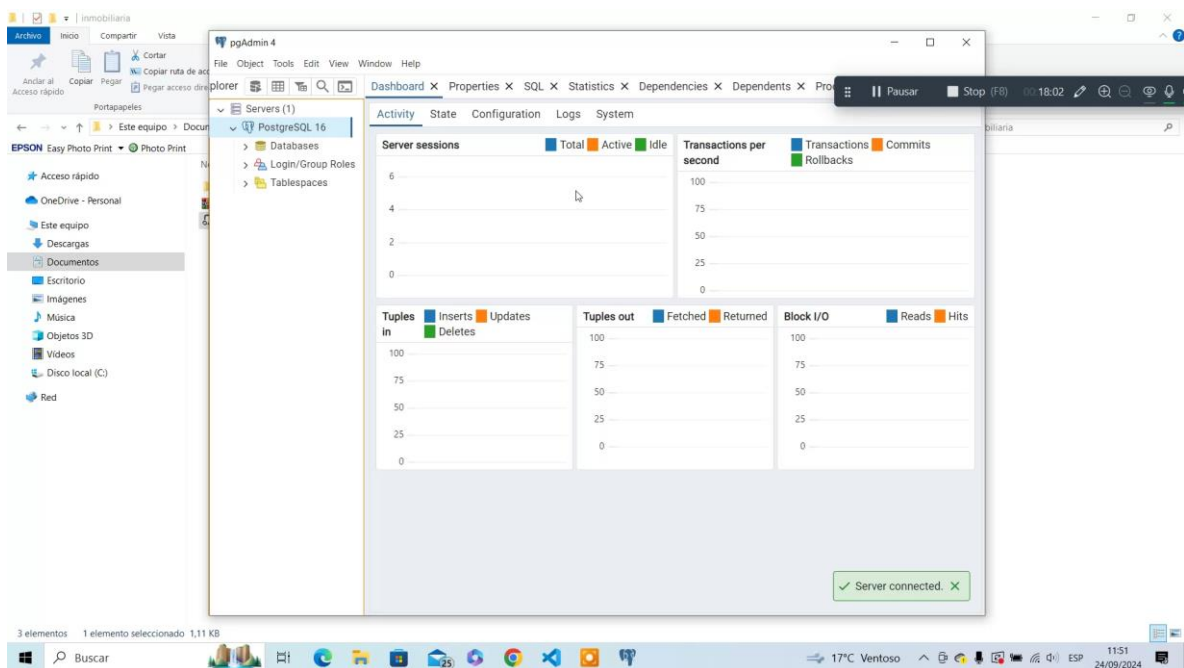


Ilustración 36. Creación de base de datos. Fuente: Autoría Propia.

