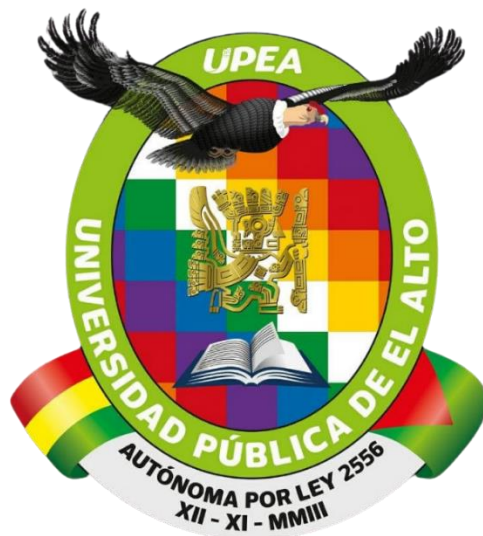


UNIVERSIDAD PÚBLICA DE EL ALTO

CARRERA INGENIERÍA DE SISTEMAS



PROYECTO DE GRADO

“SISTEMA WEB SCRAPING DE RECONOCIMIENTO COMERCIAL EN BIENES RAÍCES RESPALDADO POR INTELIGENCIA ARTIFICIAL”

“CASO: EMPRESA BI SPACE SRL”

Para optar al título de Licenciatura en Ingeniería de Sistemas

Mención: INFORMÁTICA Y COMUNICACIONES

Postulante : Juan Alberto Silva Cayo
Tutor Metodológico : M. Sc. Lic. Ing. Marisol Arguedas Balladares
Tutor Revisor : Lic. Ing. Yolanda Escobar Mancilla
Tutor Especialista : M. Sc. Lic. Ing. Freddy Félix Medina Miranda

EL ALTO – BOLIVIA
2024

DECLARACIÓN JURADA DE AUTENTICIDAD DE PERFIL DE TRABAJO DE GRADO

Yo, **Juan Alberto Silva Cayo.**, estudiante de Taller de Grado I, de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto, identificado(a) con CI. **10013618 LP.** y Registro Universitario: **200036524.** Declaro bajo juramento que:

1. Soy autor del Trabajo de Grado titulado:

“SISTEMA WEB SCRAPING DE RECONOCIMIENTO COMERCIAL EN BIENES RAÍCES RESPALDADO POR INTELIGENCIA ARTIFICIAL”.

2. El mismo que presento bajo la modalidad de Proyecto de Grado.
3. El texto de mi perfil de trabajo de grado respeta y no vulnera los derechos de terceros, incluidos los derechos de propiedad intelectual. En tal sentido, declaro que este Perfil de Trabajo de Grado no ha sido plagiado total ni parcialmente, para la cual he respetado las normas internacionales de citas y referencias de las fuentes consultadas.
4. El texto del Perfil de Trabajo de Grado que presento no ha sido publicado ni presentado antes en cualquier medio electrónico o físico.
5. Por tanto, declaro que mi Perfil de Trabajo de Grado cumple con todas las normas de la carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

El incumplimiento de lo declarado da lugar a responsabilidad del declarante, en consecuencia; a través del presente documento asumo frente a terceros, la carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto toda responsabilidad que pueda derivarse por el Perfil de Trabajo de Grado presentado.

Fecha: julio del 2024

.....
Juan Alberto Silva Cayo
CI. 10013618

DEDICATORIA

A mis padres, por enseñarme el valor del esfuerzo y la perseverancia, y por estar siempre a mi lado en cada paso de mi vida.

A mis seres queridos, por su apoyo incondicional y su fe en mis capacidades, que me dieron la fuerza para superar cada desafío de este proyecto.

A mis amigos y compañeros, quienes con su ánimo y apoyo me inspiraron a seguir adelante y completar esta meta, generando competencias para incrementar y motivar el gusto por el conocimiento

AGRADECIMIENTOS

En primer lugar, agradezco a Dios por la sabiduría y fortaleza que me ha brindado en cada paso de este proyecto, permitiéndome superar los desafíos y alcanzar esta meta.

A mis padres, quienes con su paciencia, amor y apoyo constante han sido mi inspiración y motivación a lo largo de este camino.

A mis tutores: metodológico, especialista y revisor, cuya guía y conocimientos han sido fundamentales para el desarrollo y perfeccionamiento de este trabajo. Gracias por su dedicación y compromiso en cada detalle del proceso.

A la empresa que confió en mí y me brindó la oportunidad de aplicar mis conocimientos y habilidades en un entorno profesional, permitiéndome crecer y enriquecer esta experiencia.

Finalmente, agradezco a mi universidad, que me ha brindado los recursos, el conocimiento y el espacio necesario para mi formación académica y personal.

INDICE INDICATIVO

CAPITULO I	1
1. MARCO PRELIMINAR	1
1.1 INTRODUCCION.....	1
1.2 ANTECEDENTES	2
1.2.1 <i>Antecedentes institucionales</i>	2
1.2.1.1 Misión	2
1.2.1.2 Visión	2
1.2.1.3 Objetivo.....	3
1.2.1.4 Organigrama.....	3
1.2.2 <i>Antecedentes afines a la investigación</i>	4
1.2.2.1 Antecedentes Internacionales	4
1.2.2.2 Antecedentes Nacionales	5
1.2.2.3 Antecedentes Locales	7
1.3 PLANTEAMIENTO DEL PROBLEMA	8
1.3.1 <i>Problema Principal</i>	8
1.3.2 <i>Problema Secundarios</i>	9
1.3.3 <i>Formulación del Problema</i>	9
1.4 OBJETIVOS	10
1.4.1 <i>Objetivo General</i>	10
1.4.2 <i>Objetivos Específicos</i>	10
1.5 JUSTIFICACION	11
1.5.1 <i>Justificación técnica</i>	11
1.5.2 <i>Justificación económica</i>	11
1.5.3 <i>Justificación social</i>	11
1.6 METODOLOGIA.....	12
1.6.1 <i>Metodología de desarrollo</i>	12
1.6.2 <i>Métricas de Calidad</i>	12

1.6.3	<i>Costo</i>	13
1.6.4	<i>Seguridad</i>	13
1.6.5	<i>Pruebas de Software</i>	14
1.7	MÉTODOS DE RECOLECCIÓN DE DATOS	15
1.7.1	<i>Técnicas de Investigación</i>	16
1.8	HERRAMIENTAS	16
1.8.1	<i>Desarrollo de la Inteligencia Artificial con Python y Flask</i>	16
1.8.2	<i>Procesamiento de lenguaje natural (NLP)</i>	16
1.8.3	<i>Base de datos Geoespacial con PostgreSQL y PostGIS</i>	17
1.8.4	<i>Frontend interactivo con React y Material-UI</i>	17
1.8.5	<i>Docker</i>	17
1.8.6	<i>Redis</i>	18
1.9	LIMITES Y ALCANCES	18
1.9.1	<i>Limites</i>	18
1.9.2	<i>Alcances</i>	19
1.10	APORTES	20
	CAPITULO II	21
2.	MARCO TEÓRICO	21
2.1	<i>Web Scraping</i>	21
2.2	<i>Minería de Datos (Data Mining)</i>	23
2.3	<i>Inteligencia Artificial (IA)</i>	26
2.3.1	<i>Técnicas y algoritmos de IA</i>	27
2.4	<i>Modelos de Aprendizaje Automático (Machine Learning Models)</i>	28
2.4.1	<i>Tipos de Modelos de Aprendizaje Automático</i>	29
2.4.2	<i>Proceso de Entrenamiento de Modelos</i>	29
2.5	<i>Procesamiento de Lenguaje Natural (NLP)</i>	31
2.5.1	<i>Procesos Claves en NLP</i>	33
2.6	<i>Base de Datos y Almacenamiento de Datos</i>	35
2.6.1	<i>PostgreSQL</i>	35

2.6.2	<i>PostGIS: Gestión de Datos Espaciales</i>	36
2.6.3	<i>Procesos Claves en el Almacenamiento de Datos</i>	38
2.7	<i>Análisis de datos</i>	39
2.7.1	<i>Herramientas y Técnicas</i>	40
2.8	<i>Visualización de Datos</i>	40
2.8.1	<i>Importancia en el Análisis de Datos</i>	41
2.8.2	<i>Tipos de Visualizaciones de Datos</i>	41
2.9	<i>Limpieza de Datos (Data Cleaning)</i>	42
2.9.1	<i>Importancia de la Limpieza de Datos</i>	42
2.9.2	<i>Procesos Clave en la Limpieza de Datos</i>	43
2.10	<i>Metodología CRISP-DM</i>	44
2.10.1	<i>Definición y Estructura del Modelo CRISP-DM</i>	44
2.10.1.1	<i>Comprensión del Negocio</i>	45
2.10.1.2	<i>Comprensión de los Datos</i>	45
2.10.1.3	<i>Preparación de los Datos</i>	45
2.10.1.4	<i>Modelado</i>	46
2.10.1.5	<i>Evaluación</i>	46
2.10.1.6	<i>Despliegue</i>	46
2.10.2	<i>Ventajas y Aplicabilidad de CRISP-DM</i>	47
2.11	<i>Flask</i>	47
2.12	<i>React Materia- UI</i>	49
2.13	<i>BeautifulSoup</i>	51
2.14	<i>Celery Redis</i>	53
2.15	<i>Herramienta para Web Scraping</i>	56
2.15.1	<i>Pypetter</i>	56
2.15.2	<i>Playwright</i>	56
2.15.3	<i>Selemium</i>	57
2.16	<i>Docker</i>	58
2.17	<i>Pruebas</i>	59
2.17.1	<i>Pruebas Unitarias</i>	59
2.17.2	<i>Pruebas de accesibilidad</i>	60
2.17.3	<i>Pruebas de Sistema</i>	61

2.17.3.1	Evaluación de Requisitos	61
2.18	Seguridad.....	61
2.19	Costos.....	63
2.19.1	<i>Metodología COSMIC</i>	63
2.19.1.1	Introducción a COSMIC	63
2.19.1.2	Medición de Tamaño de Software con COSMIC	64
2.19.1.3	Aplicación de COSMIC para Estimaciones de Costos.....	64
CAPITULO III		65
3.	MARCO APLICATIVO	65
3.1	CRISP-DM	65
3.1.1	<i>Comprensión del negocio</i>	65
3.1.1.1	<i>Ingeniería de requerimientos</i>	65
3.1.1.2	Análisis de la Estructura de las Páginas Web	68
3.1.1.3	Herramientas para realizar Web Scraping disponibles	71
3.1.1.4	Diseño y Creación de la Base de Datos	72
3.1.2	<i>Comprensión de los Datos</i>	75
3.1.3	<i>Preparación de los Datos</i>	77
3.1.4	<i>Modelado</i>	79
3.1.5	<i>Evaluación</i>	89
3.1.6	Despliegue	97
3.2	CONSTRUCCION DE LA INTERFAZ DE USUARIO	99
3.2.1	<i>Ingreso al sistema</i>	99
3.2.2	<i>Inicio</i>	99
3.2.3	<i>Recolección de datos</i>	101
3.2.4	<i>Completado de información</i>	105
3.2.5	<i>Historial de Datos</i>	106
3.2.6	<i>Análisis de datos</i>	107
3.2.6.1	Análisis de Precios	107
3.2.6.2	Clasificación de propiedades	107
3.2.6.3	Mapa de Calor	108
3.2.7	<i>Modelos IA</i>	108
3.2.7.1	Entrenamiento del modelo	109

3.2.7.2	Evaluación del modelo.....	110
CAPITULO IV		113
4.	CALIDAD, SEGURIDAD Y PRUEBAS	113
4.1	METRICAS DE CALIDAD	113
4.1.1	<i>Norma ISO/IEC 25010</i>	113
4.1.1.1	Funcionalidad	113
4.1.1.2	Confiabilidad	119
4.1.1.3	Usabilidad	121
4.1.1.4	Eficiencia	123
4.1.1.5	Fiabilidad	124
4.1.1.6	Portabilidad.....	126
4.1.2	<i>Resultados ISO/IEC 25010</i>	128
4.2	ESTIMACION DEL COSTO	129
4.2.1	<i>Planificación de la medición</i>	129
4.2.2	<i>Asignación y medición</i>	131
4.3	SEGURIDAD	140
4.3.1	<i>Seguridad Lógica</i>	141
4.3.1.1	Control de Acceso	141
4.3.1.2	Desactivación de Cuenta.....	142
4.3.1.3	Roles y Permisos.....	142
4.3.1.4	Control de Asignación de Roles.....	142
4.3.1.5	Protección de rutas.....	142
4.3.1.6	Control por JWT.....	142
4.3.1.7	Cifrado de contraseñas.....	142
4.3.2	<i>Seguridad Física</i>	143
4.3.2.1	Back-Ups de la base de datos	143
4.3.2.2	Firewall	143
4.4	PRUEBA AL SOFTWARE	143
4.4.1	<i>Pruebas unitarias</i>	143
4.4.2	<i>Pruebas de sistema</i>	150
4.4.3	<i>Pruebas de estrés</i>	153
4.4.4	<i>Pruebas de accesibilidad</i>	157

CAPITULO V	159
5. CONCLUSIONES Y RECOMENDACIONES.....	159
5.1 CONCLUSIONES.....	159
5.2 RECOMENDACIONES	160
REFERENCIAS BIBLIOGRAFICAS	161

INDICE DE FIGURAS

Figura 1. Organigrama institucional de la empresa Bi Space SRL	4
Figura 2. Diagrama de recolección por Web Scraping	22
Figura 3. Flujo de la minería de datos	26
Figura 4. Proceso de la metodología CRISP DM	47
Figura 5. Pagina oficial Trovit	69
Figura 6. Pagina Oficial Lamudi	70
Figura 7. Esquema Web_Scraping	73
Figura 8. Esquema Pages_Web	74
Figura 9. Esquema IA	74
Figura 10. Esquema Public	75
Figura 11. División de clases de distintas paginas	76
Figura 12. Función para eliminar duplicados	77
Figura 13. Formato de datos	78
Figura 14. Función para validar datos	79
Figura 15. Diagrama de interacciones con el modelo	80
Figura 16. Importación de herramientas	81
Figura 17. Función para realizar Web Scraping	81
Figura 18. Función de renderizado con Selenium	82
Figura 19. Función PlayWright carga rapida	82
Figura 20. Función de carga con navegador emulado	83
Figura 21. Implementación con Celery	83
Figura 22. Búsqueda de información importante	84
Figura 23. Limpieza de datos	85
Figura 24. Almacenamiento en la base de datos	86
Figura 25. Limpieza de duplicados	87
Figura 26. Completado de información importante	88
Figura 27. Función para evaluar los datos	90
Figura 28. Resultados de la calidad de datos	91
Figura 29. Función del rendimiento del Web Scraping	93
Figura 30. Resultado de la evaluación en tiempo	94
Figura 31. Test para determinar la precisión del modelo	95
Figura 32. Resultados de la evaluación	95
Figura 33. Test Tiempo de ejecución de la IA	96
Figura 34. Resultados del test	96
Figura 35. Configuración del entorno en Docker	97
Figura 36. Declaración de las APIs	98
Figura 37. Configuración del JWT	98
Figura 38. Interfaz de autenticación	99
Figura 39. Interfaz de inicio	100
Figura 40. Menú Lateral	100
Figura 41. Interfaz Visualización de datos	101
Figura 42. Interfaz lista de Registros.	102
Figura 43. Interfaz Creación de dominios	102
Figura 44. Interfaz editar dominios	103
Figura 45. Interfaz Eliminación de Registros	103
Figura 46. Interfaz Recolección de datos	104

Figura 47. Interfaz de visualización el la recolección de información	105
Figura 48. Interfaz de Completado de información	105
Figura 49. Interfaz Visualización de datos recolectados	106
Figura 50. Interfaz historial de datos	106
Figura 51. Interfaz de Análisis de precios	107
Figura 52. Visualización Interfaz de clasificación de propiedades	107
Figura 53. Interfaz Mapa de calor	108
Figura 54. Interfaz de entrenamiento del modelo paso 1	109
Figura 55. Interfaz de entrenamiento del modelo paso 2	109
Figura 56. Interfaz Entrenamiento del modelo paso 3	110
Figura 57. Interfaz de evaluación del modelo paso 1	111
Figura 58. Interfaz de evaluación del modelo paso 2	111
Figura 59. Interfaz de evaluación del modelo paso 3	112
Figura 60. Unittest para la autenticación	144
Figura 61. Resultados de las pruebas de autenticación	144
Figura 62. Unittest para la verificación de los JWT	145
Figura 63. Resultados de las pruebas para validar el JWT	145
Figura 64. Pruebas para realizar Web Scraping	146
Figura 65. Resultados del test de Web Scraping	146
Figura 66. Camino de las funciones para realizar Web Scraping	147
Figura 67. Interfaz Apache JMeter	153
Figura 68. Configuración de los hilos	154
Figura 69 Configuración de peticiones	154
Figura 70 Interfaz de creación de repostes y gráficos	155
Figura 71. Resultados de las peticiones	155
Figura 72. Resultados de las peticiones en tabla	156
Figura 73. Resultado de las peticiones en gráfico	156
Figura 74. Interfaz de axe DevTools v4	157
Figura 75. Resultados de la prueba de accesibilidad	158

INDICE DE TABLAS

Tabla 1. <i>Requerimientos funcionales</i>	66
Tabla 2 <i>Requerimientos no funcionales</i>	67
Tabla 3. <i>Elementos de las páginas web</i>	68
Tabla 4. <i>Tabla de evaluación de herramientas de Web Scraping</i>	71
Tabla 5. <i>Características de funcionalidad</i>	114
Tabla 6. <i>Parámetros de medición</i>	114
Tabla 7. <i>Factores de ponderación</i>	115
Tabla 8. <i>Cálculo de Factores de ponderación</i>	116
Tabla 9. <i>Factores de ajuste de complejidad</i>	117
Tabla 10. <i>Escala de valores de preguntas</i>	122
Tabla 11. <i>Preguntas para determinar la usabilidad</i>	122
Tabla 12. <i>Resultados ISO/IEC 25010</i>	128
Tabla 13. <i>Pasos para ejecutar Web Scraping</i>	148
Tabla 14. <i>Resultados Caja Blanca</i>	149
Tabla 15. <i>Resultados en la evaluación de los caminos</i>	150
Tabla 16. <i>Caja negra recolección de datos</i>	151
Tabla 17. <i>Caja negra Entrenamiento del modelo</i>	151
Tabla 18. <i>Caja negra evaluación de modelos</i>	152
Tabla 19. <i>Caja negra inicio de sesión</i>	152
Tabla 20. <i>Resultados por Apache JMeter</i>	157

INDICE DE ECUACIONES

<i>Ecuación 1 Calidad de datos.....</i>	<i>92</i>
<i>Ecuación 2. Funcionalidad</i>	<i>118</i>
<i>Ecuación 3. Confiabilidad</i>	<i>119</i>
<i>Ecuación 4. Usabilidad.....</i>	<i>121</i>
<i>Ecuación 5. Eficiencia</i>	<i>123</i>
<i>Ecuación 6. Disponibilidad.....</i>	<i>125</i>
<i>Ecuación 7. Portabilidad.....</i>	<i>127</i>
<i>Ecuación 8. Costos por punto de funcion.....</i>	<i>138</i>
<i>Ecuación 9. Duración del desarrollo del software</i>	<i>140</i>
<i>Ecuación 10. Complejidad Ciclomática</i>	<i>148</i>

RESUMEN

Este proyecto se basa en la recolección y completado de información de bienes raíces aplicando técnicas de Web Scraping y procesamiento de lenguaje natural. Su objetivo principal es optimizar la búsqueda y extracción de datos relevantes de páginas web de bienes raíces, como precios, ubicaciones y características de las propiedades. La metodología CRISP-DM guía el desarrollo del proyecto, asegurando un enfoque estructurado desde la recopilación de datos hasta la validación de resultados, y aplicando los estándares de calidad de la ISO/IEC 25010 para evaluar la efectividad y precisión del sistema. Para implementar el sistema, se utilizan herramientas tecnológicas como Python y Flask para el backend, PostgreSQL para la gestión de datos y React con Material-UI para el frontend, lo que permite un procesamiento eficaz y una interfaz intuitiva. El sistema emplea algoritmos de aprendizaje automático y técnicas de procesamiento de lenguaje natural (NLP) para clasificar y localizar datos específicos, automatizando la identificación de información clave para el análisis del mercado inmobiliario. Durante la validación, se llevaron a cabo pruebas funcionales y de rendimiento para evaluar la robustez del sistema en diferentes escenarios de uso. Los resultados muestran que la automatización mediante Web Scraping mejora considerablemente la eficiencia en la recopilación de datos y reduce el tiempo y los recursos necesarios para obtener información relevante, brindando a Bi Space SRL una ventaja competitiva en el sector. Este sistema sienta las bases para futuras implementaciones de Web Scraping y aprendizaje automático en el ámbito inmobiliario, destacando el potencial de estas tecnologías para optimizar la gestión y análisis de datos comerciales.

Palabras clave: Web Scraping, Procesamiento de lenguaje natural (NLP). CRISP-DM, ISO/IEC 25010

ABSTRACT

This project is based on the collection and completion of real estate information by applying Web Scraping and Natural Language Processing techniques. Its main objective is to optimize the search and extraction of relevant data from real estate web pages, such as prices, locations, and property characteristics. The CRISP-DM methodology guides the development of the project, ensuring a structured approach from data collection to results validation, and applying ISO/IEC 25010 quality standards to evaluate the effectiveness and accuracy of the system. To implement the system, technological tools such as Python and Flask are used for the backend, PostgreSQL for data management, and React with Material-UI for the frontend, allowing for efficient processing and an intuitive interface. The system employs machine learning algorithms and Natural Language Processing (NLP) techniques to classify and locate specific data, automating the identification of key information for real estate market analysis. During validation, functional and performance tests were carried out to evaluate the robustness of the system in different usage scenarios. The results show that automation through Web Scraping significantly improves the efficiency of data collection and reduces the time and resources required to obtain relevant information, giving Bi Space SRL a competitive advantage in the sector. This system lays the foundation for future implementations of Web Scraping and machine learning in the real estate sector, highlighting the potential of these technologies to optimize the management and analysis of commercial data.

Keywords: Web Scraping, Natural Language Processing (NLP). CRISP-DM, ISO/IEC 25010

CAPITULO I

MARCO PRELIMINAR

**INGENIERÍA
DE SISTEMAS**
UNIVERSIDAD PUBLICA DE EL ALTO

CAPITULO I

1. MARCO PRELIMINAR

1.1 INTRODUCCION

La industria en bienes raíces está en constante evolución, impulsada por el acceso a grandes volúmenes de información disponible en la web. En este contexto, la extracción de datos se ha vuelto crucial para comprender el mercado y tomar decisiones informadas. Bi Space SRL, una empresa de desarrollo, busca aprovechar al máximo esta oportunidad mediante el desarrollo de un sistema avanzado de reconocimiento comercial respaldado por inteligencia artificial (IA).

El proyecto se centra en abordar los desafíos actuales en la búsqueda y extracción de información en páginas web comerciales, específicamente en el mercado inmobiliario en español. Los problemas incluyen la poca de precisión y fiabilidad de los datos extraídos manualmente o con herramientas no especializadas, la diversidad de fuentes de datos con formatos y actualizaciones inconsistentes. Estos desafíos limitan la efectividad y eficiencia del Web Scraping¹ en el sector inmobiliario.

Los objetivos del proyecto son claros: automatizar las tareas complejas de análisis de datos para esto se creará un sistema que sea eficiente, adaptable y capaz de manejar grandes volúmenes de datos de manera rápida y precisa, optimizando así la eficiencia operativa de Bi Space SRL y sus clientes.

¹ **Web Scraping** es el proceso automatizado de extraer información de sitios web. Utiliza programas o scripts para acceder a páginas web y recopilar datos específicos de manera estructurada.

Con la implementación se podrá acceder y analizar información valiosa de manera eficiente. Al automatizar la extracción y el análisis de datos, se mejorará significativamente la transparencia y la confianza en las transacciones inmobiliarias.

1.2 ANTECEDENTES

1.2.1 *Antecedentes institucionales*

1.2.1.1 Misión

En Bi Space SRL, nuestra misión es impulsar la innovación a través del desarrollo y la implementación de MVPs (Productos Mínimos Viables) con enfoque tecnológico.

Nos dedicamos a mejorar continuamente la experiencia del usuario y del cliente final, utilizando un enfoque multidisciplinario que integra la experiencia de ingenieros de software, geógrafos, ingenieros de datos, científicos de datos y profesionales de marketing.

Buscamos transformar ideas en soluciones tangibles, contribuyendo así al progreso tecnológico y a la creación de experiencias excepcionales.

1.2.1.2 Visión

Nuestra visión en Bi Space SRL es ser líderes reconocidos en el desarrollo tecnológico, destacándonos por la creación constante de soluciones innovadoras que impacten positivamente la vida de nuestros usuarios.

Aspiramos a ser referentes en la industria, tanto por la calidad de nuestros productos como por nuestro compromiso con la excelencia y la mejora continua.

Buscamos no solo anticipar las necesidades del mercado, sino también influir en la forma en que la tecnología mejora la vida diaria, transformando ideas audaces en realidades exitosas.

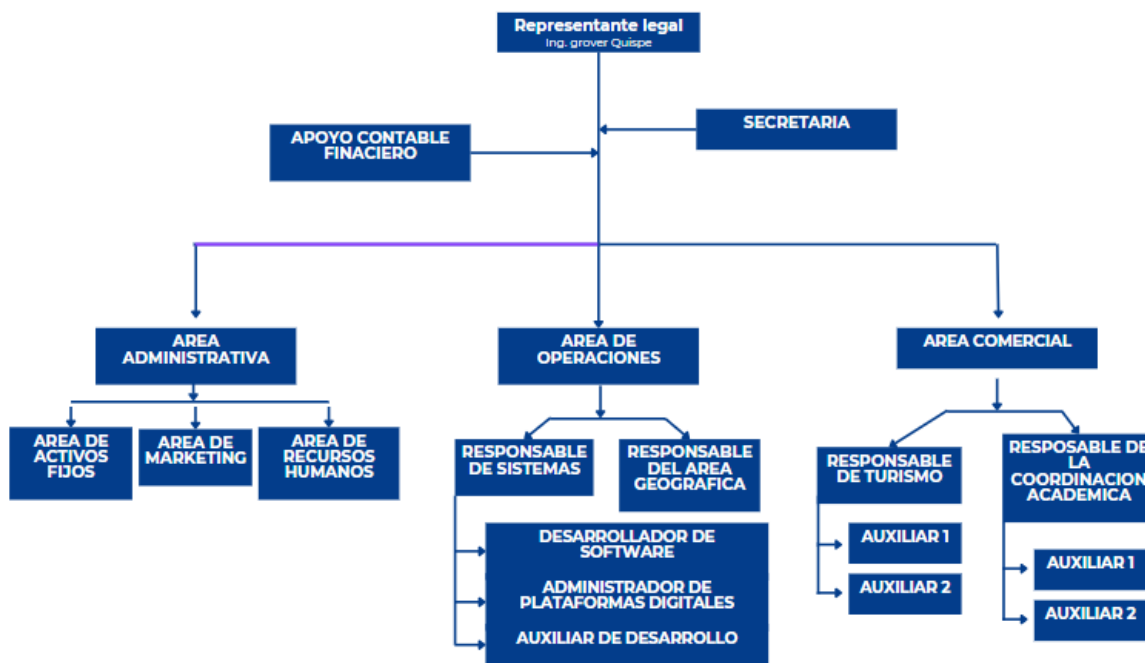
1.2.1.3 Objetivo

Bi Space SRL se enfoca en ofrecer soluciones digitales integrales que combinan la innovación tecnológica con una sólida experiencia en el desarrollo web, geomarketing y gestión catastral. Su propósito principal es transformar ideas en experiencias digitales impactantes, optimizando la presencia en línea y mejorando la toma de decisiones empresariales a través de soluciones innovadoras y personalizadas.

1.2.1.4 Organigrama

El proyecto se centra en desarrollar un sistema para el área de operaciones que tendrá como encargado al responsable de sistemas que supervisará la información extraída de diferentes páginas web de bienes raíces. Gracias a esto generara mejores resultados en la obtención de datos en bienes raíces

Figura 1.
Organigrama institucional de la empresa Bi Space SRL



Nota: El organigrama institucional representa la estructura jerárquica de la organización, destacando los principales departamentos y sus interrelaciones. Este esquema ayuda a entender la distribución de responsabilidades y la línea de autoridad dentro de la entidad.

1.2.2 Antecedentes afines a la investigación

1.2.2.1 Antecedentes Internacionales

- Revista científica “La necesidad de una herramienta de software para la automatización de la búsqueda, selección y descarga de bases de datos en la web mediante técnicas de Web Scraping” escrito por Mijangos Espinoza C., Lopez (2022), la idea central de este artículo es la necesidad de contar con una herramienta de software que utilice técnicas de Web Scraping para automatizar la búsqueda, selección y descarga de bases de datos alojadas en la web. La creciente cantidad de información en línea hace que la captura manual de datos sea lenta y complicada, lo que dificulta su uso en investigaciones y análisis de información. El sistema propuesto busca resolver este problema, permitiendo

la creación de un compendio de datos que puede ser utilizado eficientemente para análisis de información.

- Proyecto de grado “Diseño de una solución IT para el análisis de sentimientos frente a la opinión de las figuras públicas usando técnicas de Web Scraping y la Twitter API” escrito por Álvarez Correa Mercado, R., y Rodríguez González, J. (2022) este proyecto trata en abordar el constante debate político en Colombia, que se manifiesta ampliamente en Twitter, y resolver el desafío de organizar y comprender la gran cantidad de información no estructurada que se genera en esta plataforma. Para lograrlo, el equipo utiliza la Twitter API y técnicas de Machine Learning² para analizar y clasificar opiniones sobre figuras políticas según los sentimientos asociados a ellas.

1.2.2.2 Antecedentes Nacionales

- Tesis de Grado “Traductor automático español-quechua, basado en el procesamiento del lenguaje natural”, escrito por Flores Morales (2021) en la Universidad Mayor de San Andrés, el trabajo presenta un traductor automático entre los idiomas español y quechua utilizando el Procesamiento de Lenguaje Natural. Se optó por un modelo de Traducción Automática basada en redes Neuronales (NMT), implementando dos corpus paralelos con 1,919 frases alineadas en el algoritmo Transformer para el entrenamiento y aprendizaje de las redes neuronales. Se utilizaron herramientas básicas de procesamiento de lenguaje natural como tokenizadores, codificadores, normalizadores y decodificadores, siguiendo la metodología del Proceso Cíclico expuesta en el

² **Machine Learning** es una rama de la inteligencia artificial que permite a las máquinas aprender y mejorar automáticamente a partir de datos, sin ser explícitamente programadas para ello.

marco teórico. Aunque los resultados no son óptimos debido al limitado corpus bilingüe disponible, son aceptables. El trabajo concluye que el uso del quechua, como lengua aglutinada, en sistemas NMT debe ser más estudiado, pero abre posibilidades para traducir otros idiomas similares como el aimara o guaraní.

- Proyecto de grado “Método multivariado aplicado al análisis de textos de periódicos” escrito por Mayta Callisaya (2022) en la Universidad Mayor de San Andrés, el trabajo se enfoca en aplicar métodos multivariados de conglomerados y de correspondencias a textos de titulares de periódicos para un análisis estadístico sofisticado. Se recopilan automáticamente textos de titulares de periódicos de julio de 2019 a julio de 2020, clasificándolos en categorías temáticas y periodos temporales. El objetivo es crear una base de datos estructurada para analizar gráficamente los textos y las palabras, destacando el plano de máxima inercia como punto central de análisis.
- Tesis Magister Scientiarum “Modelo de análisis de normas de gobiernos autónomos, basado en minería de textos e inteligencia artificial” escrito por Oblitas Villegas (2020) en la universidad Mayor de San Andrés, en este trabajo se investiga la aplicación de técnicas de aprendizaje automático (Machine Learning) para mejorar la detección de temas en una base de datos compuesta por normas bolivianas de diferentes Gobiernos Autónomos (GA) desde 2010 hasta 2019. La base de datos se utiliza para modelar las 274 competencias distribuidas a los GA según la Constitución Política del Estado. Se emplean métodos como la Asignación de Dirichlet Latente (LDA) y Doc2vec para la detección de temas, y se comparan con redes neuronales Long Short Term Memory (LSTM) para evaluar su eficiencia en el análisis competencial de las

normas. Los resultados sugieren que el modelo propuesto mejora la precisión en la identificación de competencias dentro de la normativa boliviana.

1.2.2.3 Antecedentes Locales

- Proyecto de grado “Sistema web a la recolección de información de medio digitales mediante técnicas de Web Scraping”, escrito por Aroa Miranda (2020) en la Universidad Pública de El Alto, el proyecto propone la creación de un Sistema Web orientado a la recopilación de noticias de medios digitales legalmente establecidos en el país de origen. Para lograrlo, se emplean técnicas de Web Scraping, permitiendo así la recopilación, almacenamiento y búsqueda eficiente de noticias diarias. Este sistema se desarrolla utilizando herramientas de software libre, que abarcan lenguajes de programación, bases de datos, un editor de texto y un servidor web.
- Proyecto de grado “Sistema automatizado para la comunicación y gestión de correspondencia mediante tecnologías workflow³” escrito por Fernández Huanca (2020) en la Universidad Pública de El Alto, el proyecto se centra en la implementación de un sistema de recolección y análisis de titulares de noticias de periódicos digitales, utilizando métodos avanzados de análisis de datos textuales y gráficos. Se recopilan titulares de tres fuentes de noticias durante un año, categorizándolos en temas de salud, política y economía, y dividiéndolos en dos períodos temporales: antes y después de la cuarentena.

³ **Workflow** es una rama de la inteligencia artificial que permite a las máquinas aprender y mejorar automáticamente a partir de datos, sin ser explícitamente programadas para ello.

El objetivo es proporcionar una forma sofisticada de resumir y comprender grandes volúmenes de información textual de manera visualmente efectiva.

1.3 PLANTEAMIENTO DEL PROBLEMA

El uso cada vez más extendido del Web Scraping en el ámbito inmobiliario refleja la necesidad de acceder a datos precisos y relevantes en un entorno digital saturado de información. Sin embargo, este método se enfrenta a desafíos que limitan su efectividad y eficiencia.

Uno de los problemas críticos es la poca precisión y fiabilidad en los datos extraídos, ya sea manualmente o con herramientas no especializadas, lo que afecta la calidad de las decisiones tomadas con base en dichos datos. Además, la diversidad de fuentes de información y la inconsistencia en sus formatos y actualizaciones complican la integración y el análisis coherente de los datos extraídos.

Por otro lado, el desgaste de recursos humanos debido a las deficientes herramientas automatizadas para el Web Scraping conlleva a una pérdida de tiempo y productividad, afectando negativamente la motivación y el bienestar del personal. La dificultad en la evaluación de la disponibilidad de información en páginas web limita la capacidad de las empresas para aprovechar oportunidades de mercado de manera ágil y efectiva.

1.3.1 Problema Principal

En la empresa Bi Space SRL, las dificultades en la búsqueda y extracción de información en páginas web de bienes raíces generan retrasos significativos en las tareas, además de ocasionar problemas de incoherencia en la información al momento del envío.

1.3.2 Problema Secundarios

- La extracción manuscrita o con herramientas no especializadas puede resultar en datos inexactos o incompletos, afectando la toma de decisiones en la comercialización de bienes raíces
- Los datos obtenidos de diferentes fuentes pueden ser inconsistentes en formato, calidad y actualización, lo que dificulta su integración y análisis.
- Las herramientas tradicionales de Web Scraping pueden tener dificultades para acceder y extraer datos de sitios web complejos o dinámicos, lo que limita la capacidad de recolectar información clave.
- Las herramientas actuales son deficientes en la extracción de información, lo que obliga a los empleados a dedicar tiempo excesivo a tareas repetitivas y tediosas.
- La integración de datos de múltiples fuentes en un sistema coherente puede ser complicada sin herramientas adecuadas de procesamiento y normalización, lo que causa datos no confiables y generar conclusiones erróneas, afectando la toma de decisiones y la calidad general del proyecto.

1.3.3 Formulación del Problema

¿De qué manera se optimizará la búsqueda y extracción de datos en páginas web a través del sistema de Web Scraping de reconocimiento comercial en bienes raíces en la empresa Bi Space SRL?

1.4 OBJETIVOS

1.4.1 *Objetivo General*

Desarrollar un sistema Web Scraping de Reconocimiento Comercial en Bienes Raíces Respaldo por Inteligencia Artificial de manera que optimice la búsqueda y extracción de datos en páginas web en la empresa Bi Space SRL.

1.4.2 *Objetivos Específicos*

- Identificar las principales fuentes de información de bienes raíces en línea para en análisis de contenido de las publicaciones, incluyendo tipos de propiedades, precios, ubicaciones, características y tendencias.
- Evaluar las herramientas y tecnologías de Web Scraping disponibles en el mercado definiendo criterios de búsqueda específicos, como fuentes de datos, frecuencia de actualización, calidad de la información y relevancia.
- Elaborar algoritmos avanzados de extracción de datos que permitan obtener información de forma precisa y eficiente desde diversas páginas web de bienes raíces.
- Aplicar algoritmos de aprendizaje automático para clasificar y localizar datos relevantes para el reconocimiento comercial de bienes raíces.
- Diseñar una interfaz intuitiva que permita a los usuarios ingresar parámetros de búsqueda, como ubicación, tipo de propiedad y rango de precios.
- Elaborar las pruebas de funcionalidad que incluya casos de uso realistas y escenarios diversos.

1.5 JUSTIFICACION

1.5.1 *Justificación técnica*

La integración de inteligencia artificial, incluyendo aprendizaje automático y procesamiento de lenguaje natural, en el sistema de Bi Space SRL mejorará la calidad y relevancia de los datos de bienes raíces. Dado que la institución ya dispone de hardware, software, hosting y dominio, la implementación de estos algoritmos será más eficiente, aprovechando la infraestructura existente para automatizar la identificación de patrones y optimizar el reconocimiento comercial. Esta base tecnológica facilitará la adaptabilidad continua del sistema, asegurando un rendimiento a largo plazo, liberando recursos y mejorando la efectividad estratégica de Bi Space SRL.

1.5.2 *Justificación económica*

La implementación de inteligencia artificial en Bi Space SRL tiene una sólida justificación económica que eliminará la dependencia de servicios externos para análisis y procesamiento de datos, reduciendo costos operativos a largo plazo. La automatización mejorará la eficiencia del equipo y optimizará la asignación de recursos, identificando nuevas oportunidades de negocio y maximizando inversiones. Además, reducirá errores y promueve una gestión de riesgos más efectiva, mejorando la eficiencia operativa en general.

1.5.3 *Justificación social*

La inteligencia artificial aplicada al Web Scraping de Bi Space SRL democratiza el acceso a información inmobiliaria, beneficiando tanto al responsable de la búsqueda como a la sociedad en general. Mejorará la confianza en las transacciones y promoverá la responsabilidad social, posicionando a Bi Space como líder en innovación tecnológica y contribuyendo al desarrollo equitativo y sostenible de la comunidad.

1.6 METODOLOGIA

1.6.1 Metodología de desarrollo

La metodología CRISP-DM (Cross-Industry Standard Process for Data Mining) se seleccionó para el proyecto debido a su estructura robusta y su enfoque sistemático en el desarrollo de proyectos de minería de datos e inteligencia artificial. Esta metodología se ha establecido como un estándar en la industria debido a sus etapas bien definidas y su capacidad para adaptarse a diferentes proyectos y sectores.

CRISP-DM se enfoca en comprender el negocio y los objetivos comerciales desde el principio, lo cual es crucial para el éxito de un proyecto como el desarrollo de un sistema de reconocimiento comercial en bienes raíces. Permite identificar claramente las necesidades de información, los requisitos técnicos y las expectativas de los stakeholders⁴. Además, su enfoque iterativo permite ajustar y mejorar los procesos y modelos a lo largo del proyecto, lo que es fundamental en proyectos de inteligencia artificial donde la experimentación y la optimización son clave (CRISP-DM Consortium, 1999).

1.6.2 Métricas de Calidad

Las métricas de calidad según la norma ISO/IEC 25010 son fundamentales para evaluar un sistema de inteligencia artificial. Estas métricas abarcan aspectos como funcionalidad, confiabilidad, usabilidad, eficiencia del desempeño, mantenibilidad y portabilidad del sistema. Al cumplir con requisitos funcionales, corregir errores y garantizar facilidad de uso, el sistema se vuelve robusto y adaptable en diferentes contextos. Esta

⁴ **Stakeholders** son las personas o grupos que tienen un interés o se ven afectados por un proyecto, decisión o actividad de una organización. Incluyen a empleados, clientes, proveedores, inversores y la comunidad en general.

evaluación exhaustiva permite identificar mejoras y asegurar que el sistema opere de manera efectiva y eficiente (International Organization for Standardization. 2011).

1.6.3 Costo

Para estimar los costos del proyecto, se puede utilizar el método COSMIC, que se centra en medir el tamaño funcional del software y calcular los recursos necesarios para su desarrollo. Este enfoque considera diferentes aspectos como los costos de personal, infraestructura tecnológica, herramientas de desarrollo, capacitación y gestión del proyecto. Al aplicar el método COSMIC, se obtiene una visión detallada de los recursos requeridos en cada etapa del proyecto, desde la definición de requisitos hasta la implementación y el mantenimiento del sistema de inteligencia artificial. Esto permite una planificación financiera más precisa y una asignación eficiente de recursos para garantizar el éxito del proyecto dentro de los límites presupuestarios establecidos (COSMIC, 2015).

1.6.4 Seguridad

La norma ISO/IEC 27001 establece los lineamientos para desarrollar y mantener un sistema de gestión de seguridad de la información (SGSI) en una organización. Su propósito es proteger la información sensible, garantizando su confidencialidad, integridad y disponibilidad. Este estándar ayuda a identificar riesgos, implementar controles y políticas de seguridad, y gestionar incidentes, asegurando la protección de datos y la continuidad del negocio (International Organization for Standardization. 2013).

1.6.5 Pruebas de Software

- **Pruebas de sistema:** Son un enfoque en el que se evalúa el comportamiento de un sistema sin conocer su estructura interna. En lugar de examinar el código fuente, se centran en las entradas y salidas del sistema, así como en su funcionalidad externa. Este tipo de pruebas se basa en escenarios de uso realistas para verificar si el software cumple con los requisitos especificados, sin requerir conocimientos detallados de su implementación interna (Myers, G. J., Sandler, C., y Badgett, T. ,2011).
- **Pruebas unitarias:** Este tipo de pruebas evalúa unidades individuales de software, como funciones, métodos o módulos, de manera aislada para verificar que funcionen correctamente según lo esperado. Se realizan a nivel de código y se centran en probar la lógica y funcionalidad específica de cada unidad. Las pruebas unitarias ayudan a detectar errores en etapas tempranas del desarrollo y suelen implementarse utilizando frameworks como unittest en Python o Jest en JavaScript. Son fundamentales para garantizar la calidad del software y facilitar el mantenimiento y la evolución del código (Myers, G. J., Sandler, C., & Badgett, T., 2011).
- **Pruebas de estrés:** Las pruebas de estrés, también conocidas como pruebas de carga, evalúan la capacidad de un sistema para manejar condiciones extremas y sobrecargas. Estas pruebas simulan situaciones de alta demanda de recursos, como picos de tráfico o procesamiento intensivo, para determinar el rendimiento del sistema bajo presión máxima. Al medir la respuesta del sistema ante estas condiciones extremas, las pruebas de estrés identifican posibles cuellos de botella, fallos o degradaciones de rendimiento, permitiendo

así optimizar la capacidad y la estabilidad del sistema en entornos de uso intensivo (Sharma, A., y Sharma, P. ,2018).

- **Pruebas de Accesibilidad:** Estas pruebas evalúan si un software o una aplicación web es accesible para personas con discapacidades, asegurando el cumplimiento de estándares como las Pautas de Accesibilidad para el Contenido Web (WCAG). Se centran en verificar aspectos como la navegación mediante teclado, la compatibilidad con lectores de pantalla, el contraste de colores y el uso adecuado de etiquetas semánticas. Estas pruebas pueden realizarse manualmente, utilizando herramientas automatizadas como Axe o Lighthouse, o mediante evaluaciones realizadas por usuarios con diferentes discapacidades. Garantizan que el software sea inclusivo y cumpla con normativas legales y éticas (Myers, G. J., Sandler, C., & Badgett, T., 2011).

1.7 MÉTODOS DE RECOLECCIÓN DE DATOS

La recolección de datos en el contexto de un sistema de Web Scraping respaldado por inteligencia artificial se basa en la extracción automatizada de información de sitios web específicos del sector inmobiliario. Este proceso implica el uso de herramientas de Scraping para realizar solicitudes HTTP a las páginas web, extrayendo y procesando los datos relevantes, como descripciones de propiedades, precios y características, que posteriormente se almacenan en una base de datos estructurada. La metodología garantiza la normalización y limpieza de los datos, asegurando su calidad en el análisis y el entrenamiento de modelos de IA. (Mitchell, 2018; Han, Kamber, y Pei, 2012).

1.7.1 Técnicas de Investigación

En el contexto de un sistema de Web Scraping respaldado por inteligencia artificial, las técnicas de investigación utilizadas se centran en la minería de datos en sitios web inmobiliarios. Estas técnicas incluyen la extracción de datos no estructurados y semiestructurados, el uso de algoritmos de aprendizaje automático el completado de información (Witten, Frank, y Hall, 2017; Hastie, Tibshirani, y Friedman, 2009).

1.8 HERRAMIENTAS

1.8.1 Desarrollo de la Inteligencia Artificial con Python y Flask

El desarrollo de la inteligencia artificial (IA) se llevará a cabo utilizando Python como lenguaje principal de programación. Python es ampliamente reconocido por su versatilidad y su amplia gama de bibliotecas y herramientas para el aprendizaje automático y el procesamiento de datos. En particular, se utilizará Flask como framework⁵ web para el desarrollo de la IA. Flask es conocido por ser ligero y fácil de usar, lo que facilita la creación de aplicaciones web eficientes y escalables (Grus, J. ,2019).

1.8.2 Procesamiento de lenguaje natural (NLP)

Dentro del ámbito del procesamiento de lenguaje natural (NLP), se aplicarán técnicas y herramientas avanzadas para analizar y comprender datos textuales. Estas herramientas permiten extraer información significativa de texto no estructurado, como descripciones de propiedades inmobiliarias (Jurafsky, D., y Martin, J. H.,2023).

⁵ **Framework** es una estructura de software predefinida que proporciona un conjunto de herramientas y componentes reutilizables para facilitar el desarrollo de aplicaciones. Simplifica tareas comunes y estandariza el proceso de desarrollo.

1.8.3 Base de datos Geoespacial con PostgreSQL y PostGIS

Para gestionar datos geoespaciales de manera eficiente, se utilizará PostgreSQL como sistema de gestión de bases de datos. PostgreSQL es una opción robusta y escalable que ofrece capacidades avanzadas para el almacenamiento y manipulación de datos. Además, se aprovechará la extensión PostGIS de PostgreSQL, diseñada específicamente para trabajar con datos geoespaciales, lo que facilitará el análisis y la visualización de información relacionada con ubicaciones geográficas (Obe, R., y Hsu, L. ,2015).

