UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN

TRABAJO DE INTEGRACIÓN CURRICULAR

TEMA:

"DESARROLLO DE UNA APLICACIÓN WEB CON AGENTES DE IA PARA LA EXTRACCIÓN Y ANÁLISIS COMPARATIVO TEMPORAL DE PRECIOS DE SAAS MEDIANTE EL PROTOCOLO MCP"

Trabajo de titulación previo a la obtención del título de Ingeniería en Tecnologías de la Información.

Línea de investigación: Desarrollo, aplicación de software y cyber security (seguridad cibernética).

AUTOR:

Yoleyda Dalyana Carvajal Jaramillo.

DIRECTOR:

MSc. Pablo Andrés Landeta López.

Ibarra – Ecuador 2026

UNIVERSIDAD TÉCNICA DEL NORTE
DIRECCIÓN DE BIBLIOTECA

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO	
APELLIDOS Y NOMBRES:	Carvajal Jaramillo Yoleyda Dalyana

DATOS DE LA OBRA	
TÍTULO:	"Desarrollo de una aplicación web con agentes de IA para la extracción y análisis comparativo temporal de precios de SaaS mediante el protocolo MCP"
AUTOR (ES):	Yoleyda Dalyana Carvajal Jaramillo
FECHA:	31/08/2026
SOLO PARA TRABAJOS DE GRADO	
PROGRAMA:	☒ **PREGRADO** ☐ **POSGRADO**
TITULO POR EL QUE OPTA:	Ingeniería En Tecnologías De La Información
ASESOR /DIRECTOR:	PhD. Cathy Pamela Guevara Vega / MSc. Pablo Andrés Landeta López

2. CONSTANCIAS

El autor (es) manifiesta (n) que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto la obra es original y que es (son) el (los) titular (es) de los derechos patrimoniales, por lo que asume (n) la responsabilidad sobre el contenido de la misma y saldrá (n) en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 31 días del mes de agosto de 2026

EL AUTOR:

Validar únicamente en FirmaEC.
Firmado electrónicamente por:
YOLEYDA DALYANA CARVAJAL JARAMILLO

……
Yoleyda Dalyana Carvajal Jaramillo

CERTIFICACIÓN DEL DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Ibarra, 31 de agosto de 2026.

MSc. Pablo Andrés Landeta López
DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICA:
Haber revisado el presente informe final del trabajo de Integración Curricular, el mismo que se ajusta a las normas vigentes de la Universidad Técnica del Norte; en consecuencia, autorizo su presentación para los fines legales pertinentes.

Validar únicamente en FirmaEC.
Firmado electrónicamente por:
PABLO ANDRES LANDETA LOPEZ

MSc. Pablo Andrés Landeta López

C.C: 1002161055

APROBACIÓN DEL COMITÉ CALIFICADOR

El Comité Calificado del trabajo de Integración Curricular “ Desarrollo de una aplicación web con agentes de IA para la extracción y análisis comparativo temporal de precios de SaaS mediante el protocolo MCP ” elaborado por Yoleyda Dalyana Carvajal Jaramillo , previo a la obtención del título del INGENIERIA EN TECNOLOGÍAS DE LA INFORMACIÓN , aprueba el presente informe de investigación en nombre de la Universidad Técnica del Norte:



MSc. Pablo Andrés Landeta López

C.C: 1002161055



MSc. Cathy Pamela Guevara Vega, PhD

C.C: 1002334835

DEDICATORIA

Esta tesis está dedicada con todo mi cariño y admiración a mi familia, porque este logro es tanto suyo como mío.

A mi madre, por su inmenso amor, su confianza y su apoyo constante. Espero que este logro llene su corazón de orgullo, del mismo modo que ella ha llenado el mío de valores y fortaleza.

A mi padre, quien con su incansable esfuerzo y sacrificio me ha enseñado el verdadero significado de disciplina y dedicación.