Se llamará inmobiliaria

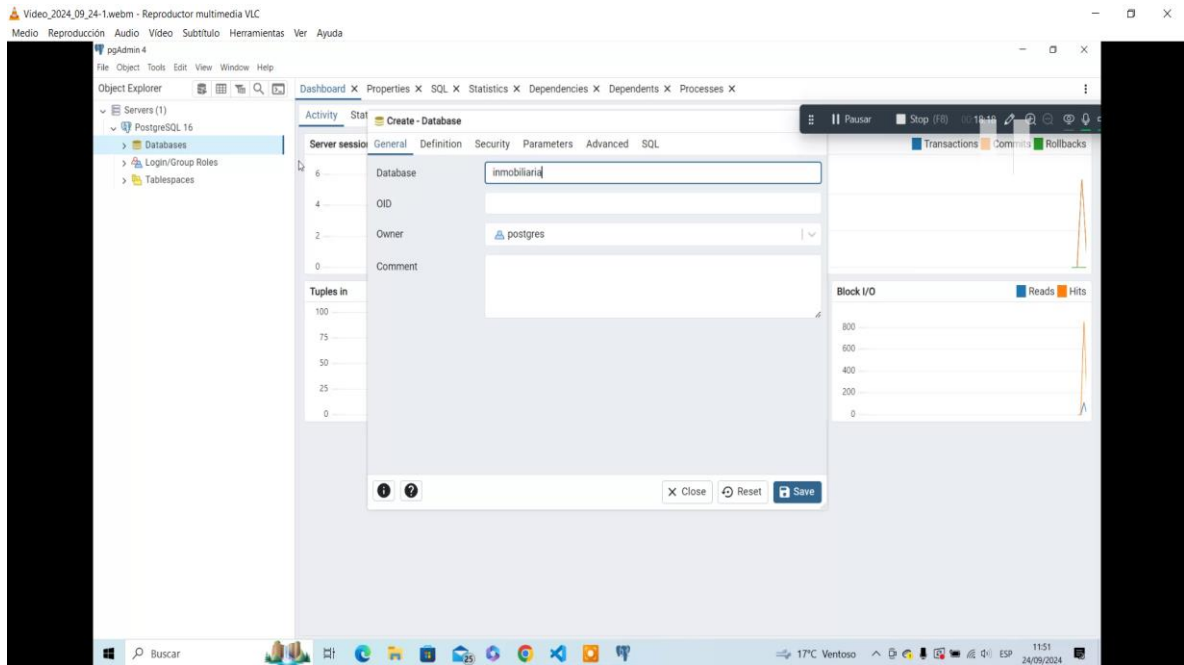


Ilustración 37. Base de datos INMOBILIARIA. Fuente: Autoría Propia.

Para que no existan problemas se crea las reglas de entrada y salida para el puerto 5432 de postgresql

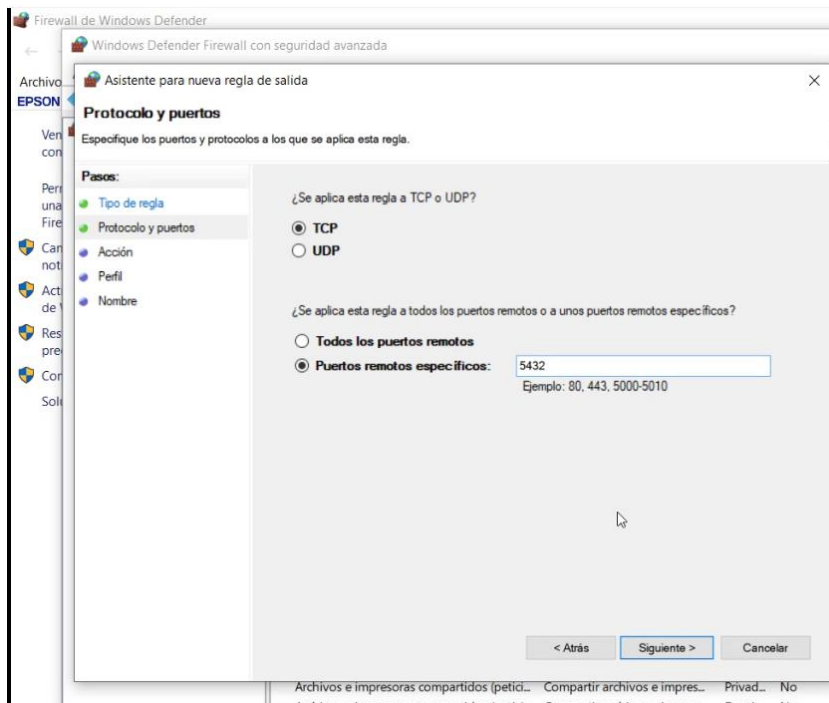


Ilustración 38. Creamos reglas de entrada y salida. Fuente: Autoría Propia.

Se instala la extensión de croppersjs `pip install django-croppersjs`

```
(baseenv) C:\Users\cris\Documents\inmobiliaria\Inmobiliaria-main>pip install django-cropperjs
Collecting django-cropperjs
  Downloading django_cropperjs-0.0.5-py3-none-any.whl.metadata (2.5 kB)
  Downloading django_cropperjs-0.0.5-py3-none-any.whl (28 kB)
Installing collected packages: django-cropperjs
Successfully installed django-cropperjs-0.0.5

(baseenv) C:\Users\cris\Documents\inmobiliaria\Inmobiliaria-main>
```

Ilustración 39. Instalación de extensión. Fuente: Autoría Propia.