1.8.4 Frontend interactivo con React y Material-UI

En cuanto al frontend⁶ de la aplicación, se empleará React como framework de JavaScript para crear una interfaz de usuario interactiva y dinámica. React es conocido por su eficiencia en la gestión de componentes y su capacidad para actualizar la interfaz de manera rápida y fluida. Para estilizar y diseñar la interfaz, se utilizará Material-UI, una biblioteca de componentes basada en React que ofrece una amplia gama de elementos predefinidos y estilos modernos para mejorar la experiencia del usuario (Wojcik, O., y Gupta, A. ,2021)

1.8.5 Docker

Docker es una plataforma de código abierto que automatiza la implementación de aplicaciones dentro de contenedores, lo que permite ejecutar aplicaciones en entornos aislados. Estos contenedores son ligeros, portátiles y consistentes, lo que facilita la ejecución de aplicaciones en diferentes entornos sin importar las diferencias de infraestructura. Docker utiliza imágenes para crear contenedores, y cada contenedor incluye todo lo necesario para

⁶ **Frontend** es la parte de una aplicación web o móvil con la que los usuarios interactúan directamente, abarcando diseño, estructura y comportamiento en el navegador o dispositivo.

ejecutar la aplicación, como el sistema operativo, las bibliotecas y las dependencias. Esto asegura que la aplicación se ejecute de la misma manera en cualquier entorno, desde desarrollo hasta producción (Merkel, 2014).

1.8.6 Redis

Redis es un sistema de gestión de bases de datos en memoria, basado en estructura de datos clave-valor, que se utiliza principalmente como base de datos, caché y agente de mensajes. Su alto rendimiento, facilidad de uso y flexibilidad lo convierten en una herramienta muy popular para mejorar la velocidad y eficiencia de las aplicaciones. Redis admite varias estructuras de datos como cadenas, listas, conjuntos, hashes y conjuntos ordenados, lo que le permite manejar datos de diferentes maneras (Antirez, 2013).

1.9 LIMITES Y ALCANCES

1.9.1 Limites

El proyecto enfrenta limitaciones tecnológicas relacionadas con la capacidad de procesamiento de la infraestructura utilizada, el tiempo requerido para entrenar modelos de IA complejos y la disponibilidad de recursos computacionales necesarios. Estas limitaciones pueden influir en la velocidad y la complejidad de la implementación de la inteligencia artificial en el proceso de extracción aplicado Web Scraping.

Además, existen limitaciones en el acceso a datos relevantes, especialmente aquellos actualizados y de alta calidad. La disponibilidad de información actualizada y precisa es crucial para el éxito de la extracción de datos y el análisis realizado por el sistema de inteligencia artificial. Asimismo, se deben considerar las restricciones de acceso a datos sensibles o

protegidos por privacidad, lo que puede requerir autorización y cumplimiento de normativas legales.

En cuanto a la implementación y el mantenimiento, se enfrentan desafíos en términos de recursos humanos capacitados para trabajar con tecnologías específicas, la integración de diferentes componentes del sistema y la necesidad de actualizaciones continuas para mantener la relevancia de los algoritmos de IA. Es importante mencionar que el enfoque del proyecto estará en páginas web de tipo comercial en español, lo que implica adaptaciones y consideraciones específicas para el análisis de datos en este contexto lingüístico y comercial.

1.9.2 Alcances

A continuación, se describen los alcances del proyecto

- **Desarrollo de Herramientas de Extracción de Datos:** El proyecto abarcará el desarrollo de herramientas de extracción de datos basadas en inteligencia artificial para la obtención de la información específica de páginas web comerciales en español. Esto incluye la implementación de técnicas de Web Scraping y procesamiento de lenguaje natural para recopilar datos relevantes relacionados con bienes raíces.
- **Integración con Plataformas Existentes:** Se realizará la integración de las herramientas desarrolladas con las plataformas existentes de Bi Space SRL, garantizando la compatibilidad y el funcionamiento eficiente del sistema. Además, se brindará soporte para la visualización y presentación de los datos procesados de manera clara y comprensible para los usuarios finales.

- **Desarrollo de Interfaces intuitivas:** Se crearán interfaces de usuario que permitan a los usuarios de Bi Space SRL interactuar fácilmente con las herramientas de extracción y análisis de datos. Esto incluye el diseño de formularios de entrada de datos, visualización de resultados
- **Capacidades de Monitoreo:** El sistema contará con capacidades de monitoreo para asegurar su rendimiento y eficiencia a lo largo del tiempo. Esto incluye la detección de posibles errores y la actualización de datos para mantener la información relevante y actualizada.
- **Seguridad y Privacidad de los Datos:** Se implementarán medidas de seguridad y privacidad para proteger la integridad y confidencialidad de los datos recopilados y procesados. Esto involucra el uso de protocolos de encriptación, gestión de accesos y cumplimiento de normativas de protección de datos para garantizar la confianza de los usuarios y la legalidad del proyecto.

1.10 APORTES

La inteligencia artificial (IA) aporta una serie de beneficios clave al proyecto de extracción de datos inmobiliarios de Bi Space SRL. Con la IA, es posible automatizar procesos complejos asociados con la extracción de información de páginas web comerciales en español, agilizando el análisis de datos y reduciendo la carga de trabajo manual. Además, la IA permite un análisis avanzado de datos, identificando patrones y tendencias ocultas en el mercado inmobiliario y generando insights⁷ valiosos para la toma de decisiones estratégicas. También contribuye a optimizar los resultados obtenidos, garantizando la calidad, relevancia y precisión

⁷ **Insights** son percepciones profundas o entendimientos útiles que se derivan del análisis de datos o información, proporcionando una comprensión nueva o perspicaz sobre un tema específico.

de la información extraída, y ofrece capacidades de personalización y adaptabilidad para satisfacer las necesidades específicas de Bi Space y sus usuarios.

CAPITULO II

MARCO TEÓRICO

**INGENIERÍA
DE SISTEMAS**
UNIVERSIDAD PUBLICA DE EL ALTO



CAPITULO II

2. MARCO TEÓRICO

2.1 Web Scraping

El Web Scraping es descrito por García (2021) en el sitio web “Web Scraping Techniques and Tools” como una técnica que permite la extracción automatizada de información de sitios web. Se utiliza para extraer grandes cantidades de datos, que luego se pueden ordenar en bases de datos o archivos CSV. Este proceso incluye el uso de bots⁸ o programas para recorrer páginas web y obtener datos específicos simulando el comportamiento que cualquier otro usuario humano realizaría manualmente. Ha ganado mucha relevancia en sectores que necesitan información en tiempo real como el mercado inmobiliario, donde los precios y las características pueden cambiar rápidamente por sitios web para evitar la recopilación de datos no autorizada.

El término "Web Scraping" está compuesto por las palabras web (que hace referencia a la red de sitios accesibles en Internet) y Scraping (que en inglés significa raspado), lo que sugiere la idea de "raspar" la información de los sitios web para extraerla. Como tecnología, el Web Scraping está en constante evolución, adaptándose a las restricciones impuestas por los sitios web para evitar la recolección no autorizada de datos.

La aplicación de Web Scraping para el reconocimiento comercial en bienes raíces está diseñada para automatizar la extracción de información clave de múltiples fuentes en línea, recopilando datos como precios, disponibilidad, características de las propiedades, y otros

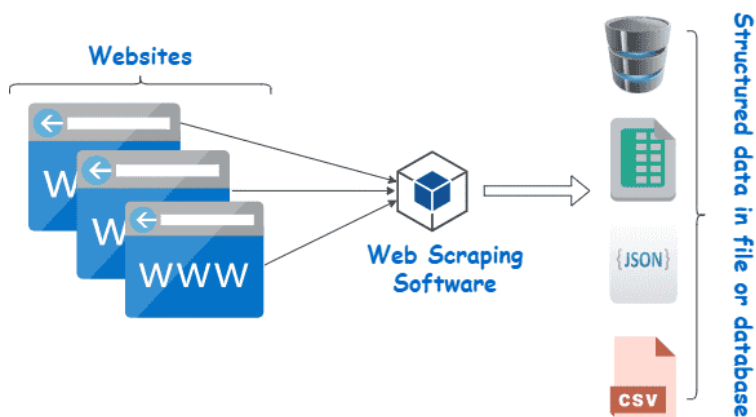
⁸ **Bots** Un bot es un programa automatizado que realiza tareas repetitivas de manera eficiente, como responder preguntas o recolectar datos.

detalles relevantes. Esta información se almacena en una base de datos estructurada, lo que permite realizar análisis detallados en diversas ubicaciones.

Posteriormente, estos datos son utilizados en estudios de mercado para identificar patrones y tendencias que revelen oportunidades de negocio en ubicaciones específicas. Esto proporciona a las empresas una ventaja estratégica, al permitirles ajustar sus ofertas y enfoques de acuerdo con la demanda del mercado en tiempo real. Así, se agiliza el proceso de toma de decisiones y se mejora la capacidad para detectar áreas de inversión y expansión con mayor precisión y rapidez.

Figura 2.

Diagrama de recolección por Web Scraping



Nota: El diagrama explica una base de cómo se trabaja con Web Scraping almacenando datos en archivos de base de datos. Extraído de Kista (2024)

2.3 Minería de Datos (Data Mining)

Mendoza, J. (2018), en el sitio web “Introducción a la Minería de Datos”, dice lo siguiente:

La minería de datos es el proceso utilizado para extraer patrones y encontrar información significativa de grandes cantidades de datos. Este campo aplica técnicas estadísticas, algoritmos y aprendizaje automático para el análisis de datos complejos con el fin de extraer conocimiento que sea beneficioso. La minería de datos encuentra aplicación en diferentes dominios como el marketing, la salud, la economía y la investigación científica al permitir a las organizaciones reconocer tendencias, pronosticar comportamientos futuros y tomar decisiones basadas en datos.

El término “minería de datos” combina la idea de “minería”, que implica la extracción de recursos valiosos, con “datos”, refiriéndose a la información que se extrae de grandes conjuntos de datos. La minería de datos implica procesos como la clasificación, la regresión, la segmentación y la asociación, cada uno con el objetivo de revelar patrones y relaciones ocultas en los datos. Esta técnica es fundamental en la era digital, donde la cantidad de información disponible es vasta y su análisis puede proporcionar una ventaja competitiva significativa.

La minería de datos no es un proceso fácil; implica muchas etapas importantes que conducen a la extracción de información de grandes conjuntos de datos. En cada paso, se debe tener sumo cuidado para llegar a un análisis de datos adecuado, relevante y útil. Los procesos comunes involucrados en la minería de datos son la preparación de los datos, así como la selección y transformación de los mismos; luego siguen el modelado, la evaluación y la interpretación.

Una etapa muy importante es la preparación de datos. Es la primera etapa de la minería de datos y posiblemente la más importante. En esta etapa, los datos crudos se recogen y limpian. Según Han, Kamber y Pei: “la preparación de datos incluye tareas, como eliminar registros duplicados, realizar manejo de casos con valores que faltan y valorar la necesidad de la normalización de datos para asegurar su consistencia y almacenamiento mejorado en la base de datos”. Todos aquellos preparativos garantizan que los datos se analicen correctamente y pueden ser sometidos a la minería avanzada.

La selección de datos consiste principalmente en la identificación de los datos necesarios para el análisis en función del propósito del proyecto. Berry y Linoff describen esta fase como la “selección de unos pocos registros o variables y la extracción de subconjuntos específicos de datos de un plato grande de datos para el modelado analítico”. Una selección adecuada de datos es importante de manera crítica, ya que permite centrarse tan estrechamente como sea posible en los puntos de datos más significativos sin enterrarlos debajo de los no esenciales que pueden confundir la imagen.

La transformación de los datos, por otro lado, se refiere a las técnicas aplicadas para convertir los datos seleccionados en un formato que sea aceptable para el análisis. La transformación de datos puede ser tan sencilla como la agregación de datos o la derivación de una nueva variable o tan complicada como transformar los datos de una coordenada a otra. La idea detrás de la transformación es asegurarse de que los datos estén en un formato que permita que los modelos analíticos sean más efectivos

En esta etapa, se aplican los algoritmos de minería de datos controlando para descubrir patrones y relaciones en los datos. Hay cuatro tipos de modelos: modelos de clasificación,

clustering⁹, regresión y reglas de asociación. Cada tipo se utiliza para un propósito específico y se elige en función de lo que se está intentando lograr con análisis. Por ejemplo, con los modelos de clasificación, se deben predecir categorías o clases. Sin embargo, con el modelo clustering, se organizan los datos en categorías similares sin etiquetas predefinidas (Jain, Murty, y Flynn, 1999).

Una vez que has diseñado tu modelo, es hora de ver qué tan bien funciona. Imagina que es como probar un nuevo auto antes de salir a la carretera. Quieres asegurarte de que funcione correctamente, ¿verdad? Bueno, en este caso, evaluar el rendimiento del modelo es como hacer una prueba de manejo.

Esta evaluación implica verificar si los resultados que obtienes son precisos y confiables. Es como revisar que el auto frene cuando lo necesitas y acelere sin problemas. Utilizamos métricas como la precisión, que es como medir cuánto se acerca el modelo a la realidad, o el recall, que sería similar a recordar cuántas veces el auto ha respondido bien a tus órdenes de frenar.

La etapa final del proceso de minería de datos implica la interpretación de los resultados del modelo. En esta etapa, los resultados son analizados para llegar a conclusiones significativas y luego presentados de manera comprensible a los interesados. La interpretación implica visualización de los datos, generación de informes y discusión de los hallazgos en relación con el problema original. Según Kaisler, Armour, Espinosa, y Money, los patrones

⁹ El **clustering** es una técnica de análisis de datos que agrupa elementos similares en conjuntos o "clusters" basados en características comunes. Se utiliza para descubrir patrones en grandes volúmenes de datos sin supervisión previa. Es útil en áreas como segmentación de mercado, análisis de imágenes y bioinformática.

extraídos deben comunicarse de forma clara para que se puedan utilizar en la toma de decisiones.

Figura 3.

Flujo de la minería de datos



Nota: En el diagrama se refleja los procesos de la minería de datos. Universidad Complutense (2022)

2.4 Inteligencia Artificial (IA)

La inteligencia artificial es el campo de la informática que se ocupa de la creación de sistemas capaces de realizar tareas que tendrían que utilizar inteligencia humana para realizarlas. Dichos sistemas están diseñados para imitar los procesos cognitivos humanos, como el aprendizaje, el razonamiento y la resolución de problemas. Según Russell y Norvig (2016), la IA puede dividirse en dos categorías principales en “Inteligencia artificial débil” diseñada para ejecutar tareas específicas, como el reconocimiento de voz y la recomendación de productos, y “IA fuerte” que tiene como objetivo imitar capacidades cognitivas generales humanas. La IA necesita integrar algoritmos y modelos de matemáticas y permitir a las máquinas procesar información y tomar decisiones basadas en ella.

La IA se aplica en múltiples áreas, cada una utilizando diferentes enfoques y técnicas para resolver problemas específicos. Una de las principales áreas que ayuda actualmente es la siguiente:

- **Procesamiento de lenguaje natural (NLP):** Esta subdisciplina de la IA se ocupa de la interacción entre las computadoras y el lenguaje humano. El NLP incluye tareas como el análisis de sentimientos, la traducción automática y la generación de texto, facilitando la comunicación entre humanos y máquinas (Jurafsky y Martin, 2020).

En el proyecto de reconocimiento comercial en bienes raíces, el procesamiento de lenguaje natural (NLP) juega un papel fundamental al facilitar la identificación de información importante en grandes volúmenes de texto. Esta técnica, aplicada en el contexto de Web Scraping, permite analizar y extraer datos clave de manera automática, como descripciones de propiedades y tendencias del mercado inmobiliario. Gracias al NLP, la aplicación puede interpretar y organizar datos textuales complejos, mejorando la precisión en el análisis de oportunidades comerciales y en la generación de informes detallados respaldados por IA.

2.4.1 Técnicas y algoritmos de IA

Las técnicas de IA abarcan una variedad de métodos y enfoques

- **Redes neuronales artificiales:** Inspiradas en el funcionamiento del cerebro humano, las redes neuronales artificiales son fundamentales en el aprendizaje profundo. Estas redes consisten en múltiples capas de nodos (neuronas) que transforman los datos de entrada en una salida a través de funciones de activación. Las redes neuronales son efectivas para tareas como el

reconocimiento de imágenes y el procesamiento de lenguaje (LeCun, Bengio, y Hinton, 2015).

- **Algoritmos de aprendizaje supervisado:** A diferencia del aprendizaje no supervisado, el aprendizaje supervisado se centra en entrenar modelos utilizando datos etiquetados, lo que permite hacer predicciones precisas sobre datos nuevos. Técnicas como la regresión y la clasificación ayudan a crear modelos que reconocen patrones y relaciones en los datos, guiados por etiquetas o resultados conocidos (Jain, Murty, y Flynn, 1999).

Gracias a estos enfoques permiten identificar información clave de manera automática, mejorar la precisión de los análisis y apoyar la toma de decisiones estratégicas en ubicaciones específicas. Al mismo tiempo, la implementación de IA en este proyecto requiere atención a aspectos éticos y regulatorios esenciales, como la transparencia, la privacidad y la equidad, asegurando que los datos se procesen de manera justa y cumpliendo con normativas como el GDPR (Reglamento General de Protección de Datos) para la protección de la información. Estos elementos en conjunto permiten que la tecnología respalde el análisis comercial de manera responsable y efectiva, maximizando las oportunidades de negocio en el sector inmobiliario.

2.5 Modelos de Aprendizaje Automático (Machine Learning Models)

El aprendizaje automático (Machine Learning, ML) es una subdisciplina de la inteligencia artificial que se centra en el desarrollo de algoritmos y modelos capaces de aprender y mejorar su rendimiento a partir de datos. Según Mitchell (1997), el aprendizaje automático se define como el campo de estudio que proporciona a los sistemas la capacidad de aprender sin ser explícitamente programados para cada tarea específica. Esto se logra

mediante la creación de modelos que identifican patrones en datos y hacen predicciones o decisiones basadas en esos patrones.

2.5.1 Tipos de Modelos de Aprendizaje Automático

Existen varios tipos de modelos en el aprendizaje automático, cada uno adecuado para diferentes tipos de problemas y datos. El que aplicaremos es el siguiente:

- **Modelos de Aprendizaje Supervisado:** Dado que el objetivo principal es el reconocimiento de entidades, se utilizarán modelos supervisados entrenados con textos etiquetados, donde las entidades relevantes (como nombres de ciudades, precios, características de propiedades) están previamente identificadas. Algoritmos como Máquinas de Vectores de Soporte (SVM) y Árboles de Decisión pueden ser útiles para entrenar el modelo a fin de reconocer patrones en los textos y asignar etiquetas a las entidades relevantes, como precios de propiedades, direcciones o características específicas (Cortes y Vapnik, 1995; Breiman et al., 1986).

2.5.2 Proceso de Entrenamiento de Modelos

El proceso de entrenamiento de un modelo de aprendizaje automático generalmente involucra los siguientes pasos:

- **Preparación de Datos:** En esta fase, se recopilan los textos extraídos mediante Web Scraping, los cuales son procesados y etiquetados para identificar las entidades clave, como precios, nombres de lugares y características de propiedades. Se realiza una limpieza exhaustiva de los datos, eliminando elementos irrelevantes o "ruido", y se lleva a cabo una normalización

para estandarizar los formatos y mejorar la precisión del modelo. Esto asegura que los datos estén listos para ser utilizados en el entrenamiento del modelo, facilitando la identificación precisa de las entidades (Kotsiantis, 2013).

- **Selección de Modelo:** Para la identificación de entidades en texto, se opta por desarrollar un modelo de Reconocimiento de Entidades Nombradas (NER) entrenado desde cero, ajustado específicamente a los elementos clave del dominio (por ejemplo, precios, direcciones, número de baños, y habitaciones). La creación de este modelo desde cero permite un entrenamiento personalizado y específico para el contexto de la aplicación, maximizando la precisión en la extracción de las entidades requeridas. Modelos de NER basados en redes neuronales profundas, como las Redes Neuronales Recurrentes (RNN) o los modelos tipo Transformer, son especialmente efectivos en esta tarea, ya que capturan la estructura contextual y las relaciones de las entidades dentro del texto (Vaswani et al., 2017). Este enfoque asegura una detección más robusta y contextualizada para el proyecto en cuestión.
- **Entrenamiento del Modelo:** En esta fase, se entrena el modelo NER utilizando un conjunto de datos etiquetado específicamente con las entidades clave de interés, como precios, direcciones, número de baños y habitaciones. Durante el entrenamiento, el modelo aprende a reconocer patrones y relaciones entre palabras que son característicos de cada entidad, mejorando su capacidad para identificar de manera precisa estos elementos en nuevos textos. Los parámetros del modelo se ajustan iterativamente para minimizar el error y optimizar la precisión, asegurando que pueda extraer de forma efectiva las entidades clave necesarias para el proyecto (Bishop, 2006).

- **Evaluación del Modelo:** Una vez entrenado, el modelo es evaluado utilizando un conjunto de datos de prueba. Las métricas comunes de evaluación, como precisión, recall y F1-score, permiten medir el rendimiento del modelo al identificar correctamente las entidades dentro del texto. Estas métricas son fundamentales para garantizar que el modelo esté extrayendo la información relevante de manera efectiva, lo que es crucial para el análisis del mercado inmobiliario en el proyecto (Powers, 2011).
- **Optimización y Ajuste:** Con base en los resultados obtenidos durante la evaluación, el modelo se optimiza para mejorar su rendimiento. Esto puede incluir la adición de más datos de entrenamiento. La optimización busca asegurar que el modelo pueda generalizar bien a nuevos textos y continúe funcionando con eficacia a medida que se procesan más datos en el futuro (Kohavi, 1995).

Con estos enfoques de selección de textos específicos aplicando aprendizaje automático, principalmente a través de modelos supervisados, se busca identificar las entidades relevantes en los datos extraídos. Esta metodología permitirá reconocer de manera más precisa elementos clave, como precios y características de propiedades, mejorando así el proceso de análisis y clasificación dentro del proyecto.

2.6 Procesamiento de Lenguaje Natural (NLP)

El Procesamiento de Lenguaje Natural (Natural Language Processing, NLP) es un campo interdisciplinario que se encuentra en la intersección de la lingüística, la informática y la inteligencia artificial. Su objetivo principal es permitir que las máquinas comprendan, interpreten y generen lenguaje humano de una manera que sea tanto útil como coherente.

Según Jurafsky y Martin (2020), NLP involucra el uso de algoritmos y modelos computacionales para manejar y analizar grandes volúmenes de datos textuales, con el fin de extraer información significativa y realizar tareas lingüísticas automatizadas.

El NLP abarca una amplia gama de objetivos y aplicaciones, cada una de las cuales aborda diferentes aspectos del lenguaje humano:

- **Reconocimiento de Entidades Nombradas (NER):** NER se encarga de identificar y categorizar entidades específicas dentro de un texto, como nombres de personas, lugares, organizaciones y fechas. Este proceso es crucial para la extracción de información y la estructuración de datos en aplicaciones como la búsqueda de información y la integración de datos (Nadeau y Sekine, 2007).

El procesamiento de lenguaje natural utiliza diversas técnicas y enfoques metodológicos, entre los cuales destacan:

- **Modelos Basados en Reglas:** En los primeros días del NLP, se utilizaban modelos basados en reglas lingüísticas explícitas para analizar y generar lenguaje. Estos modelos se basaban en un conjunto predefinido de reglas gramaticales y léxicas para interpretar el texto (Chomsky, 1957). Aunque menos flexibles, estos modelos establecieron las bases para el desarrollo de métodos más avanzados.
- **Modelos Estadísticos:** Los modelos estadísticos utilizan grandes corpus de texto y técnicas de aprendizaje automático para aprender patrones y regularidades en el lenguaje. Los modelos de n-gramas, que analizan

secuencias de n palabras, y los modelos de clasificación basados en frecuencia de términos, como TF-IDF, son ejemplos representativos de este enfoque (Manning y Schütze, 1999).

2.6.1 Procesos Claves en NLP

El procesamiento del lenguaje natural generalmente involucra varios pasos críticos:

- **Tokenización:** La tokenización es el proceso de dividir un texto en unidades más pequeñas llamadas tokens, que pueden ser palabras, frases o caracteres. Este paso es fundamental para el análisis y la manipulación del texto, permitiendo la aplicación de técnicas de análisis y modelado (Manning et al., 2008).
- **Lematización y Stemming:** La lematización y el stemming son técnicas para reducir las palabras a sus formas base o raíces. La lematización utiliza el análisis morfológico para convertir una palabra a su forma léxica base, mientras que el stemming elimina los sufijos para obtener una raíz aproximada. Ambos métodos son cruciales para la normalización del texto y la mejora del rendimiento en tareas de análisis y recuperación de información (Porter, 1980).
- **Análisis Sintáctico y Semántico:** El análisis sintáctico implica examinar la estructura gramatical del texto, identificando las relaciones entre palabras y frases para comprender su significado. El análisis semántico, por otro lado, busca interpretar el significado subyacente de las palabras y frases en su contexto, capturando la intención y el sentido general del texto (Jurafsky y Martin, 2020).
- **Extracción de Características:** La extracción de características es el proceso de identificar y extraer atributos relevantes del texto para su uso en modelos de aprendizaje automático. Este paso puede incluir la selección de términos clave, la creación de vectores de características y la aplicación de técnicas de reducción

de dimensionalidad para mejorar la eficiencia y efectividad de los modelos (Sebastiani, 2002).

El Procesamiento de Lenguaje Natural (NLP) es una disciplina clave para interpretar y extraer información útil de grandes volúmenes de texto, lo cual es esencial para el análisis de datos en proyectos como el de bienes raíces. En este contexto, el objetivo principal de NLP es permitir que las máquinas comprendan, procesen y generen texto de forma efectiva para identificar y clasificar entidades relevantes, como precios, ubicaciones y características de propiedades, a partir de los datos obtenidos mediante técnicas como el Web Scraping.

Para lograr esto, se utilizan diferentes enfoques y técnicas. Primero, se realiza la tokenización, dividiendo el texto en unidades como palabras o frases, lo que facilita su análisis posterior. Luego, se lleva a cabo la normalización del texto mediante lematización y stemming, reduciendo las palabras a su forma base para mejorar la precisión de las herramientas de análisis.

En cuanto a la extracción de información, uno de los métodos más relevantes es el Reconocimiento de Entidades Nombradas (NER), que permite identificar y clasificar automáticamente entidades como nombres de lugares, precios y características clave de las propiedades. Para ello, se emplean modelos supervisados entrenados con textos etiquetados, que aprenden a reconocer patrones en los datos para hacer predicciones sobre nuevos textos. Además, técnicas como el análisis sintáctico y semántico son útiles para comprender la estructura y el significado de los textos, lo que mejora la capacidad del modelo para interpretar correctamente la información extraída.

2.7 Base de Datos y Almacenamiento de Datos

Las bases de datos representan sistemas fundamentales para la gestión moderna de datos, facilitando el almacenamiento, recuperación, manipulación y análisis de grandes volúmenes de información. En esencia, una base de datos es una colección organizada de datos estructurados que permiten la eficiencia en el acceso y administración de la información. PostgreSQL, un sistema de gestión de bases de datos relacional (RDBMS) de código abierto, se destaca por su robustez, conformidad con estándares SQL, y extensibilidad (PostgreSQL, 2024).

El concepto de almacenamiento de datos se refiere a la técnica de guardar información en una estructura de datos que permite su recuperación y manipulación eficiente. Este almacenamiento puede ser relacional, como en PostgreSQL, o no relacional, como en bases de datos NoSQL. La capacidad de almacenar datos de manera eficaz es crucial para la operativa de aplicaciones que manejan grandes cantidades de información, desde sistemas empresariales hasta aplicaciones web y móviles (Elmasri y Navathe, 2015).

2.7.1 PostgreSQL

- **Modelo Relacional y SQL:** PostgreSQL es un RDBMS que utiliza el modelo relacional para organizar datos en tablas estructuradas en filas y columnas. Este modelo permite la implementación de relaciones entre diferentes conjuntos de datos, facilitando consultas complejas y la integridad referencial. PostgreSQL soporta el lenguaje SQL (Structured Query Language), que es el estándar para la gestión y manipulación de bases de datos relacionales. SQL permite realizar operaciones como consultas, actualizaciones y eliminaciones de datos de manera eficiente y segura (PostgreSQL, 2024).

- **Extensibilidad y Personalización:** Una de las principales ventajas de PostgreSQL es su alta extensibilidad. Los usuarios pueden definir tipos de datos personalizados, operadores y funciones, lo que permite adaptar la base de datos a necesidades específicas que van más allá de las capacidades estándar. Esta característica es especialmente útil en entornos en los que se requiere un diseño de base de datos altamente especializado o en aplicaciones que requieren operaciones personalizadas sobre los datos (PostgreSQL, 2024).
- **Soporte para Datos No Estructurados:** Además de manejar datos estructurados, PostgreSQL ofrece soporte para datos semi-estructurados y no estructurados a través de tipos de datos como JSON y XML. Esto permite la integración de información en formatos diversos y facilita la realización de consultas sobre estos datos no estructurados, ampliando las capacidades de la base de datos más allá de los límites de los datos tabulares (PostgreSQL, 2024).

2.7.2 PostGIS: Gestión de Datos Espaciales

PostGIS es una extensión de PostgreSQL que añade soporte avanzado para datos espaciales, permitiendo el almacenamiento, consulta y análisis de información geográfica. Esta extensión es crucial para aplicaciones que requieren procesamiento de datos basados en ubicaciones, tales como sistemas de información geográfica (GIS), aplicaciones de mapeo y análisis espacial. PostGIS convierte a PostgreSQL en una potente plataforma para manejar datos espaciales y realizar análisis geoespaciales complejos (PostGIS, 2024).

- **Tipos de Datos Espaciales:** PostGIS introduce tipos de datos espaciales como GEOMETRY y GEOGRAPHY. El tipo GEOMETRY se usa para almacenar objetos espaciales en un sistema de coordenadas plano, mientras que GEOGRAPHY maneja datos en un sistema de coordenadas geográficas basado en la esfericidad de la Tierra. Estos tipos de datos permiten representar diversos objetos espaciales como puntos, líneas y polígonos, y facilitan operaciones espaciales como la intersección y la proximidad (PostGIS, 2024).
- **Consultas Espaciales Avanzadas:** PostGIS proporciona una amplia gama de funciones espaciales que permiten realizar consultas avanzadas sobre los datos geográficos. Estas funciones incluyen la determinación de la distancia entre objetos, la búsqueda de áreas de influencia y la identificación de la intersección entre formas geográficas. Las consultas espaciales son fundamentales para aplicaciones que requieren análisis de proximidad, mapeo de rutas y visualización geoespacial (PostGIS, 2024).
- **Indexación Espacial:** Para optimizar el rendimiento en la búsqueda y recuperación de datos espaciales, PostGIS utiliza índices espaciales basados en estructuras de árboles R-trees, implementados a través de GIST (Generalized Search Tree). Esta indexación mejora significativamente el tiempo de respuesta en consultas que involucran grandes volúmenes de datos geográficos y operaciones espaciales complejas, permitiendo un acceso más rápido y eficiente a la información (PostGIS, 2024).

2.7.3 *Procesos Claves en el Almacenamiento de Datos*

- **Modelado de Datos:** El modelado de datos es un proceso esencial en el diseño de una base de datos, que implica la creación de un esquema que define cómo se organizarán y relacionarán los datos. Este proceso incluye la identificación de entidades, atributos y relaciones, y la implementación de un esquema físico y lógico que refleje las necesidades de la aplicación y garantice la integridad de los datos. Un buen diseño de base de datos asegura la eficiencia en el almacenamiento y la recuperación de datos, y facilita el mantenimiento a largo plazo (Elmasri y Navathe, 2015).
- **Normalización y Desnormalización:** La normalización es una técnica que organiza los datos en tablas de manera que se minimice la redundancia y se eviten anomalías en las operaciones de actualización. Este proceso implica dividir grandes tablas en tablas más pequeñas y establecer relaciones entre ellas mediante claves foráneas. Por otro lado, la desnormalización introduce redundancias controladas para mejorar el rendimiento en consultas específicas, permitiendo una optimización en el acceso a datos en aplicaciones donde la velocidad es crítica (Date, 2004).
- **Transacciones y Consistencia:** Las transacciones son unidades de trabajo que agrupan una serie de operaciones en la base de datos de manera que se garantice que todas se completen con éxito o se reviertan en caso de error. La gestión de transacciones es fundamental para mantener la consistencia y la integridad de los datos, siguiendo las propiedades ACID: Atomicidad, Consistencia, Aislamiento y Durabilidad. Estas propiedades aseguran que las operaciones sean seguras y que la base de datos mantenga un estado

coherente en presencia de fallos o errores (Hellerstein, Stonebraker, y Hamilton, 2007).

PostgreSQL y PostGIS son herramientas clave para almacenar, gestionar y analizar tanto datos estructurados como espaciales, permitiendo que el proyecto de análisis de bienes raíces maneje eficazmente grandes cantidades de información y realice consultas geoespaciales complejas.

2.8 Análisis de datos

El análisis de datos es un proceso esencial que transforma datos crudos en información valiosa mediante diversas técnicas y métodos. Este proceso es fundamental para la toma de decisiones informadas y la mejora continua en diferentes contextos, como el empresarial, el académico y el gubernamental.

El análisis de datos se define como la inspección, limpieza y modelado de datos para descubrir patrones significativos y obtener conocimientos que faciliten la toma de decisiones. McKinsey y Company (2020) describe este proceso como una metodología que transforma datos en insights aplicables, ayudando a las organizaciones a tomar decisiones basadas en evidencia y a prever futuros comportamientos.

Uno de los principales objetivos del análisis de datos es proporcionar una base sólida para decisiones fundamentadas. Davenport y Harris (2007) destacan que el análisis de datos permite descubrir patrones relevantes y prever tendencias futuras, lo que a su vez ayuda en la planificación estratégica y operativa. Este tipo de análisis facilita la identificación de oportunidades y la mitigación de riesgos, basándose en datos históricos y actuales.

2.8.1 Herramientas y Técnicas

- **Herramientas de Software:** Las herramientas de software juegan un papel crucial en el análisis de datos, permitiendo la manipulación y visualización de grandes volúmenes de información. Herramientas como Python que es ampliamente utilizada para realizar análisis detallados y generar informes. Jupyter et al. (2018) describen cómo estas herramientas facilitan el procesamiento y análisis de datos mediante scripts y programas específicos.
- **Visualización de Datos:** La visualización de datos es una parte integral del análisis, facilitando la interpretación de resultados a través de gráficos y tablas. Kirk (2016) señala que la visualización ayuda a comunicar los hallazgos de manera clara y efectiva, permitiendo a los usuarios comprender patrones y tendencias de manera visual.

2.9 Visualización de Datos

La visualización de datos se define como el proceso de crear representaciones gráficas de datos con el fin de facilitar la interpretación y el análisis de la información. Según Tufte (2006), la visualización de datos transforma datos crudos en representaciones visuales que permiten a los analistas y tomadores de decisiones identificar patrones, relaciones y anomalías que de otro modo podrían pasar desapercibidos en datos tabulares o textuales. El propósito de la visualización es proporcionar una comprensión clara y precisa de los datos, mejorando la comunicación de insights y apoyando la toma de decisiones fundamentadas.

2.9.1 Importancia en el Análisis de Datos

- **Mejora en la Interpretación de Datos:** La visualización de datos permite una interpretación más rápida y efectiva de grandes volúmenes de información al presentar los datos de manera gráfica. Healy (2018) sostiene que la representación visual facilita la identificación de patrones, tendencias y outliers, lo que es esencial para el análisis exploratorio y la toma de decisiones basada en datos.
- **Facilitación de la Comunicación:** La capacidad de comunicar resultados de manera clara y efectiva es un aspecto crucial de la visualización de datos. Few (2012) señala que las visualizaciones bien diseñadas no solo transmiten información compleja de manera accesible, sino que también facilitan la comprensión de los hallazgos por parte de audiencias no especializadas, mejorando la comunicación y la toma de decisiones en diversos contextos.

2.9.2 Tipos de Visualizaciones de Datos

- **Gráficos de Líneas:** Los gráficos de líneas son ideales para representar datos a lo largo del tiempo. Healy (2018) destaca que estos gráficos permiten la visualización de tendencias y cambios temporales, facilitando el análisis de series temporales y la identificación de patrones estacionales o cíclicos.
- **Mapas de Calor:** Los mapas de calor representan datos en forma de matrices, utilizando colores para indicar la magnitud de los valores. Wilke (2019) señala que los mapas de calor son útiles para visualizar la densidad de datos y las concentraciones de valores, proporcionando una visión general de la distribución y la intensidad de los datos en diferentes áreas.

2.10 Limpieza de Datos (Data Cleaning)

La limpieza de datos es un proceso esencial en la gestión y preparación de datos, que se enfoca en la identificación y corrección de errores, inconsistencias y anomalías en los datos. Este proceso es crucial para asegurar que los datos sean precisos, completos y utilizables, y para mejorar la calidad de los análisis y decisiones basadas en datos.

La limpieza de datos, también conocida como data cleaning o data cleansing, se define como el proceso de identificar y corregir errores e inconsistencias en un conjunto de datos. Los datos pueden contener errores, duplicados, valores faltantes o inconsistencias que pueden afectar la calidad del análisis y la interpretación de los resultados (Pyle, 1999). El propósito de la limpieza de datos es garantizar que los datos sean fiables y estén listos para su análisis, optimizando así la precisión y la integridad de las conclusiones derivadas.

2.10.1 Importancia de la Limpieza de Datos

- **Mejora de la Calidad de los Datos:** La limpieza de datos es fundamental para mejorar la calidad de los datos al corregir errores y eliminar inconsistencias. Han (2011) sostiene que la limpieza adecuada garantiza que los datos sean precisos y fiables, lo cual es esencial para obtener resultados válidos y tomar decisiones fundamentadas basadas en estos datos.
- **Reducción de Errores en el Análisis:** Los errores y las inconsistencias en los datos pueden llevar a conclusiones incorrectas. El proceso de limpieza ayuda a minimizar estos errores al identificar y corregir problemas en los datos antes de realizar el análisis (Wang et al., 2003). Esto reduce el riesgo de obtener resultados erróneos y mejora la validez del análisis.

- **Facilitación de la Integración de Datos:** La limpieza de datos también facilita la integración de datos provenientes de diferentes fuentes. Rahm y Do (2000) explican que los datos de múltiples fuentes a menudo presentan discrepancias en formatos y estructuras. La limpieza estandariza estos datos y los prepara para una integración efectiva, lo que es crucial en entornos donde se utilizan datos de diversas fuentes.

2.10.2 Procesos Clave en la Limpieza de Datos

- **Identificación de Datos Faltantes:** La identificación y manejo de datos faltantes es un aspecto importante de la limpieza de datos. Little y Rubin (2002) discuten diferentes técnicas para manejar datos faltantes, como la imputación de valores y el análisis de patrones de datos faltantes, para minimizar el impacto de la ausencia de datos en el análisis.
- **Corrección de Errores e Inconsistencias:** La corrección de errores implica identificar y rectificar errores en los datos, tales como errores tipográficos, valores incorrectos o inconsistencias en los formatos. Data cleaning software y herramientas como OpenRefine y Trifacta pueden automatizar y facilitar este proceso, garantizando que los datos se ajusten a los estándares y formatos necesarios (Hellerstein et al., 2008).
- **Eliminación de Duplicados:** La eliminación de duplicados es un proceso crucial para garantizar que cada registro en un conjunto de datos sea único. Duplicados en los datos pueden distorsionar los resultados del análisis y aumentar el riesgo de errores. Bortini et al. (2006) sugieren técnicas para la

identificación y eliminación de duplicados, como el uso de algoritmos de coincidencia y comparación de registros.

- **Normalización de Datos:** La normalización de datos implica estandarizar los datos para que sigan un formato y una estructura consistentes. Fernández et al. (2011) explican que la normalización incluye la conversión de datos a formatos uniformes y la eliminación de valores inconsistentes, lo que facilita la integración y el análisis de datos.

2.11 Metodología CRISP-DM

La metodología CRISP-DM (Cross-Industry Standard Process for Data Mining) se establece como el estándar de facto para la realización de proyectos de minería de datos, ofreciendo un marco sistemático y flexible que puede ser aplicado en diferentes contextos industriales y empresariales. Desarrollada a finales de la década de 1990 por un consorcio de empresas liderado por SPSS, CRISP-DM proporciona una estructura comprensiva para la implementación de procesos de minería de datos, abordando cada etapa del proyecto desde la comprensión inicial hasta el despliegue final del modelo.

2.11.1 Definición y Estructura del Modelo CRISP-DM

CRISP-DM es una metodología estructurada compuesta por seis fases principales: Comprensión del Negocio, Comprensión de los Datos, Preparación de los Datos, Modelado, Evaluación y Despliegue. Cada fase es iterativa, permitiendo una retroalimentación continua y una revisión del proceso a medida que se avanza. Este enfoque cíclico y flexible asegura que los modelos desarrollados sean relevantes y ajustados a los objetivos del negocio (Shearer, 2000).

2.11.1.1 Comprensión del Negocio

La fase de Comprensión del Negocio es fundamental para establecer el marco y los objetivos del proyecto de minería de datos. En esta etapa, se busca identificar las necesidades del negocio, definir el problema específico que se quiere resolver y formular los objetivos del proyecto (Chapman et al., 2000). Se realiza una evaluación detallada del contexto empresarial y se establecen criterios claros para medir el éxito del proyecto. Esta fase también implica la identificación de las restricciones y las expectativas de los stakeholders, lo cual es crucial para la alineación de los objetivos del proyecto con los resultados esperados.

2.11.1.2 Comprensión de los Datos

La Comprensión de los Datos involucra la recopilación y el análisis preliminar de los datos disponibles. Esta fase se enfoca en explorar la calidad y la estructura de los datos para identificar problemas potenciales como valores faltantes, errores o inconsistencias (Brachman y Anand, 1996). Además, se realiza un análisis descriptivo para entender las características de los datos y determinar cómo pueden ser utilizados para abordar el problema del negocio. Este análisis preliminar es esencial para guiar la preparación de datos en etapas posteriores.

2.11.1.3 Preparación de los Datos

La fase de Preparación de los Datos es crítica para la creación de un conjunto de datos limpio y estructurado que pueda ser utilizado en el modelado. Esta etapa incluye la limpieza de datos, la transformación y la integración de datos de diversas fuentes (Pyle, 1999). Los procesos típicos en esta fase son la eliminación de duplicados, la imputación de valores faltantes y la estandarización de formatos. La calidad del conjunto de datos preparado impacta directamente en la efectividad de los modelos de minería de datos.

2.11.1.4 Modelado

Durante la fase de Modelado, se seleccionan y aplican técnicas de minería de datos para construir modelos que puedan predecir o describir patrones en los datos (Han et al., 2011). Esto implica la elección de algoritmos apropiados, la configuración de parámetros y la evaluación del rendimiento del modelo. Es crucial seleccionar técnicas que sean adecuadas para los objetivos del negocio y para las características del conjunto de datos. Esta fase también incluye la validación cruzada y la evaluación de la capacidad predictiva del modelo.

2.11.1.5 Evaluación

La Evaluación es la fase en la que se revisan y analizan los resultados obtenidos para asegurar que cumplan con los objetivos del negocio establecidos (Rygielski et al., 2002). Se realiza una evaluación exhaustiva del modelo para determinar su efectividad y relevancia. Esto incluye la validación de los resultados contra los criterios del negocio y la realización de pruebas adicionales para verificar la robustez del modelo. La fase de evaluación es crucial para identificar posibles ajustes y mejoras en el modelo.

2.11.1.6 Despliegue

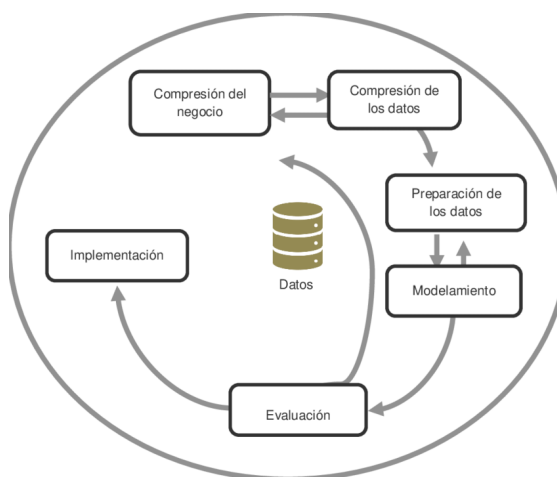
La fase de Despliegue se centra en la implementación del modelo en un entorno operativo real. Esto puede implicar la integración del modelo en sistemas de producción, la generación de informes y la comunicación de resultados a los stakeholders (Shearer, 2000). Es importante que el despliegue sea realizado de manera que facilite el uso del modelo y permita su monitoreo y mantenimiento continuo. El éxito del despliegue también depende de la capacitación del personal y del soporte técnico disponible.

2.11.2 Ventajas y Aplicabilidad de CRISP-DM

CRISP-DM se destaca por su flexibilidad y aplicabilidad en diversos contextos. Su estructura iterativa permite adaptaciones continuas y ajustes en función de los resultados y los cambios en el entorno empresarial (Chapman et al., 2000). La metodología es ampliamente utilizada en distintos sectores, incluyendo finanzas, salud, marketing y manufactura, lo que demuestra su versatilidad y eficacia en una amplia gama de aplicaciones.

Figura 4.

Proceso de la metodología CRISP DM



Nota: En el diagrama se refleja los procesos de la metodología CRISP DM. Extraído de IBM (2020)

2.12 Flask

Flask es un microframework para Python que permite desarrollar aplicaciones web de manera eficiente y flexible. Lanzado por Armin Ronacher en 2010, Flask se destaca por su simplicidad y su estructura modular. A diferencia de otros frameworks más completos como Django, Flask proporciona un núcleo básico con la posibilidad de agregar extensiones según las necesidades del proyecto, lo que lo hace ideal para proyectos de pequeño o mediano

tamaño. Es especialmente popular para la creación de APIs RESTful debido a su bajo nivel de complejidad, el control total que ofrece al desarrollador y la facilidad de integración con otros servicios (Grinberg, 2018).

Una de las características más atractivas de Flask es su capacidad de manejar aplicaciones web ligeras sin imponer una estructura rígida, lo que permite a los desarrolladores decidir cómo organizar su aplicación y qué herramientas utilizar. En términos de diseño, Flask sigue el principio de "micro", lo que significa que ofrece lo esencial para la creación de aplicaciones web y deja que el desarrollador integre solo lo necesario. Esto facilita el proceso de desarrollo sin sobrecargar al programador con dependencias innecesarias. Flask incluye soporte para enrutamiento, manejo de solicitudes HTTP, plantillas Jinja2 y desarrollo de APIs RESTful, entre otros.

Flask es particularmente adecuado para sistemas que requieren una alta flexibilidad y escalabilidad, como aplicaciones basadas en microservicios y servicios de backend como APIs. Gracias a su arquitectura minimalista, puede adaptarse a diversos casos de uso, desde pequeños proyectos hasta sistemas más complejos que se beneficiarán de la modularidad del framework (Hernández y Palacios, 2020).