A Celeste, mi hermana y mejor amiga. Tu compañía ilumina mi vida y me impulsa a seguir adelante, con la firme esperanza de demostrarte que no existen límites cuando se persigue una meta.

A Ronald, por confiar en mí, ser mi soporte y celebrar cada pequeño avance como si fuera suyo.

Yoleyda Dalyana Carvajal Jaramillo

AGRADECIMIENTO

A Dios, por sostenerme, por darme la vida y la fortaleza, por no soltar mi mano y guiar cada paso de mi camino.

A mis padres Tatiana y Romel, por su amor incondicional y ser los pilares fundamentales de mi vida.

A los distinguidos docentes que me acompañaron a lo largo de esta etapa. Gracias por compartir sus conocimientos y ser mis guías.

Al MSc. Pablo Landeta, director de tesis, por su paciencia, apoyo, dedicación y valiosa orientación, que fueron fundamentales para mi formación académica.

A la Universidad Técnica del Norte y a la carrera de Tecnologías de la Información, por darme el espacio, las herramientas y los conocimientos necesarios para crecer como profesional.

A todas aquellas personas que, de una u otra manera, hicieron posible la realización de este trabajo de investigación.

Yoleyda Dalyana Carvajal Jaramillo

ÍNDICE DE CONTENIDOS

DEDICATORIA II

AGRADECIMIENTO III

ÍNDICE DE FIGURAS IX

ÍNDICE DE TABLAS X

RESUMEN XI

ABSTRACT XII

CAPÍTULO I

1.1 Planteamiento del Problema 1

1.2 Objetivo General 2

1.2.1 Objetivos Específicos 3

1.3 Alcance 3

1.4 Justificación 3

CAPÍTULO II

2.1 Antecedentes 5

2.2 Fundamentos del Software como Servicio 6

2.2.1 Definición y Modelo de Negocio SaaS 6

2.2.2 Estrategia de Precios en SaaS 7

2.2.3 La necesidad del Análisis Comparativo de Precios 7

2.3 Extracción de Datos Web y Automatización 8

2.3.1 Conceptos Fundamentales de Web Scraping. 8
2.3.2 Técnicas y Enfoques Avanzados 9
2.3.3 Desafíos Legales y Éticos 11
2.4 Inteligencia Artificial y Agentes de IA 13
2.4.1 Conceptualización de Agentes Inteligentes. 13
2.4.2 Agentes de IA para Automatización de Tareas 14
2.4.3 Aplicación en la Extracción de Datos 16
2.5 El Protocolo MCP 18
2.5.1 Origen y propósito del MCP 18
2.5.2 Arquitectura y Componentes Clave 18
2.5.3 Casos de Uso y Relevancia para el Proyecto 19
2.6 Análisis Comparativo Temporal de Datos 19
2.6.1 Series Temporales 20
2.6.2 Técnicas de Análisis y Visualización 21
2.6.3 Herramientas y Librerías. 22
2.7 Metodologías Ágiles de Desarrollo 22
2.7.1 La Metodología Scrum 23
2.7.2 Aplicación de Scrum en Proyectos de TI 25
2.8 Evaluación de Sistemas de IA y Calidad de Datos 26
2.8.1 Métricas para Evaluar Agentes de IA 26
2.8.2 Métricas para Evaluar la Calidad de los Datos Extraídos 28
CAPÍTULO III
3.1 Descripción general 31