Procede las respectivas migraciones

`python manage.py makemigrations`

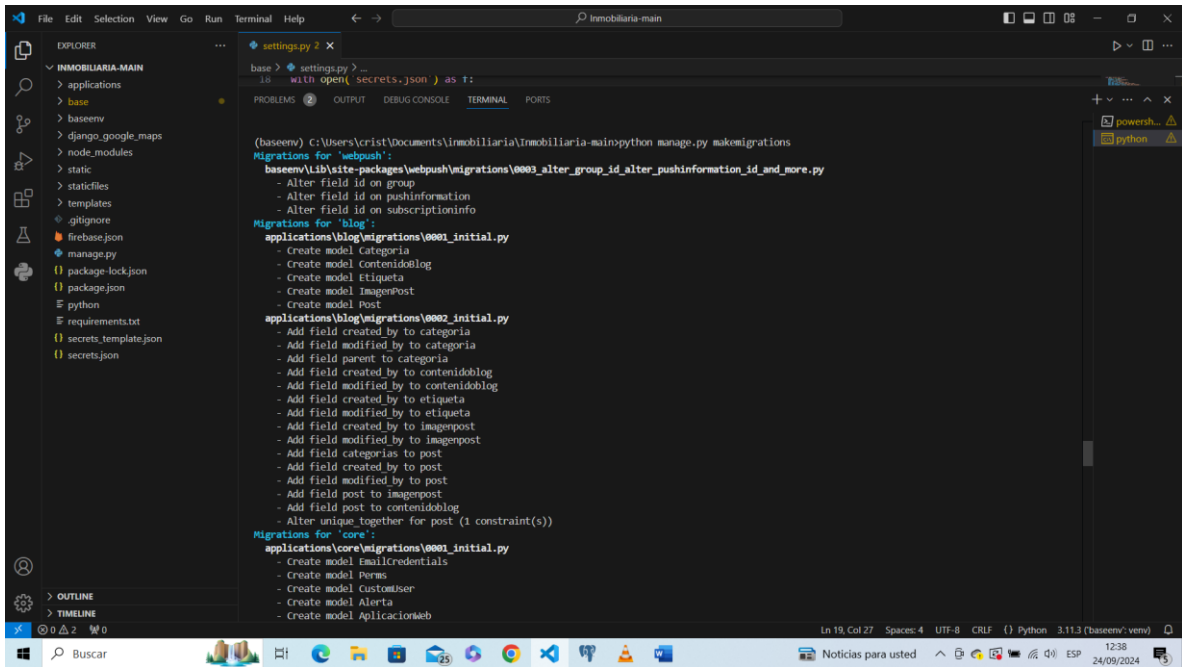


Ilustración 40. Migraciones. Fuente: Autoría Propia.