Flask se utilizará para recibir solicitudes HTTP POST, donde se enviarán los parámetros necesarios para realizar el scraping. Estos parámetros incluirán detalles como la URL de la página que se quiere analizar y los criterios específicos de búsqueda (por ejemplo, ubicación y precio). El sistema utilizará bibliotecas como BeautifulSoup para la extracción de datos del HTML de las páginas web y Pandas para limpiar y estructurar los datos extraídos, eliminando errores y formateando los valores de manera adecuada.

Con Flask, se facilitará la creación de múltiples endpoints de API que se encargarán de distintos aspectos del scraping, como la extracción de precios, ubicaciones y tipos de propiedades. Esto también permitirá integrar algoritmos de aprendizaje automático en el futuro para clasificar y organizar la información de manera más eficiente. La flexibilidad de Flask en términos de enrutamiento y manejo de solicitudes lo convierte en una herramienta ideal para este tipo de proyectos, permitiendo una fácil integración con otras tecnologías y asegurando que el sistema sea escalable y mantenible a largo plazo.

2.13 React Materia- UI

React es una biblioteca de JavaScript de código abierto, desarrollada y mantenida por Facebook, que se utiliza para construir interfaces de usuario (UI) interactivas y dinámicas. Lanzada en 2013, React permite la creación de aplicaciones web de una sola página (SPA) mediante un enfoque basado en componentes. Esta biblioteca se ha convertido en una de las más populares en el desarrollo web debido a su capacidad para crear interfaces eficientes y rápidas mediante el uso de un "DOM virtual", lo que mejora el rendimiento al minimizar las manipulaciones del DOM real (React, 2021).

El concepto clave de React es el de componentes, bloques de construcción independientes que representan una parte de la UI de una aplicación. Cada componente de React tiene su propio estado y puede recibir propiedades (props) desde sus componentes padres. Esto permite que la UI se actualice automáticamente en función del cambio de estado, lo que facilita la creación de aplicaciones web interactivas y reactivas. React hace uso de un flujo unidireccional de datos, lo que facilita la gestión del estado de la aplicación y su renderización.

Otra característica importante de React es su capacidad para integrarse fácilmente con otras bibliotecas y frameworks. En el caso de un sistema de Web Scraping como el que se implementará en Bi Space SRL, React se utiliza para crear la interfaz de usuario donde los usuarios ingresarán los parámetros de búsqueda (por ejemplo, ubicación, tipo de propiedad y rango de precios) y visualizarán los datos obtenidos de las páginas de bienes raíces.

Material-UI es una biblioteca de componentes para React que implementa las directrices de diseño de Google Material Design. Material Design es un sistema de diseño basado en principios de simplicidad, claridad y coherencia visual, lo que facilita la creación de interfaces intuitivas y accesibles. Material-UI proporciona una amplia variedad de componentes listos para usar, como botones, formularios, tablas, cuadros de texto, y muchos más, todos con un estilo consistente y personalizado según las necesidades del proyecto.

Material-UI se destaca por su facilidad de personalización y su capacidad para crear interfaces limpias y modernas sin necesidad de escribir mucho código CSS. A través de sus componentes, Material-UI permite a los desarrolladores centrarse en la funcionalidad de la aplicación mientras se asegura de que la interfaz de usuario siga las mejores prácticas de diseño. La integración con React también garantiza que los componentes sean reactivos y fáciles de manipular mediante el estado de los componentes de React.

Una de las características más útiles de Material-UI es su sistema de temas. A través de este sistema, los desarrolladores pueden definir un conjunto de estilos que se aplicarán de manera consistente en toda la aplicación. Esto incluye colores, tipografías, márgenes y otros parámetros visuales. Esto permite que la interfaz de usuario no solo sea funcional, sino también estéticamente coherente y atractiva.

En el contexto del proyecto de Web Scraping de Bi Space SRL, React y Material-UI se implementarán para crear una interfaz de usuario moderna e interactiva. React facilitará la creación de componentes reutilizables para la visualización de datos de bienes raíces, mientras que Material-UI proporcionará un conjunto de componentes listos para usar que ayudarán a acelerar el desarrollo de la UI.

Material-UI se utilizará para asegurar que la interfaz sea fácil de usar y visualmente atractiva. Los componentes como formularios de búsqueda, botones de filtrado y tablas para visualizar los resultados se estilizarán con Material-UI, lo que garantizará una experiencia de usuario consistente y agradable. Además, el sistema de temas de Material-UI permitirá que la interfaz se ajuste a las preferencias visuales de la empresa o de los usuarios finales, asegurando una coherencia estética en toda la aplicación.

Fluidez en la experiencia de usuario, junto con la modularidad de React y la personalización de Material-UI, permitirá que el sistema de Web Scraping no solo sea funcional, sino también escalable, fácil de mantener y preparado para futuras ampliaciones.

2.14 BeautifulSoup

BeautifulSoup es una biblioteca de Python que se utiliza para analizar documentos HTML y XML y extraer información útil de ellos de manera eficiente. Fue desarrollada por Leonard Richardson en 2004 y se ha convertido en una de las herramientas más populares para realizar Web Scraping debido a su facilidad de uso y flexibilidad. BeautifulSoup permite que los desarrolladores interactúen con el contenido de las páginas web de forma sencilla, sin necesidad de escribir complejos fragmentos de código de análisis o transformación (Richardson, 2004).

Una de las principales ventajas de BeautifulSoup es su capacidad para manejar documentos mal formados o con errores. Esta biblioteca tiene una gran capacidad para corregir errores comunes en los documentos HTML, como etiquetas no cerradas correctamente o estructuras mal anidadas, lo que facilita su procesamiento. BeautifulSoup se utiliza para transformar el código HTML de una página web en una estructura de árbol, lo que permite navegar y extraer datos de manera más intuitiva utilizando métodos y funciones específicas.

BeautifulSoup es especialmente útil para proyectos de Web Scraping porque permite que los desarrolladores seleccionen fácilmente los elementos de una página web utilizando diversas estrategias de búsqueda, como selectores de etiquetas, atributos o clases CSS. Además, su integración con otras bibliotecas como requests para obtener el contenido de la página o xml para mejorar el rendimiento hace que sea una herramienta muy eficiente y robusta para la extracción de datos.

- **Facilidad de uso:** BeautifulSoup tiene una sintaxis intuitiva que facilita su aprendizaje y uso, lo que permite que los desarrolladores se concentren más en el procesamiento de los datos que en los detalles técnicos del análisis HTML.
- **Compatibilidad con múltiples parsers:** BeautifulSoup puede trabajar con varios parsers de Python como html.parser, lxml y html5lib, lo que le permite adaptarse a las necesidades del proyecto en cuanto a rendimiento y compatibilidad.
- **Manejo de documentos mal formados:** Gracias a su capacidad para manejar documentos HTML mal formados, BeautifulSoup es una opción ideal cuando se trabaja con páginas web que no siguen un estándar de codificación estricto.

- **Selección fácil de elementos:** Permite seleccionar fácilmente elementos por etiquetas, clases, ids o atributos específicos, lo que facilita la extracción de datos sin necesidad de escribir código complejo.

En el sistema de Web Scraping, BeautifulSoup se utilizará para procesar el contenido HTML de las páginas web de bienes raíces que se encuentren dentro de los parámetros definidos por el usuario. Cuando un usuario ingrese criterios de búsqueda (como ubicación, tipo de propiedad y precio), el sistema utilizará BeautifulSoup para analizar las páginas web correspondientes y extraer información relevante. Por ejemplo, se extraerán detalles sobre el tipo de propiedad, la ubicación, el precio y las características de la propiedad. Posteriormente, los datos extraídos se organizarán y almacenarán en la base de datos para su posterior análisis y presentación a los usuarios.

BeautifulSoup facilitará la navegación dentro del código HTML de las páginas, permitiendo localizar los elementos que contienen la información necesaria. Utilizando métodos como `find()`, `find_all()`, o `select()`, el sistema será capaz de identificar etiquetas HTML específicas, como `<div>`, `<span>`, o `<a>`, y extraer los datos asociados a ellas, como los precios de las propiedades o sus descripciones. Esta herramienta es crucial para asegurar que la extracción de datos se realice de manera eficiente y confiable, incluso en páginas web con estructuras HTML complejas o mal formadas.

2.15 Celery Redis

Redis es una base de datos en memoria, clave-valor, de código abierto, conocida por su rapidez, flexibilidad y eficiencia. A diferencia de las bases de datos tradicionales que se basan en disco, Redis almacena los datos en la memoria RAM, lo que le permite acceder y manipular datos de manera extremadamente rápida, lo que lo convierte en una herramienta

ideal para sistemas que requieren procesamiento en tiempo real y alta disponibilidad (Pietraszek, 2019).

Una de las características clave de Redis es su capacidad para manejar estructuras de datos avanzadas, como cadenas, listas, conjuntos, mapas, y otros tipos complejos. Esto lo hace muy útil en escenarios que requieren alta disponibilidad y rendimiento, como almacenamiento de sesiones, colas de tareas y almacenamiento de caché. Redis también proporciona persistencia opcional, lo que significa que los datos pueden guardarse en disco para que no se pierdan en caso de que se reinicie el sistema.

En el contexto de Web Scraping, Redis se utiliza para almacenar de manera eficiente las solicitudes y respuestas de páginas web, así como para gestionar las colas de tareas que se deben ejecutar en segundo plano. Esto es particularmente útil cuando se gestionan grandes volúmenes de solicitudes a diversas páginas web, ya que permite procesarlas de manera eficiente sin sobrecargar el sistema.

Celery es una biblioteca distribuida de Python que se utiliza para la ejecución asíncrona de tareas. Permite a los desarrolladores gestionar tareas en segundo plano, como el procesamiento de datos o la ejecución de funciones de manera asíncrona, lo que mejora la escalabilidad y la eficiencia de las aplicaciones. Celery se integra fácilmente con sistemas de almacenamiento de colas, como Redis, para manejar y distribuir tareas de manera eficiente entre varios trabajadores (Schwörer, 2018).

Celery se utiliza comúnmente en sistemas donde las tareas de procesamiento son independientes y deben ejecutarse sin bloquear la ejecución principal de la aplicación. En el caso de Web Scraping, Celery es ideal para manejar tareas como la descarga de múltiples páginas web o la extracción de datos desde diversas fuentes. Cada tarea se coloca en una

cola gestionada por Redis, y los trabajadores de Celery se encargan de procesarlas de manera eficiente.

El uso de Celery en un sistema de Web Scraping permite que las tareas se realicen de manera asíncrona, lo que significa que la aplicación principal puede seguir operando sin esperar a que se completen las tareas de scraping. Esto es esencial en proyectos donde el scraping de páginas web es intensivo en recursos y tiempo, ya que mejora significativamente el rendimiento y la escalabilidad del sistema.

En el contexto de extracciones, Redis y Celery se integrarán para mejorar la eficiencia y escalabilidad del sistema de Web Scraping. Redis será utilizado como sistema de cola para almacenar las solicitudes de scraping y las respuestas obtenidas, permitiendo que las tareas se distribuyan entre múltiples trabajadores de manera eficiente. Además, Redis se utilizará para almacenar información en caché, como los resultados de scraping previos, lo que reducirá la carga en el servidor y acelerará las respuestas.

Por otro lado, Celery se encargará de gestionar las tareas asíncronas de Web Scraping. Cuando el sistema reciba una solicitud de scraping, Celery encolará la tarea, y los trabajadores de Celery comenzarán a procesarla de manera asíncrona. Esto permitirá que el sistema pueda manejar múltiples solicitudes simultáneamente sin bloquear la ejecución del servidor. Además, Celery ayudará a controlar el flujo de tareas, asegurando que se gestionen correctamente las dependencias entre ellas, y permitiendo la ejecución de tareas en paralelo.

En el sistema de Web Scraping, Redis será utilizado como una herramienta para gestionar las colas de trabajo, almacenar los datos obtenidos de las páginas web de manera eficiente y mejorar el rendimiento del sistema. Por ejemplo, cuando se realiza una solicitud de scraping, Redis se encargará de almacenar las URLs de las páginas a procesar y las

respuestas obtenidas, lo que permitirá que el sistema maneje grandes volúmenes de datos sin sobrecargar el servidor.

Por su parte, Celery se integrará para gestionar las tareas asíncronas de scraping. Cada tarea de scraping será colocada en una cola de Redis y será procesada por uno de los trabajadores de Celery. Este enfoque permitirá que el sistema ejecute múltiples tareas de scraping en paralelo, mejorando el rendimiento y la escalabilidad del sistema. Además, Celery se encargará de manejar los tiempos de espera y los errores en el proceso, garantizando que las tareas se realicen de manera eficiente y confiable.

2.16 Herramienta para Web Scraping

2.16.1 Pypetter

PyPetter se destaca por su sencillez y rapidez al acceder a contenido estático de las páginas web. Sin embargo, cuando se trata de interactuar con contenido dinámico cargado por JavaScript, la herramienta podría no ser tan efectiva como otras más avanzadas. No obstante, para aplicaciones de scraping en sitios web relativamente sencillos, PyPetter es una opción atractiva por su bajo nivel de complejidad (PyPetter, 2022).

2.16.2 Playwright

Playwright es una herramienta poderosa de automatización de navegadores para Python, diseñada específicamente para interactuar con sitios web que requieren una interacción más compleja, como el manejo de JavaScript dinámico, formularios y cookies. Desarrollada por Microsoft, Playwright es comparable a Selenium, pero ofrece varias ventajas en términos de velocidad, estabilidad y la capacidad de manejar múltiples navegadores a la vez, como Chromium, Firefox y WebKit.

Una de las características destacadas de Playwright es su capacidad para capturar contenido dinámico de sitios web sin necesidad de configuraciones adicionales. Esto es útil para proyectos de Web Scraping que requieren extraer datos de páginas interactivas, como sitios de bienes raíces, que actualizan constantemente el contenido mediante JavaScript. Playwright también permite realizar pruebas automatizadas y simular la interacción de un usuario real con las páginas, lo que lo convierte en una herramienta versátil para el desarrollo de aplicaciones complejas (Playwright, 2023).

2.16.3 Selenium

A pesar de su flexibilidad y capacidades avanzadas, Selenium puede ser relativamente lento en comparación con otras herramientas debido a que controla un navegador completo para realizar las interacciones. Sin embargo, su capacidad para manejar sitios web dinámicos cargados con JavaScript lo convierte en una herramienta ideal para proyectos de Web Scraping que requieren una simulación precisa de la navegación humana. En proyectos como el sistema de reconocimiento comercial de bienes raíces, Selenium es útil para extraer datos de propiedades en tiempo real, navegar entre páginas de resultados y realizar acciones complejas como el filtrado y la búsqueda avanzada (Selenium, 2020).

Para optimizar el proceso de Web Scraping, se implementarán varias herramientas de scraping, cada una elegida según la complejidad de las páginas web a analizar. PyPettet se empleará para sitios web estáticos, mientras que Playwright y Selenium se utilizarán para manejar sitios dinámicos e interactivos que requieren simulación de navegación. Axios, por otro lado, se usará para acceder rápidamente a APIs o páginas simples sin necesidad de simular interacciones del usuario. Esta combinación de herramientas permitirá que el sistema de Web Scraping sea ágil, eficiente y capaz de adaptarse a diversas fuentes de datos en el sector inmobiliario.

2.17 Docker

Docker es una plataforma de contenedorización que permite a los desarrolladores empaquetar aplicaciones y sus dependencias en contenedores ligeros, garantizando un entorno de ejecución consistente sin importar la máquina o el sistema operativo donde se ejecuten. Al empaquetar tanto el frontend como el backend de una aplicación web en contenedores separados, Docker facilita el despliegue, la escalabilidad y el mantenimiento del sistema, eliminando problemas de compatibilidad entre distintos entornos de desarrollo y producción (Merkel, 2014).

Al crear contenedores independientes para el frontend y el backend, Docker permite que cada componente tenga sus propias configuraciones, dependencias y versiones de software, lo cual es especialmente útil en proyectos que utilizan tecnologías diferentes en el lado del cliente y del servidor. Además, Docker permite definir redes internas para que los contenedores del frontend y el backend puedan comunicarse entre sí de manera segura y eficaz. Esto proporciona una gran flexibilidad, ya que el sistema puede escalar horizontalmente aumentando el número de contenedores del backend o frontend de acuerdo con la demanda del usuario.

En el proyecto de reconocimiento comercial en bienes raíces, Docker se utiliza para empaquetar tanto el frontend como el backend en contenedores separados, lo que permite crear entornos de ejecución controlados y uniformes. La arquitectura se basa en la contenedorización del sistema: se implementan archivos Dockerfile específicos para el backend (que ejecuta la API con Flask) y el frontend (desarrollado con React y Material-UI), además de un archivo de configuración docker-compose.yml para orquestar ambos contenedores. Esto establece redes internas que facilitan la comunicación entre servicios y garantiza que cada parte de la aplicación esté aislada y optimizada con sus propias

dependencias. Este enfoque permite ejecutar y escalar el sistema en diferentes entornos sin preocuparse por problemas de compatibilidad o configuración. Docker, al proporcionar esta consistencia y flexibilidad, optimiza el despliegue, mantenimiento y escalabilidad del sistema, lo cual es crucial en proyectos complejos donde el frontend y el backend necesitan comunicarse de forma confiable y segura en cualquier entorno.

2.18 Pruebas

Las pruebas en el desarrollo de software son fundamentales para asegurar que el sistema cumple con los requisitos especificados, identifica errores y garantiza que el software funciona correctamente en todas las condiciones previstas. Esta etapa es crucial para asegurar la calidad y confiabilidad del producto final.

2.18.1 Pruebas Unitarias

Las pruebas unitarias son el primer nivel de pruebas que se enfocan en verificar la funcionalidad de componentes individuales del software, como funciones y métodos específicos (Beck, 2003). Este tipo de prueba se realiza en aislamiento, sin la influencia de otros componentes del sistema. Su propósito es detectar errores en el nivel más granular, lo que permite corregir fallos rápidamente durante el desarrollo.

Las herramientas de pruebas unitarias, como Jest para JavaScript y unittest para Python, automatizan la ejecución de pruebas repetitivas y proporcionan informes detallados sobre el estado de las pruebas (Crispin y Gregory, 2009). Estas herramientas ayudan a mantener una alta cobertura de código y permiten una detección temprana de errores, facilitando un ciclo de desarrollo ágil y eficiente. Además, al integrarse con otros sistemas de integración continua, contribuyen a un flujo de trabajo más rápido y confiable, lo que mejora la

calidad del software en proyectos de desarrollo como los que involucran tecnologías web y backend.

2.18.2 Pruebas de accesibilidad

Las pruebas de accesibilidad son un componente crítico en el desarrollo de software inclusivo, diseñado para garantizar que personas con discapacidades puedan interactuar de manera efectiva con sistemas digitales. Estas pruebas se basan en las Pautas de Accesibilidad para el Contenido Web (WCAG) y otros estándares internacionales, como la Sección 508 en Estados Unidos, que establecen criterios específicos para evaluar la percepción, operabilidad, comprensión y solidez de las interfaces. Se enfocan en aspectos técnicos, como la navegación mediante teclado, la compatibilidad con tecnologías de asistencia como lectores de pantalla, el contraste adecuado entre texto y fondo, y el uso correcto de etiquetas semánticas en el HTML. Más allá de los requisitos legales, estas pruebas promueven la equidad digital al hacer que los sistemas sean accesibles para todos los usuarios, independientemente de sus capacidades físicas o cognitivas.

El proceso de evaluación combina métodos manuales y automatizados, utilizando herramientas como Axe, Lighthouse o WAVE, que identifican problemas comunes de accesibilidad. Sin embargo, las pruebas realizadas por usuarios reales con discapacidades proporcionan una perspectiva más profunda sobre la usabilidad y las barreras específicas que enfrentan. Estas pruebas no solo aseguran el cumplimiento normativo, sino que también mejoran la experiencia de usuario en general, lo que puede traducirse en una mayor satisfacción y alcance del producto. En este sentido, la accesibilidad no debe considerarse únicamente un requisito técnico, sino un principio ético y de diseño universal que beneficia a todas las partes interesadas (Myers, G. J., Sandler, C., & Badgett, T., 2011).

2.18.3 Pruebas de Sistema

Las pruebas de sistema son un nivel más amplio que abarca la evaluación del sistema en su totalidad para asegurar que cumpla con los requisitos funcionales y no funcionales (Pressman, 2014). Estas pruebas se realizan en un entorno que simula las condiciones reales de operación y examinan la interacción del sistema con el entorno y los usuarios finales.

2.18.3.1 Evaluación de Requisitos

Durante las pruebas de sistema, se revisan tanto los requisitos funcionales, como la funcionalidad del sistema, como los requisitos no funcionales, como el rendimiento y la seguridad. Este enfoque integral asegura que todos los aspectos del sistema sean evaluados y validados antes de su implementación en producción (Dean y Ghemawat, 2008).

2.19 Seguridad

La seguridad en el desarrollo de software es fundamental para proteger los sistemas y datos frente a accesos no autorizados, ataques y otros riesgos que puedan comprometer la integridad, confidencialidad y disponibilidad de la información. Aplicando la norma ISO/IEC 27001, se asegura una gestión estructurada y continua de la seguridad de la información, estableciendo políticas y controles que ayudan a proteger los activos de información críticos y a mantener la confianza de los usuarios y las organizaciones. Implementar prácticas de seguridad eficaces permite resguardar información sensible y asegurar que el sistema esté protegido contra diversas amenazas y vulnerabilidades.

La confidencialidad es uno de los principios clave de seguridad, garantizando que solo las personas o sistemas autorizados accedan a la información. Para esto, se aplican técnicas como el cifrado de datos y controles de acceso, limitando el acceso a datos sensibles y

protegiéndolos de entidades no autorizadas. De igual forma, la integridad es fundamental para evitar alteraciones no autorizadas en los datos, ya sea durante su almacenamiento, transmisión o procesamiento. Mecanismos como las firmas digitales y sumas de verificación se utilizan para identificar y evitar modificaciones en los datos, manteniendo su precisión y fiabilidad. La disponibilidad asegura que tanto los sistemas como la información estén accesibles para los usuarios autorizados en el momento que lo necesiten. Esto se logra mediante la implementación de sistemas redundantes, planes de recuperación ante desastres y protección contra ataques de denegación de servicio (DoS), lo que garantiza que las aplicaciones sean accesibles de manera continua.

Las amenazas a la seguridad pueden ser internas, como empleados que abusan de sus privilegios, o externas, como ataques cibernéticos, malware y otras amenazas que explotan vulnerabilidades en el sistema. Las vulnerabilidades, como errores de programación y configuraciones inadecuadas, representan puntos débiles que los atacantes pueden aprovechar. La identificación y remediación de estas vulnerabilidades es crucial para proteger el sistema de ataques. Por ejemplo, el cifrado convierte la información en un formato codificado que solo puede ser descifrado con una clave adecuada, protegiendo así los datos en reposo y en tránsito. Algoritmos como AES y RSA son ampliamente utilizados para garantizar la confidencialidad de la información incluso si es interceptada. Los controles de acceso, que incluyen la autenticación y autorización de usuarios, limitan los permisos de acceso a recursos específicos, lo que reduce el riesgo de accesos no autorizados y protege los datos sensibles.

La norma ISO/IEC 27001 también establece procesos de evaluación y gestión de riesgos como el análisis de riesgos, que ayuda a identificar amenazas, evaluar vulnerabilidades y entender el impacto potencial en el sistema. Este análisis es la base para establecer medidas de mitigación efectivas y asignar recursos a áreas críticas de seguridad.

Además, la planificación de la recuperación ante desastres define los procedimientos para restaurar el funcionamiento del sistema tras un incidente. Este enfoque incluye la creación de copias de seguridad periódicas y planes de contingencia, asegurando que los sistemas puedan recuperarse rápidamente y mantener la continuidad operativa frente a fallos o ataques. Estas prácticas refuerzan la resiliencia del sistema y permiten una respuesta eficaz ante incidentes de seguridad.

2.20 Costos

El análisis y gestión de costos en el desarrollo de software es crucial para garantizar la viabilidad económica de los proyectos. La metodología COSMIC (Common Software Measurement International Consortium) proporciona un marco estructurado para medir y gestionar el tamaño y los costos del software de manera efectiva. Esta metodología ayuda a estimar los costos asociados con el desarrollo y mantenimiento del software basándose en métricas cuantitativas.

2.20.1 Metodología COSMIC

2.20.1.1 Introducción a COSMIC

COSMIC es una metodología de medición del tamaño del software que se basa en la funcionalidad que el software proporciona a sus usuarios. Esta metodología mide el tamaño del software en términos de "Function Points" (puntos de función), que representan la cantidad de funcionalidad proporcionada. COSMIC se utiliza para estimar los costos de desarrollo y mantenimiento, facilitando la planificación y el control financiero (COSMIC, 2014).

2.20.1.2 Medición de Tamaño de Software con COSMIC

COSMIC mide el tamaño del software mediante la identificación de las funciones y el análisis de las transacciones y datos que el software maneja. Esta metodología clasifica las funciones en cuatro categorías principales: Entradas de Datos, Salidas de Datos, Lecturas de Datos y Escrituras de Datos. Cada función se evalúa para determinar su impacto en el tamaño total del software, lo que permite calcular los costos asociados con su desarrollo y mantenimiento (COSMIC, 2014).

2.20.1.3 Aplicación de COSMIC para Estimaciones de Costos

Una vez que se ha medido el tamaño del software utilizando COSMIC, los resultados se utilizan para estimar los costos del proyecto. Estos costos se calculan a partir de la relación entre el tamaño del software y las tarifas de desarrollo por punto de función. La metodología COSMIC permite ajustar estas estimaciones en función de factores como la complejidad del proyecto y las tasas de productividad del equipo de desarrollo (Gérard, 2005).

CAPITULO III

MARCO APLICATIVO

**INGENIERÍA
DE SISTEMAS**
UNIVERSIDAD PÚBLICA DE EL ALTO

CAPITULO III

3. MARCO APLICATIVO

En el desarrollo de un sistema de Scraping inteligente para la recolección de datos en el sector inmobiliario, es fundamental implementar metodologías que faciliten la planificación, desarrollo, y despliegue efectivo del sistema. Para lograr esto, el marco aplicativo emplea CRISP-DM (Cross-Industry Standard Process for Data Mining) el cual proporciona una estructura sólida y metodológica para abordar las fases de análisis de datos.

3.1 CRISP-DM

3.1.1 *Comprensión del negocio*

La primera fase del desarrollo del sistema de extracción de datos inmobiliarios basado en inteligencia artificial se centra en la comprensión del problema de negocio, lo cual implica analizar la estructura de las páginas web de bienes raíces y definir una base de datos adecuada para almacenar la información extraída. Además, en esta etapa se integran los requerimientos funcionales y no funcionales identificados en la **Ingeniería de Requerimientos**, asegurando que el sistema cumpla con las especificaciones y necesidades del negocio.

3.1.1.1 *Ingeniería de requerimientos*

La Ingeniería de Requerimientos constituye la base del desarrollo del sistema, proporcionando una descripción clara de las funcionalidades y características esperadas. En esta etapa se han definido tanto los requerimientos funcionales, que describen las operaciones y procesos principales que debe realizar el sistema, como los requerimientos no funcionales, que especifican los atributos de calidad deseados, como la seguridad, escalabilidad y usabilidad.

- **Requerimientos funcionales:**

Tabla 1.*Requerimientos funcionales*

ID	Requerimiento Funcional	Descripción
RF1	Gestión de configuraciones de Scraping	El sistema debe permitir al usuario crear, editar, listar y eliminar configuraciones para realizar Scraping en diferentes sitios web de bienes raíces.
RF2	Extracción de datos de propiedades	El sistema debe extraer información clave de las propiedades, incluyendo precio, área, ubicación y descripción, a partir de las configuraciones definidas.
RF3	Almacenamiento de datos	El sistema debe almacenar los datos extraídos en una base de datos estructurada para su consulta posterior.
RF4	Procesamiento y limpieza de datos	El sistema debe procesar y limpiar los datos extraídos para homogeneizar formatos (ejemplo: precios, áreas, ubicaciones).
RF5	Completar información faltante mediante IA	En caso de que alguna información esté ausente, el sistema debe emplear un modelo de inteligencia artificial con técnicas de procesamiento de lenguaje natural (NLP) para extraer esta información desde las descripciones de las propiedades.
RF6	Visualización de datos en frontend	El sistema debe ofrecer una interfaz web para visualizar los datos recolectados, permitiendo búsquedas y filtros por diferentes criterios.
RF7	Generación de reportes	El sistema debe generar reportes sobre la información recolectada y mostrar estadísticas de interés para el usuario.
RF8	Integración con APIs de mapas	El sistema debe permitir la visualización de las propiedades en un mapa interactivo mediante integración con una API de mapas (ej., Google Maps, Leaflet).
RF9	Autenticación y autorización de usuarios	El sistema debe gestionar la autenticación y autorización de usuarios para garantizar la seguridad y privacidad de los datos.

- **Requerimientos no funcionales:**

Tabla 2*Requerimientos no funcionales*

ID	Requerimiento No Funcional	Descripción
RNF1	Seguridad	El sistema debe asegurar que los datos sensibles de los usuarios y las configuraciones de Scraping estén protegidos mediante protocolos de autenticación (JWT) y almacenamiento seguro de contraseñas.
RNF2	Escalabilidad	El sistema debe ser escalable, permitiendo añadir nuevos sitios web de Scraping y manejar grandes volúmenes de datos de propiedades sin afectar el rendimiento.
RNF3	Eficiencia en la extracción de datos	La extracción de datos debe realizarse en un tiempo razonable, y los procesos de Scraping no deben consumir recursos de manera excesiva.
RNF4	Tolerancia a fallos	El sistema debe ser robusto frente a errores de conexión, cambios en la estructura de las páginas web y fallos en la aplicación de los modelos de IA.
RNF5	Usabilidad	La interfaz web debe ser intuitiva y fácil de usar, permitiendo que usuarios sin conocimientos técnicos puedan visualizar los datos recolectados y acceder a la IA para completar información.
RNF6	Precisión del modelo de IA	El modelo de IA con NLP debe garantizar un alto grado de precisión y confiabilidad al extraer y completar información faltante en las descripciones de las propiedades.
RNF7	Mantenimiento y extensibilidad	El sistema debe estar diseñado para facilitar su mantenimiento, permitiendo actualizaciones en la lógica de Scraping, IA y la base de datos.
RNF8	Soporte de múltiples formatos de datos	El sistema debe soportar la extracción de datos en diversos formatos (texto, enlaces) y realizar la conversión según sea necesario.

3.1.1.2 Análisis de la Estructura de las Páginas Web

En esta etapa se realiza un análisis exhaustivo de las páginas web inmobiliarias que serán objeto de extracción de datos. Dado que cada sitio puede variar en términos de estructura HTML, se hace necesario un estudio detallado para identificar patrones y selectores específicos que permitirán acceder a la información de cada propiedad.

Para cada página, se identifican los siguientes elementos:

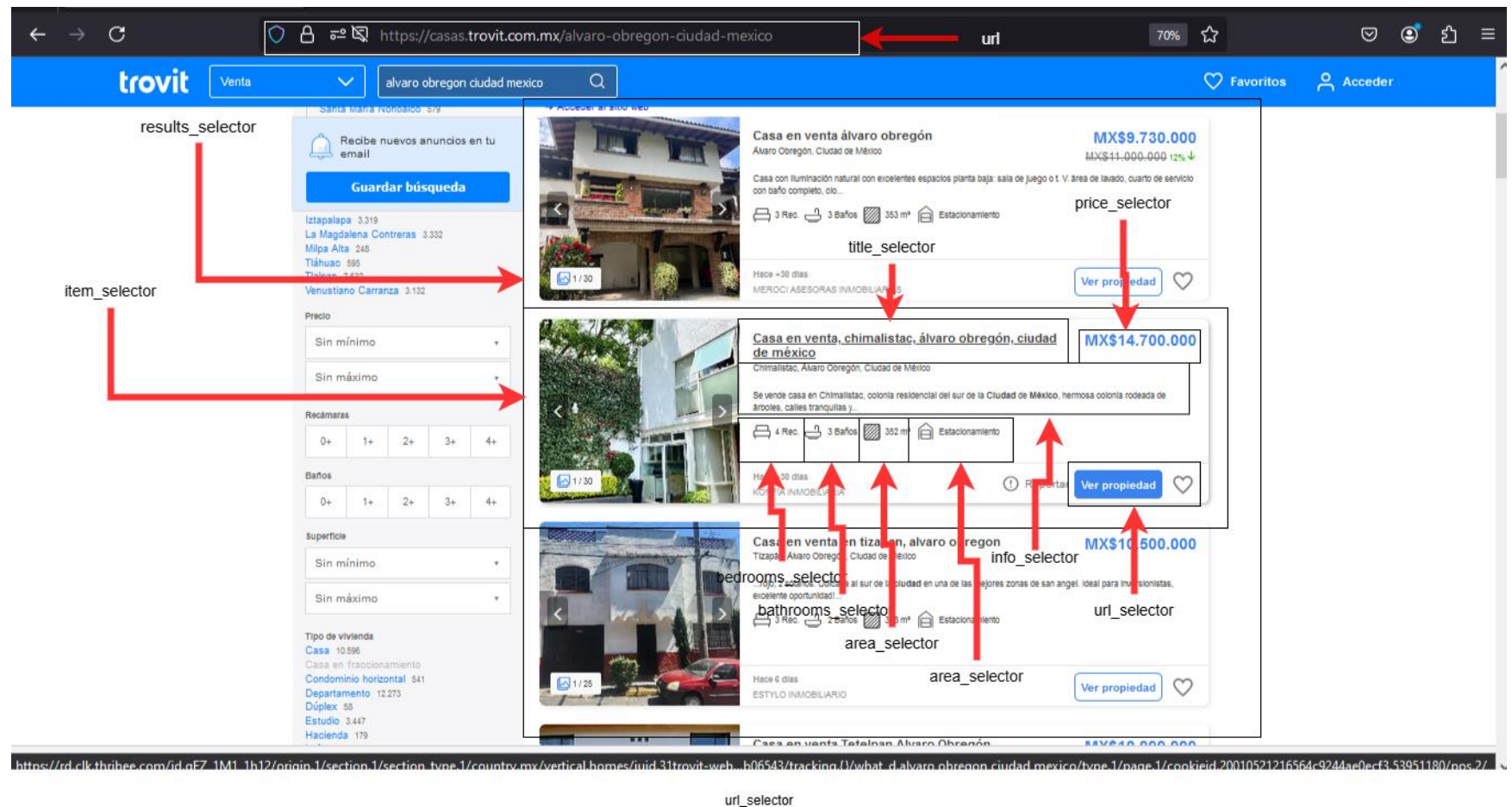
Tabla 3.
Elementos de las páginas web

Campo	Descripción
page_name	Nombre de la página o sitio web de donde se extraen los datos. Esto permite llevar un control de las distintas fuentes utilizadas en el sistema.
url	URL principal del sitio web que contiene las publicaciones de propiedades. Sirve como punto de partida para las solicitudes de Scraping.
results_selector	Selector del contenedor que agrupa los resultados de búsqueda de propiedades. Este elemento permite acceder a la lista completa de publicaciones en la página.
item_selector	Selector del elemento que representa cada propiedad individual dentro del contenedor de resultados. Es fundamental para aislar y extraer la información de cada propiedad.
price_selector	Selector del elemento que muestra el precio de la propiedad. Este valor es clave para cualquier análisis posterior sobre los precios en el mercado inmobiliario.
amount_selector	Selector que identifica el costo o monto total de la propiedad, útil en algunos sitios que muestran precios totales en lugar de precios por unidad de medida.
info_selector	Selector que apunta a la sección de descripción de la propiedad, donde se detalla información adicional sobre las características de la propiedad.
url_selector	Selector del enlace específico que redirige a la página de detalle de cada propiedad. Este enlace es crucial para extraer más datos detallados cuando sea necesario.
area_selector	Selector del campo que indica el área total de la propiedad, en metros cuadrados o cualquier otra unidad de medida especificada.
bedrooms_selector	Selector del elemento que muestra la cantidad de habitaciones de la propiedad. Este dato es fundamental para clasificar el tipo y tamaño de la propiedad.
bathrooms_selector	Selector para el número de baños en la propiedad, un dato comúnmente requerido por los usuarios en el sector inmobiliario.
location_selector	Selector que indica la ubicación o dirección de la propiedad. En algunos casos, se espera que el sistema identifique coordenadas geográficas para integrarse con mapas.
title_selector	Selector que identifica el título o encabezado de la publicación, que suele resumir las características principales de la propiedad.
parking_selector	Selector que indica si la propiedad cuenta con espacio para estacionamiento, un detalle importante en áreas urbanas.

Nota: Tabla de selectores de elementos en distintas páginas web donde se reconoce las distintas fuentes de información en las páginas web.

Caso Trovit.com.mx

Figura 5.
Pagina oficial Trovit



Nota: Estructura de la página web de Trovit con sus respectivas etiquetas y selectores en donde se reconoce la información relevante para ser almacenada.

Caso lamudi.com

Figura 6.
Pagina Oficial Lamudi

The screenshot shows the Lamudi website interface with several red arrows pointing to specific elements, each labeled with a selector:

- url**: Points to the browser address bar showing `https://www.lamudi.com.mx/for-sale/`.
- results_selector**: Points to the 'En venta' dropdown menu under 'TIPO DE OFERTA'.
- item_selector**: Points to the first property listing card.
- title_selector**: Points to the title 'Casa en Venta en Juriquilla'.
- price_selector**: Points to the price '\$ 4,150,000 MXN'.
- area_selector**: Points to the area '245 m²'.
- info_selector**: Points to the 'Ver mas info' button.
- url_selector**: Points to the title 'Casa en Venta en Del Valle'.

The website layout includes a navigation bar with filters (TIPO DE OFERTA, UBICACIÓN, TIPO DE PROPIEDAD, PRECIO, RECÁMARAS) and a main content area displaying property listings with images, titles, prices, and details.

Nota: Estructura de la página web de Lamudi con sus respectivas etiquetas y selectores en donde se reconoce la información relevante para ser almacenada.

Esta información es fundamental para configurar el sistema de Scraping, adaptando los selectores y métodos de extracción a las particularidades de cada sitio. El análisis detallado asegura que el sistema sea flexible y adaptable a distintos formatos y estructuras de datos.

3.1.1.3 Herramientas para realizar Web Scraping disponibles

Tabla 4.

Tabla de evaluación de herramientas de Web Scraping

Herramienta	Descripción	¿Es apto?
BeautifulSoup	Biblioteca para extraer datos de archivos HTML y XML, muy utilizada por su facilidad de uso.	SI
Scrapy	Framework completo para scraping que permite la extracción de datos de sitios web y automatización.	NO
Selenium	Herramienta para la automatización de navegadores web que permite interactuar con JavaScript.	SI
Requests-HTML	Biblioteca para scraping que permite manejar JavaScript de manera sencilla.	NO
PyQuery	Permite usar una sintaxis similar a jQuery para realizar scraping sobre documentos HTML.	NO
LXML	Biblioteca rápida para trabajar con XML y HTML.	NO
Splash	Servicio web para renderizar JavaScript y obtener contenido de páginas dinámicas.	NO
Pyppeteer	Variante en Python del popular Puppeteer para automatizar el navegador Chrome.	SI
Puppeteer	Herramienta para controlar Chrome mediante Node.js, ideal para scraping de sitios JS-heavy.	NO
Grab	Framework de scraping que utiliza una sintaxis simple para crear spiders.	NO
Goose3	Extrae el contenido principal de un artículo (texto, imágenes, etc.) de una página web.	NO
PlayWright	Renderiza la página más rápidamente, ayuda con la extracción de manera optima y eficiente	SI

3.1.1.4 Diseño y Creación de la Base de Datos

La base de datos es fundamental para almacenar y gestionar la información extraída de las propiedades mediante web Scraping, y debe ser capaz de manejar grandes volúmenes de datos de manera eficiente. Para ello, se diseñará una base de datos relacional utilizando PostgreSQL, organizando los datos en esquemas específicos: **WEBSCRAPING** para la configuración de Scraping y almacenamiento de datos, **PAGES_WEB** para la configuración de la páginas web, **AI** para las predicciones generadas por la inteligencia artificial y **PUBLIC** para la gestión de usuarios. Cada esquema y tabla estará interconectada mediante claves primarias y foráneas, garantizando la integridad de los datos y facilitando las consultas complejas.

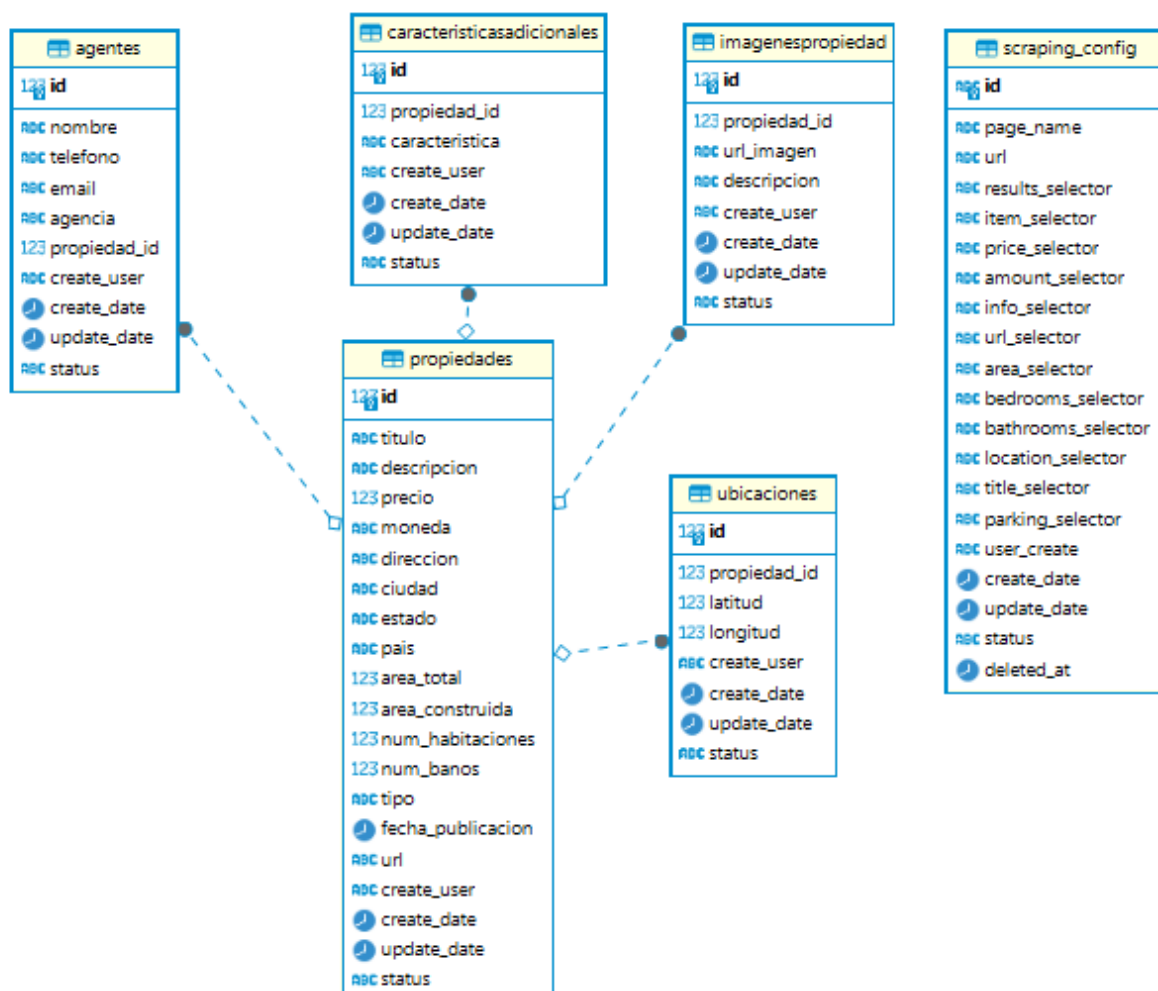
Este diseño asegura un rendimiento óptimo, permitiendo la recuperación rápida de datos y la escalabilidad a medida que el volumen de información crece. Además, el sistema estará optimizado mediante la normalización de las tablas, eliminando redundancias y mejorando la eficiencia en las consultas. PostgreSQL, con su capacidad de manejar consultas complejas y ofrecer alta seguridad, es la opción ideal para este proyecto, garantizando la fiabilidad y la integridad de los datos a largo plazo

Estructura de la Base de Datos:

Web_Scraping:

Figura 7.

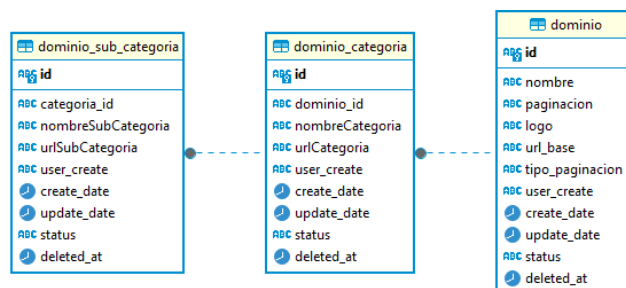
Esquema Web_Scraping



Nota: Estructura de las tablas dentro del esquema en donde se almacenará la información extraída de las distintas páginas web

Pages_Web:

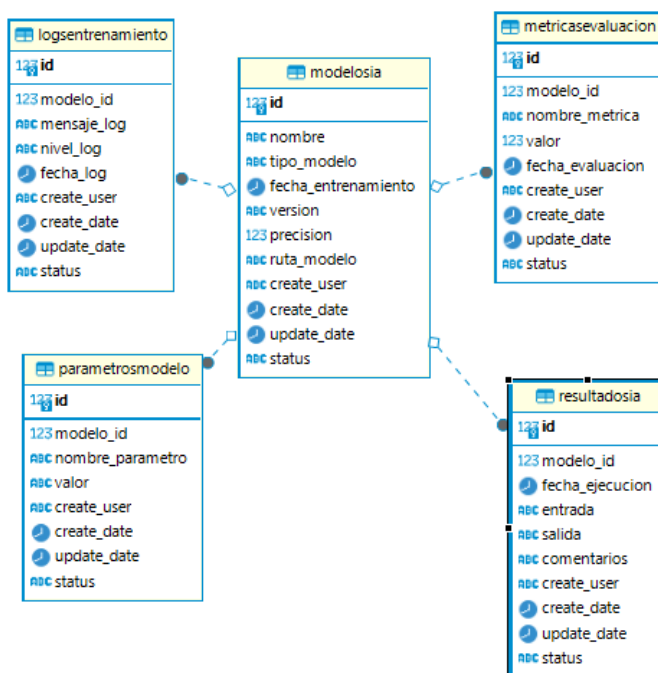
Figura 8.
Esquema Pages_Web



Nota: Estructura de las tablas dentro del esquema Page_Web en donde se registrará la información de páginas de donde se extraerá la información.

IA:

Figura 9.
Esquema IA

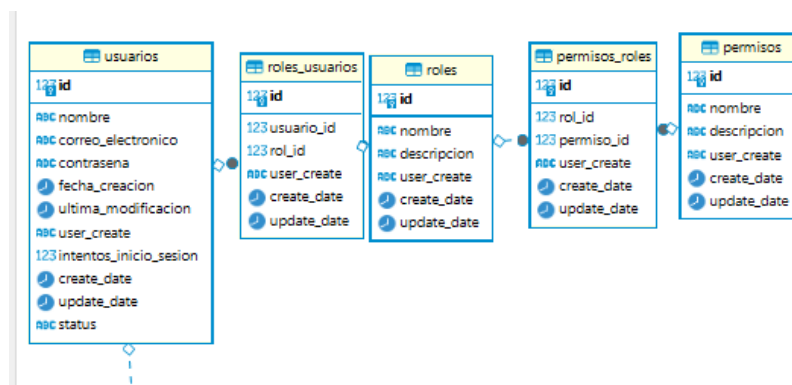


Nota: Estructura de las tablas dentro del esquema IA en donde se almacenará la información referente al modelo como los resultados del entrenamiento y antecedentes referentes al modelo de IA

Public:

Figura 10.