3.2 Arquitectura del sistema 31
3.3 Implementación del Protocolo MCP 33
3.3.1 Servidor MCP (playwright mcp server) 33
3.3.2 Cliente MCP (MCPAgent) 33
3.4 Flujo Completo de Extracción de Precios 34
3.5 Stack tecnológico 35
3.5.1 Backend 35
3.5.2 Frontend 36
3.5.3 Base de datos e infraestructura 36
3.5.4 Inteligencia Artificial y protocolo MCP 37
3.6 Metodología de Desarrollo: Scrum 38
3.6.1 Roles del Equipo 38
3.6.2 Product Backlog 38
3.7 Planificación de Sprints 40
3.7.1 Sprint 1 — Infraestructura y API Base 40
3.7.2 Sprint 2 — Integración MCP y Claude Haiku 41
3.7.3 Sprint 3 — Frontend y Dashboard 42
3.7.4 Sprint 4 — Historial Comparativo y Mejoras 43
3.8 Resumen de Sprints 44
3.9 Estructura del Proyecto 45
3.10 Desafíos Técnicos y Soluciones Implementadas 47

3.10.1 Compatibilidad de Playwright con Windows y Uvicorn. 47
3.10.2 Optimización de Costos en el Modelo de Lenguaje 48
3.10.3 Orden de Carga de Variables de Entorno 48
3.10.4 Implementación MCP en Entorno Asíncrono Windows. 48
CAPÍTULO IV
4.1 Introducción 49
4.1.1 Uso de DeepEval. 49
4.1.2 Modelo Evaluador (GPT-4o-mini) 50
4.1.3 Evaluación Subjetiva y Objetiva. 51
4.2 Configuración de la Evaluación 52
4.3 Evaluación Subjetiva (Manual) 52
4.3.1 Metodología de Evaluación Manual 52
4.3.2 Resultados de la Evaluación Subjetiva 53
4.3.3 Análisis de la Evaluación Subjetiva 54
4.4 Evaluación Objetiva con DeepEval 55
4.4.1 Métricas de Evaluación Aplicadas 55
4.4.2 Resultados por SaaS 56
4.4.3 Resumen Estadístico por Métrica 57
4.4.4 Análisis de Resultados por Métrica. 59
4.4.5 Análisis por Categoría de SaaS. 61
4.5 Discusión de Resultados 63

4.5.1 Fortalezas del Sistema 63
4.5.2 Limitaciones identificadas. 64
4.5.3 Comparación con el Objetivo de la Investigación 64
4.5.4 Análisis inferencial complementario. 65
4.6 Resumen de Resultados 67
CONCLUSIONES 70
RECOMENDACIONES 72
BIBLIOGRAFÍA 74
ANEXOS 80

ÍNDICE DE FIGURAS

Figura 1 Árbol de problemas 2

Figura 2 Web Scraping (Adaptado de: [22]) 8
Figura 3 Diagrama de 3 fases con supervisión humana (Adaptado de [21]) 11
Figura 4 Arquitectura de un web crawler (Adaptado de [19]) 15
Figura 5 Modelo Extracción Planes (Adaptado de [23]) 17
Figura 6 Ciclo Scrum (Adaptado de [36]) 24
Figura 7 Flujo de trabajo de evaluación de MCPEval (Adaptado de [41]) 27

Figura 8 Arquitectura general de la página web 32
Figura 9 Proceso de desarrollo basado en Scrum 40
Figura 10 Organización del directorio 46
Figura 11 Organización del directorio 47

Figura 12 Scores promedio por métrica de evaluación 59
Figura 13 Faithfulness por SaaS evaluado 60
Figura 14 Planes de precios extraídos por SaaS 63
Figura 15 Dispersión: Planes extraídos vs. Faithfulness (Spearman $\rho = -0.22$, $p = 0.43$) 66
Figura 16 Perfil del agente MCP por métrica de evaluación 68
Figura 17 Página de inicio 80
Figura 18 Página de analizar 80
Figura 19 Página de dashboard 81
Figura 20 Historial comparativo con datos reales 81
Figura 21 Consola de logs mostrando la sesión MCP activa 82