python manage.py migrate

python manage.py collectstatic

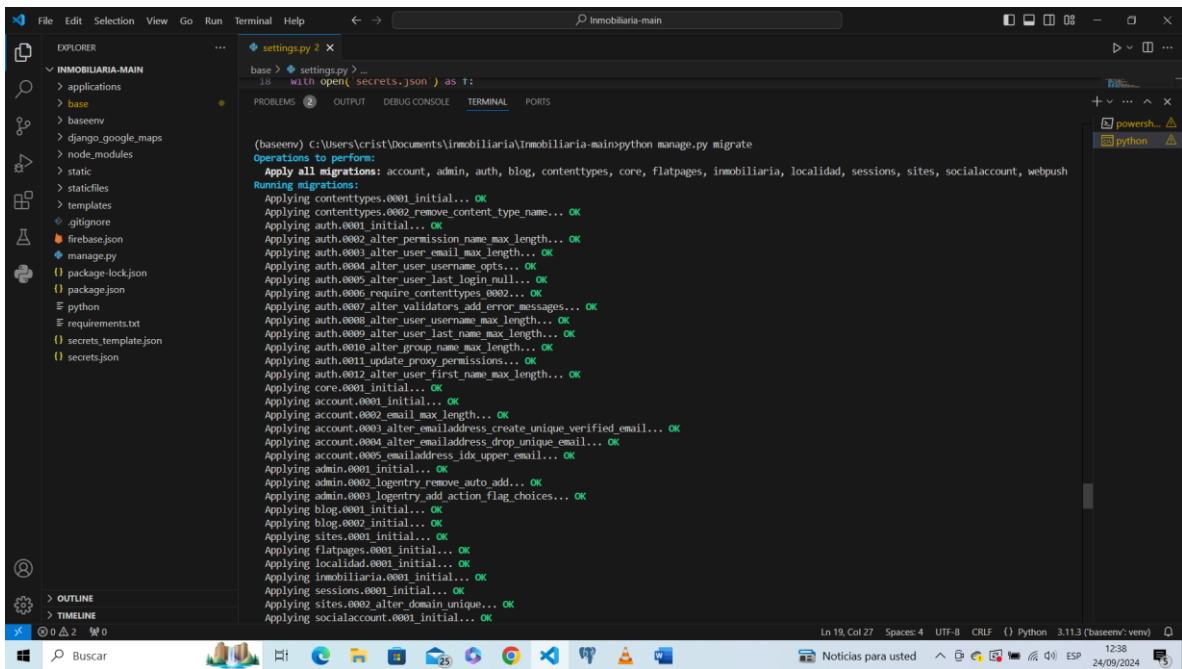


Ilustración 41. Migraciones. Fuente: Autoría Propia.

Se crea un superusuario python manage.py createsuperuser y se llena los datos correspondientes.

```
(baseenv) C:\Users\cris\Documents\inmobiliaria\Inmobiliaria-main>python manage.py createsuperuser
Nombre de usuario: alexandra2024
Correo electrónico: alexandra2024@gmail.com
Password:
Password (again):
La contraseña es demasiado similar a la de nombre de usuario.
Bypass password validation and create user anyway? [y/N]: n
Password:
Password (again):
La contraseña es demasiado similar a la de nombre de usuario.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.

(baseenv) C:\Users\cris\Documents\inmobiliaria\Inmobiliaria-main>python manage.py runserver
```

Ilustración 42. Creación del superusuario. Fuente: Autoría propia.

Se inicia el servidor python manage.py runserver

```
(baseenv) C:\Users\cris\Documents\inmobiliaria\Inmobiliaria-main>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
September 24, 2024 - 12:11:18
Django version 5.0.4, using settings 'base.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.

[24/Sep/2024 12:11:30] "GET / HTTP/1.1" 200 22561
[24/Sep/2024 12:11:30] "GET /static/assets/css/style.css HTTP/1.1" 200 2108
[24/Sep/2024 12:11:30] "GET /static/bootstrap/styles.css HTTP/1.1" 200 289974
[24/Sep/2024 12:11:30] "GET /static/bootstrap/bootstrap.bundle.min.js HTTP/1.1" 200 80721
[24/Sep/2024 12:11:30] "GET /static/assets/js/functions.js HTTP/1.1" 200 7431
[24/Sep/2024 12:11:30] "GET /static/assets/images/home/2.png HTTP/1.1" 200 867514
[24/Sep/2024 12:11:30] "GET /static/assets/images/home/3.png HTTP/1.1" 200 1228476
[24/Sep/2024 12:11:30] "GET /static/assets/images/home/1.png HTTP/1.1" 200 164631
[24/Sep/2024 12:11:30] "GET /static/assets/images/home/6.png HTTP/1.1" 200 459712
[24/Sep/2024 12:11:31] "GET /static/assets/images/home/7.png HTTP/1.1" 200 321582
[24/Sep/2024 12:11:31] "GET /static/assets/images/home/9.png HTTP/1.1" 200 788438
[24/Sep/2024 12:11:31] "GET /static/assets/images/home/5.png HTTP/1.1" 200 241926
[24/Sep/2024 12:11:31] "GET /static/assets/images/home/10.png HTTP/1.1" 200 201262
[24/Sep/2024 12:11:31] "GET /static/assets/images/home/8.png HTTP/1.1" 200 228721
[24/Sep/2024 12:11:31] "GET /static/assets/images/home/4.png HTTP/1.1" 200 1996301
Not Found: /favicon.ico
[24/Sep/2024 12:11:31] "GET /favicon.ico HTTP/1.1" 404 5156
[24/Sep/2024 12:12:12] "GET / HTTP/1.1" 200 22561
```