Esquema Public



Nota: Estructura de las tablas dentro del esquema Public en donde se almacenará los datos referentes a los usuarios y accesos a la aplicación

Implementación: El sistema utiliza PostgreSQL con la extensión PostGIS para gestionar datos geoespaciales, como las ubicaciones de las propiedades. Esta tecnología permite almacenar y consultar coordenadas geográficas, lo que facilita búsquedas avanzadas basadas en la ubicación y el análisis espacial de las propiedades en el sistema.

3.1.2 Comprensión de los Datos

- Análisis de la estructura de las Páginas Web**

En esta fase, el primer paso fue comprender la estructura de las páginas web de donde se extraen los datos. Para ello, se identificaron los selectores CSS y XPath que apuntan a los elementos relevantes en cada página web. Los selectores son esenciales para la localización de la información que se desea extraer, como el precio de la propiedad, el número de habitaciones, el área, y otros detalles específicos.

Se utilizaron las siguientes estrategias:

page_name: Identificación de la fuente de datos, que indica de qué página se están extrayendo los datos.

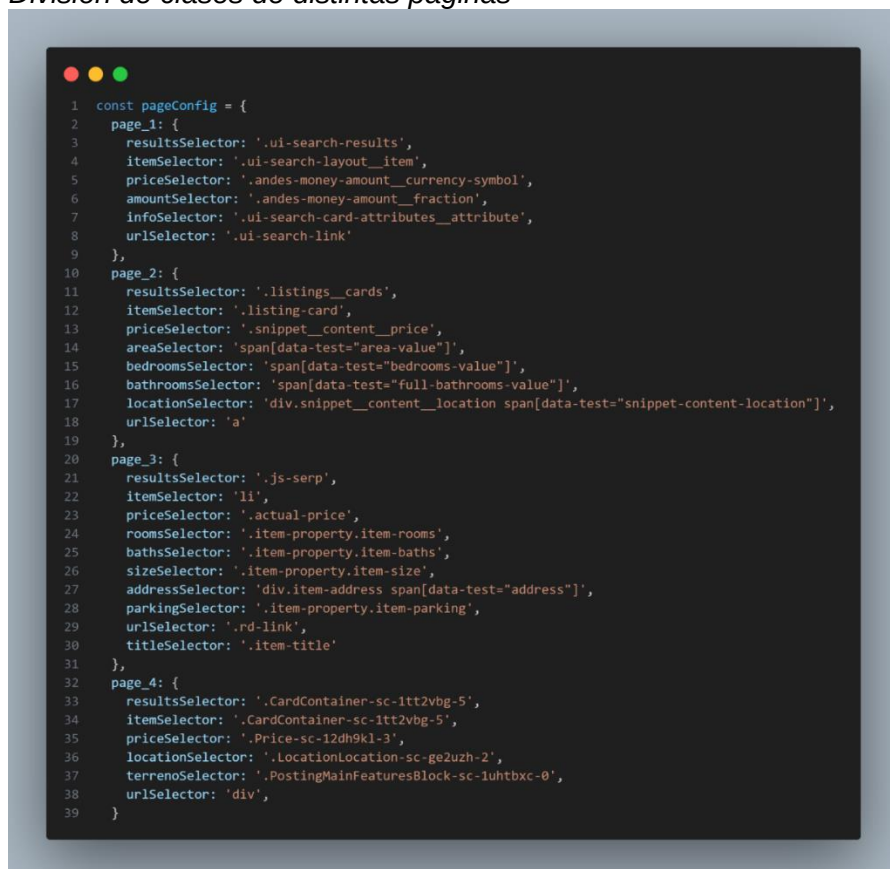
url: Se definió la URL principal que actúa como punto de partida para la recolección de datos.

Selectores Específicos: Se definieron selectores como price_selector, area_selector, location_selector, entre otros, que permiten acceder a las partes específicas de cada página web donde se encuentran los datos relevantes. Por ejemplo, el selector para obtener el precio de una propiedad puede ser algo como. property-price o un XPath como //span[@class='price'].

Aquí un ejemplo en formato json de cómo se reconoció las distintas clases

Figura 11.

División de clases de distintas paginas



```

1  const pageConfig = {
2    page_1: {
3      resultsSelector: '.ui-search-results',
4      itemSelector: '.ui-search-layout__item',
5      priceSelector: '.andes-money-amount__currency-symbol',
6      amountSelector: '.andes-money-amount__fraction',
7      infoSelector: '.ui-search-card-attributes__attribute',
8      urlSelector: '.ui-search-link'
9    },
10   page_2: {
11     resultsSelector: '.listings__cards',
12     itemSelector: '.listing-card',
13     priceSelector: '.snippet__content__price',
14     areaSelector: 'span[data-test="area-value"]',
15     bedroomsSelector: 'span[data-test="bedrooms-value"]',
16     bathroomsSelector: 'span[data-test="full-bathrooms-value"]',
17     locationSelector: 'div.snippet__content__location span[data-test="snippet-content-location"]',
18     urlSelector: 'a'
19   },
20   page_3: {
21     resultsSelector: '.js-serp',
22     itemSelector: 'li',
23     priceSelector: '.actual-price',
24     roomsSelector: '.item-property.item-rooms',
25     bathsSelector: '.item-property.item-baths',
26     sizeSelector: '.item-property.item-size',
27     addressSelector: 'div.item-address span[data-test="address"]',
28     parkingSelector: '.item-property.item-parking',
29     urlSelector: '.rd-link',
30     titleSelector: '.item-title'
31   },
32   page_4: {
33     resultsSelector: '.CardContainer-sc-1tt2vbg-5',
34     itemSelector: '.CardContainer-sc-1tt2vbg-5',
35     priceSelector: '.Price-sc-12dh9kl-3',
36     locationSelector: '.LocationLocation-sc-ge2uzh-2',
37     terrenoSelector: '.PostingMainFeaturesBlock-sc-1uhtbxc-0',
38     urlSelector: 'div',
39   }
40 }

```

Nota: Estructura de las clases encontradas en distintas fuentes con sus respectivos selectores.

3.1.3 Preparación de los Datos

La fase de preparación de los datos es crucial para garantizar que el conjunto de datos esté listo para ser usados. En esta etapa, se lleva a cabo el proceso de limpieza, transformación y estructuración de los datos para que puedan ser procesados adecuadamente por los algoritmos de minería de datos. El proceso de preparación de los datos incluye varias actividades como la eliminación de duplicados, la imputación de valores faltantes, la normalización de datos y la transformación de formatos.

Limpieza de Datos:

Eliminación de Duplicados: Es común que los datos extraídos contengan entradas repetidas. En esta subfase, se identifican y eliminan las filas duplicadas para evitar sesgos en los resultados de los modelos.

Figura 12.

Función para eliminar duplicados



```

1 def remove_duplicates():
2     duplicates = db.session.query(Property.url, db.func.min(Property.id).label('min_id')) \
3         .group_by(Property.url) \
4         .having(db.func.count(Property.url) > 1).all()
5
6     for duplicate in duplicates:
7         duplicate_ids = db.session.query(Property.id).filter(Property.url == duplicate.url).all()
8         ids_to_delete = [id[0] for id in duplicate_ids]
9         ids_to_delete.remove(duplicate.min_id)
10
11     # Eliminar registros duplicados
12     db.session.query(Property).filter(Property.id.in_(ids_to_delete)).delete(synchronize_session=False)
13
14     db.session.commit()
15

```

Nota: Función para poder eliminar datos duplicados asegurando la confiabilidad de la información.

Transformación de Datos:

Estandarización de Formatos: Los datos extraídos pueden estar en diferentes formatos (por ejemplo, precios con diferentes símbolos de moneda, fechas en formatos distintos, áreas con unidades no uniformes). Se realiza la estandarización de estos formatos para que toda la información sea homogénea y compatible con el análisis.

Integración de Datos: Los datos provenientes de diferentes fuentes o páginas web deben ser integrados de forma coherente. Asegurarse de que las columnas y los valores sean compatibles entre sí es vital. En algunos casos, los datos extraídos de diferentes páginas pueden tener estructuras ligeramente diferentes, por lo que será necesario transformarlos para que encajen en una estructura común.

Figura 13.

Formato de datos



```
1 def integrate_data_from_other_sources():
2     # Suponiendo que tienes una tabla 'new_properties' con nuevas propiedades
3     new_properties = db.session.query(NewProperty).all()
4
5     for new_property in new_properties:
6         # Insertar los nuevos datos de propiedades en la tabla principal
7         property = Property(
8             url=new_property.url,
9             price=new_property.price,
10            area=new_property.area,
11            location=new_property.location,
12            bedrooms=new_property.bedrooms,
13            bathrooms=new_property.bathrooms
14        )
15        db.session.add(property)
16
17    db.session.commit()
18
```

Nota: Formateo de datos para obtener un estándar para su posterior almacenamiento.

Validación de la Calidad de los Datos: Tras realizar la limpieza y transformación, se lleva a cabo una validación de calidad. Se verifica que todos los valores sean razonables y que no haya errores o inconsistencias en los datos. Por ejemplo, los precios no deben ser negativos, y las áreas deben estar dentro de rangos lógicos.

Figura 14.

Función para validar datos

A screenshot of a code editor with a dark background and light-colored text. The code is a Python function named `validate_data_quality()`. It uses SQLAlchemy's `db.session.query()` to filter properties where the price is less than 0 or the area is less than or equal to 0. It then iterates over the resulting `invalid_data` and prints a message for each item showing its ID, price, and area. Finally, it commits the session.

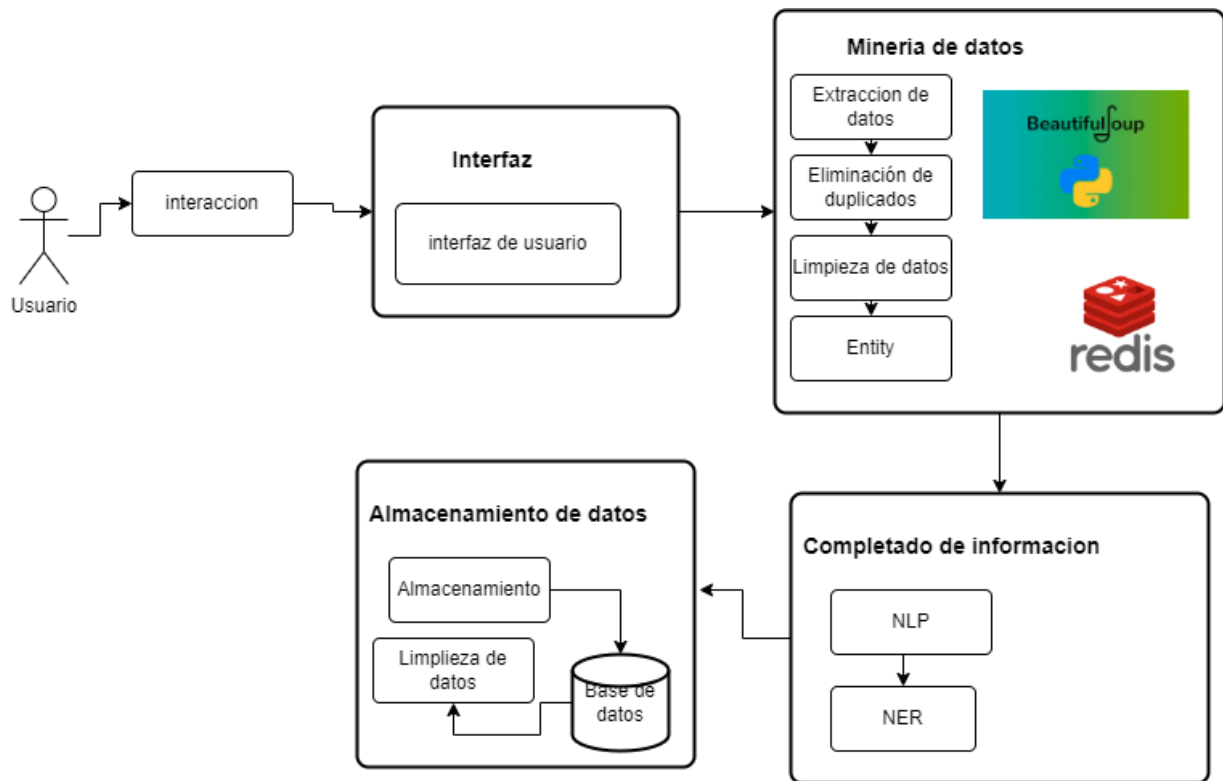
```
1 def validate_data_quality():
2     invalid_data = db.session.query(Property).filter(
3         (Property.price < 0) | (Property.area <= 0)
4     ).all()
5
6     if invalid_data:
7         for item in invalid_data:
8             print(f"Invalid data found: {item.id} - {item.price} - {item.area}")
9
10    db.session.commit()
11
```

Nota: Validación de precios para no tener valores negativos en los precios

3.1.4 Modelado

El propósito principal de esta fase es emplear técnicas de Procesamiento de Lenguaje Natural (NLP) para inferir y completar los campos faltantes a partir de las descripciones textuales de las propiedades. Si una propiedad tiene, por ejemplo, un campo vacío para el número de habitaciones o el precio, pero cuenta con una descripción detallada, podemos utilizar un modelo NLP para identificar y extraer la información relevante y llenar estos campos automáticamente.

Figura 15.
Diagrama de interacciones con el modelo



Fase de Modelado para un Sistema de Web Scraping

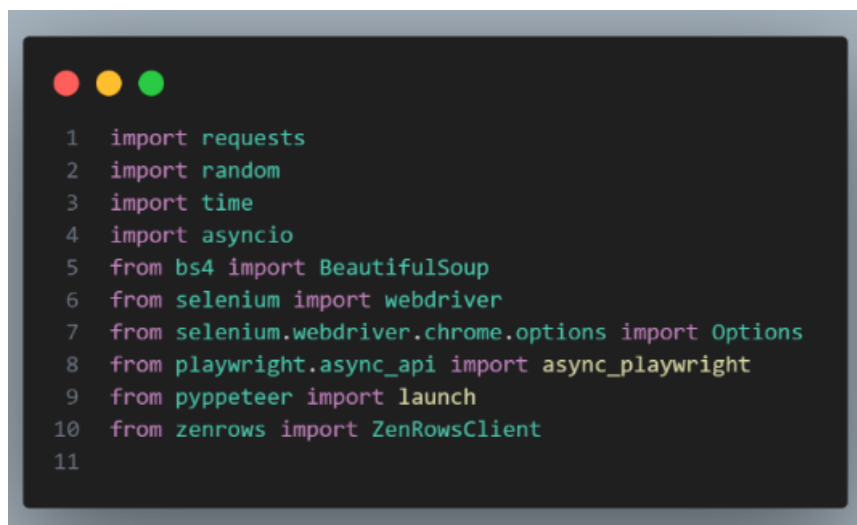
- **Extracción de datos (Web Scraping):** En esta parte se incluyó herramientas de Web Scraping como ser: Axios, Playwright, Selenium, Puppeteer.

Aquí el ejemplo del código

Importamos las herramientas

Figura 16.

Importación de herramientas



```
1  import requests
2  import random
3  import time
4  import asyncio
5  from bs4 import BeautifulSoup
6  from selenium import webdriver
7  from selenium.webdriver.chrome.options import Options
8  from playwright.async_api import async_playwright
9  from pyppeteer import launch
10 from zenrows import ZenRowsClient
11
```

Nota: Importaciones de herramientas fundamentales para cargar las páginas web.

Creamos las funciones para el Web Scraping

Figura 17.

Función para realizar Web Scraping



```
1  async def axios_scraping(url):
2      headers = {
3          "User-Agent": get_random_user_agent(),
4          "Accept": "text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8"
5      }
6      proxy = get_random_proxy()
7      try:
8          response = await asyncio.to_thread(requests.get, url, headers=headers, proxies={"http": proxy, "https": proxy}, timeout=10)
9          response.raise_for_status()
10         return response.text
11     except requests.RequestException as e:
12         print(f"Axios scraping failed:")
13         return None
14
```

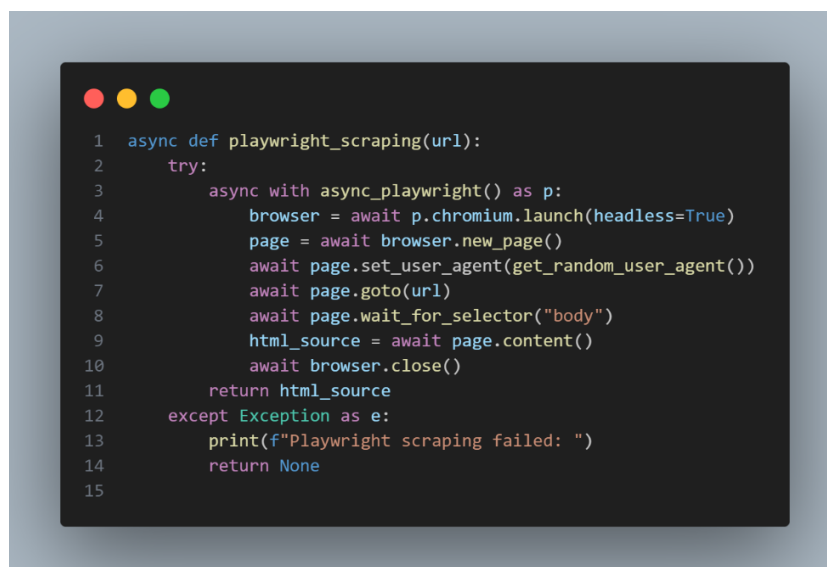
Nota: Función para recibir una respuesta dentro de un sitio web.

Figura 18.*Función de renderizado con Seleniuin*


```

1  async def selenium_scraping(url):
2      chrome_options = Options()
3      chrome_options.add_argument("user-agent=" + get_random_user_agent())
4      chrome_options.add_argument("--no-sandbox")
5      chrome_options.add_argument("--disable-dev-shm-usage")
6      chrome_options.add_argument("--disable-infobars")
7      chrome_options.add_argument("--disable-extensions")
8      proxy = get_random_proxy()
9
10     # Configuración del proxy para Selenium
11     chrome_options.add_argument(f'--proxy-server={proxy}')
12
13     try:
14         driver = await asyncio.to_thread(webdriver.Chrome, options=chrome_options)
15         driver.get(url)
16         time.sleep(3) # Opcional, para dar tiempo a cargar la página
17         html_source = driver.page_source
18         driver.quit()
19         return html_source
20     except Exception as e:
21         print(f"Selenium scraping failed: ")
22         return None
23

```

Nota: Función de renderizado aplicando Seleniuin para cargar la imagen de un sitio web.**Figura 19.***Función Playwright carga rapida*


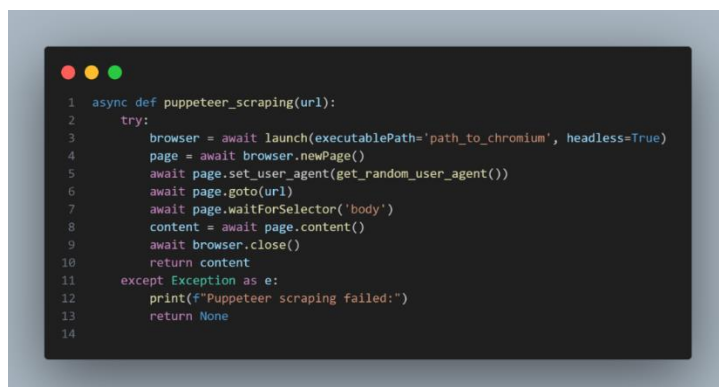
```

1  async def playwright_scraping(url):
2      try:
3          async with async_playwright() as p:
4              browser = await p.chromium.launch(headless=True)
5              page = await browser.new_page()
6              await page.set_user_agent(get_random_user_agent())
7              await page.goto(url)
8              await page.wait_for_selector("body")
9              html_source = await page.content()
10             await browser.close()
11             return html_source
12     except Exception as e:
13         print(f"Playwright scraping failed: ")
14         return None
15

```

Nota: Aplicando Playwright para carga rápida la URL

Figura 20.
Función de carga con navegador emulado



```

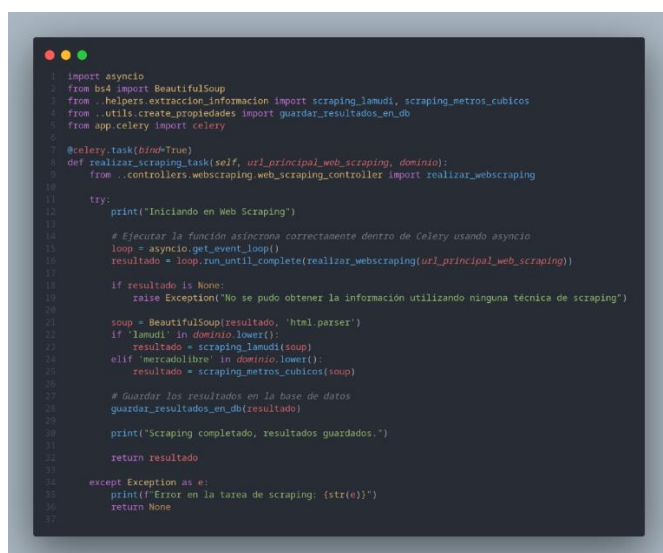
1  async def puppeteer_scraping(url):
2      try:
3          browser = await launch(executablePath='path_to_chromium', headless=True)
4          page = await browser.newPage()
5          await page.set_user_agent(get_random_user_agent())
6          await page.goto(url)
7          await page.waitForSelector('body')
8          content = await page.content()
9          await browser.close()
10         return content
11     except Exception as e:
12         print(f"Puppeteer scraping failed:")
13         return None
14

```

Nota: Función de que carga la página con un emulador de navegador

Para optimizar este proceso y evitar que interfiera con otras tareas que se ejecutan de manera consecutiva, se implementa la librería Celery. Está gestionando las tareas de forma eficiente y sin sobrecargar el sistema, lo que garantiza que el rendimiento general no se vea afectado.

Figura 21 .
Implementación con Celery



```

1  import asyncio
2  from bs4 import BeautifulSoup
3  from .helpers.extraccion_informacion import scraping_lamudi, scraping_metros_cubicos
4  from .utils.create_propiedades import guardar_resultados_en_db
5  from app.celery import celery
6
7  @celery.task(bind=True)
8  def realizar_scraping_task(self, url_principal_web_scraping, dominio):
9      from .controllers.webscraping.web_scraping_controller import realizar_webscraping
10
11      try:
12          print('Iniciando en Web Scraping')
13
14          # Ejecutar la función asíncrona correctamente dentro de Celery usando asyncio
15          loop = asyncio.get_event_loop()
16          resultado = loop.run_until_complete(realizar_webscraping(url_principal_web_scraping))
17
18          if resultado is None:
19              raise Exception("No se pudo obtener la información utilizando ninguna técnica de scraping")
20
21          soup = BeautifulSoup(resultado, 'html.parser')
22          if 'lamudi' in dominio.lower():
23              resultado = scraping_lamudi(soup)
24          elif 'mercadolibre' in dominio.lower():
25              resultado = scraping_metros_cubicos(soup)
26
27          # Guardar los resultados en la base de datos
28          guardar_resultados_en_db(resultado)
29
30          print('Scraping completado, resultados guardados.')
31
32          return resultado
33
34      except Exception as e:
35          print(f'Error en la tarea de scraping: {str(e)}')
36          return None
37

```

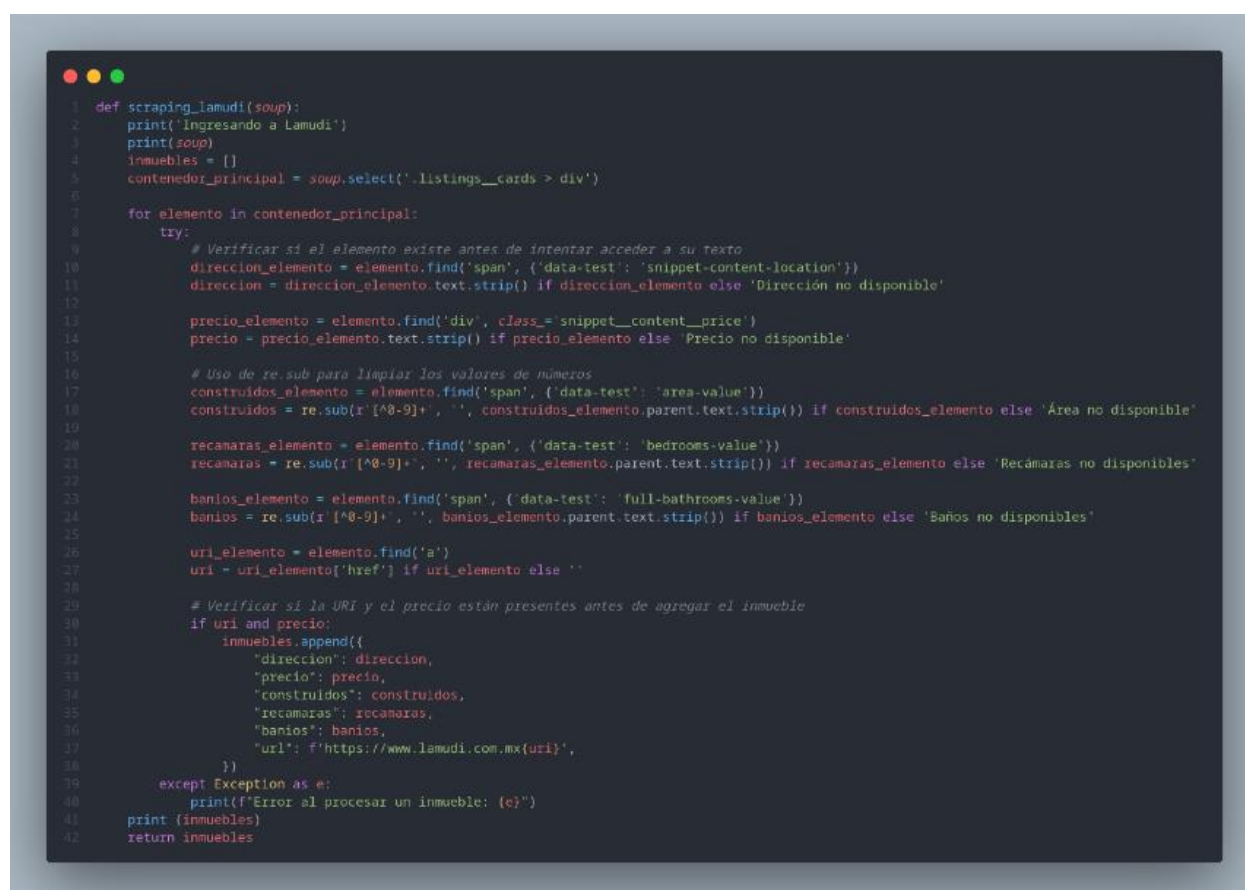
Nota: Función aplicando Celery para trabajo en segundo plano.

- **Selección de información Relevante**

Para evitar confusiones al extraer información de diferentes páginas web, se asignaron clases específicas a cada una de ellas. Esto permite un filtrado preciso de los datos que deben ser extraídos, garantizando que solo se obtenga la información relevante de cada página. Al aplicar este filtrado, se mejora la precisión y eficiencia del proceso de Web Scraping, asegurando que los datos no se mezclen entre diferentes fuentes.

Figura 22.

Búsqueda de información importante



```

1 def scraping_lamudi(soup):
2     print('Ingresando a Lamudi')
3     print(soup)
4     inmuebles = []
5     contenedor_principal = soup.select('.listings_cards > div')
6
7     for elemento in contenedor_principal:
8         try:
9             # Verificar si el elemento existe antes de intentar acceder a su texto
10            direccion_elemento = elemento.find('span', {'data-test': 'snippet-content-location'})
11            direccion = direccion_elemento.text.strip() if direccion_elemento else 'Dirección no disponible'
12
13            precio_elemento = elemento.find('div', class_='snippet_content_price')
14            precio = precio_elemento.text.strip() if precio_elemento else 'Precio no disponible'
15
16            # Uso de re.sub para limpiar los valores de números
17            construidos_elemento = elemento.find('span', {'data-test': 'area-value'})
18            construidos = re.sub(r'[^\d-]+', '', construidos_elemento.parent.text.strip()) if construidos_elemento else 'Área no disponible'
19
20            recamaras_elemento = elemento.find('span', {'data-test': 'bedrooms-value'})
21            recamaras = re.sub(r'[^\d-]+', '', recamaras_elemento.parent.text.strip()) if recamaras_elemento else 'Recámaras no disponibles'
22
23            banios_elemento = elemento.find('span', {'data-test': 'full-bathrooms-value'})
24            banios = re.sub(r'[^\d-]+', '', banios_elemento.parent.text.strip()) if banios_elemento else 'Baños no disponibles'
25
26            uri_elemento = elemento.find('a')
27            uri = uri_elemento['href'] if uri_elemento else ''
28
29            # Verificar si la URI y el precio están presentes antes de agregar el inmueble
30            if uri and precio:
31                inmuebles.append({
32                    "direccion": direccion,
33                    "precio": precio,
34                    "construidos": construidos,
35                    "recamaras": recamaras,
36                    "banios": banios,
37                    "url": f'https://www.lamudi.com.mx{uri}',
38                })
39            except Exception as e:
40                print(f'Error al procesar un inmueble: {e}')
41        print(inmuebles)
42    return inmuebles

```

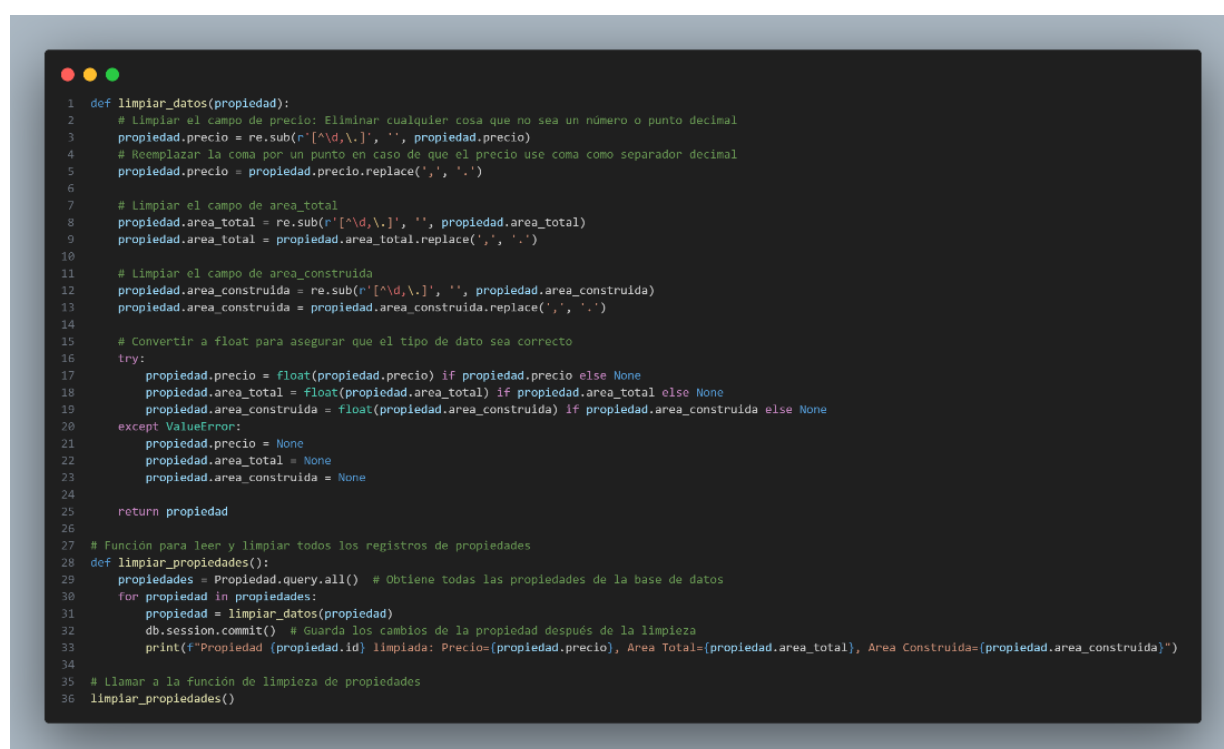
Nota: Esta función buscara las etiquetas luego las limpiara para poder almacenarlas luego.

- **Limpieza de datos inicial**

Para la limpieza de datos inicial se lo aplica para poder tener la información más estandarizada ya que al tener diferentes fuentes los formatos de información cambian y para eso se aplica las funciones de para poder tener más control sobre los datos

Figura 23.

Limpieza de datos



```

1 def limpiar_datos(propiedad):
2     # Limpiar el campo de precio: Eliminar cualquier cosa que no sea un número o punto decimal
3     propiedad.precio = re.sub(r'[^d\.]', '', propiedad.precio)
4     # Reemplazar la coma por un punto en caso de que el precio use coma como separador decimal
5     propiedad.precio = propiedad.precio.replace(',', '.')
6
7     # Limpiar el campo de area_total
8     propiedad.area_total = re.sub(r'[^d\.]', '', propiedad.area_total)
9     propiedad.area_total = propiedad.area_total.replace(',', '.')
10
11    # Limpiar el campo de area_construida
12    propiedad.area_construida = re.sub(r'[^d\.]', '', propiedad.area_construida)
13    propiedad.area_construida = propiedad.area_construida.replace(',', '.')
14
15    # Convertir a float para asegurar que el tipo de dato sea correcto
16    try:
17        propiedad.precio = float(propiedad.precio) if propiedad.precio else None
18        propiedad.area_total = float(propiedad.area_total) if propiedad.area_total else None
19        propiedad.area_construida = float(propiedad.area_construida) if propiedad.area_construida else None
20    except ValueError:
21        propiedad.precio = None
22        propiedad.area_total = None
23        propiedad.area_construida = None
24
25    return propiedad
26
27 # Función para leer y limpiar todos los registros de propiedades
28 def limpiar_propiedades():
29     propiedades = Propiedad.query.all() # Obtiene todas las propiedades de la base de datos
30     for propiedad in propiedades:
31         propiedad = limpiar_datos(propiedad)
32         db.session.commit() # Guarda los cambios de la propiedad después de la limpieza
33         print(f"Propiedad {propiedad.id} limpiada: Precio={propiedad.precio}, Area Total={propiedad.area_total}, Area Construida={propiedad.area_construida}")
34
35 # Llamar a la función de limpieza de propiedades
36 limpiar_propiedades()

```

Nota: Esta función limpiara los datos en caso de que exista algunas excepciones.

- **Almacenamiento en la base de datos**

Todos los datos que se encuentren se almacenaran en la base de datos para poder tener el control y siempre tener un respaldo de la información además de poder visualizar los datos recolectados

Figura 24.

Almacenamiento en la base de datos

```

1  db = SessionLocal()
2  try:
3      # Iteramos sobre todas las propiedades obtenidas
4      for resultado in resultados:
5          # Crear una nueva propiedad con los datos obtenidos del scraping
6          propiedad = Propiedad(
7              precio=resultado.get('precio'),
8              direccion=resultado.get('direccion'),
9              estado=resultado.get('estado'),
10             area_total=resultado.get('construidos'),
11             area_construida=resultado.get('construidos'),
12             num_habitaciones=resultado.get('recamaras'),
13             num_banos=resultado.get('banios'),
14             num_estacionamiento=resultado.get('estacionamiento'),
15             url=resultado.get('url'),
16         )
17         db.add(propiedad)
18         db.commit()
19         db.refresh(propiedad)
20         if resultado.get('latitud') and resultado.get('longitud'):
21             ubicacion = Ubicacion(
22                 propiedad_id=propiedad.id,
23                 latitud=resultado.get('latitud'),
24                 longitud=resultado.get('longitud')
25             )
26             db.add(ubicacion)
27             db.commit()
28
29
30 except Exception as e:
31     db.rollback()
32     print(e)
33

```

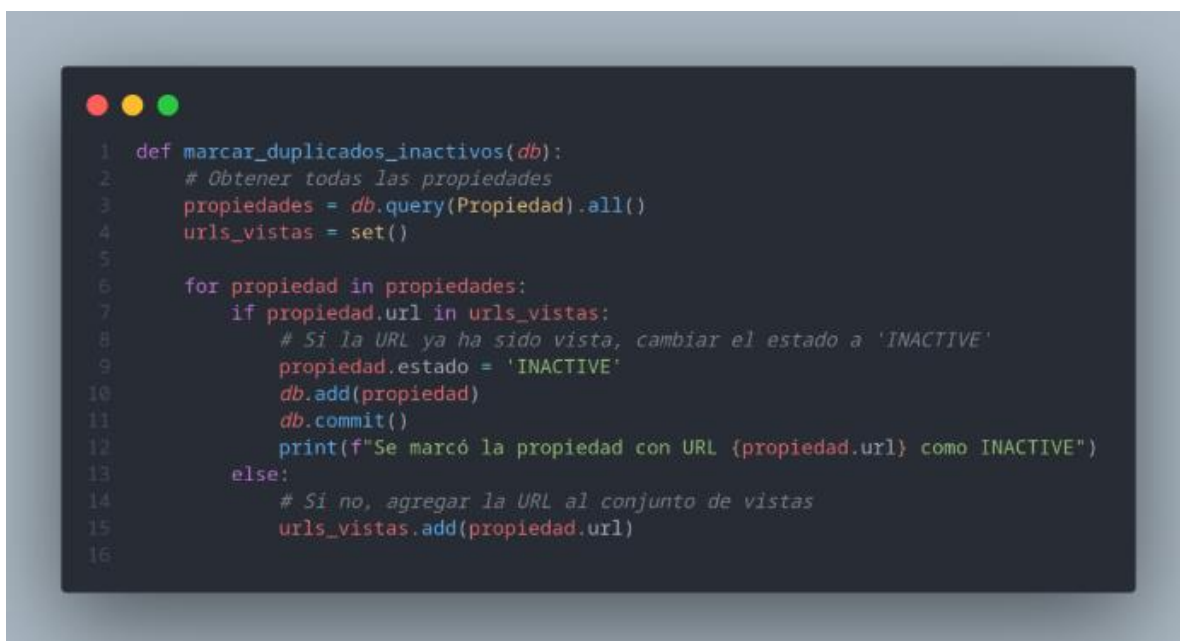
Nota: Una vez encontrado toda la información se procedió con el almacenado en la base de datos.

- **Limpieza de duplicados**

Al almacenar en una base de datos no se garantiza que no exista duplicados por eso se implementó esta función ya que puede existir doble registros de una sola propiedad garantizando así una fuente de información confiable y certera

Figura 25.

Limpieza de duplicados



```
1 def marcar_duplicados_inactivos(db):
2     # Obtener todas las propiedades
3     propiedades = db.query(Propiedad).all()
4     urls_vistas = set()
5
6     for propiedad in propiedades:
7         if propiedad.url in urls_vistas:
8             # Si la URL ya ha sido vista, cambiar el estado a 'INACTIVE'
9             propiedad.estado = 'INACTIVE'
10            db.add(propiedad)
11            db.commit()
12            print(f"Se marcó la propiedad con URL {propiedad.url} como INACTIVE")
13        else:
14            # Si no, agregar la URL al conjunto de vistas
15            urls_vistas.add(propiedad.url)
16
```

Nota: Función Limpia datos duplicados en la base de datos.

- **Completado de información Faltante aplicando IA**

Para completar la información faltante, ya se vio que en las distintas fuentes no siempre se coloca la información en las etiquetas seleccionadas dando campos vacíos.

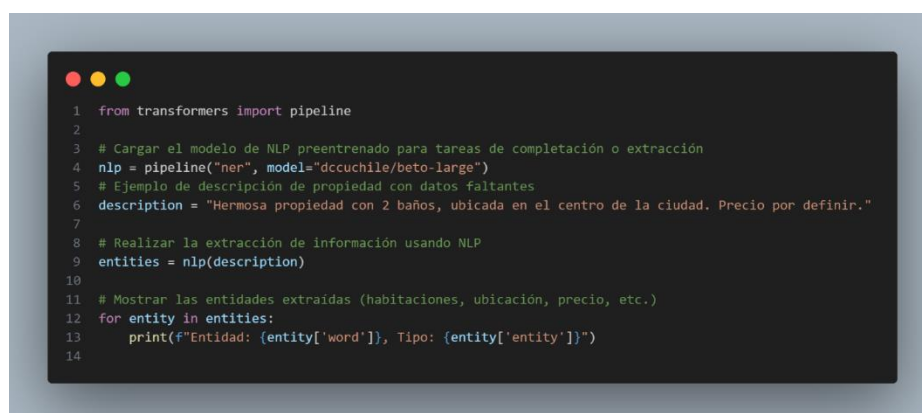
Con esto en mente se vio que la información (no en todas las páginas web) la información faltante se encuentra en la parte de descripción de la propiedad así que aplicar NLP para poder extraer esta información.

A continuación, detallare cómo funciona internamente la herramienta NLP

El código aplicado es el siguiente

Figura 26.

Completado de información importante



```
1 from transformers import pipeline
2
3 # Cargar el modelo de NLP preentrenado para tareas de completación o extracción
4 nlp = pipeline("ner", model="dccuchile/beto-large")
5 # Ejemplo de descripción de propiedad con datos faltantes
6 description = "Hermosa propiedad con 2 baños, ubicada en el centro de la ciudad. Precio por definir."
7
8 # Realizar la extracción de información usando NLP
9 entities = nlp(description)
10
11 # Mostrar las entidades extraídas (habitaciones, ubicación, precio, etc.)
12 for entity in entities:
13     print(f"Entidad: {entity['word']}, Tipo: {entity['entity']}")
14
```

Nota: La función analizará el texto entrante y sacará etiquetas con la información requerida o encontrada.

Por el exterior solo parece un simple código, pero internamente aplica lo siguiente:

Tokenización: Consiste en dividir el texto en unidades más pequeñas, como palabras o frases (tokens). La tokenización es un paso inicial necesario para la mayoría de las tareas de procesamiento, ya que permite dividir el texto en partes que pueden ser más fácilmente analizadas.

Si tenemos el texto: "Hermosa casa con 3 habitaciones."

Los tokens serían: ['Hermosa', 'casa', 'con', '3', 'habitaciones'].

Etiquetado de Partes del Habla (POS tagging): Es el proceso de etiquetar las palabras en una oración con su respectiva categoría gramatical (sustantivo, verbo, adjetivo, etc.). Esto ayuda a identificar entidades importantes como números (por ejemplo, "3 habitaciones") o sustantivos relacionados con las propiedades (por ejemplo, "baño", "precio").

Reconocimiento de Entidades Nombradas (NER): El NER permite identificar y clasificar entidades clave dentro de un texto, como nombres de lugares, personas, números, fechas, etc. se usaría para identificar elementos específicos como "número de habitaciones", "precio", "ubicación", etc., dentro de las descripciones de las propiedades.

3.1.5 Evaluación

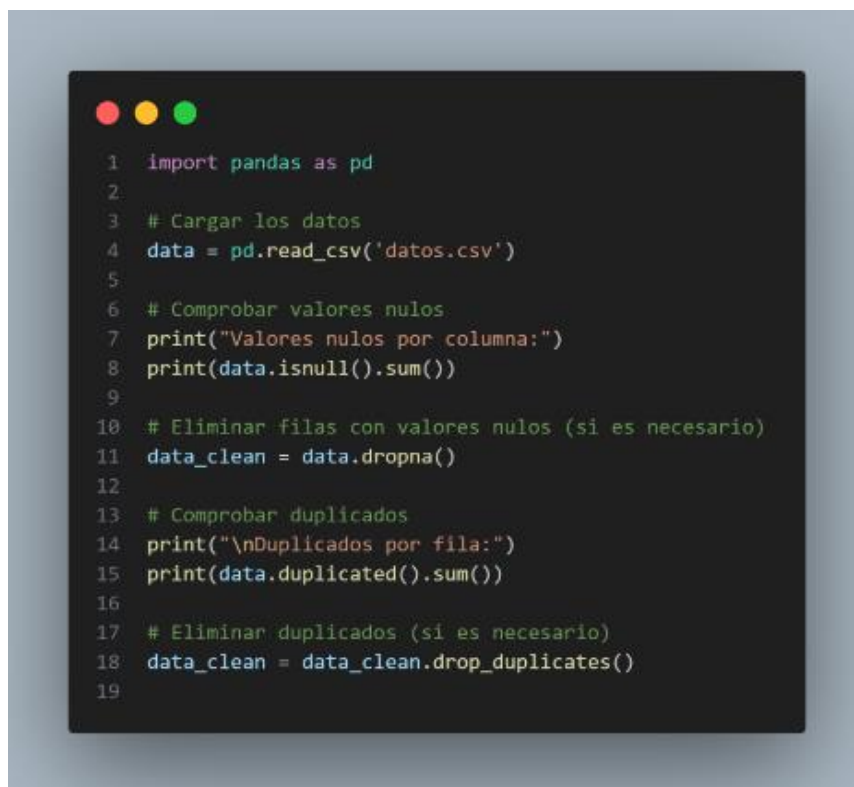
En esta parte analizaremos y aseguraremos que los datos extraídos sean completos, correctos y consistentes

- **Evaluación de la calidad de los datos:**

En esta fase evaluaremos los datos registrados en el sistema para ello descargaremos el archivo csv y lo evaluaremos con la herramienta de pandas, buscaremos principalmente los nulos y duplicados.

Figura 27.

Función para evaluar los datos



```
1  import pandas as pd
2
3  # Cargar los datos
4  data = pd.read_csv('datos.csv')
5
6  # Comprobar valores nulos
7  print("Valores nulos por columna:")
8  print(data.isnull().sum())
9
10 # Eliminar filas con valores nulos (si es necesario)
11 data_clean = data.dropna()
12
13 # Comprobar duplicados
14 print("\nDuplicados por fila:")
15 print(data.duplicated().sum())
16
17 # Eliminar duplicados (si es necesario)
18 data_clean = data_clean.drop_duplicates()
19
```

Nota: La función evaluará los datos extraídos y contara cuantos datos faltan completar.

Para ello lo aplicaremos a total de registros encontrados siendo una cantidad de 4600 registros

Dando el siguiente resultado

Figura 28.*Resultados de la calidad de datos*

```

1  Valores nulos por columna:
2  area_construida      3
3  area_total          3
4  ciudad              2
5  create_date         0
6  deleted_at          20
7  descripcion         0
8  direccion           1
9  estado              0
10 fecha_publicacion    0
11 id                  0
12 num_banos           1
13 num_estacionamiento  1
14 num_habitaciones     1
15 precio              0
16 status              0
17 titulo              0
18 ubicacion           1
19 update_date         0
20 url                 0
21 user_create         0
22 latitud             1
23 longitud            1
24 dtype: int64
25
26 Duplicados por fila:
27 0

```

Nota: Resultado de la función para evaluar datos en donde se refleja la cantidad de datos vacíos y duplicados en la base de datos

Con estos datos sacamos los cálculos para ver el nivel de calidad de los datos

Ecuación 1 Calidad de datos

$$CD = 1 - \frac{\sum \frac{\text{cantidad de nulos}}{\text{cantidad de datos}}}{\text{cantidad de campos}}$$

Calculamos la sumatoria

$$\sum \frac{\text{cantidad de nulos}}{\text{cantidad de datos}}$$

$$\frac{17}{2300}$$

Calculamos la calidad de los datos

$$CD = 1 - \frac{\frac{17}{2300}}{22}$$

$$CD = 0.9997$$

$$CD = 99.97\%$$

Con esto mencionamos que un 99.97% de los datos corresponden a la calidad establecida.