ÍNDICE DE TABLAS

Tabla I Stack tecnológico del backend 35
Tabla II Stack tecnológico del frontend 36
Tabla III Infraestructura, IA y herramientas 37
Tabla IV Roles Scrum del proyecto 38
Tabla V Product Backlog del proyecto 39
Tabla VI Sprint 1 — Infraestructura base 41
Tabla VII Sprint 2 — MCP e Inteligencia Artificial 42
Tabla VIII Sprint 3 — Interfaz y visualización 43
Tabla IX Sprint 4 — Historial y mejoras finales 44
Tabla X Resumen de sprints y velocidad del equipo 45

Tabla XI Configuración de la evaluación 52
Tabla XII Criterios de evaluación 53
Tabla XIII Expectativa por SaaS 54
Tabla XIV Métricas de evaluación aplicadas 56
Tabla XV Resultados de evaluación por SaaS. 57
Tabla XVI Resumen estadístico por métrica. * Hallucination: score alto indica presencia de contradicciones 58
Tabla XVII Variables analizadas para el análisis de correlación 65
Tabla XVIII Resultados del análisis de correlación de Spearman 66
Tabla XIX Resumen final de resultados. * Hallucination: score alto indica contradicciones detectadas 69

RESUMEN

El presente trabajo describe el desarrollo e implementación de una aplicación web que automatiza la extracción, almacenamiento y análisis comparativo temporal de precios de servicios Software as a Service (SaaS) mediante agentes de inteligencia artificial coordinados con el Model Context Protocol (MCP). El problema abordado es la dificultad que enfrentan equipos de desarrollo y usuarios para monitorear de forma eficiente la alta volatilidad y opacidad de los precios en el mercado SaaS, proceso que actualmente se realiza de forma manual, ineficiente y propensa a errores. El sistema fue desarrollado bajo la metodología ágil Scrum en cuatro sprints de dos semanas, empleando Python 3.10, FastAPI como framework de backend, Streamlit como interfaz de usuario, PostgreSQL como base de datos relacional y el SDK oficial de MCP versión 1.26 para la coordinación real de agentes. La arquitectura implementa el Protocolo MCP con una estructura cliente-servidor donde el MCPAgent actúa como cliente MCP, playwright_mcp_server.py como servidor MCP y navigate_and_extract como herramienta expuesta que utiliza Playwright con Chromium headless para navegar páginas web dinámicas. El análisis semántico del contenido extraído se realiza mediante el modelo de lenguaje Claude 3 Haiku accedido a través de la plataforma OpenRouter. La evaluación del sistema se realizó con el framework DeepEval sobre 15 servicios SaaS reales, aplicando cinco métricas especializadas. Los resultados demuestran un desempeño excelente en Answer_Relevancy con 0.996, Contextual_Recall con 0.933 y Contextual_Precision con 0.867, confirmando que el agente recupera y responde con alta pertinencia y cobertura. La métrica Faithfulness obtuvo 0.222, limitación atribuible a la pérdida de estructura visual durante la extracción de texto plano. El análisis inferencial complementario mediante la correlación de Spearman entre el número de planes extraídos y la Faithfulness obtuvo $\rho = -0{,}22$ con $p = 0{,}43$, confirmando que las dificultades de extracción no son estadísticamente predecibles mediante indicadores simples de complejidad visual. El sistema extrajo exitosamente 49 planes de precios de los 15 SaaS evaluados con un promedio de 3.27 planes por servicio, almacenándolos con soporte para análisis comparativo temporal e identificación automática de cambios de precio.

Palabras clave: Model Context Protocol, agentes de inteligencia artificial, web scraping, SaaS, extracción de precios, DeepEval, FastAPI, Playwright, Claude Haiku