Ilustración 43. Inicio del servidor Python. Fuente: Autoría Propia.

Verificar que este ejecutando correctamente el la ip correspondiente.

```
base > settings.py > ...  
with open('secrets.json') as f:  
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS  
System check identified no issues (0 silenced).  
September 24, 2024 - 12:11:18  
Django version 5.0.4, using settings 'base.settings'  
Starting development server at http://127.0.0.1:8000/  
Quit the server with CTRL-BREAK.  
[24/Sep/2024 12:11:30] "GET / HTTP/1.1" 200 22561  
[24/Sep/2024 12:11:30] "GET /static/assets/css/style.css HTTP/1.1" 200 2108  
[24/Sep/2024 12:11:30] "GET /static/bootstrap/styles.css HTTP/1.1" 200 289974  
[24/Sep/2024 12:11:30] "GET /static/bootstrap/bootstrap.bundle.min.js HTTP/1.1" 200 80721  
[24/Sep/2024 12:11:30] "GET /static/assets/js/functions.js HTTP/1.1" 200 7431  
[24/Sep/2024 12:11:30] "GET /static/assets/images/home/2.png HTTP/1.1" 200 867514  
[24/Sep/2024 12:11:30] "GET /static/assets/images/home/3.png HTTP/1.1" 200 1228476  
[24/Sep/2024 12:11:30] "GET /static/assets/images/home/1.png HTTP/1.1" 200 164631  
[24/Sep/2024 12:11:30] "GET /static/assets/images/home/6.png HTTP/1.1" 200 459712  
[24/Sep/2024 12:11:31] "GET /static/assets/images/home/7.png HTTP/1.1" 200 321582  
[24/Sep/2024 12:11:31] "GET /static/assets/images/home/9.png HTTP/1.1" 200 788438  
[24/Sep/2024 12:11:31] "GET /static/assets/images/home/5.png HTTP/1.1" 200 241926  
[24/Sep/2024 12:11:31] "GET /static/assets/images/home/10.png HTTP/1.1" 200 201262  
[24/Sep/2024 12:11:31] "GET /static/assets/images/home/8.png HTTP/1.1" 200 228721  
[24/Sep/2024 12:11:31] "GET /static/assets/images/home/4.png HTTP/1.1" 200 1996301  
Not Found: /favicon.ico  
[24/Sep/2024 12:11:31] "GET /favicon.ico HTTP/1.1" 404 5156  
[24/Sep/2024 12:12:12] "GET / HTTP/1.1" 200 22561  
[24/Sep/2024 12:12:13] "GET / HTTP/1.1" 200 22561  
[24/Sep/2024 12:12:15] "GET / HTTP/1.1" 200 22561  
[24/Sep/2024 12:12:16] "GET / HTTP/1.1" 200 22561  
[24/Sep/2024 12:12:17] "GET / HTTP/1.1" 200 22561  
[24/Sep/2024 12:12:22] "GET / HTTP/1.1" 200 22561  
[24/Sep/2024 12:12:26] "GET / HTTP/1.1" 200 22561  
[24/Sep/2024 12:12:29] "GET / HTTP/1.1" 200 22561  
[24/Sep/2024 12:13:12] "GET /accounts/login/ HTTP/1.1" 200 18957  
[24/Sep/2024 12:15:18] "GET /accounts/signup/ HTTP/1.1" 200 11513  
Internal Server Error: /accounts/signup/  
Traceback (most recent call last):  
File "C:/Users/crist/Documents/inmobiliaria/inmobiliaria-main/baseenv\Lib/site-packages/django/core/handlers/exception.py", line 55, in inner
```