Ahora con la verificación de duplicados vemos un total de duplicados de 0 en todos los registros con lo cual aseguramos que los datos son únicos y confiables.

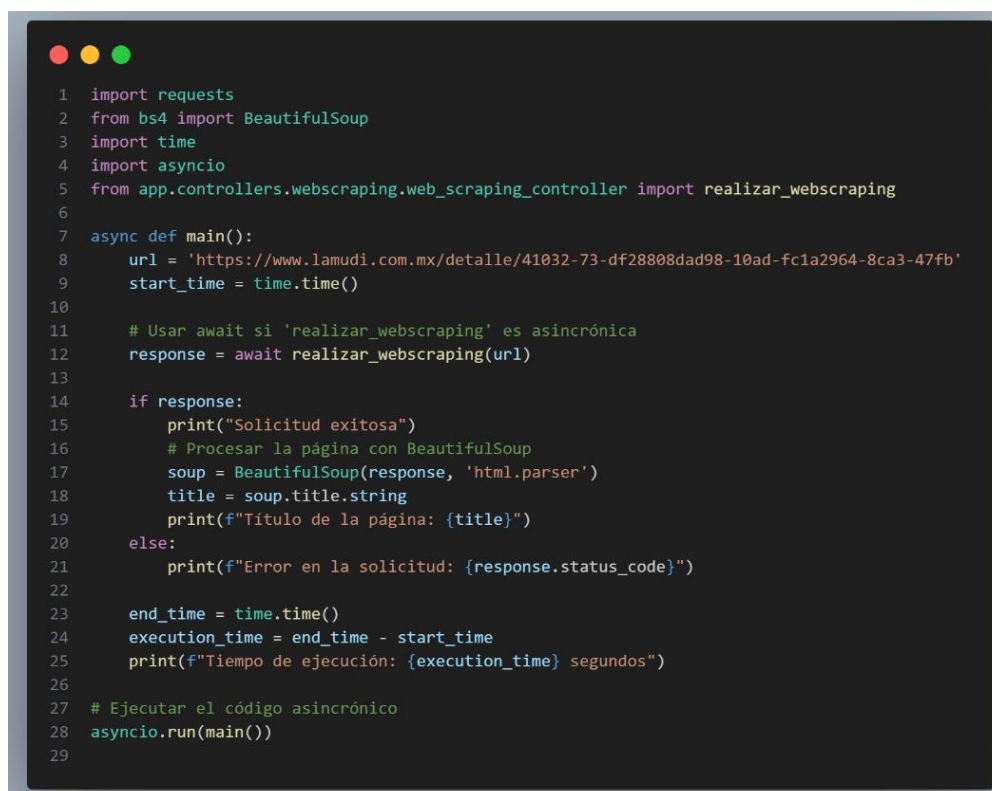
- **Evaluación de la eficiencia del Web Scraping**

Para ello evaluaremos las funciones ya registradas para Web Scraping y el tiempo que tarda en ejecutar, considerando que normalmente se tarda 5sec al cargar la página y extraer la información.

Para ello usaremos el siguiente código

Figura 29.

Función del rendimiento del Web Scraping

A screenshot of a code editor with a dark background and light-colored text. The code is written in Python and is numbered from 1 to 29. It imports 'requests', 'BeautifulSoup' from 'bs4', 'time', and 'asyncio'. It then imports 'realizar_web scraping' from 'app.controllers.web scraping.web_scraping_controller'. The code defines an asynchronous 'main' function that takes a URL, starts a timer, calls 'realizar_web scraping' with 'await', processes the response with 'BeautifulSoup', prints the title, and calculates the execution time. Finally, it runs the 'main' function using 'asyncio.run'.

```
1 import requests
2 from bs4 import BeautifulSoup
3 import time
4 import asyncio
5 from app.controllers.webscraping.web_scraping_controller import realizar_web scraping
6
7 async def main():
8     url = 'https://www.lamudi.com.mx/detalle/41032-73-df28808dad98-10ad-fc1a2964-8ca3-47fb'
9     start_time = time.time()
10
11     # Usar await si 'realizar_web scraping' es asíncrona
12     response = await realizar_web scraping(url)
13
14     if response:
15         print("Solicitud exitosa")
16         # Procesar la página con BeautifulSoup
17         soup = BeautifulSoup(response, 'html.parser')
18         title = soup.title.string
19         print(f"Título de la página: {title}")
20     else:
21         print(f"Error en la solicitud: {response.status_code}")
22
23     end_time = time.time()
24     execution_time = end_time - start_time
25     print(f"Tiempo de ejecución: {execution_time} segundos")
26
27 # Ejecutar el código asíncrono
28 asyncio.run(main())
29
```

Nota: La función usará el método de realizar Web Scraping y evaluará el tiempo que tarda en ejecutarse.

Resultado

Figura 30.

Resutaldo de la evaluación en tiempo

```
https://www.lamudi.com.mx/detalle/41032-73-df28808dad98-10ad-fc1a2964-8ca3-47fb
Solicitud exitosa
Título de la página: Casa en venta, Cuernavaca, Acapatzingo, Morelos | Lamudi.com.mx
Tiempo de ejecución: 3.3675715923309326 segundos
```

Nota: Resultados de la función aplicada.

Siendo $3\text{sec} < 5\text{sec}$ afirmamos que los datos se están extrayendo eficientemente ahorrando tiempo y recursos

- **Evaluación del rendimiento de la IA**

Para esta sección evaluaremos puntos importantes que son la precisión, y el tiempo de ejecución

Evaluación de la precisión, recall, F1-Score:

Para ello se aplicará el siguiente código:

Figura 31.*Test para determinar la precisión del modelo*

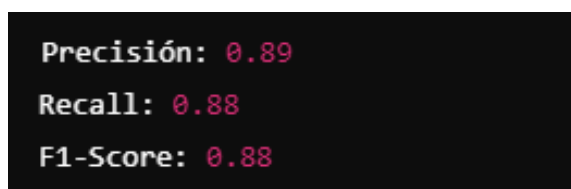

```

1 from sklearn.metrics import precision_score, recall_score, f1_score
2 import numpy as np
3 import pandas as pd
4
5 # Leer los archivos CSV con los valores reales y las predicciones
6 df_y_true = pd.read_csv('valores_reales.csv') # Archivo con valores reales
7 df_y_pred = pd.read_csv('valores_predicidos.csv') # Archivo con predicciones
8
9 # Asegúrate de que las columnas de los CSV se llamen 'y_true' y 'y_pred' o ajusta los nombres si son diferentes
10 y_true = df_y_true['y_true'].values # Convertir la columna de valores reales a un array de numpy
11 y_pred = df_y_pred['y_pred'].values # Convertir la columna de predicciones a un array de numpy
12
13 # Calcular precisión, recall y F1-score
14 precision = precision_score(y_true, y_pred)
15 recall = recall_score(y_true, y_pred)
16 f1 = f1_score(y_true, y_pred)
17
18 # Imprimir los resultados
19 print(f"Precisión: {precision:.2f}")
20 print(f"Recall: {recall:.2f}")
21 print(f"F1-Score: {f1:.2f}")
22

```

Nota: La función evaluará los resultados de la IA con el método de RECALL y F1-SCORE.

Dando como resultados

Figura 32.*Resultados de la evaluación*


```

Precisión: 0.89
Recall: 0.88
F1-Score: 0.88

```

Precisión (0.89): De todas las veces que el modelo predijo positivo, un 89% fueron correctas.

Recall (0.88): De todos los casos positivos reales, el modelo detectó el 88%.

F1-Score (0.88): El F1-score es un equilibrio entre precisión y recall, y en este caso, es 0.88, lo que indica que el modelo tiene un rendimiento razonable, pero podría mejorar en cuanto a la detección de los positivos reales.

Para el tiempo de ejecución se aplicará el siguiente código

Figura 33.

Test Tiempo de ejecución de la IA



```

1 import time
2 import spacy
3
4 # Cargar el modelo de spaCy para NER
5 start_time = time.time() # Iniciar el temporizador
6
7 nlp = spacy.load("valor_propiedad_ia") # Cargar el modelo preentrenado de spaCy para NER
8
9 # Medir el tiempo de carga del modelo
10 load_time = time.time() - start_time
11 print(f"Tiempo de carga del modelo: {load_time:.4f} segundos")
12
13 # Texto de ejemplo para realizar NER
14 text = ""
15 MODELO TOLEDO Doble Altura Terreno 237 m2 Acabados de lujo 3 habitaciones con baño completo y clóset cada una Jardín privado Cocina equipada Estacionamiento para 3 autos uno techado
16 ""
17
18 # Iniciar el temporizador para la predicción
19 start_time = time.time()
20
21 # Procesar el texto para NER
22 doc = nlp(text)
23
24 # Medir el tiempo de predicción
25 predict_time = time.time() - start_time
26 print(f"Tiempo de predicción: {predict_time:.4f} segundos")

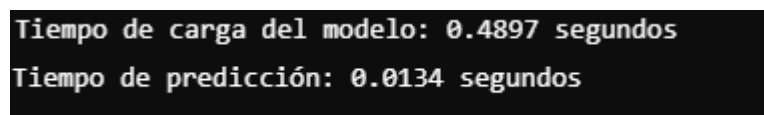
```

Nota: El test se encargará de medir el tiempo de carga del modelo y el tiempo de predicción.

Dando el siguiente resultado

Figura 34.

Resultados del test



```

Tiempo de carga del modelo: 0.4897 segundos
Tiempo de predicción: 0.0134 segundos

```

Con esto confirmamos que el modelo responde a las exigencias requeridas.

3.1.6 Despliegue

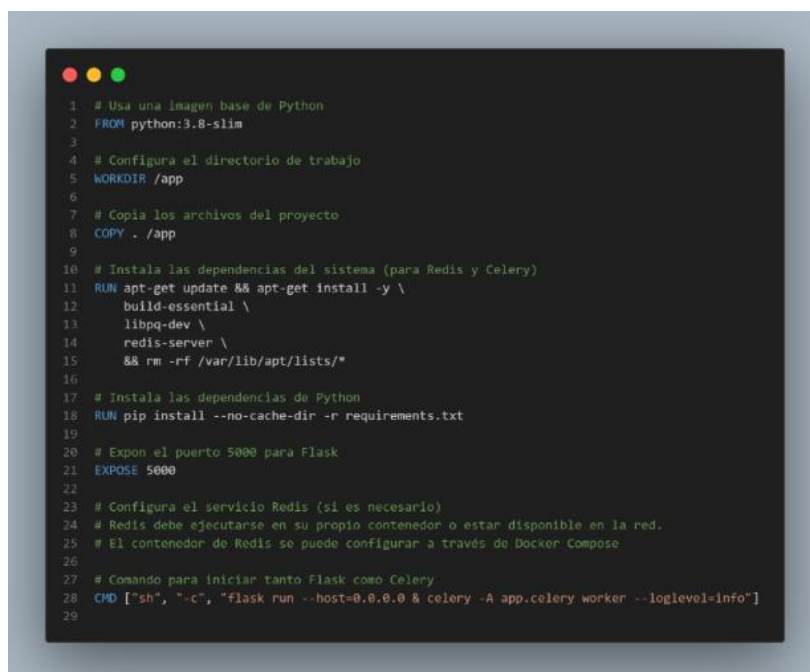
Una vez completado y configurado el modelo para la extracción de información, será llevar el sistema implementando API REST full en el backend (usando flask) que permita la interacción del frontend con el sistema de Web Scraping y completado de datos, además de garantizar la seguridad y eficiencia del sistema

- **Preparación del entorno de despliegue**

Se implementará el despliegue con Docker siendo este un contenedor que no afectara a los demás sistemas que se encuentran en producción para eso aplicaremos la siguiente configuración

Figura 35.

Configuración del entorno en Docker



```

1  # Usa una imagen base de Python
2  FROM python:3.8-slim
3
4  # Configura el directorio de trabajo
5  WORKDIR /app
6
7  # Copia los archivos del proyecto
8  COPY . /app
9
10 # Instala las dependencias del sistema (para Redis y Celery)
11 RUN apt-get update && apt-get install -y \
12     build-essential \
13     libpq-dev \
14     redis-server \
15     && rm -rf /var/lib/apt/lists/*
16
17 # Instala las dependencias de Python
18 RUN pip install --no-cache-dir -r requirements.txt
19
20 # Expon el puerto 5000 para Flask
21 EXPOSE 5000
22
23 # Configura el servicio Redis (si es necesario)
24 # Redis debe ejecutarse en su propio contenedor o estar disponible en la red.
25 # El contenedor de Redis se puede configurar a través de Docker Compose
26
27 # Comando para iniciar tanto Flask como Celery
28 CMD ["sh", "-c", "flask run --host=0.0.0.0 & celery -A app.celery worker --loglevel=info"]
29

```

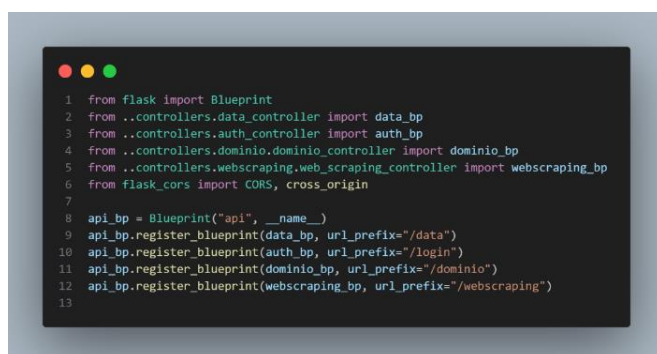
Nota: Configuración del entorno necesario para ejecutar la aplicación.

Siendo que en requirements.txt se tendrá todas las dependencias que se usará en el sistema

- **Nombramiento de las rutas de las APIs en Flask**

Figura 36.

Declaración de las APIs



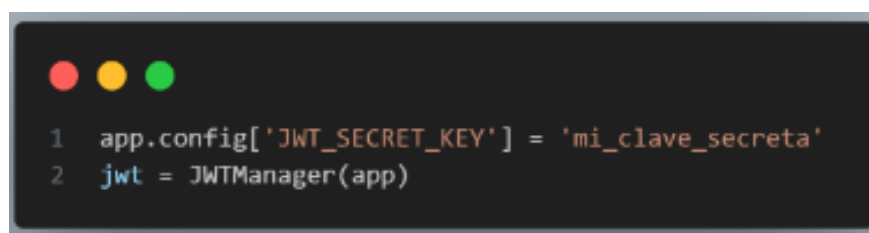
```
1 from flask import Blueprint
2 from ..controllers.data_controller import data_bp
3 from ..controllers.auth_controller import auth_bp
4 from ..controllers.dominio.dominio_controller import dominio_bp
5 from ..controllers.webscraping.web_scraping_controller import webscraping_bp
6 from flask_cors import CORS, cross_origin
7
8 api_bp = Blueprint("api", __name__)
9 api_bp.register_blueprint(data_bp, url_prefix="/data")
10 api_bp.register_blueprint(auth_bp, url_prefix="/login")
11 api_bp.register_blueprint(dominio_bp, url_prefix="/dominio")
12 api_bp.register_blueprint(webscraping_bp, url_prefix="/webscraping")
13
```

Nota: Determinación de las rutas configuradas a distintos controladores.

- **Implementación de seguridad en las Apis**

Figura 37.

Configuración del JWT



```
1 app.config['JWT_SECRET_KEY'] = 'mi_clave_secreta'
2 jwt = JWTManager(app)
```

Nota: Configuración básica de JWT, con esto ya tenemos el método de cifrado dentro del sistema aplicando Json Web Tokens.

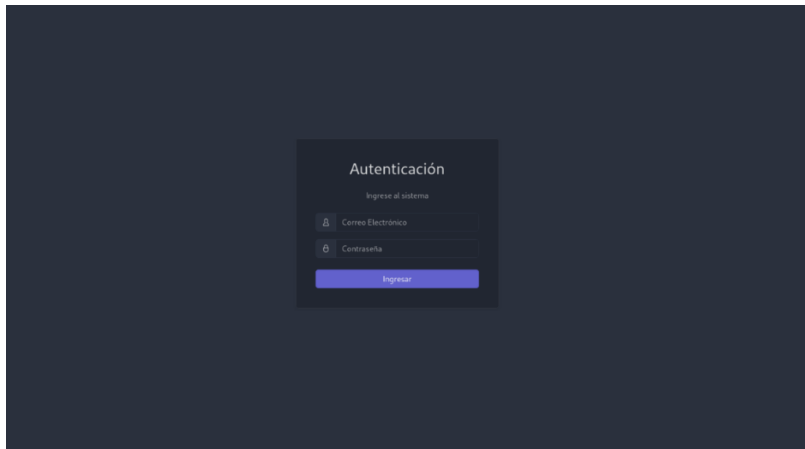
3.2 CONSTRUCCION DE LA INTERFAZ DE USUARIO

Para la construcción de la interfaz de usuario se implementará las APIs que se crearon en el lado del backend para ello utilizaremos el framework de JavaScript en frontend REACT con su extensión Material UI que nos dará estilos e iconos para poder aplicar en la interfaz de usuario

3.2.1 Ingreso al sistema

Figura 38.

Interfaz de autenticación



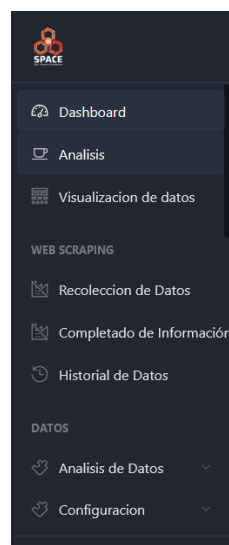
Nota: Interfaz de autenticación, el usuario podrá ingresar al sistema utilizando su correo corporativo y contraseña.

3.2.2 Inicio

En esta sección se vera la parte inicial del sistema una vez iniciada la sesión donde se podrá visualizar diagramas contadores y datos estadísticos extraídos desde el backend por consulta de APIs

Figura 39*Interfaz de inicio*

Además de poder ver los las siguientes pestañas

Figura 40*Menú Lateral*

Análisis:

Visualización de Datos:

Se podrá visualizar los datos que se están añadiendo y además se podrá descargar en los formatos requeridos para su posterior análisis incluyendo un filtro de datos para solo sacar lo necesario.

Figura 41.

Interfaz Visualización de datos

Dashboard

Inicio / Web Scraping

Buscar propiedades...

Descargar Excel Descargar CSV Descargar JSON Descargar TXT

ID	Título	Precio	Ciudad	Acciones
0a61fd3f-5d2f-4a71-a658-2a2d5f2e4294	Proyecto de Vivienda ALFIERI \$ 5,221,700 MXN Lamudi.com.mx	Desde \$ 5,221,700 MXN		Descripción
8f18e6c8-52a0-48f2-9122-131f0ba0785	¡BAJA DE PRECIO! Excelente Casa con Inmejorable Ubicación y Súper Distribución Lamudi.com.mx	\$ 10,400,000 MXN		Descripción
25dfaf3b-ba74-418c-acce-1316b4aaab12	🔥 CASA EN LAGOS RESIDENCIAL 🔥 Lamudi.com.mx	\$ 13,400,000 MXN		Descripción
da23b7a8-93f7-44e4-b730-63db47a722f1	🔥 SE VENDE CASA EN ZONA PLATEADA 🔥 Lamudi.com.mx	\$ 7,730,000 MXN		Descripción
a537e929-6bdc-454f-8272-8e9a45a1c890	PARAISO COUNTRY CLUB Lamudi.com.mx	\$ 16,200,000 MXN		Descripción
2590ad01-9fe2-4b9e-ae85-f714448a7a59	PARAISO COUNTRY CLUB Lamudi.com.mx	\$ 15,500,000 MXN		Descripción
057662d0-3f38-4167-a4be-8dbd34ed53be	Últimas Casas en Pre-venta! A 2 min de La Cacho, Perch by Cosanq Lamudi.com.mx	USD 329,000		Descripción
e9567b4e-fcf4-422a-9eb1-ae2dc391f091	Element; exclusiva mini-privada: Últimas casas! Lamudi.com.mx	USD 329,000		Descripción
ce9764e2-9e81-419c-ad03-a6be7c39f326	Estrena casa en privada residencial Lamudi.com.mx	\$ 3,858,000 MXN		Descripción
ade3b300-8da2-4976-9975-0200957880a8	Casa con recámara en planta baja Lamudi.com.mx	\$ 3,650,000 MXN		Descripción

« Anterior Siguiente »

3.2.3 Recolección de datos

En esta sección se comenzará con la extracción de información además de poder crear o insertar sitios web para la extracción de información de bienes raíces aplicando Web Scraping

- **Agregar Sitio Web:**

En esta sección se podrá registrar ver editar los distintos dominios que existen para extraer información

Lista de registros

Figura 42.

Interfaz lista de Registros.

Dashboard

Inicio / Web Scraping

Agregar Página Web Iniciar Web Scraping

#	Dominio	Logo	Tipo de Paginación	Categorías
1	https://www.inmuebles24.com		pagina	Ver Categorías
2	https://inmuebles.mercadolibre.com.mx		Desde	Ver Categorías
3	https://casas.trovit.com.mx		infinito	Ver Categorías
4	https://www.lamudi.com.mx		param	Ver Categorías

<<1>>

Crear nuevo dominio

Figura 43.

Interfaz Creación de dominios

Dashboard

Inicio / Web Scraping

Agregar Página Web Iniciar Web Scraping

Agregar Nueva Página Web

Dominio

Logo (URL)

Paginación

True

Tipo de Paginación

Slash

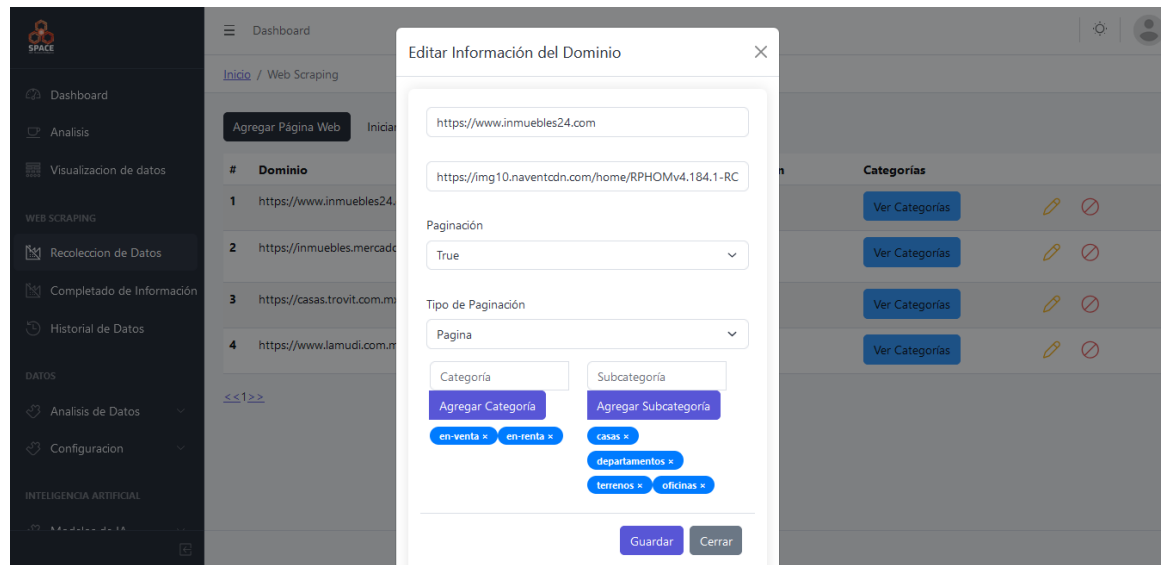
Categoría Subcategoría

Agregar Categoría Agregar Subcategoría

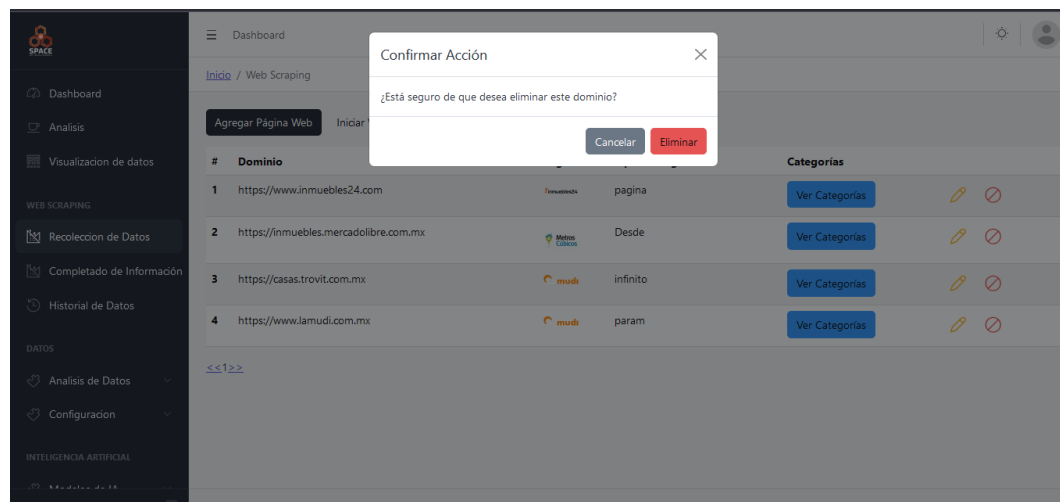
Registrar Dominio

Cerrar

Editar el registro

Figura 44.*Interfaz editar dominios*

Eliminar el registro

Figura 45.*Interfaz Eliminación de Registros*

- **Iniciar Web Scraping**

En esta sección se podrá visualizar un formulario para poder iniciar el Web Scraping, donde se seccionará el dominio, la categoría y la subcategoría.

Además de tener un límite de páginas web junto con la fecha (No se podrá ingresar fechas posteriores a la actual)

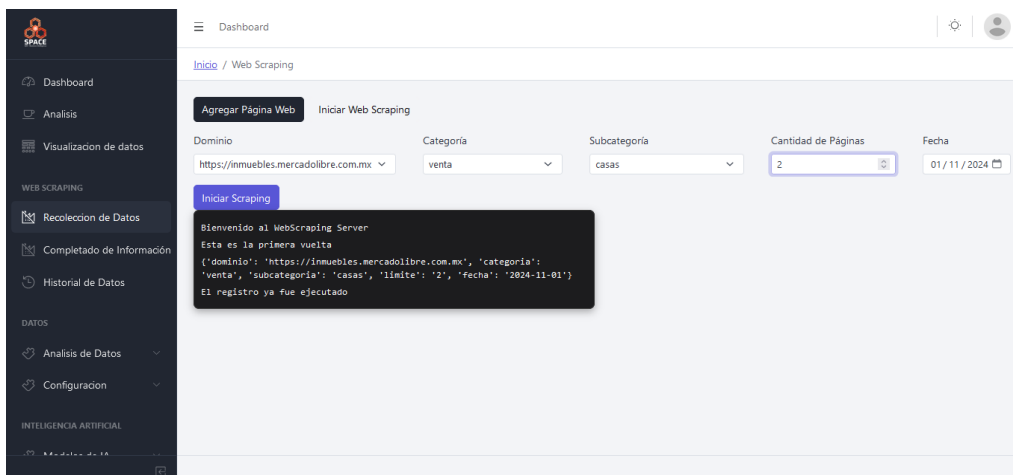
Figura 46.

Interfaz Recolección de datos

The screenshot displays the 'Iniciar Web Scraping' section of the SPACE application. On the left, a dark sidebar contains navigation links: Dashboard, Analisis, Visualización de datos, WEB SCRAPING (highlighted), Recoleccion de Datos, Completado de Información, Historial de Datos, DATOS, Analisis de Datos, Configuración, and INTELIGENCIA ARTIFICIAL. The main content area has a header 'Dashboard' and a sub-header 'Inicio / Web Scraping'. Below this, there are two buttons: 'Agregar Página Web' and 'Iniciar Web Scraping'. The 'Iniciar Web Scraping' button is highlighted in blue. A form is present with the following fields: 'Dominio' (a text input with 'https://inmuebles.mercadolibre.com.mx'), 'Categoría' (a dropdown menu with 'venta'), 'Subcategoría' (a dropdown menu with 'casas'), 'Cantidad de Páginas' (a text input with '42'), and 'Fecha' (a date picker with '01/11/2024'). A 'Iniciar Scraping' button is located below the form. A dark toast message at the bottom of the form area reads 'Bienvenido al WebScraping Server'.

Una vez ejecutándose se comenzará a mostrar la información en el cuadro de abajo sobre la información encontrada y como se está almacenando con esto también se visualizará en si es que los datos ingresados ya han sido procesados con anterioridad.

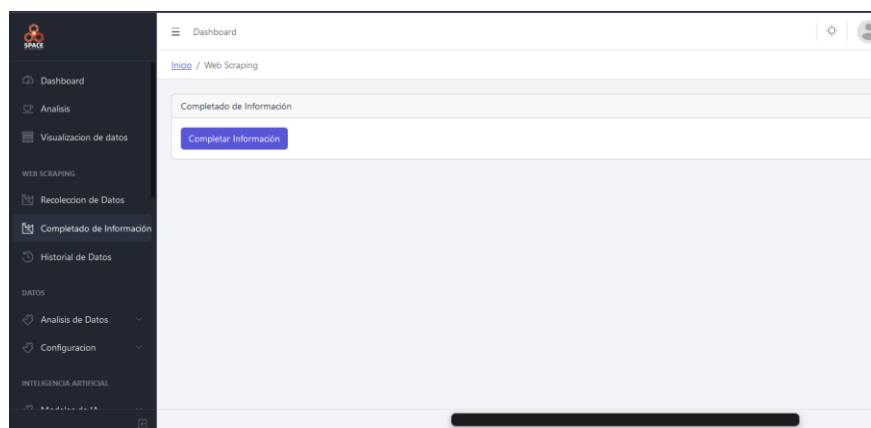
Figura 47.
Interfaz de visualización el la recoleccion de información



3.2.4 Completado de información

Cuando la información ya ha sido creada y llenada existe la opción de que algunos datos no se llenaron en este caso los de latitud y longitud, por eso se creó esta función que no ayudara a completar la información faltante

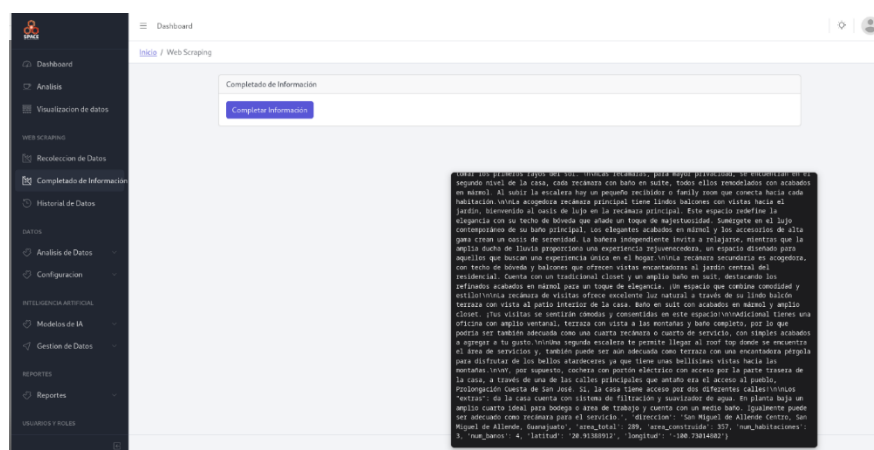
Figura 48.
Interfaz de Completado de información



Al iniciar con el completado si visualizara los datos que se están completando y si no existe datos que completar entonces informara sobre esto

Figura 49.

Interfaz Visualización de datos recolectados

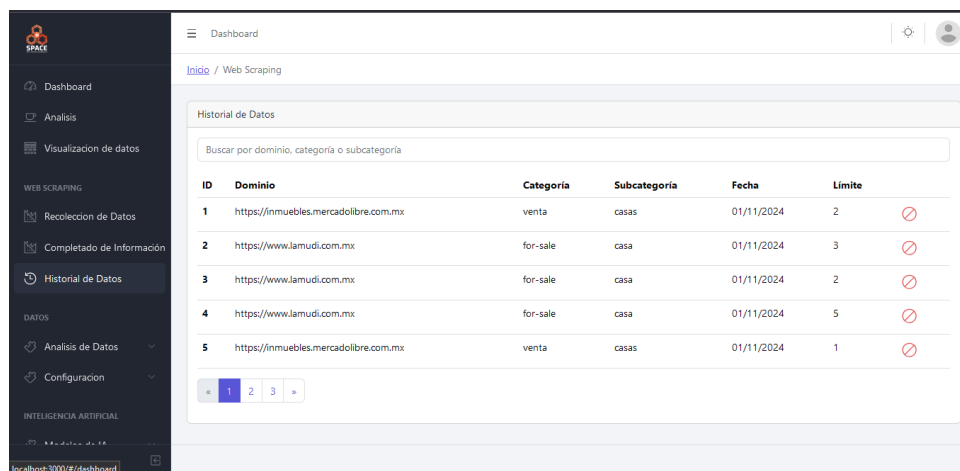


3.2.5 Historial de Datos

Visualizaremos el historial de los registros ingresados y solo se podrá eliminar si es necesario y querer volver a iniciar de nuevo la extracción

Figura 50.

Interfaz historial de datos

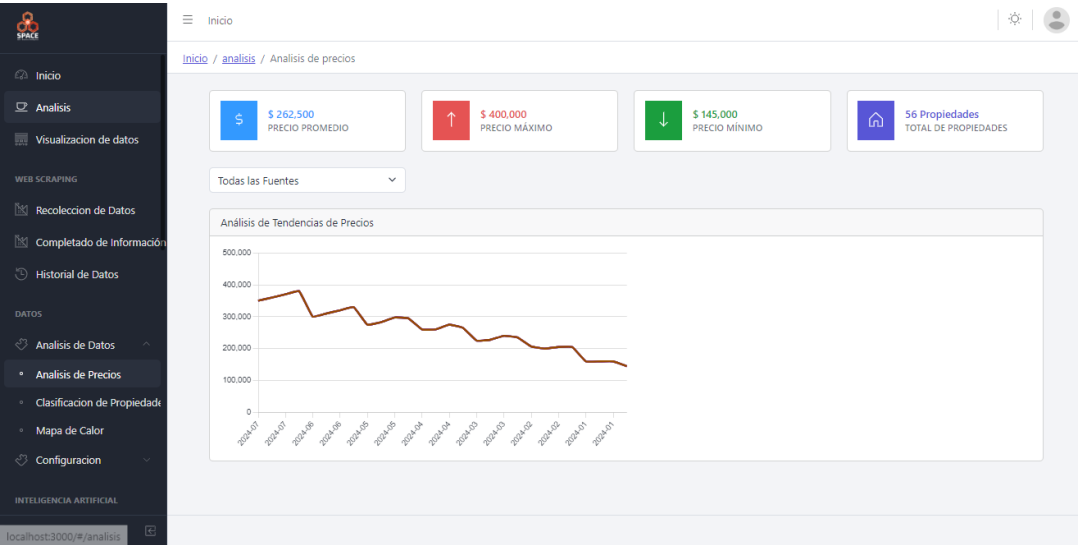


3.2.6 *Análisis de datos*

3.2.6.1 *Análisis de Precios*

Figura 51.

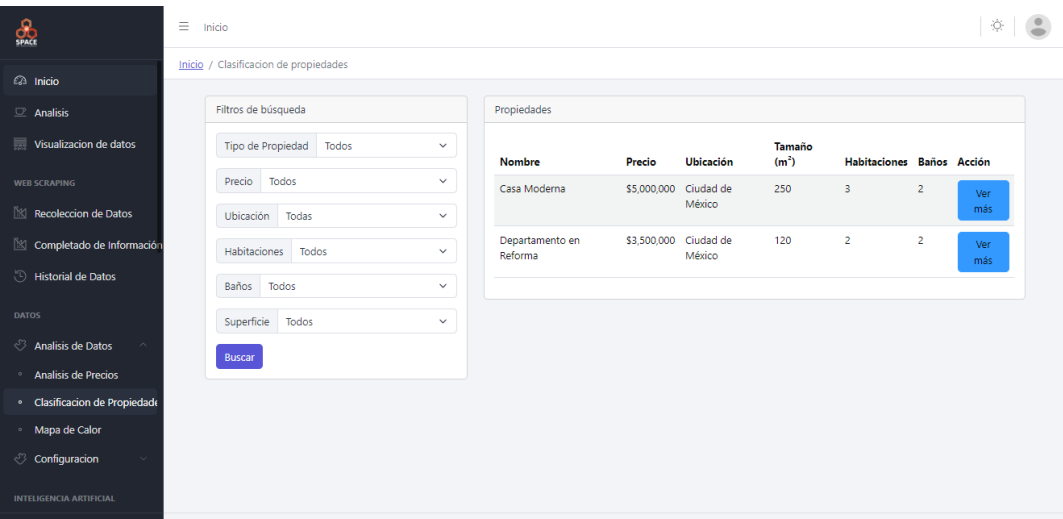
Interfaz de Análisis de precios



3.2.6.2 *Clasificación de propiedades*

Figura 52.

Visualización Interfaz de clasificación de propiedades

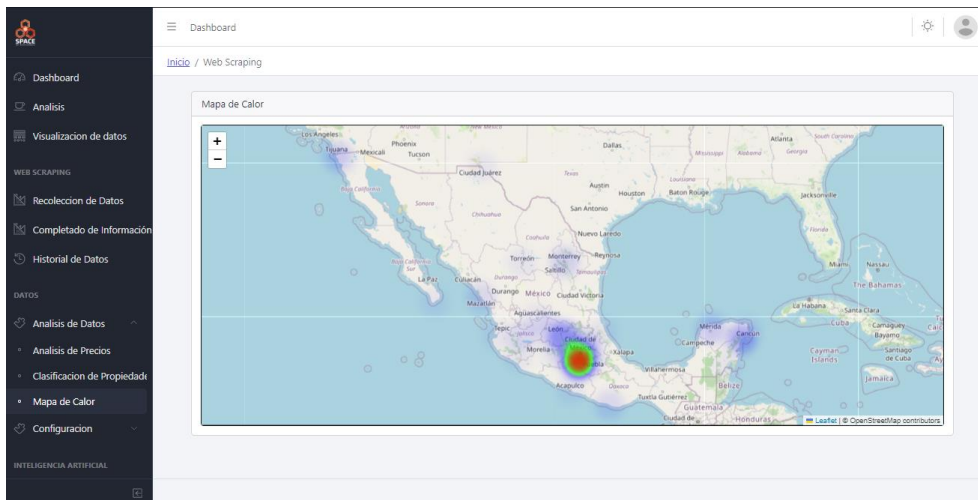


3.2.6.3 Mapa de Calor

Para este módulo veremos los resultados en un mapa de calor para ver en lugares son donde se mueve el mercado.

Figura 53.

Interfaz Mapa de calor



Nota: Interfaz de mapa de calor que muestra los focos de calor de propiedades registradas.

3.2.7 Modelos IA

En esta sección solo tendrá acceso el administrador y el encargado del entrenamiento.

Iniciamos con el entrenamiento del modelo de reconocimiento de entidades

3.2.7.1 Entrenamiento del modelo

Figura 54.

Interfaz de entrenamiento del modelo paso 1

The screenshot shows the SPARK application interface. On the left is a dark sidebar with a menu containing: 'Completado de Información', 'Historial de Datos', 'DATOS' (with sub-items 'Análisis de Datos' and 'Configuración'), 'INTELIGENCIA ARTIFICIAL' (with sub-items 'Modelos de IA', 'Entrenamiento de modelo', 'Evaluación de Modelos', and 'Aplicación de IA'), 'Gestión de Datos', 'REPORTES' (with sub-item 'Reportes'), and 'USUARIOS Y ROLES'. The main content area has a header 'Inicio' and a breadcrumb 'Inicio / Entrenamiento'. The title is 'Paso 1: Ingresa el texto' with the instruction 'Ingresa el texto que quieres etiquetar:'. Below this is a text input field with the placeholder 'Escribe el texto aquí...' and a blue 'Siguiente' button.

Para esta sección el sistema pedirá que se ingrese el texto de donde se encuentra la información importante para el entrenamiento

Figura 55.

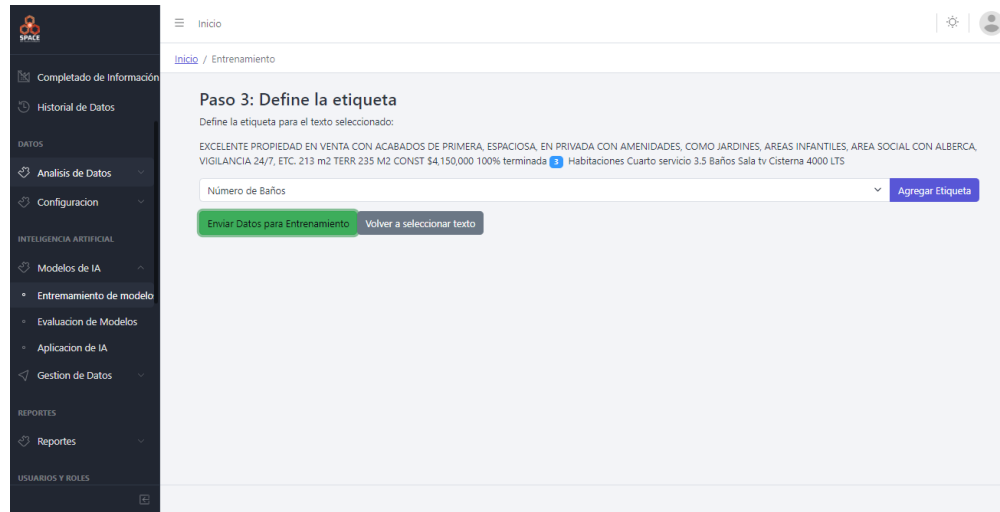
Interfaz de entrenamiento del modelo paso 2

The screenshot shows the SPARK application interface at Step 2. The sidebar is identical to the previous figure. The main content area has the title 'Paso 2: Selecciona el texto relevante' with the instruction 'Selecciona el texto relevante dentro del área de texto.'. Below this is a text area containing the sample text: 'EXCELENTE PROPIEDAD EN VENTA CON ACABADOS DE PRIMERA, ESPACIOSA, EN PRIVADA CON AMENIDADES, COMO JARDINES, AREAS INFANTILES, AREA SOCIAL CON ALBERCA, VIGILANCIA 24/7, ETC. 213 m2 TERR 235 M2 CONST \$4.150.000 100% terminada 3 Habitaciones Cuarto servicio 3.5 Baños Sala tv Cisterna 4000 LTS'. Below the text area is a dropdown menu labeled 'Selecciona una etiqueta'. At the bottom are two buttons: 'Agregar Texto Seleccionado' (blue) and 'Volver a escribir texto' (grey).

Luego pedirá que se resalte el texto importante y le asignemos una etiqueta

Figura 56.

Interfaz Entrenamiento del modelo paso 3



Te permitirá adicionar más etiquetas luego te llevará a enviar los datos al modelo para entrenarlo

3.2.7.2 Evaluación del modelo

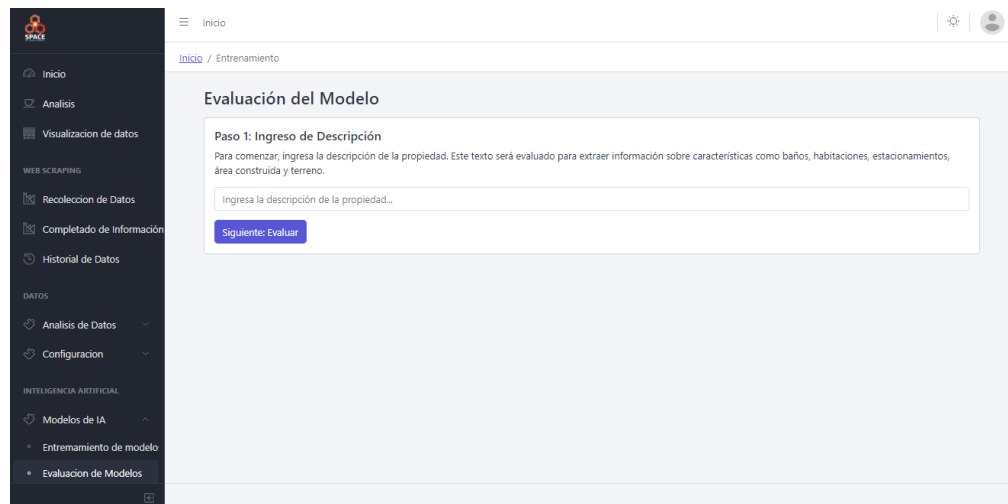
Al igual que la vista anterior solo tendrá acceso el administrador y el encargado del entrenamiento.

En esta sección se permitirá ver la eficiencia del modelo

Se va por pasos, primero ingresamos el texto que queremos evaluar

Figura 57.

Interfaz de evaluación del modelo paso 1



Luego seleccionamos que se evaluara el texto

Figura 58.

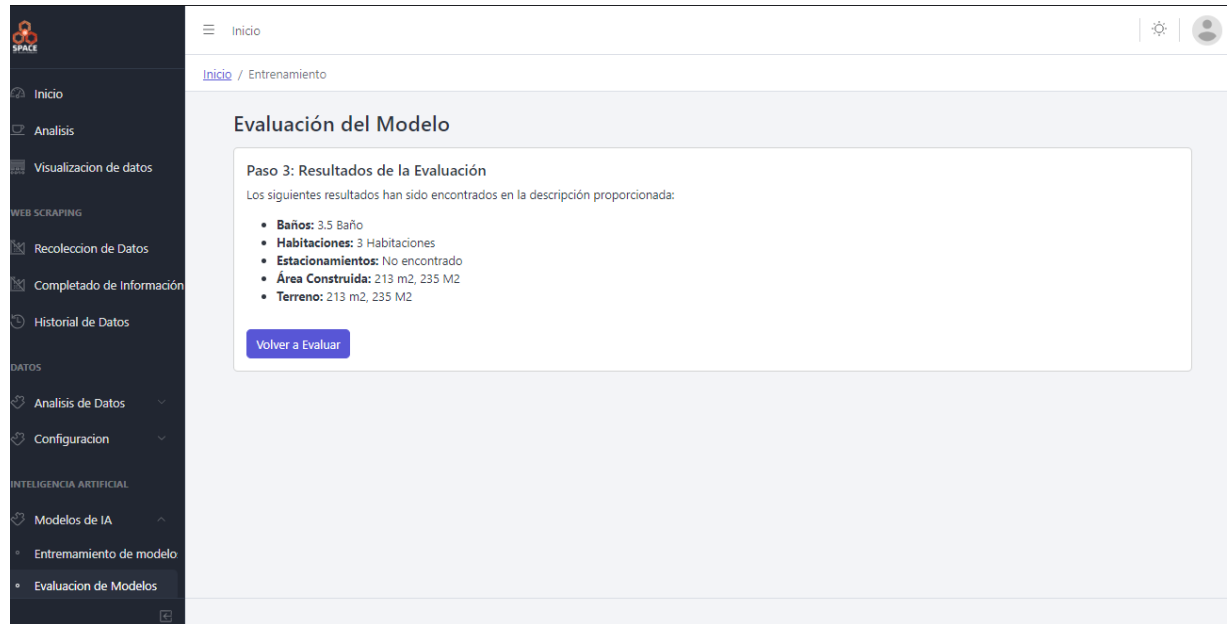
Interfaz de evaluación del modelo paso 2



Visualizamos los resultados

Figura 59.