ABSTRACT

This work describes the development and implementation of a web application that automates the extraction, storage, and temporal comparative analysis of Software as a Service (SaaS) pricing through artificial intelligence agents coordinated via the Model Context Protocol (MCP). The problem addressed is the difficulty faced by development teams and users in efficiently monitoring the high volatility and opacity of prices in the SaaS market, a process currently performed manually in an inefficient and error-prone manner. The system was developed using the Scrum agile methodology across four two-week sprints, employing Python 3.10, FastAPI as the backend framework, Streamlit as the user interface, PostgreSQL as the relational database, and the official MCP SDK version 1.26 for real agent coordination. The implemented architecture follows the MCP Protocol with a client-server structure where MCPAgent acts as the MCP client, playwright_mcp_server.py as the MCP server, and navigate_and_extract as the exposed tool that uses Playwright with headless Chromium to navigate dynamic web pages. Semantic analysis of the extracted content is performed using the Claude 3 Haiku language model accessed through the OpenRouter platform. System evaluation was conducted using the DeepEval framework on 15 real SaaS services, applying five specialized metrics. Results demonstrate excellent performance in Answer_Relevancy at 0.996, Contextual_Recall at 0.933, and Contextual_Precision at 0.867, confirming that the agent retrieves and responds with high relevance and coverage. The Faithfulness metric obtained 0.222, a limitation attributable to the loss of visual structure during plain text extraction. A complementary inferential analysis using Spearman correlation between the number of extracted plans and Faithfulness obtained $\rho = -0{,}22$ with $p = 0{,}43$, confirming that extraction difficulties are not statistically predictable through simple visual complexity indicators. The system successfully extracted 49 pricing plans from the 15 evaluated SaaS services with an average of 3.27 plans per service, storing them with support for temporal comparative analysis and automatic price change detection.

Keywords: Model Context Protocol, artificial intelligence agents, web scraping, SaaS, price extraction, DeepEval, FastAPI, Playwright, Claude Haiku

CAPÍTULO I

INTRODUCCIÓN

1.1 Planteamiento del Problema

En la era digital, los servicios Software as a Service (SaaS) se han afianzado como una columna vertebral para empresas y usuarios, la relevancia de estos servicios SaaS en el mercado radica en su capacidad para transformar los modelos tradicionales de adquisición y uso de software, ofreciendo beneficios sustanciales como la reducción de costos, la escalabilidad y la flexibilidad operativa según *Corral de la Mata* [1].

Los servicios Software as a Service (SaaS) son modelos en el que el software no se instala en un computador, sino que son usados a través de internet, generalmente con una suscripción, esto permite eliminar la necesidad de comprar licencias de software, hardware costoso e infraestructura de TI, convirtiendo los gastos de capital en gastos operativos predecibles y recurrentes. Sin embargo, el mercado de estos servicios es dinámico, con muchas combinaciones en los precios y regularmente confuso.

El principal problema radica en la dificultad para obtener y comparar precios de manera eficiente; la rápida expansión del mercado SaaS ha generado una gran complejidad para los equipos de DevOps, quienes deben gestionar manualmente las estructuras de precios y planes, los cuales son modificados constantemente. por ello, la alta volatilidad y opacidad en la fijación de precios representan un desafío significativo para la toma de decisiones estratégicas, de acuerdo a *Cavero* [2].

Los métodos actuales para la extracción y análisis de precios no logran mitigar este problema de manera efectiva, los enfoques manuales son extremadamente ineficientes, propensos a errores humanos y no escalables para un mercado con miles de proveedores. Esto conduce a decisiones de adquisición subóptimas y a una desventaja competitiva para empresas que dependen de una evaluación precisa de los costos de software

En consecuencia, existe una necesidad latente en el mercado y es necesaria una solución automatizada que permita la extracción, almacenamiento y el análisis comparativo de precios de SaaS en tiempo real. En este contexto, se propone el desarrollo de una aplicación web que emplee agentes de inteligencia artificial mediante MCP. Esta herramienta busca superar las limitaciones actuales, ofreciendo una extracción de precios automatizada que los almacene de manera organizada y realice un análisis comparativo temporal, de esta manera no solo se optimizará el proceso de investigación para los usuarios, sino que también ofrecerá una visión clara y actualizada del mercado de precios de SaaS.