09

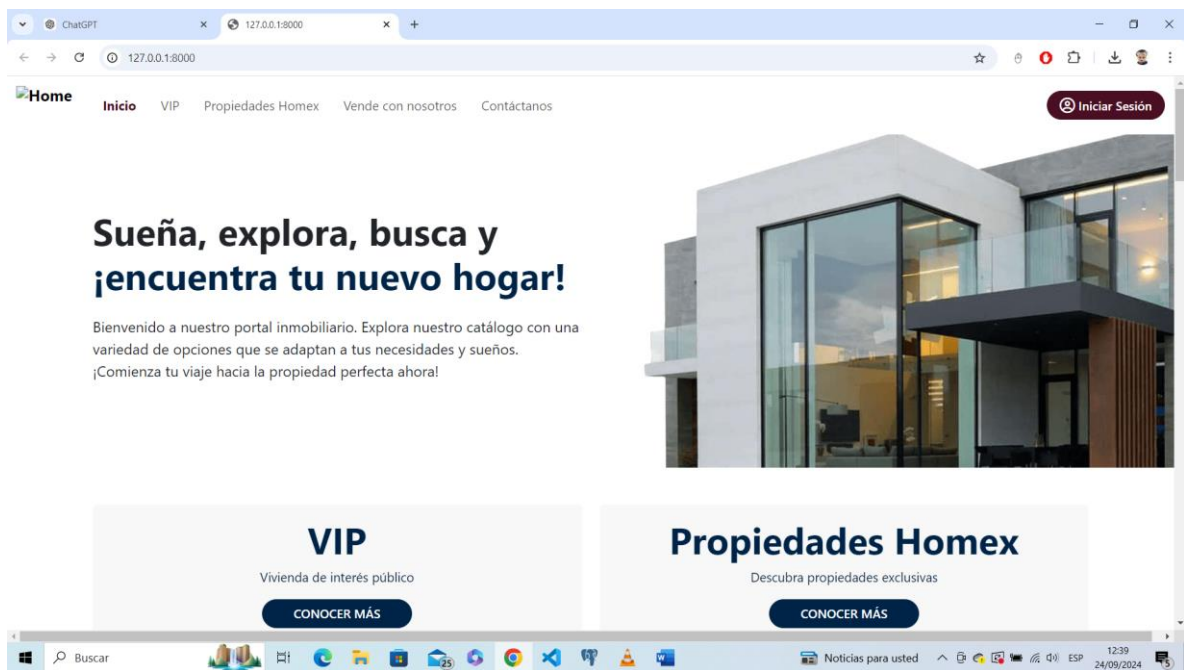


Ilustración 44. Ejecución de la IP. Fuente: Autoría Propia.



Narcisa Alexandra Guamán Duchi portador(a) de la cédula de ciudadanía N° **0302422894** En calidad de autor/a y titular de los derechos patrimoniales del trabajo de titulación **Desarrollo de un sistema de Información para la gestión Inmobiliaria de Burís y bienes raíces, basado en metodologías ágiles**, de conformidad a lo establecido en el artículo 114 Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación, reconozco a favor de la Universidad Católica de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos y no comerciales. Autorizo además a la Universidad Católica de Cuenca, para que realice la publicación de este trabajo de titulación en el Repositorio Institucional de conformidad a lo dispuesto en el artículo 144 de la Ley Orgánica de Educación Superior.

Cañar, **27 de noviembre de 2024**

F:

Narcisa Alexandra Guamán Duchi

C.I. 0302422894