Interfaz de evaluación del modelo paso 3



CAPITULO IV

CALIDAD, SEGURIDAD Y PRUEBAS

**INGENIERÍA
DE SISTEMAS**
UNIVERSIDAD PÚBLICA DE EL ALTO



CAPITULO IV

4. CALIDAD, SEGURIDAD Y PRUEBAS

En este capítulo, se detallan los aspectos de calidad, seguridad y pruebas implementados en el sistema de Web Scraping. Dado que el sistema está orientado a la recopilación y procesamiento de datos de propiedades inmobiliarias, asegurar la calidad de los datos y la estabilidad del sistema es fundamental para garantizar resultados precisos y confiables.

4.1 METRICAS DE CALIDAD

Se realizará las métricas de calidad de software mediante la norma ISO 25010 con las características que son funcionalidad, confiabilidad, usabilidad, eficiencia, fiabilidad, mantenibilidad y portabilidad del sistema

4.1.1 Norma ISO/IEC 25010

4.1.1.1 Funcionalidad

En la norma ISO/IEC 25010, la funcionalidad de un software puede evaluarse considerando su nivel de complejidad. Para medir esta característica en el presente proyecto, se emplea la métrica de Punto de Función, un método efectivo para cuantificar el tamaño y la funcionalidad del sistema. Este enfoque requiere determinar las siguientes características clave, permitiendo una evaluación precisa de la funcionalidad y el alcance del software.

Después de haber definido las características y aplicando el modelo se obtiene los siguientes datos:

Tabla 5.*Características de funcionalidad*

Características	Descripción
Entrada de Datos del Usuario (EI)	Área en la interfaz donde los usuarios introducen criterios como ubicación, rango de precios o tipo de propiedad para realizar búsquedas de propiedades específicas.
Resultados de Información (EO)	Generación de reportes detallados con los datos obtenidos de las propiedades, como su ubicación, características principales y un precio aproximado.
Filtros y Consultas (EQ)	Funcionalidad que permite a los usuarios acceder a propiedades específicas o aplicar filtros avanzados en los resultados de búsqueda para obtener coincidencias exactas.
Base de Datos de Propiedades (ILF)	Total, de propiedades capturadas y almacenadas en el sistema, incluyendo datos clave como precios, ubicación y superficie para análisis y visualización.
Conexiones Externas (EFL)	Integración del sistema con otras fuentes de información externas para completar datos faltantes y validar detalles de propiedades a través de servicios adicionales.

Al tener definido las características, contamos en el sistema cuantos módulos cumplen con las características y se obtiene los siguientes datos

Tabla 6.*Parámetros de medición*

Características	Descripción
Entrada de Datos del Usuario (EI)	2
Resultados de Información (EO)	4
Filtros y Consultas (EQ)	30
Base de Datos de Propiedades (ILF)	20
Conexiones Externas (EFL)	1

Para ejecutar los cálculos según los puntos de función se debe obtener los datos de la cuenta con los factores de ponderación que se especifican en la siguiente tabla.

Tabla 7.

Factores de ponderación

Variable	Simple	Medio	Complejo
Entrada externa (EI)	3	4	6
Salida externa (EO)	4	5	7
Consulta externa (EQ)	3	4	6
Archivo lógico interno (ILF)	7	10	15
Archivo lógico externo (EFL)	5	7	10

Al analizar el modelo, se decidió utilizar el factor de medición "Medio". Al aplicar este factor a los datos obtenidos del conteo realizado en los diferentes factores de medición, se obtienen los valores correspondientes al punto de función. Este cálculo permite estimar la complejidad y el tamaño del sistema de manera cuantitativa, lo que ayuda en la evaluación del esfuerzo y recursos necesarios para el desarrollo del software.

Tabla 8.*Cálculo de Factores de ponderación*

Variable	Cuenta	Factor	Total
Entrada externa (EI)	1	4	4
Salida externa (EO)	4	5	20
Consulta externa (EQ)	30	4	120
Archivo lógico interno (ILF)	30	10	300
Archivo lógico externo (EFL)	1	7	7
Total			451

Ahora que se ha obtenido la sumatoria total de los factores multiplicados por los factores de ponderación, se procede a determinar los valores de complejidad. Para ello, se asignan valores específicos de complejidad a cada uno de los factores, lo que permite evaluar de manera más precisa el nivel de dificultad del sistema. Estos valores son esenciales para estimar el esfuerzo necesario para el desarrollo, ajustando la carga de trabajo y los recursos según el grado de complejidad identificado en cada factor.

Con los valores obtenidos, procederemos a calcular los puntos de función mediante la siguiente fórmula:

Ecuación 2. Funcionalidad

$$PF = CuentaTotal * 0.65 + 0.01 * \sum F_i)$$

Donde:

- **CuentaTotal** es el resultado de la sumatoria de los productos entre los factores de ponderación y los valores de los parámetros.
- $\sum F_i$ es la sumatoria de los valores de ajustes de complejidad.

Aplicando los valores proporcionados:

$$PF = 451 * 0.65 + 0.01 * 48)$$

$$PF = 451 * 1.13$$

$$PF = 509.63$$

Considerando el ajuste máximo de complejidad, donde $\sum F_i = 48$, calculamos el nivel de confianza al 100% de la siguiente forma:

$$PF_{max} = CuentaPF_{max} = CuentaTotal * (0,65 + 0,01 * \sum F_i)$$

$$PF_{max} = 509 * (0,65 + 0,01 * 48)$$

$$PF_{max} = 509 * 1.13$$

$$PF_{max} = 575.17$$

Para obtener el porcentaje de funcionalidad, relacionamos los resultados obtenidos de la siguiente manera:

$$Funcionalidad = \frac{PF}{PF_{max}} * 100$$

$$Funcionalidad = \frac{509.63}{575.17} * 100$$

$$Funcionalidad = 0,8860 * 100$$

$$Funcionalidad = 88,6\%$$

Este cálculo indica que la funcionalidad del sistema, según los puntos de función estimados, alcanza un 88.6%.

4.1.1.2 Confiabilidad

La confiabilidad del sistema de información es la probabilidad de operación libre de fallos de un programa o computadora.

Tenemos en cuenta que:

Ecuación 3. Confiabilidad

$$P(T \leq t) = F(t) \Rightarrow \text{probabilidad de fallas}$$

$$P(T \leq t) = 1 - F(t) \Rightarrow \text{probabilidad de trabajo sin fallas}$$

A continuación. Obtenemos la confiabilidad del sistema con la siguiente información.

$$(t) = f * e^{-\frac{u}{10} * t}$$

Donde:

- t = Tiempo de prueba del Sistema
- u = Probabilidad de error del sistema
- f = Funcionalidad del sistema

Se considera un periodo de 30 días como tiempo de prueba donde se define que cada 20 ejecuciones se presenta 1 falla.

Aplicamos en la formula

$$F(t) = 0.886 * e^{-\frac{2}{20} * 30}$$

$$F(t) = 0.886 * 0.05$$

$$F(t) = 0.0443$$

$$F(t) = 4.43\%$$

Remplazamos en la ecuación

$$P(T \leq t) = 1 - F(t)$$

$$P(T \leq t) = 1 - 0.0443$$

$$P(T \leq t) = 0.9557$$

$$P(T \leq t) = 95.57\%$$

Este resultado concluye que la confiabilidad del sistema durante un período de prueba de 30 días es del 95.57%, lo que es un valor considerablemente alto y refleja una muy buena estabilidad y fiabilidad en su desempeño bajo las condiciones especificadas.

4.1.1.3 Usabilidad

La usabilidad mide cuan eficiente los usuarios puede alcanzar sus objetivos de manera efectiva y satisfactoria

Para calcular la usabilidad se aplicará la siguiente formula

Ecuación 4. Usabilidad

$$FU = \left[\frac{\sum x_i}{n} * 100 \right]$$

Donde

X_i = Valores de la sumatoria

n = El número de preguntas

Tabla 10.*Escala de valores de preguntas*

Escala	Valor
Muy bueno	5
Bueno	4
Regular	3
Malo	2

Tabla 11.*Preguntas para determinar la usabilidad*

Nro.	PREGUNTAS	SI	NO	Xi
1	¿Le resulta fácil utilizar el sistema?	5	0	1
2	¿Puede realizar las tareas deseadas de manera rápida y eficiente?	5	0	1
3	¿Considera que las respuestas del sistema son complicadas?	4	1	0.8
4	¿Te resulta agradable la interfaz visual del sistema?	4	1	0.8
5	¿El sistema responde de manera satisfactoria y rápida a tus acciones, sin demora?	4	1	0.8
6	¿Le resulta complicadas las funciones del sistema, o son fáciles de comprender y utilizar?	5	0	1
7	¿Los resultados que obtienes del sistema son precisos y confiables?	5	0	1
8	¿Has experimentado algún error o problema mientras utilizabas el sistema durante tu interacción?	4	1	0.8
9	¿Consideras que los resultados que obtienes del sistema hacen más fácil tu trabajo?	5	0	1
10	¿El uso del sistema reduce el tiempo de trabajo?	5	0	1
Total				9.2

Calculamos la usabilidad

$$FU = \left[\frac{9.2}{10} * 100 \right]$$

$$FU = [0.92 * 100]$$

$$FU = 92\%$$

Afirmamos que el sistema tiene una usabilidad satisfactoria con un índice de 92%

4.1.1.4 Eficiencia

La eficiencia mide el rendimiento y la capacidad del sistema para procesar transacciones de forma rápida para ello la ISO/IEC 25010 evalúa mediante el tiempo de respuesta promedio y la utilización de recursos computacionales

El tiempo de respuesta promedio, es el tiempo que tarda el sistema en procesar una transacción desde que se envía hasta que se completa.

Para ello aplicaremos las siguiente formula:

Ecuación 5. Eficiencia

$$\text{Tiempo de Respuesta Promedio} = \frac{\text{Suma de Tiempos de Respuesta}}{\text{Numero total de transacciones}}$$

$$\text{Tiempo de Respuesta Promedio} = \frac{2500 \text{ ms}}{50}$$

$$\text{Tiempo de Respuesta Promedio} = 50\text{ms}$$

Fórmula para la eficiencia en el rendimiento

$$Eficiencia\ en\ el\ Rendimiento = \frac{Tiempo\ de\ respuesta\ Objetivo}{tiempo\ de\ respuesta\ promedio} * 100$$

Tiempo de respuesta objetivo: 50ms

Tiempo de respuesta promedio Calculado: 50ms

$$Eficiencia\ en\ el\ Rendimiento = \frac{50}{50} * 100$$

$$Eficiencia\ en\ el\ Rendimiento = 100\%$$

Esto indica que, con un tiempo promedio de 50 ms, el sistema procesa cada transacción en aproximadamente 0.05 segundos. Este rendimiento es rápido y adecuado para asegurar una experiencia de usuario fluida y eficiente.

4.1.1.5 Fiabilidad

La fiabilidad del sistema refleja su capacidad para funcionar sin interrupciones y recuperarse de fallos, permitiendo que los usuarios accedan a la información de manera continua y segura. La disponibilidad del sistema es la métrica que evalúa este aspecto, representando el porcentaje de tiempo en el que el sistema se mantiene operativo y accesible.

Disponibilidad

La disponibilidad mide el tiempo en que el sistema está en funcionamiento sin interrupciones. Mantener una alta disponibilidad es esencial para proporcionar acceso confiable y constante a los datos. Para calcular la disponibilidad, se utiliza la siguiente fórmula:

Ecuación 6. Disponibilidad

$$Disponibilidad = \frac{TMEF}{TMEF + TMDR} * 100$$

Donde:

- TMEF: Tiempo Medio entre Fallos, que representa la duración promedio entre fallos del sistema.
- TMDR: Tiempo Medio de Reparación, que mide el tiempo promedio necesario para restaurar el sistema después de un fallo.

Durante la fase de evaluación, con los datos entrenados el sistema de Web Scraping registro un total de 1 fallos en un periodo de 300 horas de uso continuo (casi 3 semanas). El tiempo total requerido para reparar las fallas es de 3 horas.

Cálculo del TMEF

$$TMEF = \frac{Tiempo\ Total\ de\ Operacion}{Numero\ de\ Fallos}$$

$$TMEF = \frac{300}{1}$$

$$TMEF = 300$$

Cálculo del TMDR:

$$TMDR = \frac{Tiempo\ Total\ para\ Reparar\ Fallos}{Numero\ de\ Fallos}$$

$$TMDR = \frac{3}{1}$$

$$TMDR = 3$$

Cálculo de la Disponibilidad:

$$Disponibilidad = \frac{300}{300 + 3} * 100$$

$$Disponibilidad = 99.01$$

Este resultado muestra que el sistema alcanza un nivel de disponibilidad del 99.01%, lo que refleja una fiabilidad elevada. Esto implica que el sistema de Web Scraping permanece operativo y accesible para los usuarios casi en todo momento, garantizando así un acceso constante y confiable.

4.1.1.6 Portabilidad

Para el análisis de la portabilidad en el sistema web, se considera su capacidad de adaptación y funcionalidad en diversos entornos de despliegue, tanto en el servidor como en el cliente. Los aspectos clave incluyen:

Adaptabilidad: Evaluamos la capacidad del sistema para ajustarse de manera eficiente y eficaz a distintos entornos operacionales, hardware y plataformas de software. Este aspecto es esencial para asegurar que el sistema pueda ser implementado en una variedad de configuraciones sin requerir modificaciones sustanciales.

Capacidad de instalación: Analizamos la facilidad con la que el sistema puede ser instalado y desinstalado en cualquier entorno definido, asegurando que el proceso de despliegue sea directo y permita la configuración rápida en diferentes entornos.

Capacidad de Sustitución: Consideramos la capacidad del sistema para ser reemplazado por otro con funciones similares en el mismo entorno, garantizando así la posibilidad de cambios sin afectar la funcionalidad o el desempeño general del sistema.

Para ello aplicaremos la siguiente formula

Ecuación 7. Portabilidad

$$P = 1 - \frac{ET}{EI}$$

Donde:

ET = Esfuerzo para trasladar el sistema

EI = Esfuerzo para implementar el sistema

P = Portabilidad

Siendo que se está aplicando Docker para el traslado de dependerá de un total de 1 hora y el esfuerzo para implementar el sistema considerando instalaciones de dependencias se da un total de 8 horas para implementar desde cero

Remplazamos en los datos

$$P = 1 - \frac{1}{8}$$

$$P = 1 - 0.13$$

$$P = 0.87$$

$$Portabilidad = 87\%$$

La portabilidad del sistema es de un 87%, al momento de realizar una migración a otro servidor.

4.1.2 Resultados ISO/IEC 25010

En conclusión, se representa los valores obtenidos en la siguiente tabla.

Tabla 12.

Resultados ISO/IEC 25010

Características	Resultados
Funcionalidad	88.6%
Confiabilidad	95.57%
Usabilidad	92%
Eficiencia	100%
Fiabilidad	99.01%
Portabilidad	87%
TOTAL	93.67%

4.2 ESTIMACION DEL COSTO

El proyecto de desarrollo del sistema web scraping de reconocimiento comercial en bienes raíces tiene como objetivo automatizar la recolección y análisis de información sobre propiedades inmobiliarias desde diversas fuentes en línea. Utilizando inteligencia artificial (IA), el sistema podrá recopilar datos faltantes de las propiedades como ser el precio, la ubicación, la cantidad de habitaciones o el tamaño de las propiedades.

Este documento utiliza el estándar COSMIC (Common Software Measurement International Consortium) para estimar el tamaño, esfuerzo y costo de desarrollo del software. COSMIC se enfoca en la cantidad de datos que el software manipula, permitiendo una estimación más precisa y adaptable a proyectos de software como este.

Con la integración de IA, las entidades de datos clave se amplían, ya que se deben tener en cuenta tanto las entradas externas (scraping de datos) como las acciones de la IA para completar los datos faltantes.

Para obtener las métricas de estimación de costos utilizando el método COSMIC, se llevaron a cabo tres etapas que incluyeron lo siguiente:

- **Etapas: Planificación de la medición**
- **Etapas: Asignación**
- **Etapas: Medición**

4.2.1 *Planificación de la medición*

Para este proyecto, se determinó que la medición se basará en los módulos del sistema, con el objetivo de estimar los recursos necesarios para su desarrollo y mantenimiento.

El propósito principal de la medición es obtener una evaluación precisa del tamaño y la complejidad del sistema, lo que ayudará a establecer un presupuesto adecuado y gestionar los tiempos de desarrollo.

Módulos funcionales del proyecto:

Se describirá de manera general los módulos y a que usuarios afecta para mejor entendimiento.

- **Módulo Recolección de Datos:** El técnico podrá listar, modificar, eliminar y registrar sitios web del cual se desea extraer información además podrá inicial la recolección de información configurando parámetros de recolección.
- **Módulo Completado de Información:** El técnico podrá realizar el completado de información faltante de las fuentes registradas con parámetros incompletos, tendrá la visualización de datos con una descripción de los datos que se están almacenando
- **Modulo Historial de Datos:** El técnico podrá ver los historiales de extracción y con que parámetros se realizó el Web Scraping. El Administrador podrá eliminar algún registro que se esté queriendo volver a extraer por motivos de confusión u otros.
- **Módulo Configuración:** El técnico podrá realizar las configuraciones de parámetros del scraping como ser las clases de donde extraer la información relevante, además de poder configurar direcciones ip de proxies para mejorar la eficiencia en la extracción de datos.

- **Módulo Modelos de IA:** El encargado de IA podrá entrenar al modelo registrado con los datos e información relevante además de evaluar el modelo y mejorar la fiabilidad en la extracción de datos, con esto podrá aplicar el modelo para completar información faltante en las propiedades registradas
- **Módulo Gestión de Datos:** El administrador podrá ver la estructura de la base de datos además de poder realizar el BACKUP correspondiente para asegurar la información registrada, El técnico podrá realizar la limpieza de datos manual para poder asegurar la fiabilidad en los datos
- **Módulo Reportes:** El técnico y administrador podrá generar repostes según la fecha y pagina web registrada además de tener un histórico de reportes realizados para tener mejor comprensión en los repostes realizados
- **Módulo Usuarios y Roles:** El administrador podrá crear listar editar y eliminar usuarios además de poder asignar roles específicos a cada usuario dentro del sistema

4.2.2 Asignación y medición

Control de usuarios

Movimiento de datos

Entrada: Ingresa los datos correo y contraseña

Lectura: validación de los datos ingresados

Lectura: obtener los datos de usuarios registrados

Salida: Mostrar lista de usuario registrados

Entrada: modificar datos de los usuarios

Puntos de función COSMIC: 5 CFP

Control de usuarios

Movimiento de datos

Entrada: Ingresa los datos correo y contraseña

Lectura: validación de los datos ingresados

Lectura: obtener los datos de usuarios registrados

Salida: Mostrar lista de usuario registrados

Entrada: modificar datos de los usuarios y contraseñas

Puntos de función COSMIC: 5 CFP

Control de roles

Movimiento de datos

Entrada: selecciona asignación de roles

Entrada: ingresa datos de rol

Salida: se visualiza la creación del nuevo rol

Entrada: selecciona un usuario

Entrada: asignamos los roles determinados

Lectura obtenemos una lista de los roles

Salida: veremos el usuario con el nuevo rol

Puntos de función COSMIC: 7 CFP

Recolección de datos

Movimiento de datos

Entrada: selecciona Iniciar Web Scraping

Entrada: ingresa los parámetros para el Web Scraping

Salida: se visualiza los datos de las páginas web extraídas

Lectura: obtenemos los las URL ingresadas

Entrada: selecciona Completado de Información

Entrada: selecciona Completar información

Salida: visualización de páginas ingresadas

Lectura: registro de datos registrados y completados

Entrada: ingreso de la url para extraer

Salida: datos registrados en la base de datos

Entrada: descripción de la información

Salida: visualización de las páginas registradas

Lectura: lista de páginas registradas y datos encontrados

Puntos de función COSMIC: 13 CFP

Control de páginas web

Movimiento de datos

Entrada: selecciona Agregar Página Web

Entrada: ingresa datos correspondientes para página web

Salida: registro de la página web

Lectura: lista de las páginas web y atributos creados

Entrada: selecciona Ver Categorías

Lectura: lista de las categorías y subcategorías

Entrada: selecciona editar página web

Entrada: selecciona eliminar página web

Entrada: selecciona historial de datos

Salida le visualiza la lista del historial de extracciones

Entrada: selecciona eliminar historial

Lectura: se visualiza la eliminación

Puntos de función COSMIC: 12 CFP

Entrenamiento de la IA

Movimiento de datos

Entrada: selecciona modelos de IA

Entrada: selecciona Entrenamiento del modelo

Entrada: se llena los datos de entrenamiento

Entrada: se selecciona las palabras clave para el entrenamiento

Lectura: se visualiza los datos registrados

Salida: vista de entrenamiento logrado

Entrada: evaluación de modelos

Entrada: se ingresa texto con la información de búsqueda

Salida: se ve los datos encontrados

Lectura: texto con datos relevantes

Entrada: selecciona Aplicación de IA

Lectura: lista de datos registrados

Salida: registros actualizados y completados con la información obtenida

Puntos de función COSMIC: 13 CFP

Configuración de Scraping

Movimiento de datos

Entrada: selecciona configuración

Entrada: selecciona parámetros de scraping

Entrada: ingresa datos de ubicación de datos en clases

Lectura: parámetros de búsqueda de clases

Salida: aceptar cambiar la estructura

Entrada: configuración de proxies

Lectura: registro y validación de datos

Salida: lista de proxies almacenados

Puntos de función COSMIC: 5 CFP

Control de reportes

Movimiento de datos

Entrada: selecciona reportes

Entrada: selecciona generar informe

Entrada: ingresa fecha y fuente

Lectura: validación de fecha y fuente

Salida: informe con los parámetros mencionados

Entrada: selecciona historial de reportes

Entrada: selecciona descarga de historial de reportes

Salida: lista de reportes generados

Puntos de función COSMIC: 8 CFP

De esta forma se determina que el proyecto tiene una medición de:

68 puntos de función COSMIC: 68 CFP

Para estimar el costo del desarrollo de una unidad de medida para el tamaño del software, es necesario considerar información proveniente de diversas fuentes adicionales. Esto incluye factores como la experiencia del desarrollador, la complejidad del proyecto y los recursos requeridos.

Costo Estimado

El sueldo de un desarrollador varía según su nivel de experiencia. En este caso, se ha establecido un valor de 2,400 Bs como el sueldo mensual mínimo de un desarrollador web, según la oferta actual en el mercado laboral.

Tras realizar los cálculos correspondientes para implementar el sistema, se ha elaborado una tabla con el costo total estimado del proyecto.

Costo por Unidad de Medida

Para determinar el costo de desarrollar cada punto de función, se emplea la siguiente fórmula:

Ecuación 8. Costos por punto de función

$$\text{Costos por punto de función} = \frac{\text{costos por mes del equipo de trabajo}}{\text{punto de función mes}}$$

Donde se vio que al mes se cumple un total de 15 puntos de función y basándonos al costo mensual de un equipo de trabajo es de 4000 remplazamos los datos en la formula

$$\text{Costos por punto de funcion} = \frac{4000 \text{ bs}}{15 \text{ puntos de funcion}} = 267 \text{ Bs/CPF}$$

Esta fórmula tiene como base la relación entre el costo del desarrollador y el tamaño estimado del software, lo cual permite calcular el costo de cada unidad de medida de manera precisa.

Estimación de costos del proyecto de software

Para poder determinar el costo del proyecto de software se aplicará la siguiente formula:

$$\text{costos del proyecto de Software} = \text{tamaño del software} \times \text{costo por punto de funcion}$$

El tamaño del software se va representando por el total de todos los puntos de función obtenidos.

Remplazamos en la formula

$$\text{costos del proyecto de Software} = 68 \text{ CFP} \times 267 \text{ Bs/CFP}$$

$$\text{costos del proyecto de Software} = 18,156 \text{ Bs}$$

Según los cálculos se determinó una estimación de costo del sistema desarrollado de 18,156 bolivianos

Tiempo de duración del desarrollo de software

Una vez teniendo los puntos de función se puede aplicar para determinar el tiempo de duración del desarrollo del sistema, aplicaremos la siguiente formula

Ecuación 9. Duración del desarrollo del software

$$Duración\ del\ desarrollo\ del\ software = \frac{puntos\ de\ funcion}{puntos\ de\ funcion\ mes}$$

Remplazamos los datos

$$Duración\ del\ desarrollo\ del\ software = \left\lceil \frac{68}{15} \right\rceil \approx 5\ meses$$

Conclusión

De eta forma se determina el desarrollo y costo del proyecto del software:

Durará 5 meses en desarrollarse

Costará 18156 Bs (bolivianos)

4.3 SEGURIDAD

La norma ISO/IEC 27001 es un estándar internacional diseñado para la gestión de la seguridad de la información, garantizando la confidencialidad, integridad y disponibilidad de los datos dentro de una organización. Este estándar proporciona un marco sistemático para

identificar y evaluar riesgos, y aplicar controles específicos que ayuden a mitigar estos riesgos de forma efectiva.

El cumplimiento de ISO/IEC 27001 refleja el compromiso de la organización en mejorar continuamente la seguridad de la información. Implementar esta norma contribuye a reducir la probabilidad de incidentes de seguridad y asegura la protección de los activos de información, estableciendo una sólida base para proteger la organización y generar confianza entre sus partes interesadas.

4.3.1 Seguridad Lógica

El sistema cuenta con los siguientes parámetros de seguridad.

4.3.1.1 Control de Acceso

Al ser una aplicación web tiene accesos restringidos para los usuarios asignándole a cada persona un usuario y contraseña para poder acceder al sistema

El nombre del usuario es el correo corporativo que se le asigna al momento de comenzar con sus actividades en la empresa.

La contraseña del personal se la asigna por defecto, dentro del sistema le permitirá cambiar la contraseña si el usuario lo ve necesario.

En caso de que el usuario se olvidara la contraseña el administrador tendrá la opción de restablecer la contraseña a la de por defecto.

4.3.1.2 Desactivación de Cuenta

El administrador tendrá la opción de desactivar a cuenta del usuario cuando ya no vea necesario sus servicios o la conclusión de su contrato, esta se podrá volver a activar en caso de que vuelva a desempeñar a misma tarea asignada.

4.3.1.3 Roles y Permisos

Los usuarios registrados tendrán roles específicos dentro del sistema para sus tareas asignadas.

4.3.1.4 Control de Asignación de Roles

En administrador otorgara roles específicos a cada usuario para que comience con sus actividades dentro del sistema.

4.3.1.5 Protección de rutas

Al tener middewares en cada petición al backend se asegura la confidencialidad de la información protegiendo información relevante.

4.3.1.6 Control por JWT

Con el control de JWT (Json Web Tockens) aseguramos la comunicación entre el usuario y el servidor dando un tiempo límite y asegurarnos de que es una petición valida

4.3.1.7 Cifrado de contraseñas

Al tener la contraseña de acceso de los usuarios cifrada con HASH aseguramos que esta sea difícilmente descifrada asegurando la integridad de los datos del usuario

4.3.2 Seguridad Física

4.3.2.1 Back-Ups de la base de datos

Se tendrá la opción de realizar un back up de la base de datos por motivos de preservación de la información y además de poder evitar la pérdida de información

4.3.2.2 Firewall

El servidor está configurado para protegerse de intrusos y además asegurar la protección de los servicios desplegados e instalados

4.4 PRUEBA AL SOFTWARE

4.4.1 Pruebas unitarias

Para las pruebas unitarias usaremos la herramienta unittest en Python para determinar las funciones de las APIs y validando los distintos escenarios que puedan ocurrir.

Figura 60.
Unittest para la autenticación

```

1 import unittest
2 from app import create_app
3 import json
4
5 class APITestLogin(unittest.TestCase):
6     def setUp(self):
7         self.app, self.socketio = create_app() # Ahora ambas instancias son recibidas
8         self.client = self.app.test_client() # Obtiene el cliente de prueba
9         self.app.config['TESTING'] = True
10
11     # INGRESO DE USUARIO VERIFICADO
12     def test_login_endpoint(self):
13         login_data = {
14             "email": "juansilva@bispase.tech",
15             "password": "12345678"
16         }
17         response = self.client.post('/api/login/', data=json.dumps(login_data), content_type='application/json')
18         self.assertEqual(response.status_code, 200)
19         print(response)
20         self.assertIn('token', response.json) # Asegúrate de que el token esté presente
21
22     # INGRESO DE USUARIO NO REGISTRADO
23     def test_login_user_incorrect_endpoint(self):
24         login_data = {
25             "email": "usuario1@bispase.tech",
26             "password": "usuario1"
27         }
28         response = self.client.post('/api/login/', data=json.dumps(login_data), content_type='application/json')
29         self.assertEqual(response.status_code, 404)
30         print(response)
31         self.assertEqual(response.json['message'], 'Correo no registrado')
32
33     # INGRESO DE USUARIO CON CONTRASEÑA INCORRECTA
34     def test_login_password_incorrect_endpoint(self):
35         login_data = {
36             "email": "juansilva@bispase.tech",
37             "password": "INVALID_PASSWORD"
38         }
39         response = self.client.post('/api/login/', data=json.dumps(login_data), content_type='application/json')
40         self.assertEqual(response.status_code, 403)
41         print(response)
42         self.assertEqual(response.json['message'], 'Contraseña incorrecta')
43
44 if __name__ == '__main__':
45     unittest.main()
46

```

Figura 61.
Resultados de las pruebas de autenticación

```

PS C:\Users\juans\OneDrive\Escritorio\II-2024\taller 2 proyecto de grado\Proyecto\webscraping\webscraping-backend> python
niit_test.test_app
Ingreso al sistema
{'email': 'juansilva@bispase.tech', 'password': '12345678'}
validamos la contraseña
<app.models.public_model.Usuarios object at 0x00000287BFC43880>
1 juansilva@bispase.tech ADMINISTRADOR
<WrapperTestResponse streamed [200 OK]>
Ingreso al sistema
{'email': 'juansilva@bispase.tech', 'password': 'INVALID_PASSWORD'}
validamos la contraseña
<WrapperTestResponse streamed [403 FORBIDDEN]>
Ingreso al sistema
{'email': 'usuario1@bispase.tech', 'password': 'usuario1'}
<WrapperTestResponse streamed [404 NOT FOUND]>
.
-----
Ran 3 tests in 10.077s

OK
PS C:\Users\juans\OneDrive\Escritorio\II-2024\taller 2 proyecto de grado\Proyecto\webscraping\webscraping-backend>

```

Figura 62.*Unittest para la verificación de los JWT*

```

1 import unittest
2 from app import create_app
3 import json
4
5 class APITestJWT(unittest.TestCase):
6     def setUp(self):
7         self.app, self.socketio = create_app() # Ahora ambas instancias son recibidas
8         self.client = self.app.test_client() # Obtiene el cliente de prueba
9         self.app.config['TESTING'] = True
10
11     # PETICION CON UN JWT VALIDO
12     def test_jwt_valido(self):
13         token = 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIIOjE5ImVtYWIzIjoianVhbnNpbHhZbGQ3PC3BhY2UudGVjaCIsInVjbGUiOiJBRRE1JTklTVFJBRE9SIiwudmtyVBLzUUpKxQjQjQhbnDY5sW4wvTMD2U_ep-1m-kDg'
14         dominio_data = {
15             "nombre": "NOMBRE DEL DOMINIO",
16             "paginacion": "PAGINACION",
17             "logo": "URL DEL LOGO",
18             "url_base": "URL BASE",
19             "tipo_paginacion": "TIPO DE PAGINACION",
20         }
21         response = self.client.post('/api/dominio/', data=json.dumps(dominio_data),
22                                     content_type='application/json',
23                                     headers={'Authorization': f'Bearer {token}'})
24         self.assertEqual(response.status_code, 200)
25         print(response)
26         self.assertEqual(response.json['message'], 'Dominio creado correctamente')
27
28     # PETICION CON UN JWT NO VALIDO
29     def test_jwt_invalido(self):
30         token = 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIIOjE5ImVtYWIzIjoianVhbnNpbHhZbGQ3PC3BhY2UudGVjaCIsInVjbGUiOiJBRRE1JTklTVFJBRE9SIiwudmtyVBLzUUpKxQjQjQhbnDY5sW4wvTMD2U_ep-1m-kDg'
31         dominio_data = {
32             "nombre": "NOMBRE DEL DOMINIO",
33             "paginacion": "PAGINACION",
34             "logo": "URL DEL LOGO",
35             "url_base": "URL BASE",
36             "tipo_paginacion": "TIPO DE PAGINACION",
37         }
38         response = self.client.post('/api/dominio/', data=json.dumps(dominio_data),
39                                     content_type='application/json',
40                                     headers={'Authorization': f'Bearer {token}'})
41         self.assertEqual(response.status_code, 401)
42         print(response)
43         self.assertEqual(response.json['message'], 'Invalid token!')
44
45     # PETICION SIN UN JWT
46     def test_jwt_undefined(self):
47         token = ''
48         dominio_data = {
49             "nombre": "NOMBRE DEL DOMINIO",
50             "paginacion": "PAGINACION",
51             "logo": "URL DEL LOGO",
52             "url_base": "URL BASE",
53             "tipo_paginacion": "TIPO DE PAGINACION",
54         }
55         response = self.client.post('/api/dominio/', data=json.dumps(dominio_data),
56                                     content_type='application/json',
57                                     headers={'Authorization': f'Bearer {token}'})
58         self.assertEqual(response.status_code, 401)
59         print(response)
60         self.assertEqual(response.json['message'], 'Token is missing!')
61
62 if __name__ == '__main__':
63     unittest.main()
64

```

Figura 63.*Resultados de las pruebas para validar el JWT*

```

PS C:\Users\juans\OneDrive\Escritorio\II-2024\taller 2 proyecto de grado\Proyecto\webscraping\webscraping-backend> python -m unittest test.niit_test.json_test
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIIOjE5ImVtYWIzIjoianVhbnNpbHhZbGQ3PC3BhY2UudGVjaCIsInVjbGUiOiJBRRE1JTklTVFJBRE9SIiwudmtyVBLzUUpKxQjQhbnDY5sW4wvTMD2U_ep-1m-kDg
invalid
<WrapperTestResponse streamed [401 UNAUTHORIZED]>
.
<WrapperTestResponse streamed [401 UNAUTHORIZED]>
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIIOjE5ImVtYWIzIjoianVhbnNpbHhZbGQ3PC3BhY2UudGVjaCIsInVjbGUiOiJBRRE1JTklTVFJBRE9SIiwudmtyVBLzUUpKxQjQhbnDY5sW4wvTMD2U_ep-1m-kDg
{'sub': 1, 'email': 'juansilva@bispacedev.tech', 'role': 'ADMINISTRADOR'}
<WrapperTestResponse streamed [200 OK]>
.
-----
Ran 3 tests in 8.279s

OK
PS C:\Users\juans\OneDrive\Escritorio\II-2024\taller 2 proyecto de grado\Proyecto\webscraping\webscraping-backend>

```

Figura 64.
Pruebas para realizar Web Scraping

```

1 import unittest
2 from app import create_app
3 import json
4
5 class APITestWebScraping(unittest.TestCase):
6     def setUp(self):
7         self.app, self.socketio = create_app() # Ahora ambas instancias son recibidas
8         self.client = self.app.test_client() # Obtiene el cliente de prueba
9         self.app.config['TESTING'] = True
10
11     # PETICION PARA REALIZAR WEB SCRAPING
12     def test_web_scraping_valido(self):
13         token = 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJlbnVtYmVlIjoianVhbnNpbHZhZGQ3PC3BhY2UudG9jaCIsIn3vbmG1O1JBRE1JTk1TVF3BRE9SiIn0.mdyVBLzUUpKxQjqlhndY5sVW4wVtMD2U_ep-1m-KDg'
14         dominio_data = {
15             "dominio": "https://inmuebles.mercadolibre.com.mx",
16             "categoria": "venta",
17             "subcategoria": "otros",
18             "limite": '2',
19             "fecha": '2024-11-01',
20         }
21         response = self.client.post('/api/web_scraping/create', data=json.dumps(dominio_data),
22                                     content_type='application/json',
23                                     headers={"Authorization": f"Bearer {token}"})
24         self.assertEqual(response.status_code, 200)
25         print(response)
26         self.assertEqual(response.json['message'], 'Extracción completada')
27
28     # PETICION PARA REALIZAR WEB SCRAPING CON URL NO REGISTRADAS
29     def test_web_scraping_invalido(self):
30         token = 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJlbnVtYmVlIjoianVhbnNpbHZhZGQ3PC3BhY2UudG9jaCIsIn3vbmG1O1JBRE1JTk1TVF3BRE9SiIn0.mdyVBLzUUpKxQjqlhndY5sVW4wVtMD2U_ep-1m-KDg'
31         dominio_data = {
32             "dominio": "URL DEL DOMINIO",
33             "categoria": "venta",
34             "subcategoria": "otros",
35             "limite": '22',
36             "fecha": '2024-11-01',
37         }
38         response = self.client.post('/api/web_scraping/create', data=json.dumps(dominio_data),
39                                     content_type='application/json',
40                                     headers={"Authorization": f"Bearer {token}"})
41         self.assertEqual(response.status_code, 404)
42         print(response)
43         self.assertEqual(response.json['message'], 'Dominio no encontrado')
44
45     # PETICION PARA REALIZAR CON DATOS INCOMPLETOS
46     def test_WEB_SCRAPING_INCOMPLETO(self):
47         token = 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJlbnVtYmVlIjoianVhbnNpbHZhZGQ3PC3BhY2UudG9jaCIsIn3vbmG1O1JBRE1JTk1TVF3BRE9SiIn0.mdyVBLzUUpKxQjqlhndY5sVW4wVtMD2U_ep-1m-KDg'
48         dominio_data = {
49             "dominio": "URL DEL DOMINIO",
50             "categoria": "venta",
51             "subcategoria": "otros",
52             "limite": '22',
53         }
54         response = self.client.post('/api/web_scraping/create', data=json.dumps(dominio_data),
55                                     content_type='application/json',
56                                     headers={"Authorization": f"Bearer {token}"})
57         self.assertEqual(response.status_code, 410)
58
59 if __name__ == '__main__':
60     unittest.main()
61

```

Figura 65.
Resultados del test de Web Scraping

```

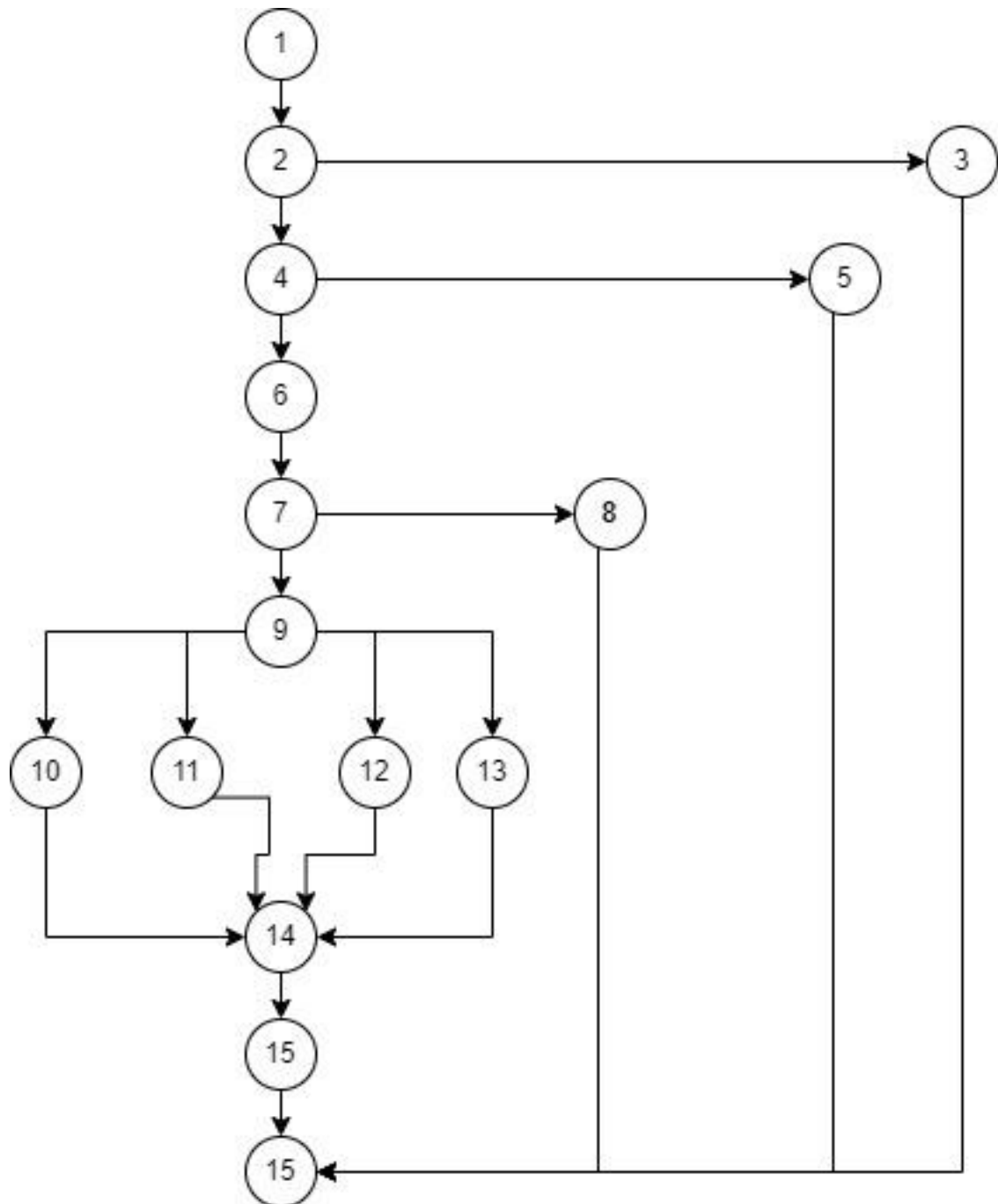
PS C:\Users\juans\OneDrive\Escritorio\II-2024\taller 2 proyecto de grado\Proyecto\webscraping\webscraping-backend> py
nit_test.web_scraping test
Esta es la primera vuelta
.Esta es la primera vuelta
<WrapperTestResponse streamed [404 NOT FOUND]>
.<WrapperTestResponse streamed [404 NOT FOUND]>
.
-----
Ran 3 tests in 9.078s

OK
PS C:\Users\juans\OneDrive\Escritorio\II-2024\taller 2 proyecto de grado\Proyecto\webscraping\webscraping-backend>

```

Figura 66.

Camino de las funciones para realizar Web Scraping



Nota: camino de ejecución para realizar Web Scraping.

Donde

Tabla 13.

Pasos para ejecutar Web Scraping

Numero	Función
1	Primera vuelta general
2	Validación de campos
3	Error en el envío de datos
4	Verificación de nueva extracción
5	Error extracción ya realizada
6	Registro de la extracción
7	Verificación de Dominio existente
8	Error El dominio no existe
9	Selección de formato de URL Params
10	Selección de formato de URL Desde
11	Selección de formato de URL query
12	Selección de formato de URL paginado
13	Extracción de datos
14	Guardar datos en la base de datos
15	Respuesta con mensaje al frontend

Analizamos el grafo generado a parte de las características del sistema, ahora se procede a determinar la complejidad ciclomática del grafo

Ecuación 10. Complejidad Ciclomática

$$V(G) = (A - N) + 2$$

Donde

A = 21 Aristas

N= 15 Nodos

Remplazamos en la formula

$$V(G) = (20 - 15) + 2$$

$$V(G) = 5 + 2 = 7$$

Los caminos que deben ser aprobados dado el valor resultante es de 7 caminos que son los siguientes:

Tabla 14.

Resultados Caja Blanca

pruebas	
Camino 1	1,2,3,16
Camino 2	1,2,4,5,16
Camino 3	1,2,4,6,7,8,16
Camino 4	1,2,4,6,7,9,10,14,15,16
Camino 5	1,2,4,6,7,9,11,14,15,16
Camino 6	1,2,4,6,7,9,12,14,15,16
Camino 7	1,2,4,6,7,9,13,14,15,16

Para lo cual se ejecutará ciertos caminos y se debe evaluar y establecer las condiciones en las que se ejecutan los nodos establecidos en el camino

Tabla 15.

Resultados en la evaluación de los caminos

pruebas	
Camino 1	Error campos ingresado inválidos
Camino 2	Extracción ya fue ejecutada
Camino 3	Dominio no registrado
Camino 4	Datos registrados y almacenados
Camino 5	Datos registrados y almacenados
Camino 6	Datos registrados y almacenados
Camino 7	Datos registrados y almacenados

4.4.2 Pruebas de sistema

Las pruebas de sistema se centran en evaluar las funcionalidades requeridas de un módulo, enfocándose en verificar si cumple con sus especificaciones. Este tipo de pruebas se basa en analizar el comportamiento externo del sistema, sin considerar su estructura interna. Estas pruebas buscan identificar casos específicos que aseguren que el sistema responde adecuadamente a los requisitos planteados. Para realizar estas pruebas, se establecerán funciones clave y se analizarán los resultados obtenidos para validar el correcto funcionamiento del sistema.