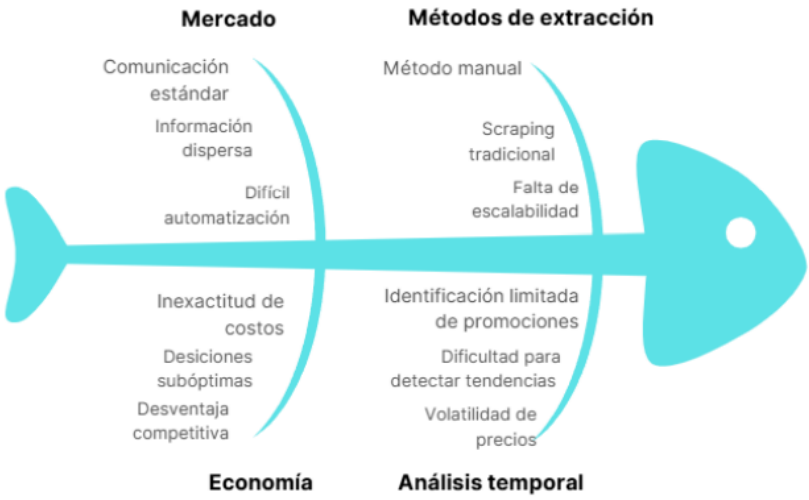


Figura 1.
Árbol de problemas

La Figura 1 ilustra los principales desafíos identificados en el proceso de obtención y análisis de información del mercado SaaS, organizados en cuatro dimensiones interconectadas:

- **Mercado:** Se observa una comunicación estándar y desestructurada, información dispersa en múltiples fuentes, y dificultades para la automatización de procesos.
- **Métodos de extracción:** Los enfoques actuales se limitan a métodos manuales y técnicas de scraping tradicionales, lo que genera falta de escalabilidad e inexactitud en la captura de datos.
- **Economía:** Las limitaciones técnicas derivan en consecuencias económicas directas, como la inexactitud en los costos, toma de decisiones subóptima y desventaja competitiva en el mercado.
- **Análisis temporal:** Finalmente, se identifican problemas en el análisis continuo, incluyendo identificación limitada de promociones, dificultad para detectar tendencias y alta volatilidad de precios no monitoreada adecuadamente.

1.2 Objetivo General

Desarrollar una aplicación web que emplee agentes de inteligencia artificial coordinados mediante el protocolo MCP para automatizar la extracción, almacenamiento estructurado y análisis comparativo temporal de precios de servicios SaaS seleccionados.

1.2.1 Objetivos Específicos

- Investigar el protocolo MCP y arquitecturas de agentes IA para extracción web de datos identificando componentes clave para la comunicación clienteservidor.
- Desarrollar una aplicación web para la extracción de precios de servicios SaaS utilizando agentes IA que se comunican con el protocolo MCP.
- Evaluar la aplicación web con métricas específicas para agentes de inteligencia artificial y de precisión de datos.

1.3 Alcance

El alcance de este proyecto se centra en el desarrollo de una aplicación web funcional que sirva como interfaz para los usuarios permitiendo visualizar los resultados de una extracción y realizar un análisis comparativo de precios, además se utilizarán agentes de inteligencia artificial usando el protocolo MCP que actúa como un intermediario estandarizado que centraliza y simplifica el acceso a diversas herramientas y sitios web de proveedores de SaaS para extraer información exacta sobre los precios [3].

Para el desarrollo de este proyecto se ha seleccionado la metodología Scrum, este es un marco de trabajo que está diseñado para manejar proyectos con una alta flexibilidad y adaptabilidad que encajan perfectamente con los tiempos de desarrollo de este, la metodología promueve la disposición para aceptar cambios en las características del producto durante el desarrollo, lo que permite ajustarse a nuevas condiciones o requerimientos de forma eficiente [4].