Tabla 16.*Caja negra recolección de datos*

Prueba Nro. 1 Recolección de datos			
Pruebas	Entrada o acción de usuarios correctos	Resultados de sistema	¿Resultado?
P1	Selecciona Agregar Página Web ingresa nuevos datos	El sistema validara los datos del formulario y llenara los datos el backend con éxito	SI
P2	Selecciona Agregar Página Web ingresa datos incompletos	El sistema alertara sobre completar la información	SI
P3	Selecciona Iniciar Web Scraping e ingresa datos nuevos para el registro	El sistema iniciara el Web Scraping con los datos proporcionados	SI
P4	Selecciona Iniciar Web Scraping e ingresa datos ya registrados	El sistema informara sobre el registro encontrado	Si
P5	Selecciona Iniciar Web Scraping e ingresa datos incompletos	El sistema le pedirá que complete la información faltante	SI

Tabla 17.*Caja negra Entrenamiento del modelo*

Prueba Nro. 2 Entrenamiento del modelo			
Pruebas	Entrada o acción de usuarios correctos	Resultados de sistema	¿Resultado?
P1	Ingresa el texto para entrenar, selecciona datos relevantes, entrena al modelo	El sistema pedirá los datos según los pasos y confirmará los datos ingresados	SI
P2	No ingresa ningún texto	El sistema no dejara pasar al segundo paso sin texto que procesar	SI
P3	No se selecciona alguna etiqueta para el entrenamiento	El sistema no dejara pasar datos sin tener una etiqueta definida	SI

Tabla 18.*Caja negra evaluación de modelos*

Prueba Nro. 3 Evaluación de modelos			
Pruebas	Entrada o acción de usuarios correctos	Resultados de sistema	¿Resultado?
P1	Ingresa el texto de la descripción a evaluar	El sistema dejara pasar al siguiente paso para evaluar	SI
P2	Selecciona a comienza la evaluación	El sistema mostrara la información encontrada	SI
P3	No ingresa una descripción a evaluar	El sistema no le dejara pasar al siguiente paso	SI
P4	Ingresa información que no tiene datos para encontrar	El sistema mostrara los resultados sin datos ya que no existen	Si

Tabla 19.*Caja negra inicio de sesión*

Prueba Nro. 4 Inicio de sesión			
Pruebas	Entrada o acción de usuarios correctos	Resultados de sistema	¿Resultado?
P1	Ingresa el correo correcto y la contraseña	El sistema validara los datos del formulario y dejara el acceso al sistema	SI
P2	El usuario no ingresa un correo electrónico	El sistema informara que ingrese un correo electrónico	SI
P3	El usuario ingresa datos incorrectos	El sistema informara sobre los datos incorrectos	SI
P4	El usuario no ingresa ningún registro	El sistema informara que complete la información	Si
P5	El usuario ingresa una contraseña menos de 6 caracteres	El sistema informara sobre la cantidad minima de caracteres	SI

4.4.3 Pruebas de estrés

Para realizar las pruebas de estrés del software, utilizaremos la herramienta JMeter, que permite simular cargas de solicitudes al sistema para evaluar su capacidad de respuesta bajo condiciones de alta demanda. Configuraremos los parámetros de carga adecuados y ejecutaremos pruebas incrementales, comenzando con la simulación de un usuario por segundo, para observar y analizar el rendimiento del sistema ante distintas cargas.

Figura 67.
Interfaz Apache JMeter

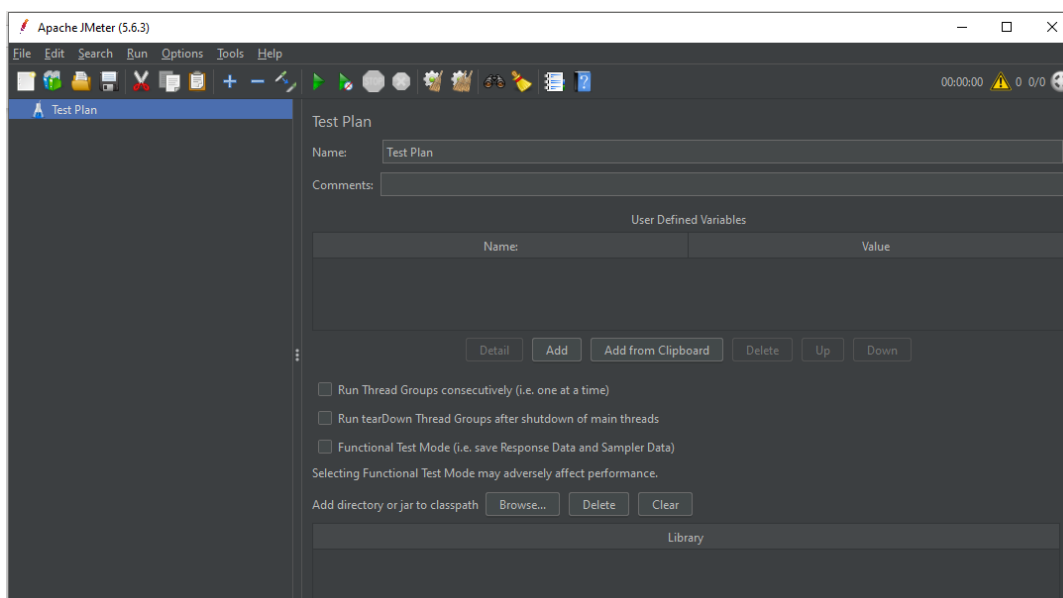
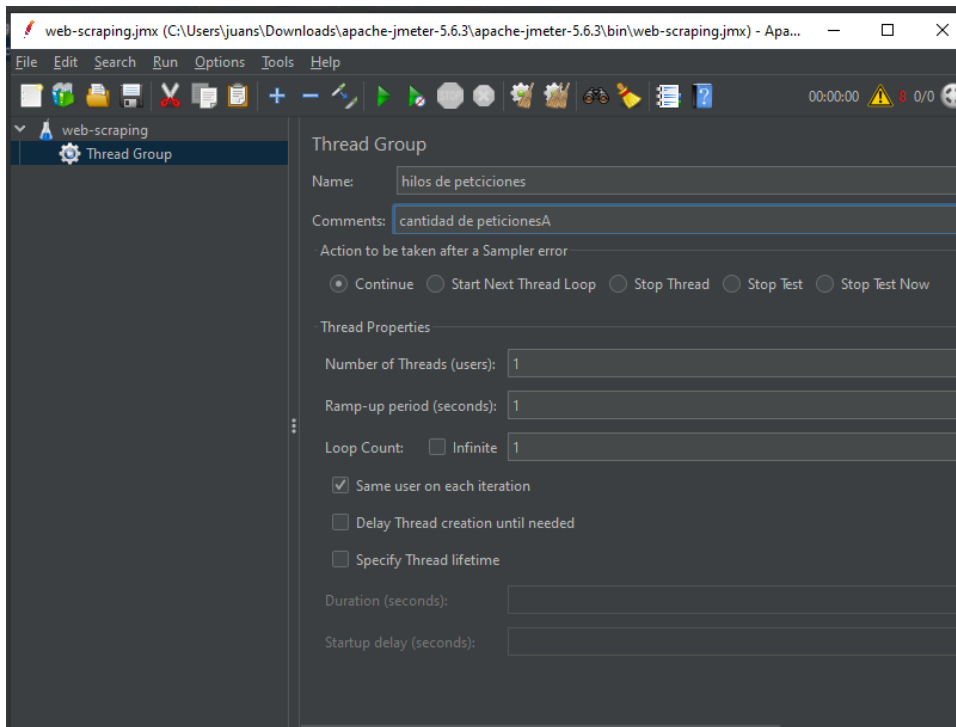
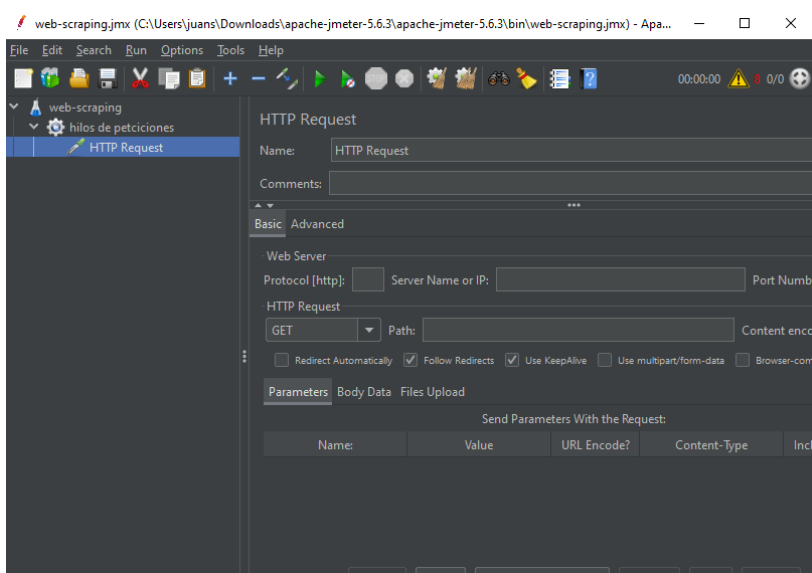


Figura 68.
Configuración de los hilos



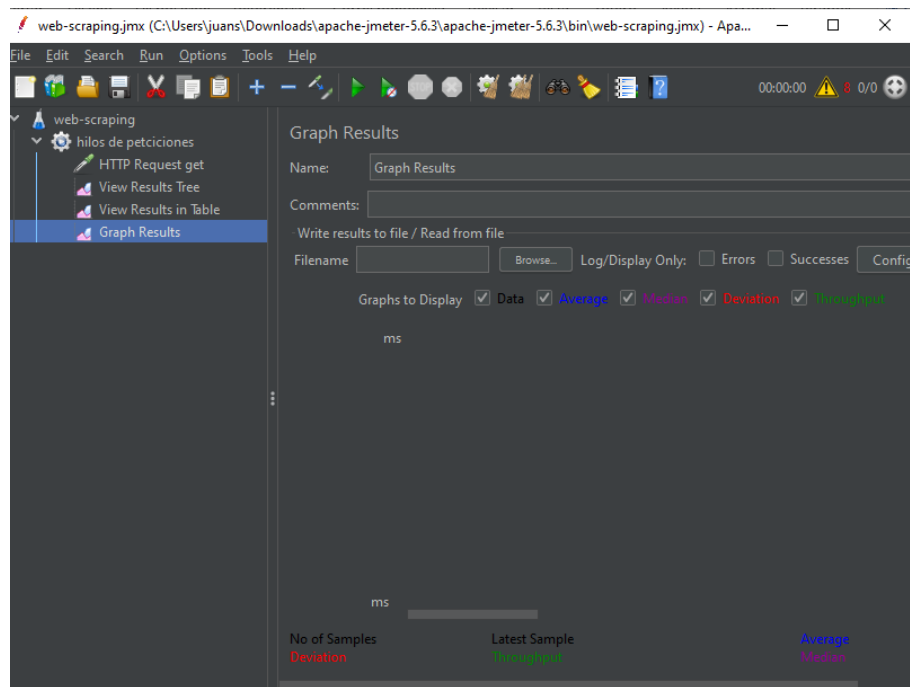
Nota: configuración de cantidad de hilos en peticiones.

Figura 69
Configuración de peticiones



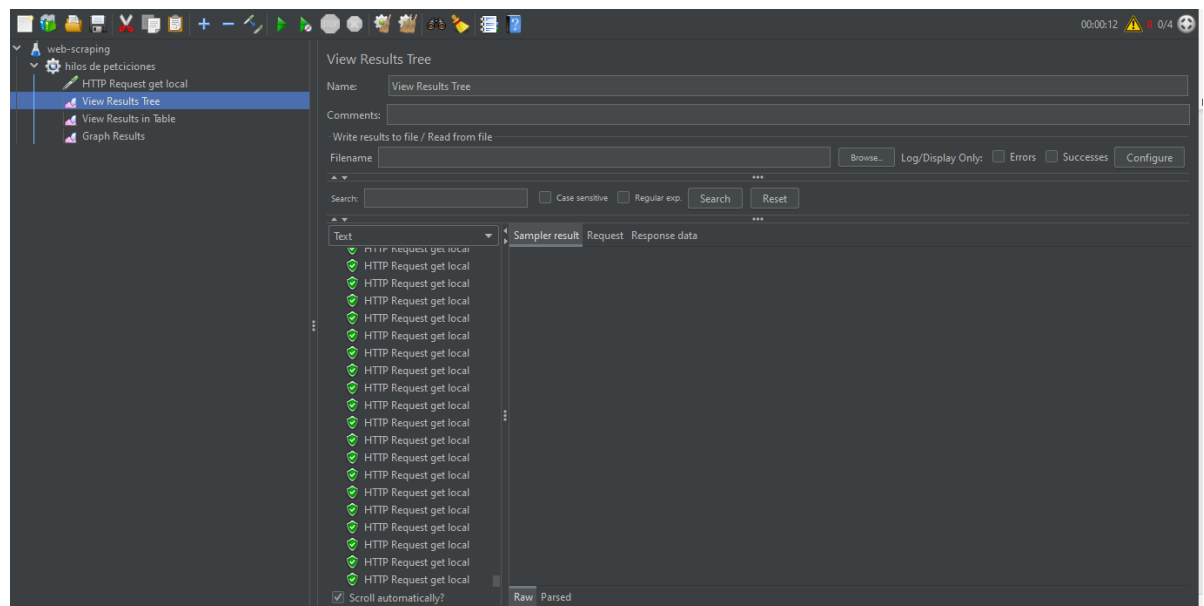
Nota: Configuración de las peticiones al servidor

Figura 70
Interfaz de creación de repostes y gráficos



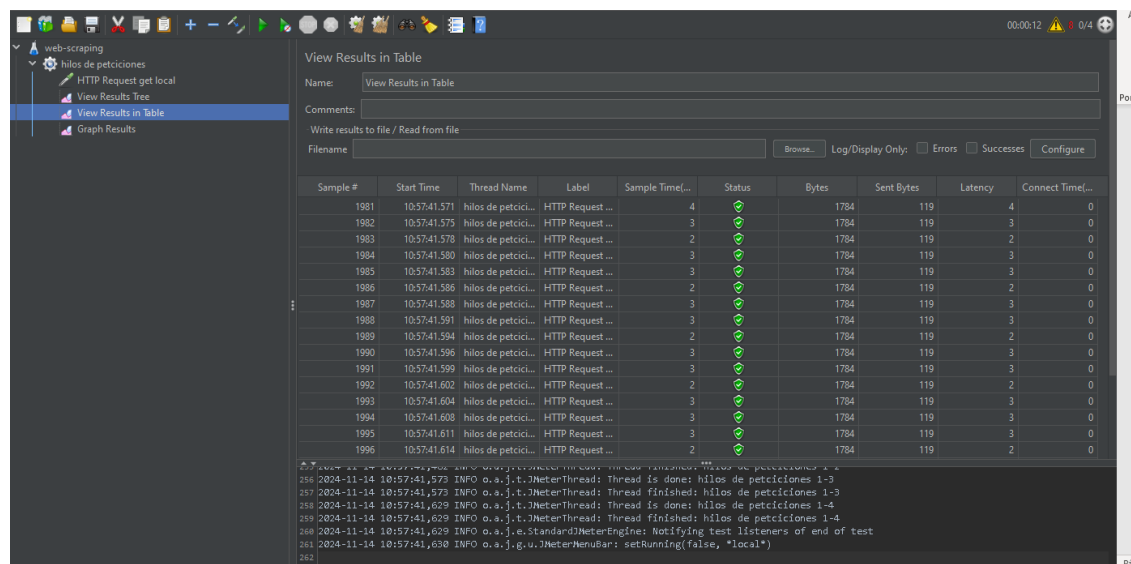
Nota: Interfaz de creación de reportes y gráficos.

Figura 71.
Resultados de las peticiones



Nota: Resultado en los estados 200.

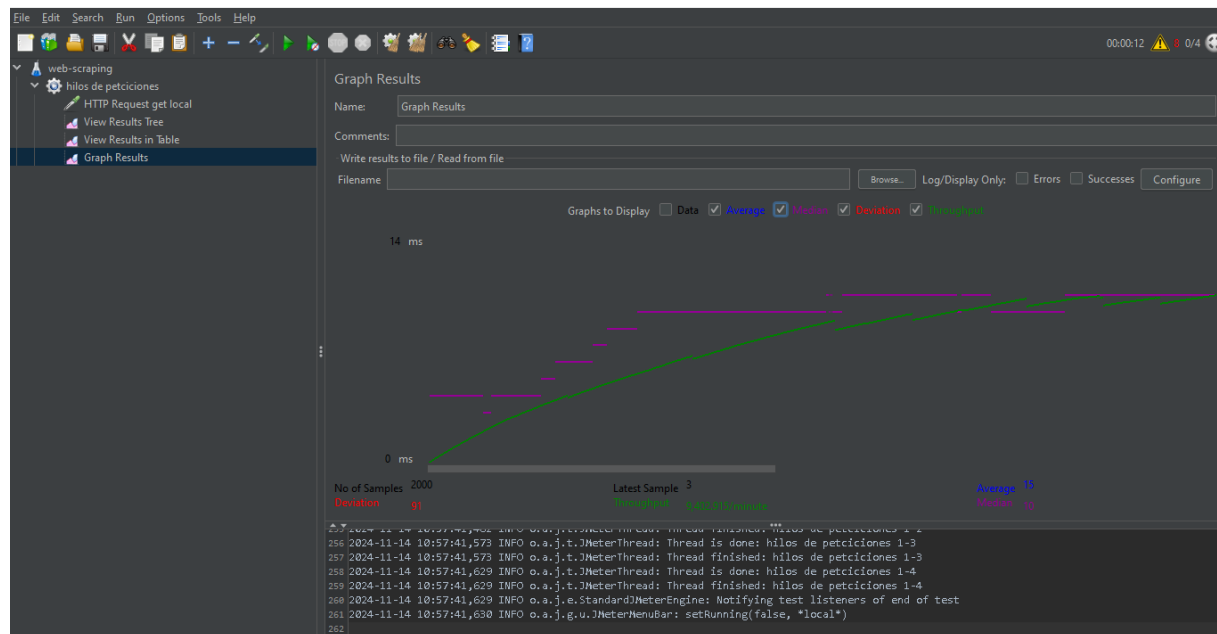
Figura 72.
Resultados de las peticiones en tabla



Sample #	Start Time	Thread Name	Label	Sample Time...	Status	Bytes	Sent Bytes	Latency	Connect Time...
1981	10:57:41.571	hilos de petici...	HTTP Request ...	4	Success	1784	119	4	0
1982	10:57:41.575	hilos de petici...	HTTP Request ...	3	Success	1784	119	3	0
1983	10:57:41.578	hilos de petici...	HTTP Request ...	2	Success	1784	119	2	0
1984	10:57:41.580	hilos de petici...	HTTP Request ...	3	Success	1784	119	3	0
1985	10:57:41.583	hilos de petici...	HTTP Request ...	3	Success	1784	119	3	0
1986	10:57:41.586	hilos de petici...	HTTP Request ...	2	Success	1784	119	2	0
1987	10:57:41.588	hilos de petici...	HTTP Request ...	3	Success	1784	119	3	0
1988	10:57:41.591	hilos de petici...	HTTP Request ...	3	Success	1784	119	3	0
1989	10:57:41.594	hilos de petici...	HTTP Request ...	2	Success	1784	119	2	0
1990	10:57:41.596	hilos de petici...	HTTP Request ...	3	Success	1784	119	3	0
1991	10:57:41.599	hilos de petici...	HTTP Request ...	3	Success	1784	119	3	0
1992	10:57:41.602	hilos de petici...	HTTP Request ...	2	Success	1784	119	2	0
1993	10:57:41.604	hilos de petici...	HTTP Request ...	3	Success	1784	119	3	0
1994	10:57:41.608	hilos de petici...	HTTP Request ...	3	Success	1784	119	3	0
1995	10:57:41.611	hilos de petici...	HTTP Request ...	3	Success	1784	119	3	0
1996	10:57:41.614	hilos de petici...	HTTP Request ...	2	Success	1784	119	2	0

Nota: Resultado de datos en tabla donde se refleja el consumo por petición además de la latencia

Figura 73.
Resultado de las peticiones en gráfico



Nota: Resultado de datos en gráfico en donde se visualiza la relación de la latencia con el tiempo en función a las peticiones realizadas

Tabla 20.*Resultados por Apache JMeter*

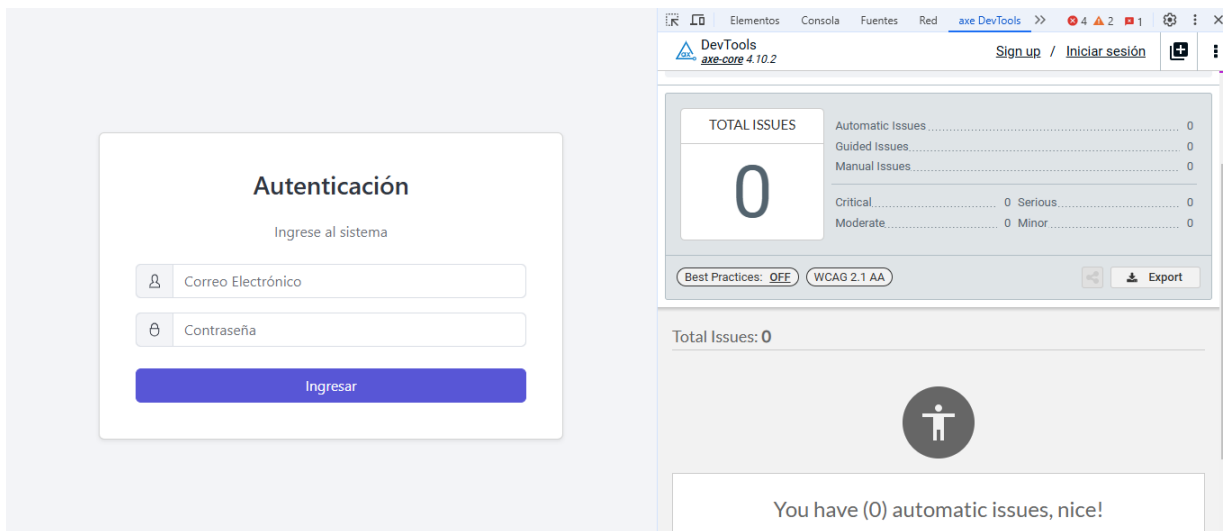
Cantidad de solicitudes	Tiempo	Porcentaje de error
10 usuarios	0.05 segundos	0%
100 usuarios	12 segundos	2%
500 usuarios	180 segundos	10%

4.4.4 Pruebas de accesibilidad

Para realizar las pruebas de accesibilidad se aplicará la herramienta de AXE DEVTOOLS, que permite evaluar las interacciones del usuario además de las configuraciones de la interfaz de usuario con sus respectivas etiquetas.

Figura 74.*Interfaz de axe DevTools v4*

Figura 75.
Resultados de la prueba de accesibilidad



CAPITULO V

CONCLUSIONES

Y

RECOMENDACIONES

**INGENIERÍA
DE SISTEMAS**
UNIVERSIDAD PÚBLICA DE EL ALTO

CAPITULO V

5. CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

El proyecto cumplió con el objetivo satisfactoriamente, con el desarrollo del Sistema Web Scraping de reconocimiento comercial en bienes raíces respaldado por Inteligencia Artificial. El desarrollo del proyecto se llevó a cabo siguiendo las metodologías, herramientas de Scraping logrando optimizar y dando confiabilidad de la información recolectada, así mismo reduciendo tiempos y esfuerzo para estas tareas.

- Se logro identificar las principales fuentes de información de bienes raíces en línea comprendiendo mejor las estructuras de la información dentro de las páginas web. **Figura 5 Figura 6**
- Se evaluó y analizo las herramientas de Web Scraping disponibles. **Tabla 4**
- Se desarrollo algoritmos de extracción de para obtener información de forma precisa y eficiente. **Figura 17**
- Se aplico técnicas de limpieza y preprocesamiento de datos asegurando la consistencia de la información. **Figura 22 Figura 23**
- Se aplico algoritmos de aprendizaje automático para clasificar y localizar datos relevantes de las distintas fuentes. **Figura 26**
- Se diseño e implemento una interfaz de fácil manejo para que los usuarios puedan interactuar con el sistema. **Figura 39**
- Finalmente se elaboró las pruebas de funcionalidad con casos realistas en diversos eventos. **Figura 60**

5.2 RECOMENDACIONES

Con el cumplimiento del presente proyecto se procede a mencionar las recomendaciones consideradas:

- Se recomienda entrenar el modelo de inteligencia artificial con datos válidos para ello capacitar a los nuevos usuarios que manejaran este sistema.
- Se propone generar back up en un periodo de 3 a 4 meses por la cantidad de datos almacenados asegurando así un mejor rendimiento de la aplicación.
- Se recomienda a los usuarios cambiar periódicamente las credenciales con el fin de asegurar el acceso al sistema.
- Se recomienda realizar extracciones una vez al mes como es previsto y no así seguido para mantener la integridad de los datos
- Para mejorar el funcionamiento de la recolección de información y poder encontrar información faltante en otros aspectos como la localización en coordenadas se recomienda y sugiere que se implemente un módulo de búsqueda de localización con los campos de ciudad y dar un aproximado de la localización así para tener mejor precisión al momento de analizar la información encontrada
- Si en un futuro se realiza una actualización del sistema se sugiere revisar la documentación para tomar decisiones y ejecutar la acción satisfactoriamente.

BIBLIOGRAFIA

**INGENIERÍA
DE SISTEMAS**
UNIVERSIDAD PÚBLICA DE EL ALTO



REFERENCIAS BIBLIOGRAFICAS

- Álvarez Correa Mercado Ricardo, y Rodríguez González Jose. (2022). Diseño de una solución IT para el análisis de sentimientos frente a la opinión de las figuras públicas usando técnicas de web scraping y la Twitter API. 2022-06-06.
- Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., y Wirth, R. (2000). *CRISP-DM 1.0: Step-by-step data mining guide*. CRISP-DM Consortium.
- COSMIC. (2015). *COSMIC Measurement Manual, Version 4.0.1: The COSMIC Functional Size Measurement Method*. The Common Software Measurement International Consortium.
- Grus, J. (2019). *Data Science from Scratch: First Principles with Python*. O'Reilly Media.
- International Organization for Standardization. (2011). *ISO/IEC 25010:2011 Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models*. International Organization for Standardization.
- International Organization for Standardization. (2013). *ISO/IEC 27001:2013 Information technology -- Security techniques -- Information security management systems -- Requirements*. International Organization for Standardization.
- Jurafsky, D., y Martin, J. H. (2023). *Speech and Language Processing* (3rd ed.). Pearson.
- Mijangos-Espinosa, R., Martínez-Rebollar, A., Estrada-Esquivel, H., y Hernández-Pérez, Y. (2022). *Use of Web Scraping Techniques for Automatic Capturing of Databases Located in the Web*.

- Flores Morales R. (2021). Traductor automático español-quechua, basado en el procesamiento del lenguaje natural. Universidad Mayor de San Andres
- Oblitas Villegas C. G. (2020). “Modelo De Análisis De Normas De Gobiernos Autónomos, Basado En Minería De Textos E Inteligencia Artificial”, Universidad Mayor de San Andres
- Myers, G. J., Sandler, C., y Badgett, T. (2011). *The Art of Software Testing*. John Wiley y Sons.
- Obe, R., y Hsu, L. (2015). *PostGIS in Action* (2nd ed.). Manning Publications.
- Sharma, A., y Sharma, P. (2018). *Software Testing: Principles and Practices*. Oxford University Press.
- Wojcik, O., y Gupta, A. (2021). *React Material-UI Cookbook*. Packt Publishing.
- Breiman, L., Friedman, J. H., Olshen, R. A., y Stone, C. J. (1986). Classification and regression trees. CRC Press.
- Cortes, C., y Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273–297.
<https://doi.org/10.1007/BF00994018>
- Ester, M., Kriegel, H. P., Sander, J., y Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*, 226–231.
- Jolliffe, I. T. (2002). *Principal component analysis* (2nd ed.). Springer.
- Grinberg, M. (2018). *Flask Web Development: Developing Web Applications with Python*. O'Reilly Media.

Hernández, A., y Palacios, J. (2020). Introducción al desarrollo web con Flask: Fundamentos y aplicaciones. Editorial UOC.

React. (2021). React - A JavaScript library for building user interfaces. <https://reactjs.org/>

Material-UI. (2020). Material-UI - React components that implement Google's Material Design. <https://mui.com/>

Pietraszek, J. (2019). Redis in Action: A Comprehensive Guide to Redis and Its Use Cases. Manning Publications.

Schwörer, M. (2018). Celery: Distributed Task Queue for Python.
<https://docs.celeryproject.org/en/stable/>

Playwright. (2023). Playwright: Node library to automate Chromium, Firefox, and WebKit with a single API. <https://playwright.dev>

PyPetter. (2022). PyPetter: A simple tool for web scraping.
<https://pyppeteer.github.io/pyppeteer/reference.html>

Selenium. (2020). Selenium: WebDriver for Python.
<https://www.selenium.dev/documentation/en/webdriver/>

Merkel, D. (2014). Docker: Lightweight Linux Containers for Consistent Development and Deployment. Linux Journal, 2014(239), 2.

BIBLIOGRAFIA

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000).

CRISP-DM 1.0 Step-by-step data mining guide. CRISP-DM Consortium.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to Algorithms. MIT Press.

Duckett, J. (2014). HTML and CSS: Design and Build Websites. Wiley.

Facebook. (2024). Jest Documentation. Jest. Disponible en <https://jestjs.io>

Few, S. (2012). Show Me the Numbers: Designing Tables and Graphs to Enlighten. Analytics Press.

Fowler, M. (2015). Patterns of Enterprise Application Architecture. Addison-Wesley.

Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media.

Grinberg, M. (2018). Flask Web Development: Developing Web Applications with Python. O'Reilly Media.

Han, J., Pei, J., & Kamber, M. (2011). Data Mining: Concepts and Techniques. Elsevier.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... & Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825), 357–362.
<https://doi.org/10.1038/s41586-020-2649-2>

McKinney, W. (2012). Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython. O'Reilly Media.

Postman. (2024). Postman Learning Center. Postman. Disponible en <https://learning.postman.com>

Pressman, R. S., & Maxim, B. R. (2019). Software Engineering: A Practitioner's Approach. McGraw-Hill.

Provost, F., & Fawcett, T. (2013). Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking. O'Reilly Media.

Python Software Foundation. (2024). unittest — Unit testing framework. Python Docs. Disponible en <https://docs.python.org>

Russell, S., & Norvig, P. (2021). Artificial Intelligence: A Modern Approach. Pearson.

Stuttard, D., & Pinto, M. (2011). The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws. Wiley.

Tanenbaum, A. S., & Bos, H. (2014). Modern Operating Systems. Pearson.

The Apache Software Foundation. (2024). Apache JMeter User Manual. Apache. Disponible en <https://jmeter.apache.org>

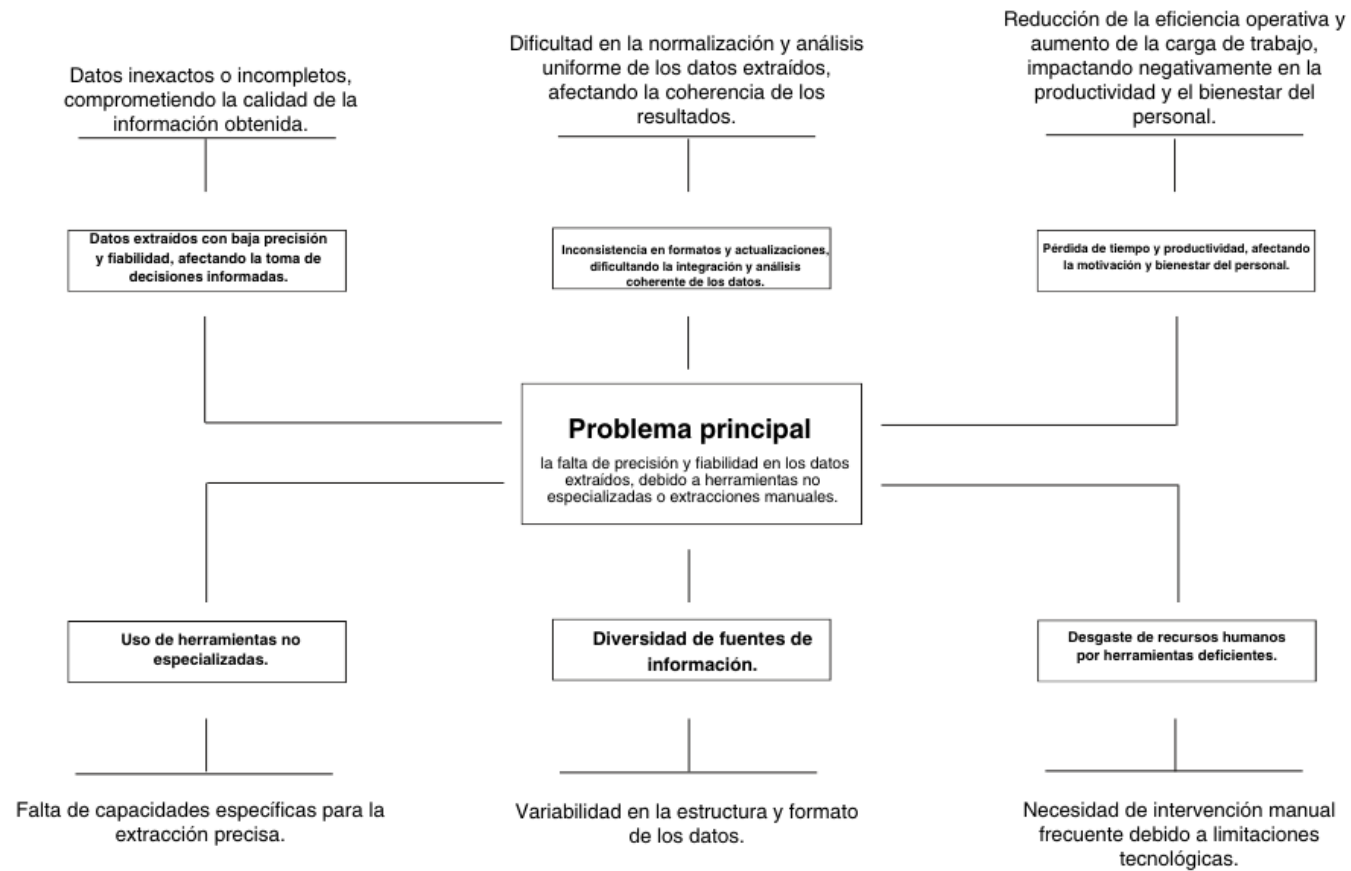
Van Rossum, G., & Drake, F. L. (2009). Python 3 Reference Manual. PythonLabs.

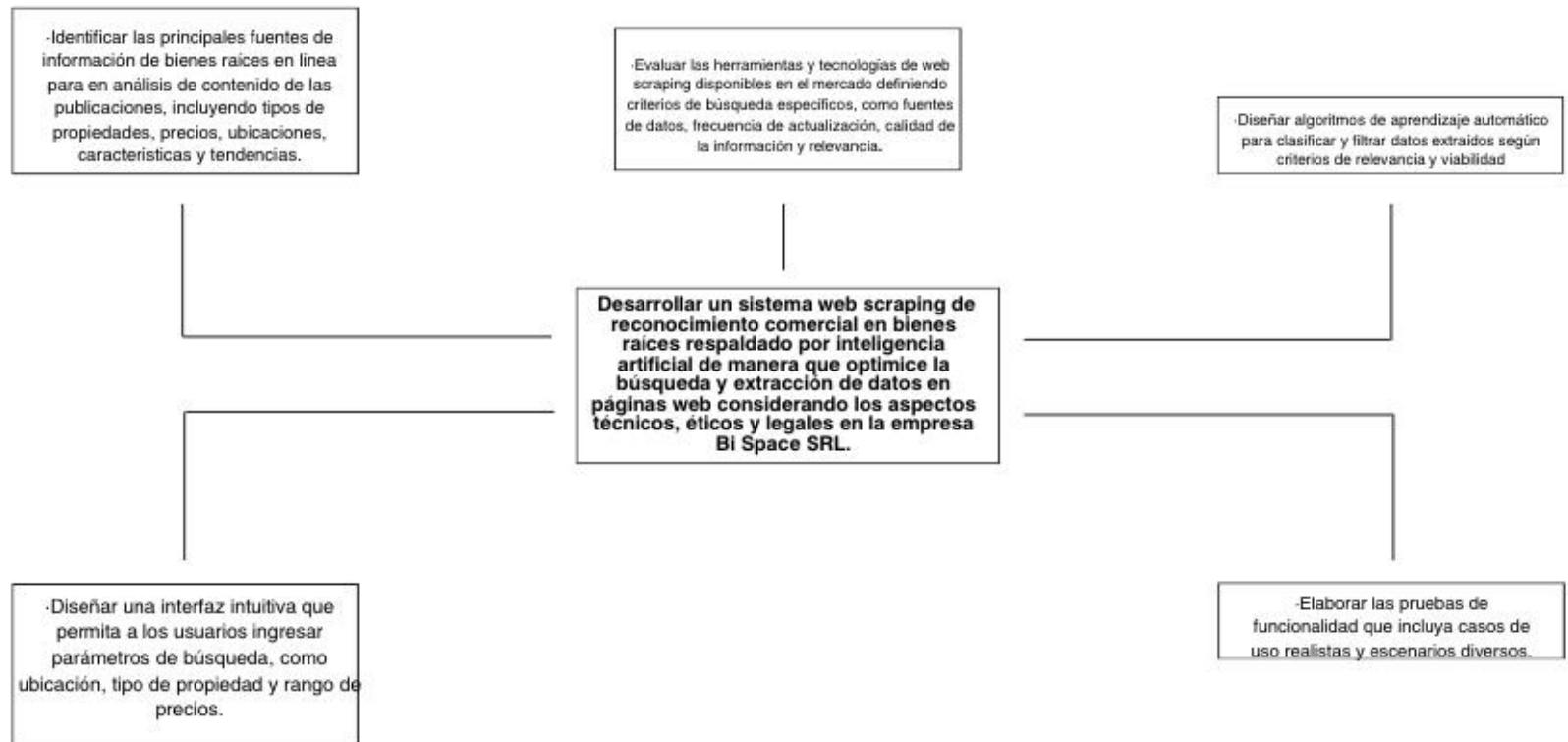
ANEXOS

**INGENIERÍA
DE SISTEMAS**
UNIVERSIDAD PÚBLICA DE EL ALTO



ANEXO A: ÁRBOL DE PROBLEMAS



ANEXO B. ÁRBOL DE OBJETIVOS

AVAL DE CONFORMIDAD

(TUTOR ESPECIALISTA)

El Alto, noviembre de 2024

Señora:

M. Sc. Lic. Ing. Marisol Arguedas Balladares

TUTOR METODOLÓGICO

TALLER DE GRADO II

Presente. –

REF. AVAL DE CONFORMIDAD

Distinguido tutor metodológico:

Mediante la presente tengo a bien comunicarle mi conformidad del Trabajo de Grado:

TITULO: SISTEMA WEB SCRAPING DE RECONOCIMIENTO COMERCIAL EN BIENES RAÍCES RESPALDADO POR INTELIGENCIA ARTIFICIAL

CASO: EMPRESA BI SPACE SRL

MODALIDAD: PROYECTO DE GRADO

Universitario: Juan Alberto Silva Cayo

Registro Universitario: 200036524

Cedula de Identidad: 10013618 LP

Para su defensa pública y evaluación correspondiente a la materia de Taller de Grado II, de acuerdo al Reglamento Vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Atentamente,

.....
M. Sc. Lic. Ing. Freddy Félix Medina Miranda
TUTOR ESPECIALISTA

El TUTOR ESPECIALISTA MSc. Lic. Ing. Freddy Félix Medina Miranda

Ha cumplido el Reglamento de Titulación de Modalidades de Graduación a nivel de Licenciatura de la carrera de Ingeniería de Sistemas.

- a) Orientar y guiar al postulante en el cumplimiento del desarrollo del Trabajo de Grado en los aspectos de: metodología de desarrollo, uso de herramientas para la construcción del sistema, modelo, prototipos, en cumplimiento con los objetivos planteados en el Perfil de Trabajo de Grado. Los puntos a trabajar son: Reconocimiento, formalización, análisis del problema y objetivos. Recolección de información relacionada con el tema del Trabajo de Grado.
- b) Realizar las revisiones previas a la defensa y/o trabajo de graduación para evaluar y determinar el nivel de competencia que permita a los postulantes ingresar a la siguiente etapa, según corresponda.
- c) Concluida el proceso de revisión de manera satisfactoria, deberá emitir el Aval de conformidad del Trabajo de Grado.

En ese sentido y con el objetivo de mantener la calidad de formación universitaria en la carrera de Ingeniería de Sistemas, como constancia de su cumplimiento, es que firma al pie de esta nota.

Nota. Quedando pendiente asistir a la Defensa Pública del Trabajo de Grado.

.....

M. Sc. Lic. Ing. Freddy Félix Medina Miranda

TUTOR ESPECIALISTA

AVAL DE CONFORMIDAD

(TUTOR REVISOR)

El Alto, noviembre de 2024

Señora:

M. Sc. Lic. Ing. Marisol Arguedas Balladares

TUTOR METODOLÓGICO

TALLER DE GRADO II

Presente. –

REF. AVAL DE CONFORMIDAD

Distinguido tutor metodológico:

Mediante la presente tengo a bien comunicarle mi conformidad del Trabajo de Grado:

TITULO: SISTEMA WEB SCRAPING DE RECONOCIMIENTO COMERCIAL EN BIENES RAÍCES RESPALDADO POR INTELIGENCIA ARTIFICIAL

CASO: EMPRESA BI SPACE SRL

MODALIDAD: PROYECTO DE GRADO

Universitario: Juan Alberto Silva Cayo

Registro Universitario: 200036524

Cedula de Identidad: 10013618 LP

Para su defensa pública y evaluación correspondiente a la materia de Taller de Grado II, de acuerdo al Reglamento Vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Atentamente,

.....
Lic. Ing. Yolanda Escobar Mancilla
TUTOR REVISOR

El **TUTOR REVISOR** Lic. Ing. Yolanda Escobar Mancilla

Cumplido con el Artículo 18.- (FUNCIONES DEL TUTOR) del Reglamento de Titulación de Modalidades de Graduación a nivel de Licenciatura de la carrera de Ingeniería de Sistemas.

- Orientar y asesorar en lo relativo a los aspectos de forma metodológica, que incluyen: técnicas, fuentes, redacción, ortografía, sintaxis, forma, estructura y coherencia del trabajo y la presentación del mismo. También otros aspectos metodológicos, en relación a:
 - a) Diseño de la Investigación;
 - b) Realización de la investigación;
 - c) Revisión de redacción, ortografía, sintaxis, coherencia y estructura del trabajo;
 - d) Elaboración de conclusiones;
 - e) Bibliografía bajo formato APA 7ma edición excepto el margen a la izquierda 3cm y paginas justificadas.
 - f) Asistir a la pre defensa.
 - g) Podrán proponer correcciones al Trabajo de Grado las veces que consideren necesarias en el formulario de seguimiento.
 - h) Concluida el proceso de revisión de manera satisfactoria, deberá emitir el **Aval de conformidad** del Trabajo de Grado.

En ese sentido y con el objetivo de mantener la calidad de formación universitaria en la carrera de Ingeniería de Sistemas, como constancia de su cumplimiento, es que firma al pie de esta nota.

Nota. Quedando pendiente asistir a la Defensa Pública del Trabajo de Grado.

.....
Lic. Ing. Yolanda Escobar Mancilla
TUTOR REVISOR

AVAL DE CONFORMIDAD

(TUTOR METODOLÓGICO)

El Alto, noviembre de 2024

Señor:
Lic. Ing. William Roque Roque
DIRECTOR CARRERA INGENIERÍA DE SISTEMAS
UNIVERSIDAD PÚBLICA DE EL ALTO
Presente. –

REF. AVAL DE CONFORMIDAD

Distinguido director:

Mediante la presente tengo a bien comunicarle mi conformidad del Trabajo de Grado:

TITULO: SISTEMA WEB SCRAPING DE RECONOCIMIENTO COMERCIAL EN BIENES RAÍCES RESPALDADO POR INTELIGENCIA ARTIFICIAL

CASO: EMPRESA BI SPACE SRL

MODALIDAD: PROYECTO DE GRADO

Universitario: Juan Alberto Silva Cayo

Registro Universitario: 200036524

Cedula de Identidad: 10013618 LP

Para que se le otorgue fecha y hora de su defensa pública y evaluación correspondiente a la materia de Taller de Grado II, de acuerdo al Reglamento Vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Atentamente,

.....
M. Sc. Lic. Ing. Marisol Arguedas Balladares
TUTOR METODOLÓGICO



MANUAL DE USUARIO

SISTEMA WEB SCRAPING DE RECONOCIMIENTO COMERCIAL RESPALDADO
POR INTELIGENCIA ARTIFICIAL

AUTOR: JUAN ALBERTO SILVA CAYO
EMPRESA: BI SPACE SRL

Contenido

1. INTRODUCCION	2
2. OBJETIVO	2
3. HARDWARE Y SOFTWARE	3
4. FUNCIONALIDAD DEL SISTEMA.....	3

1. INTRODUCCION

Bienvenido al Manual de Usuario del Sistema Web Scraping de Reconocimiento Comercial. Este sistema ha sido desarrollado con el objetivo de facilitar la extracción, análisis y procesamiento de información comercial desde diversas fuentes web, con un enfoque especial en el sector inmobiliario. Gracias a su capacidad para automatizar la recolección de datos, este sistema permite a los usuarios obtener información clave sobre precios, características de propiedades, ubicación, y más, directamente desde sitios web de bienes raíces.

Este manual le guiará paso a paso en el uso del sistema, desde la configuración inicial hasta la generación de reportes y análisis. Con esta herramienta, podrá optimizar sus procesos de recolección de datos, obtener insights comerciales valiosos y tomar decisiones informadas para mejorar la competitividad en el mercado inmobiliario.

2. OBJETIVO

El objetivo principal de este manual es proporcionar una guía detallada y clara sobre cómo utilizar el **Sistema Web Scraping de Reconocimiento Comercial** de manera efectiva y eficiente. A través de este manual, los usuarios aprenderán a:

- **Configurar y personalizar** el sistema para adaptarlo a sus necesidades.
- **Realizar la extracción de datos** desde diversas páginas web de bienes raíces.
- **Analizar los datos recolectados** para obtener insights útiles sobre el mercado inmobiliario.
- **Generar reportes y visualizaciones** para ayudar en la toma de decisiones comerciales.

Este manual está diseñado para asegurar que los usuarios puedan aprovechar todas las funcionalidades del sistema, mejorando así su capacidad para realizar un web scraping de alta calidad, optimizando su estrategia comercial y reduciendo el tiempo dedicado a la recolección manual de datos.

3. HARDWARE Y SOFTWARE

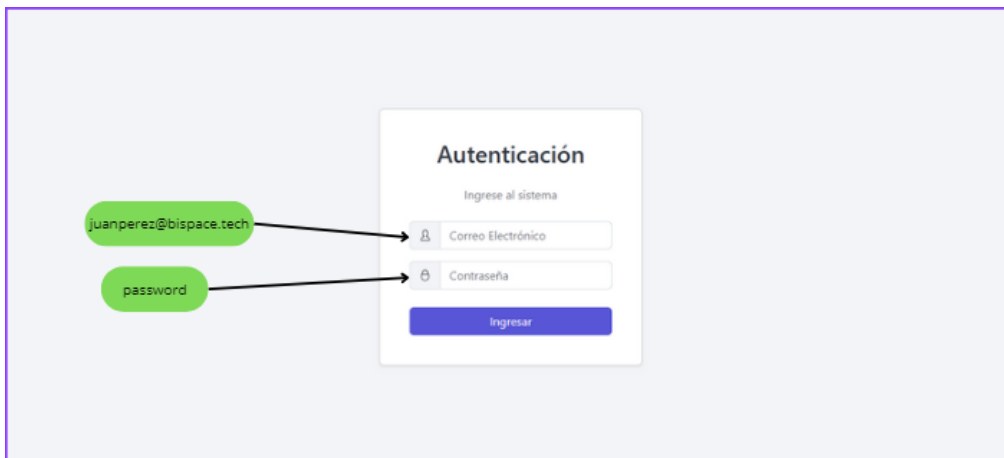
Para hacer uso del sistema el usuario requiere contar con:

- Una computadora o dispositivo móvil conectada a internet
- Un navegador de internet, como ser OPERA, BRAVE, MOZILLA FIREFOX, GOOGLE CHROME, EDGE, para Windows 7 y versiones superiores y en dispositivos móviles desde Android 8 para adelante

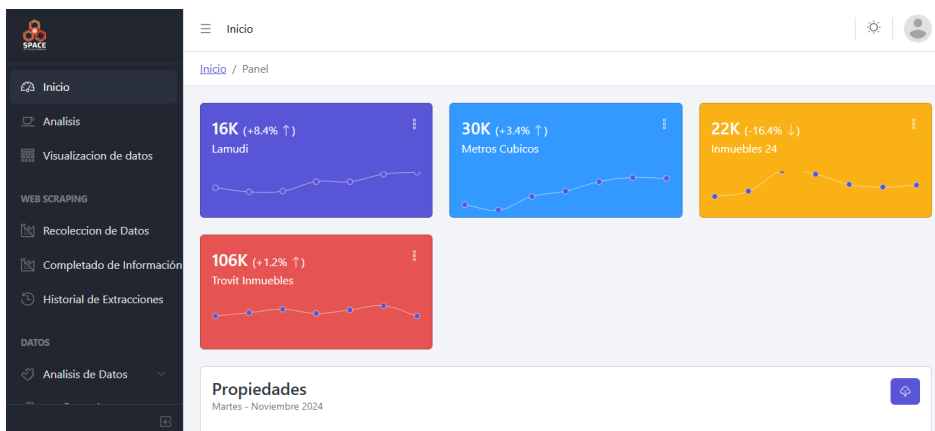
4. FUNCIONALIDAD DEL SISTEMA

Pantalla inicial

En la pantalla inicial el sistema te pedirá tus datos de autenticación como ser tu usuario y contraseña



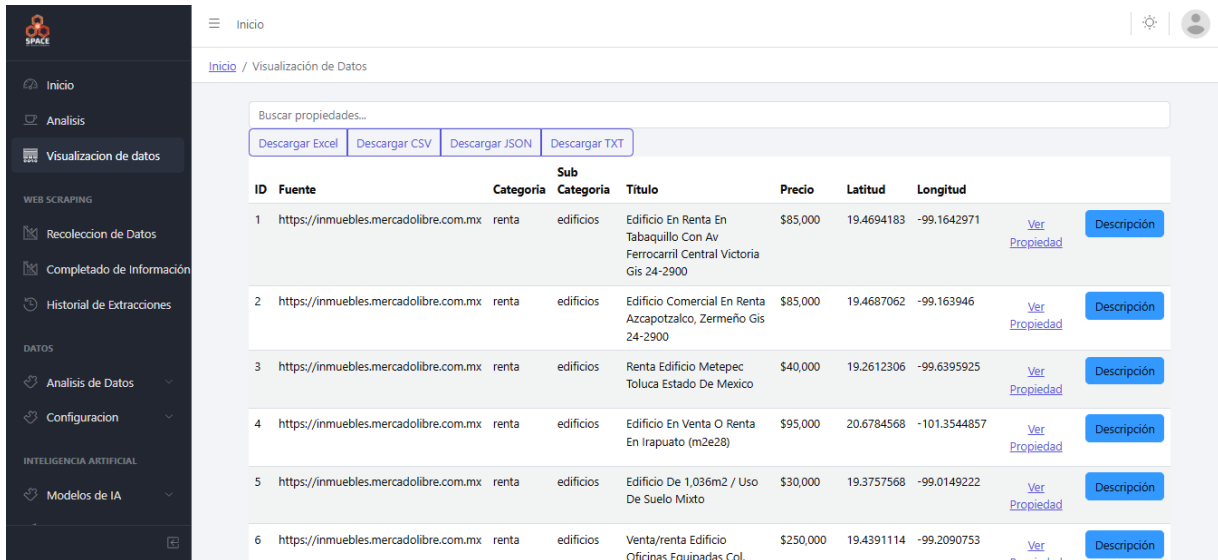
En donde te llevara a la pantalla de inicio del sistema



En donde se podrá visualizar los datos referentes a las extracciones

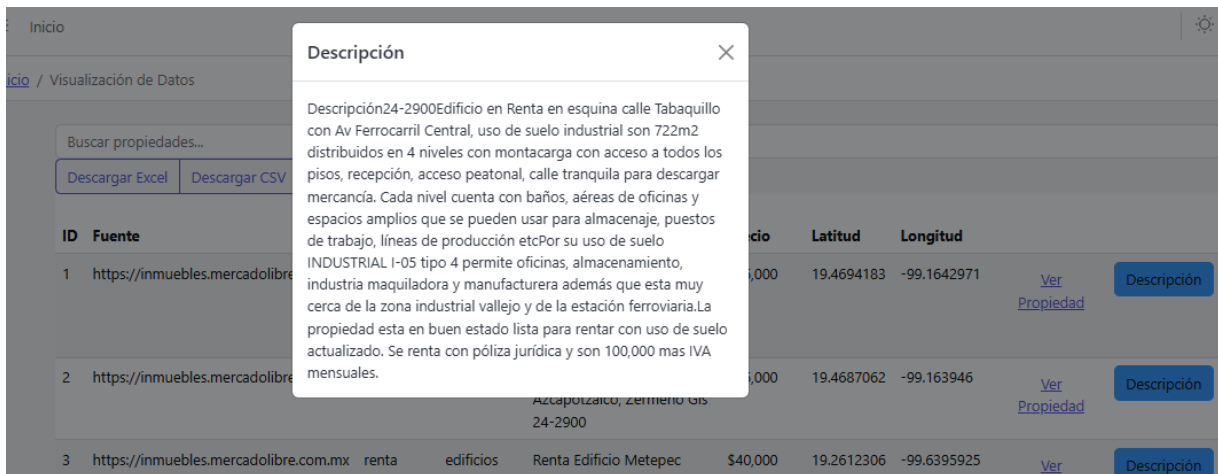
VISUALIZACION DE DATOS

En esta sección se podrá visualizar los datos extraídos de las distintas fuentes



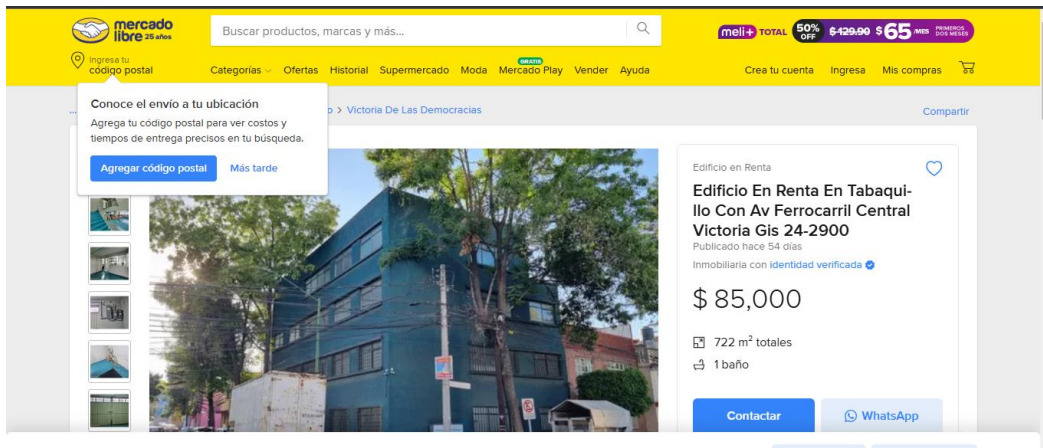
ID	Fuente	Sub Categoría	Categoría	Título	Precio	Latitud	Longitud		
1	https://inmuebles.mercadolibre.com.mx	renta	edificios	Edificio En Renta En Tabaquillo Con Av Ferrocarril Central Victoria Gis 24-2900	\$85,000	19.4694183	-99.1642971	Ver Propiedad	Descripción
2	https://inmuebles.mercadolibre.com.mx	renta	edificios	Edificio Comercial En Renta Azcapotzalco, Zermeno Gis 24-2900	\$85,000	19.4687062	-99.163946	Ver Propiedad	Descripción
3	https://inmuebles.mercadolibre.com.mx	renta	edificios	Renta Edificio Metepec Toluca Estado De Mexico	\$40,000	19.2612306	-99.6395925	Ver Propiedad	Descripción
4	https://inmuebles.mercadolibre.com.mx	renta	edificios	Edificio En Venta O Renta En Irapuato (m2e28)	\$95,000	20.6784568	-101.3544857	Ver Propiedad	Descripción
5	https://inmuebles.mercadolibre.com.mx	renta	edificios	Edificio De 1,036m2 / Uso De Suelo Mixto	\$30,000	19.3757568	-99.0149222	Ver Propiedad	Descripción
6	https://inmuebles.mercadolibre.com.mx	renta	edificios	Venta/renta Edificio Oficinas Enuinardas Col.	\$250,000	19.4391114	-99.2090753	Ver Propiedad	Descripción

En donde se podrá visualizar la descripción de la propiedad extraída

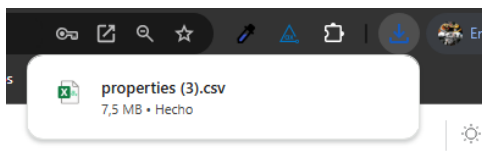
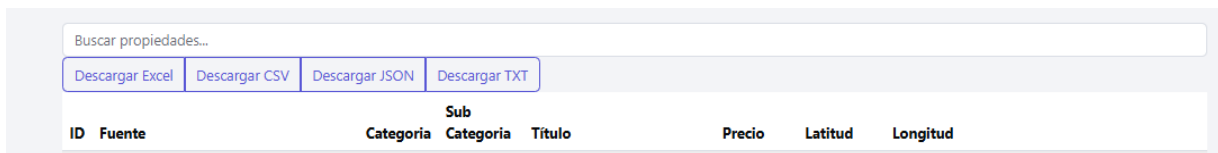


ID	Fuente	Sub Categoría	Categoría	Título	Precio	Latitud	Longitud		
1	https://inmuebles.mercadolibre.com.mx	renta	edificios	Edificio En Renta En Tabaquillo Con Av Ferrocarril Central Victoria Gis 24-2900	\$85,000	19.4694183	-99.1642971	Ver Propiedad	Descripción
2	https://inmuebles.mercadolibre.com.mx	renta	edificios	Edificio Comercial En Renta Azcapotzalco, Zermeno Gis 24-2900	\$85,000	19.4687062	-99.163946	Ver Propiedad	Descripción
3	https://inmuebles.mercadolibre.com.mx	renta	edificios	Renta Edificio Metepec Toluca Estado De Mexico	\$40,000	19.2612306	-99.6395925	Ver Propiedad	Descripción

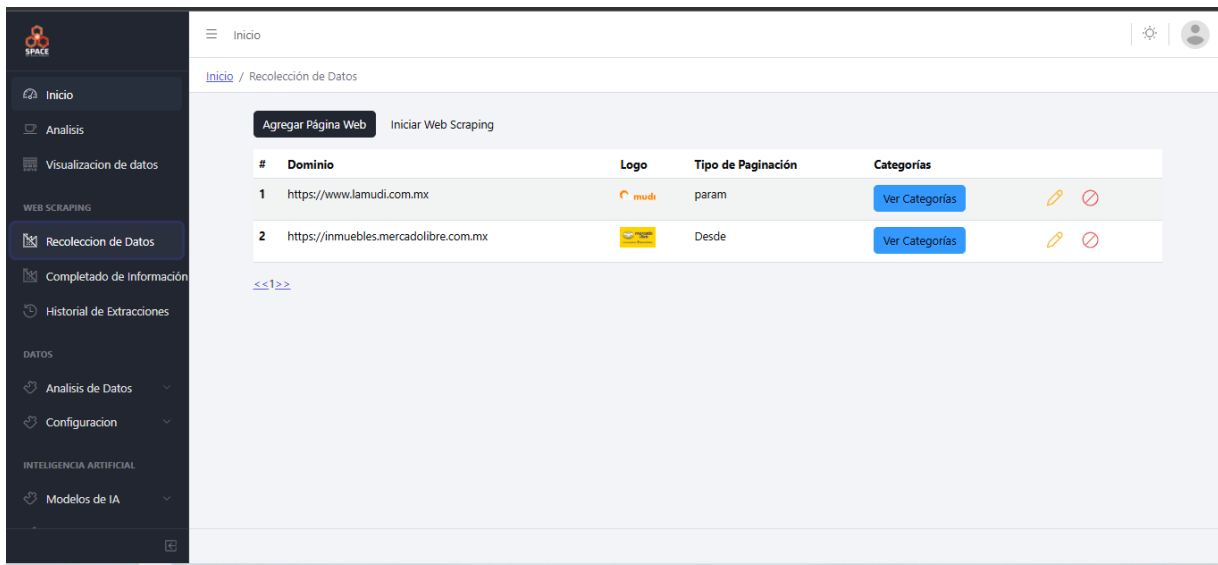
Además de poder visualizar la fuente de donde se extrajo la información



Esta información se podrá descargar en distintos formatos como Excel, CSV, JSON, TXT

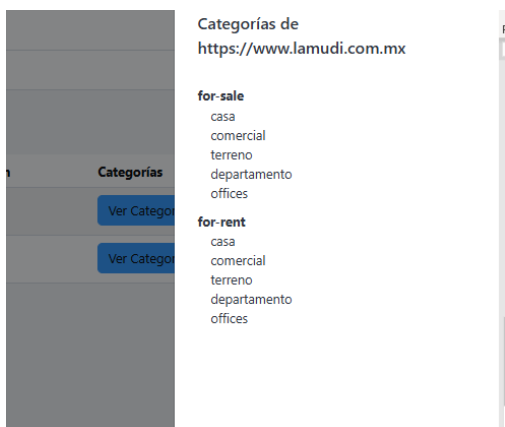


RECOLECCION DE DATOS



En donde se podrá visualizar las fuentes registradas además de poder registrar nuevas fuentes para extraer información

Visualización de categorías

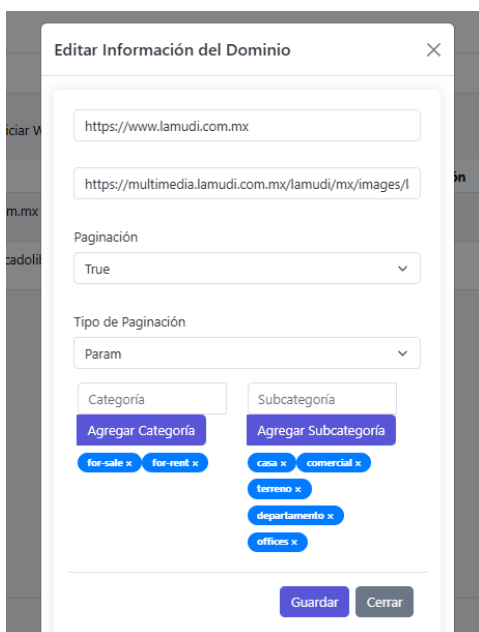


Para registrar una nueva fuente se mostrará el siguiente formulario

A screenshot of a modal form titled 'Agregar Nueva Página Web'. The form contains several input fields and buttons. At the top, there is a 'Dominio' input field. Below it is a 'Logo (URL)' input field. Then, a 'Paginación' dropdown menu is set to 'True'. Below that is a 'Tipo de Paginación' dropdown menu set to 'Slash'. There are two input fields for 'Categoría' and 'Subcategoría', each with a corresponding blue button: 'Agregar Categoría' and 'Agregar Subcategoría'. At the bottom of the form is a large blue button labeled 'Registrar Dominio'. A 'Cerrar' button is located at the bottom right of the modal.

En donde se registrara la url del dominio además de la imagen en url , la paginación y adicionar categorías y subcategorías.

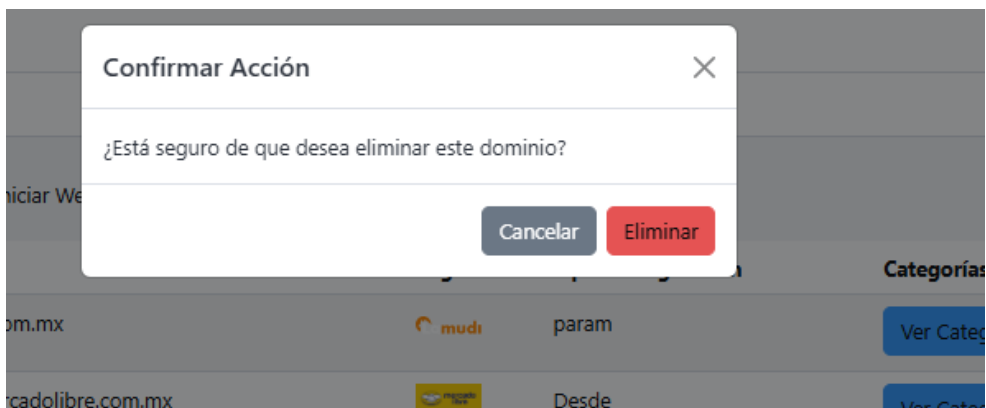
Editar dominio: en esta parte se podrá editar el dominio ya sea cambiando las categorías o subcategorías además de poder añadir o eliminar las categorías creadas



The screenshot shows a modal window titled "Editar Información del Dominio" with a close button (X) in the top right corner. The form contains the following elements:

- Two text input fields for domain URLs: "https://www.lamudi.com.mx" and "https://multimedia.lamudi.com.mx/lamudi/mx/images/l".
- A "Paginación" dropdown menu set to "True".
- A "Tipo de Paginación" dropdown menu set to "Param".
- Two input fields for "Categoría" and "Subcategoría".
- Two buttons: "Agregar Categoría" and "Agregar Subcategoría".
- Two columns of category tags with an "x" to remove them:
 - Under "Categoría": "for-sale x", "for-rent x".
 - Under "Subcategoría": "casa x", "comercial x", "terreno x", "departamento x", "offices x".
- At the bottom right, two buttons: "Guardar" and "Cerrar".

Estos registros de los dominios se podrán eliminar sin antes confirmar la eliminación de este.



The screenshot shows a modal dialog titled "Confirmar Acción" with a close button (X) in the top right corner. The dialog contains the text "¿Está seguro de que desea eliminar este dominio?" and two buttons at the bottom: "Cancelar" and "Eliminar".

INICIAR WEB SCRAPING

Inicio / Recolección de Datos

Agregar Página Web **Iniciar Web Scraping**

Dominio: Selecciona un dominio

Cantidad de Páginas:

Fecha: dd/mm/aaaa

Iniciar Scraping

En donde se podrá iniciar el proceso de extracción seleccionando el dominio las categorías, las subcategorías, las paginas a extraer información y por ultimo la fecha

Inicio / Recolección de Datos

Agregar Página Web **Iniciar Web Scraping**

Dominio: https://www.lamudi.com.mx

Categoría: for-sale

Subcategoría: casa

Cantidad de Páginas: 3

Fecha: 01/11/2024

Iniciar Scraping

Esta es la primera vuelta

```
{
  "dominio": "https://www.lamudi.com.mx",
  "categoria": "for-sale",
  "subcategoria": "casa",
  "limite": "3",
  "fecha": "2024-11-01"
}
```

En la tarjeta se visualizará la información recolectada y posibles errores.

En caso de que ya se realizó la extracción de información en una fecha específica y valores similares se mostrara el siguiente mensaje

The screenshot shows a web scraping interface with three dropdown menus: 'Dominio' (Domain) set to 'https://www.lamudi.com.mx', 'Categoría' (Category) set to 'for-sale', and 'Subcategoría' (Subcategory) set to 'casa'. Below these is a blue button labeled 'Iniciar Scraping'. A dark modal box is displayed over the button, containing the following text:

```
Esta es la primera vuelta  
{'dominio': 'https://www.lamudi.com.mx', 'categoria': 'for-sale',  
'subcategoria': 'casa', 'limite': '3', 'fecha': '2024-11-01'}  
El registro ya fue ejecutado
```

En caso de volver a extraer la información de una fecha específica se recomienda contactar con el administrado para que elimine el registro y se habilite la extracción

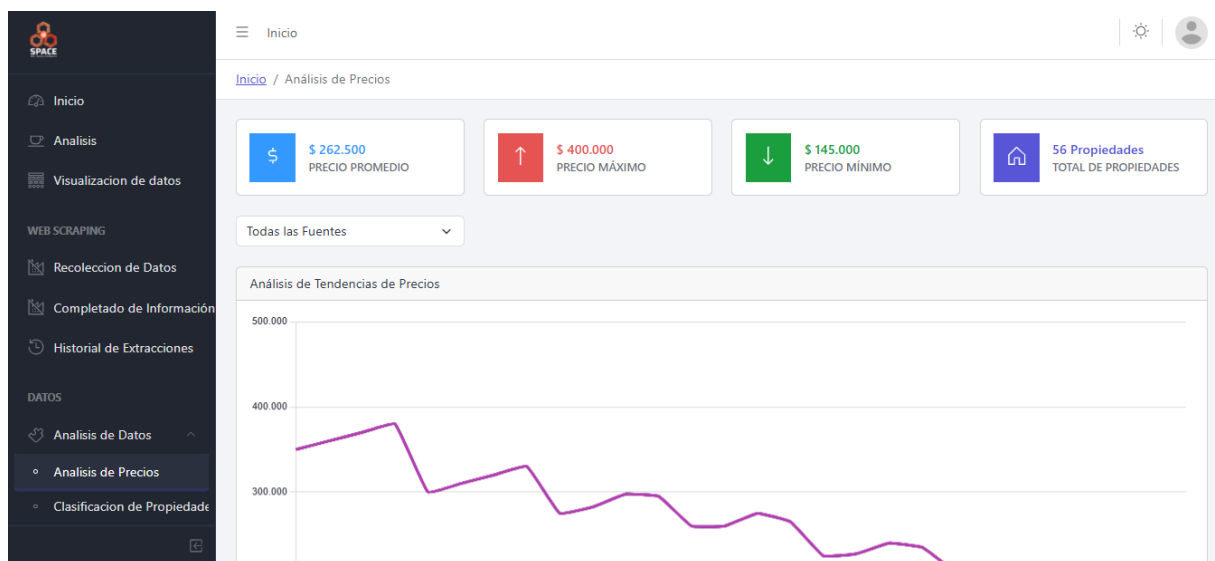
The screenshot shows the 'Historial de Extracciones' page. On the left is a sidebar with navigation links: 'Inicio', 'Análisis', 'Visualización de datos', 'WEB SCRAPING', 'Recolección de Datos', 'Completado de Información', 'Historial de Extracciones', 'DATOS', 'Análisis de Datos', 'Configuración', and 'INTELIGENCIA ARTIFICIAL'. The main content area has a search bar 'Buscar por dominio, categoría o subcategoría' and a table with the following data:

ID	Dominio	Categoría	Subcategoría	Fecha	Límite	
1	https://inmuebles.mercadolibre.com.mx	renta	edificios	01/11/2024	7	⛔
2	https://www.lamudi.com.mx	for-sale	casa	01/11/2024	3	⛔

Below the table is a pagination control showing '1', '2', and '3' with arrows.

Esta vista estará disponible pero el administrador tendrá el derecho de poder eliminar algún registro que se ejecutó en consideración con el técnico que se encargara de la extracción

ANALISIS DE DATOS



Esta es una vista en donde el usuario podrá ver los resúmenes de los datos extraídos en donde se visualizara el precio promedio el precio máximo el mínimo y la cantidad de registros además de poder filtrar según las distintas fuentes

Clasificación de propiedades

The dashboard includes search filters and a table of properties:

Filtros de búsqueda:

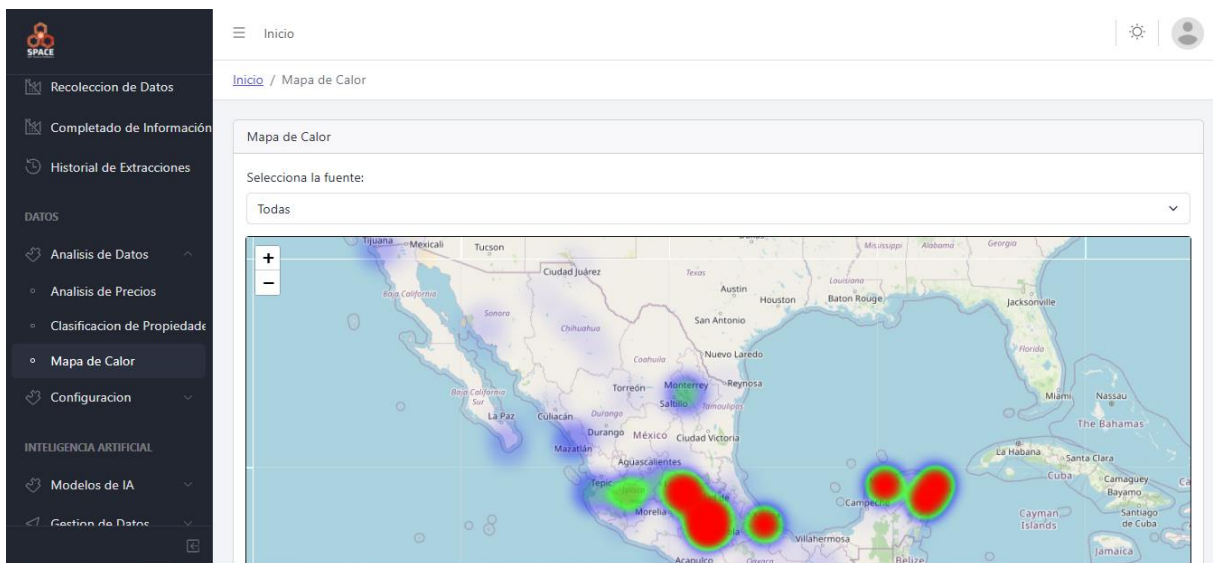
- Tipo de Propiedad: Todos
- Precio: Todos
- Ubicación: Todas
- Habitaciones: Todos
- Baños: Todos
- Superficie: Todos

Propiedades:

Nombre	Precio	Ubicación	Tamaño (m ²)	Habitaciones	Baños	Acción
Casa Moderna	\$5.000.000	Ciudad de México	250	3	2	Ver más
Departamento en Reforma	\$3.500.000	Ciudad de México	120	2	2	Ver más

Esta vista mostrara y filtrara datos según los datos recolectados para poder comprender mejor la calidad de datos y poder tomar decisiones con datos realistas y actualizadas

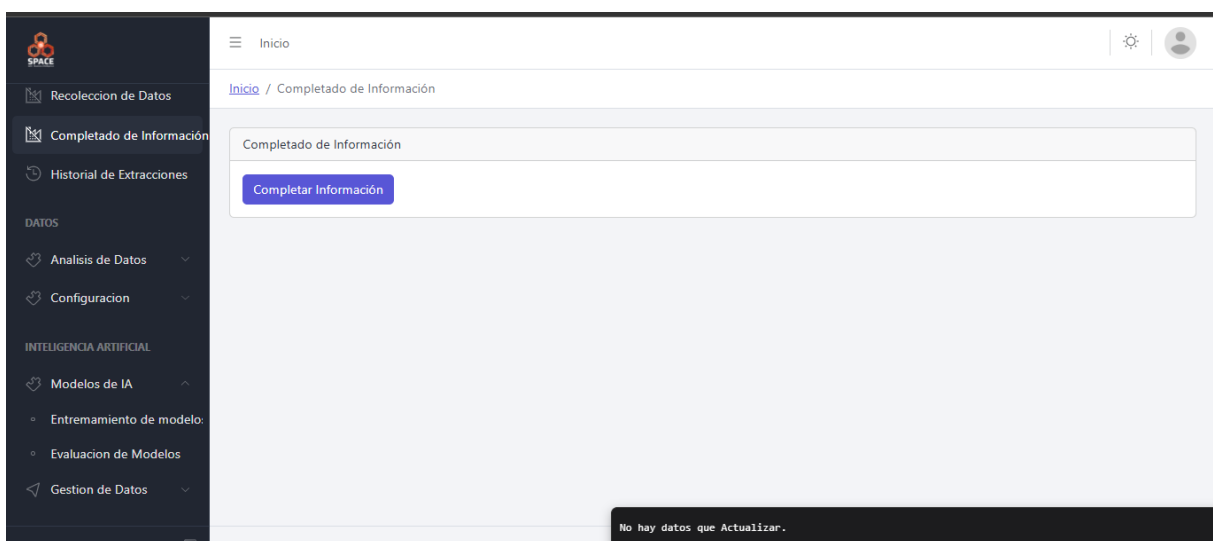
Mapa de calor



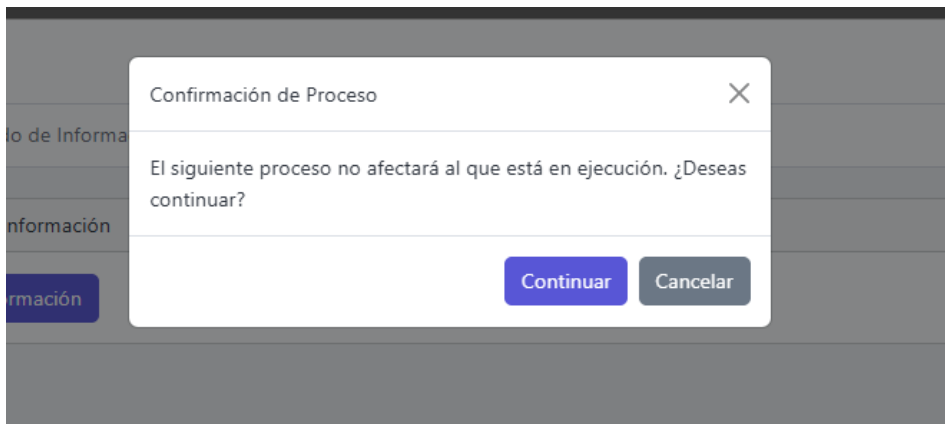
Esta sección es de visualización de datos según las fuentes en un mapa interactivo con datos reales y actuales, permitirá analizar los focos de calor en venta de propiedades se podrá filtrar por fuente recolectada.

COMPLETADO DE INFORMACIÓN

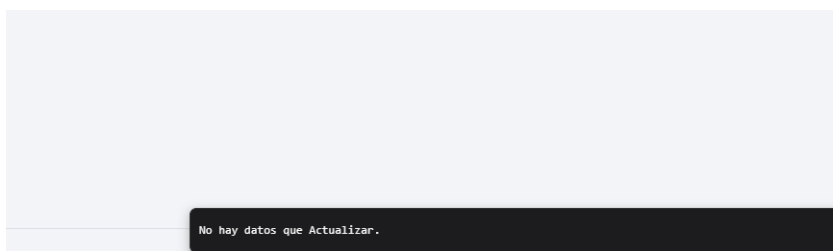
Para esta parte no se asegura al 100% que la información recolectada este completa así que se añadió esta vista para poder recolectar información precisa de los datos ya almacenados



En donde se preguntará e informara que el proceso no alterara o perjudicara a otros procesos en camino.



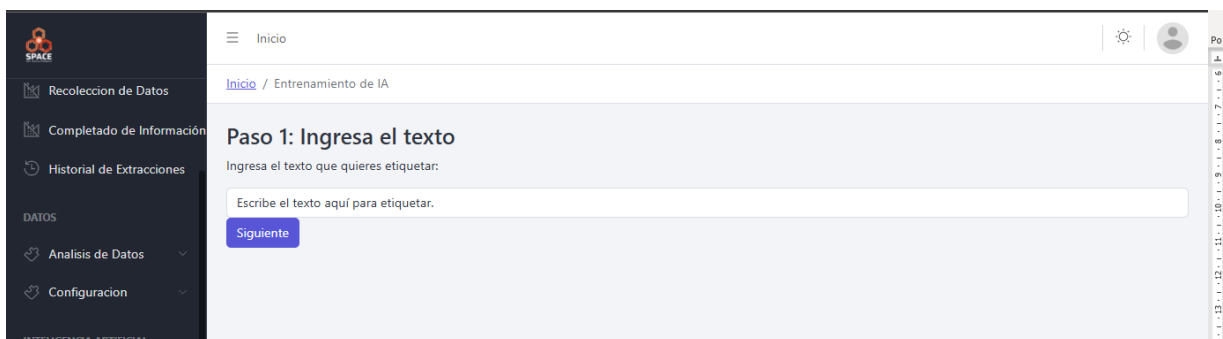
Si no hay datos que actualizar entonces mostrar que no existe datos para actualizar



MODELOS DE IA

Entrenamiento del modelo

En esta sección se vera como se entrena el modelo de IA con datos reales para poder asegurar y encontrar información relevante.



Ingresa el texto para etiquetar las palabras clave ej: Bonita casa con 3 habitaciones

Selecciona siguiente.

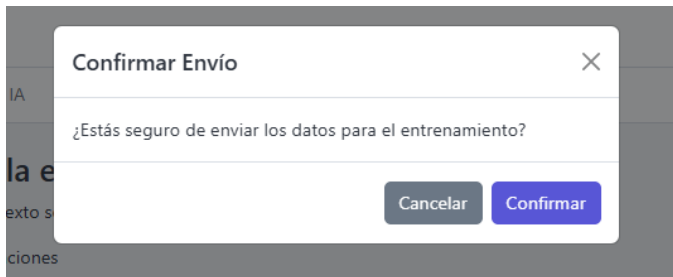
The screenshot shows a web application interface. On the left is a dark sidebar with a logo and a menu containing items like 'Recoleccion de Datos', 'Completado de Información', 'Historial de Extracciones', and sections for 'DATOS' and 'INTELIGENCIA ARTIFICIAL'. The main content area has a header 'Inicio' and a breadcrumb 'Inicio / Entrenamiento de IA'. The title is 'Paso 2: Selecciona el texto relevante' with the instruction 'Selecciona el texto relevante dentro del área de texto.' Below this, the text 'Bonita casa con 3 habitaciones' is displayed. A dropdown menu is open, showing 'Selecciona una etiqueta' with a downward arrow. Below the dropdown are two buttons: 'Agregar Texto Seleccionado' (purple) and 'Volver a escribir texto' (grey).

Selecciona la información relevante con la etiqueta correcta

This is a close-up of the dropdown menu from the previous screenshot. The title 'Paso 2: Selecciona el texto relevante' and the instruction 'Selecciona el texto relevante dentro del área de texto.' are visible at the top. The text 'Bonita casa con 3 habitaciones' is also present. The dropdown menu is open, showing a list of labels: 'Selecciona una etiqueta', 'Área Construida', 'Número de Baños', and 'Número de Habitaciones'. The 'Número de Habitaciones' option is highlighted with a blue background.

Se podrá seleccionar más etiquetas o también enviar datos para el entrenamiento

The screenshot shows the next step in the process. The breadcrumb is 'Inicio / Entrenamiento de IA'. The title is 'Paso 3: Define la etiqueta' with the instruction 'Define la etiqueta para el texto seleccionado:'. The text 'Bonita casa con 3 habitaciones' is displayed, with the number '3' highlighted in a blue circle. Below this, a dropdown menu shows 'Número de Habitaciones' with a downward arrow. To the right of the dropdown is a purple button labeled 'Agregar Etiqueta'. At the bottom, there are two buttons: 'Enviar Datos para Entrenamiento' (green) and 'Volver a seleccionar texto' (grey).



Se confirma el envío de datos para el entrenamiento

A form with a light green header bar labeled 'Datos llevados'. Below it, the title 'Paso 1: Ingresa el texto' is displayed. The instruction 'Ingresa el texto que quieres etiquetar:' is followed by a text input field containing 'Bonita casa con 3 habitaciones'. A blue button labeled 'Siguiete' is at the bottom.

los datos son llevados y se reinicia el proceso.

Evaluación del modelo

A screenshot of a web application interface. On the left is a dark sidebar with the 'SPACE' logo and a menu with options like 'Recolección de Datos', 'Completado de Información', 'Historial de Extracciones', 'DATOS', 'Análisis de Datos', 'Configuración', 'INTELIGENCIA ARTIFICIAL', 'Modelos de IA', 'Entrenamiento de modelo', 'Evaluación de Modelos', and 'Gestión de Datos'. The main area has a header with 'Inicio' and a settings/user icon. Below the header, the breadcrumb 'Inicio / Evaluación de Modelos' is shown. The main title is 'Evaluación del Modelo'. Under 'Paso 1: Ingreso de Descripción', there is an instruction: 'Para comenzar, ingresa la descripción de la propiedad. Este texto será evaluado para extraer información sobre características como baños, habitaciones, estacionamientos, área construida y terreno.' Below this is a text input field with the placeholder 'Ingresa la descripción de la propiedad...'. A blue button labeled 'Siguiete: Evaluar' is at the bottom.

Al igual que en paso anterior te sugiere ingresar datos para evaluar el modelo ej.: Bonita casa con 3 habitaciones

Evaluación del Modelo

Paso 1: Ingreso de Descripción

Para comenzar, ingresa la descripción de la propiedad. Este texto será evaluado para extraer información sobre características como baños, habitaciones, estacionamientos, área construida y terreno.

Bonita casa con 3 habitaciones

Siguiente: Evaluar

Se aceptara la evaluación

Evaluación del Modelo

Paso 2: Evaluando la Descripción

El sistema está evaluando la descripción ingresada. Haz clic en el botón para comenzar la evaluación.

Evaluar Descripción

Regresar

Muestra de resultados con el modelo para el reconocimiento de las entidades

Evaluación del Modelo

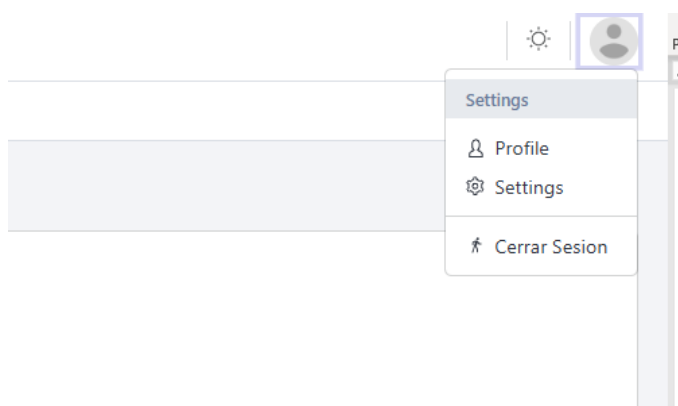
Paso 3: Resultados de la Evaluación

Los siguientes resultados han sido encontrados en la descripción proporcionada:

- **Baños:** No encontrado
- **Habitaciones:** 3 habitaciones
- **Estacionamientos:** No encontrado
- **Área Construida:** No encontrado
- **Terreno:** No encontrado

Volver a Evaluar

CERRAR SESION



Se podrá cerrar sesión y lo enviará al inicio o autenticación para ingresar al sistema



MANUAL DE TECNICO-

SISTEMA WEB SCRAPING DE RECONOCIMIENTO COMERCIAL RESPALDADO POR INTELIGENCIA ARTIFICIAL

AUTOR: JUAN ALBERTO SILVA CAYO
EMPRESA: BI SPACE SRL

Contenido

1. INTRODUCCION	2
2. OBJETIVO	2
3. HARDWARE Y SOFTWARE	2
4. ESTUCTURA Y ARQUITECTURA	3
5. DESPLIEGUE	5

1. INTRODUCCION

Este documento está dirigido a desarrolladores, ingenieros de software y personal técnico responsable de la implementación, configuración y mantenimiento del sistema. Diseñado para automatizar la recolección y análisis de información comercial desde diversas páginas web, este sistema emplea herramientas avanzadas de web scraping y procesamiento de datos.

2. OBJETIVO

El objetivo de este manual es proporcionar una guía técnica completa que permita a los especialistas en tecnología comprender y gestionar el Sistema Web Scraping de Reconocimiento Comercial. Este manual cubre:

- Requisitos técnicos necesarios para la instalación y configuración del sistema.
- Estructura y arquitectura del sistema, incluyendo el backend desarrollado en Flask y el frontend en React.
- Procedimientos para la implementación y despliegue en entornos locales o en la nube.

Este manual técnico está diseñado para garantizar la correcta operación del sistema.

3. HARDWARE Y SOFTWARE

El sistema requiere un servidor con características específicas para garantizar el funcionamiento eficiente y estable. El entorno se configuró utilizando las siguientes especificaciones:

- Sistema Operativo: Centos 7
- Procesador: 2vCPU
- Memoria RAM:GB
- Almacenamiento: 100GB

Con estas características se asegura que el servidor pueda manejar las cargas de trabajo relacionadas con el Web Scraping y el procesamiento de entidades relacionandas

Para el software el sistema requiere:

- Docker: para levantar el sistema
- Docker compose para la orquestación de servicios en contenedores
- NVM para gestionar las versiones de node
- Python 3 para levantar el backend
- Git para la clonación del repositorio
- Celery para gestionar tareas en segundo plano

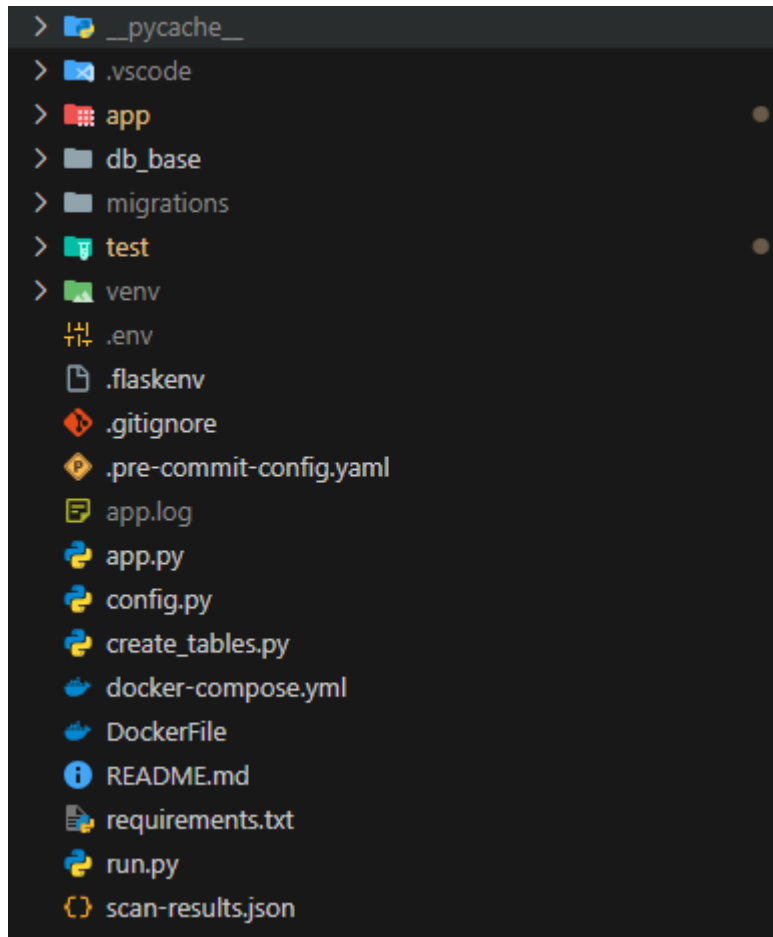
Finalizando con nano que nos ayudara a editar archivos de configuración en el servidor

4. ESTRUCTURA Y ARQUITECTURA

El backend esta conformado por llamadas a APIs protegidas por JWT para asegurar la confidencialidad de los datos la estructura de estas APIs es la siguiente

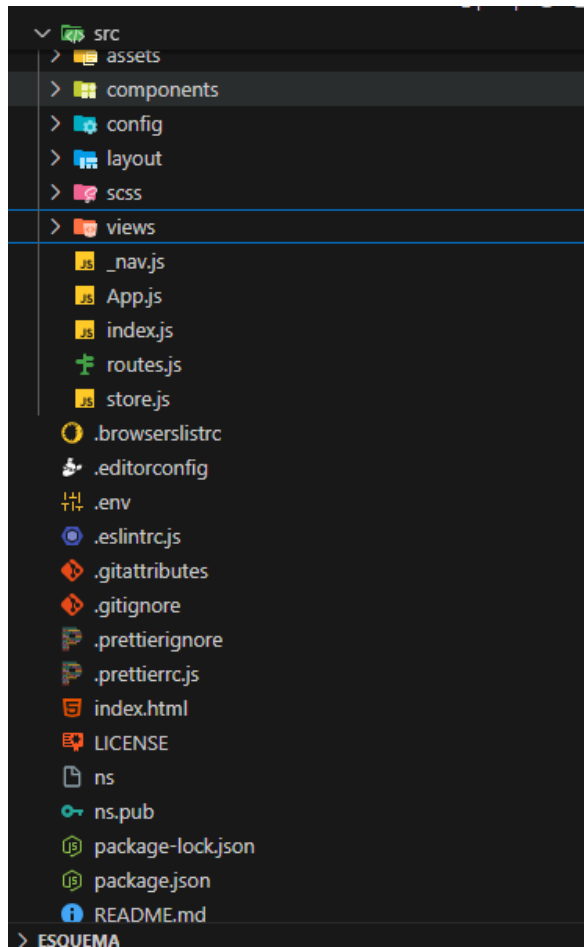
Función	API
data	/api/data/
login	/api/login/
dominio	/api/dominio/
webscraping	/api/webscraping
ubicaciones	/api/webscraping
propiedades	/api/propiedades

Estas son las rutas para poder interactuar con el backend considerado que todas tienen las funciones de GET, POST, PUT y DELETE con los cuales se espera el JWT para poder procesar la información



En donde `app` maneja los controladores, modelos y rutas para gestionar mejor la aplicación

Por otro lado está el frontend que tiene la siguiente estructura de archivos



Donde src maneja los componentes vistas y las configuraciones para su respectivo armado.

5. DESPLIEGUE

Para el despliegue es mejor preparar el servidor para que acepte entornos Docker para el despliegue, siendo HOSTINGER un servicio de host ya implementado en la empresa, se vio que ya esta implementado para el despliegue optimo en Docker

En una terminal nos conectamos con el servidor mediante ssh utilizando la ip pública y la contraseña que se proporcione al momento de tener la cuenta

```
o ssh root@ 31.220.31.160
```

instalamos nvm como gestor de versiones de node y usamos la versión 20

- `curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.5/install.sh | bash`
- Cargar NVM en la sesión actual `o source ~/.bashrc`
- Instalar Node.js versión 20 `o nvm install 20`

Clonamos el repositorio del backend y frontend

<https://gitlab.com/silvacayojuanalberto/web scraping-backend>

<https://gitlab.com/silvacayojuanalberto/web scraping-frontend>

Ejecutamos el docker-compose del backend habilitando el puerto 5000

```
pm2 start server.js --name backend --watch --env production
```

Con esto ya tenemos desplegado el backend con redis adicionando de forma local

Ahora configuramos el frontend

Ejecutamos

```
npm install
npm run build
pm2 serve build 3000 --name frontend -spa
```

Con esto tenemos desplegado el frontend

Para habilitar el escucha desde el servidor usaremos pm2 y nginx como gestor de proxy inverso

Abrimos la configuración de nginx

```
sudo nano /etc/nginx/conf.d/default.conf
```

configuramos como proxy inverso

```
server {
    listen 80;
    server_name yourdomain.com;

    location /api {
        proxy_pass http://localhost:5000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }

    location / {
        proxy_pass http://localhost:3000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }

    error_page 404 /404.html;
    location = /404.html {
        root /usr/share/nginx/html;
        internal;
    }
}
```

Reiniciamos el Nginx

```
sudo systemctl restart nginx
```

Habilitamos el FireWall con los puertos

```
sudo firewall-cmd --permanent --add-service=http  
sudo firewall-cmd --permanent --add-service=https  
sudo firewall-cmd --reload
```

```
sudo firewall-cmd --permanent --add-port=5000/tcp  
sudo firewall-cmd --permanent --add-port=3000/tcp  
sudo firewall-cmd --reload
```

Con esto aseguramos el despliegue oficial de la aplicación en el servidor de la empresa