La aplicación estará diseñada para extraer datos de precios, planes y características clave de un conjunto predefinido y limitado de servicios SaaS, la selección inicial de estos servicios se basará en su popularidad y relevancia en el mercado, lo cual permite a los usuarios identificar tendencias, cambios y diferencias entre los distintos planes y proveedores.

Por último, se llevará a cabo una evaluación del sistema que mida la precisión de los datos extraídos y la eficiencia de los agentes de IA mediante métricas específicas.

1.4 Justificación

Actualmente, el desarrollo de software ha aumentado considerablemente debido a la administración, control, y gestión rápida que proporcionan, logrando así que la mayoría de las organizaciones opten por automatizar sus procesos y actividades utilizando estas herramientas [5].

El desarrollo de una aplicación web con agentes de inteligencia artificial para el análisis comparativo de precios de SaaS se justifica desde una perspectiva práctica, teórica y tecnológica,

el presente proyecto aborda una problemática real al tiempo que innova en la aplicación de tecnologías de vanguardia y contribuye al conocimiento en el campo de la inteligencia artificial.

Desde un punto de vista práctico, surge la necesidad de enfrentar la opacidad y la volatilidad del mercado de servicios SaaS. La tarea manual de monitorear y comparar los precios de estos servicios es un proceso tedioso, ineficiente y propenso a errores, lo que lleva a usuarios y empresas a tomar decisiones subóptimas y a incurrir en gastos innecesarios, en este contexto, la solución propuesta es una aplicación web automatizada que ofrece una herramienta tangible y eficaz para superar estas limitaciones. Al automatizar la extracción y el análisis de datos, se proporciona a los usuarios información transparente, actualizada y comparativa, permitiéndoles tomar decisiones financieras más inteligentes.[6]

Este trabajo investigativo espera aportar de manera significativa en el campo de la inteligencia artificial y la arquitectura de sistemas multi-agente explorando y demostrando la viabilidad de utilizar el Model Context Protocol como una tecnología emergente con una evolución natural de los grandes modelos de lenguaje, diseñada para otorgar a los agentes de IA la capacidad de interactuar directamente con aplicaciones en lugar de responder únicamente en texto [7].

A diferencia de otros protocolos, el uso de MCP permite a los agentes operar con un entendimiento compartido del problema, de esta forma se optimiza la extracción de información. El proyecto servirá como un caso de estudio empírico para validar la efectividad de esta arquitectura en un escenario real de extracción de datos web, contribuyendo al cuerpo de conocimiento sobre la implementación y el comportamiento de sistemas de agentes basados en protocolos de contexto compartido.

En el ámbito tecnológico, el proyecto representa una innovación al integrar una aplicación web moderna con la implementación de agentes de inteligencia artificial que operan bajo un protocolo de comunicación específico. Esta combinación va más allá del simple web scraping o extracción web automatizado, ya que los agentes, al utilizar MCP, pueden adaptarse a cambios en las estructuras de las páginas web y operar de manera más robusta e inteligente mostrando un modelo escalable y eficiente para la automatización de tareas de recolección y análisis de datos en entornos digitales.

CAPÍTULO II

MARCO TEÓRICO

2.1 Antecedentes

La irrupción de los Modelos de Lenguaje Grande (LLMs) y los Agentes de IA abrió la posibilidad de crear sistemas capaces de navegar e interpretar la web de forma autónoma, emulando el razonamiento humano para encontrar información específica. Esto representó un potencial salto tecnológico para automatizar tareas complejas como la extracción de información de sitios web dinámicos.

No obstante, los Agentes de IA basados únicamente en LLMs presentan limitaciones críticas para una aplicación confiable. En 2024, los autores *Fu y Xie* identificaron que estos modelos carecen de acceso a datos en tiempo real y son propensos a la alucinación, generando información plausible pero incorrecta, lo que los hace inherentemente no confiables para tareas que demandan precisión absoluta, como el análisis financiero o de precios [8].

Para superar estas limitaciones, a finales de 2024 surgió el MCP como un estándar abierto. Los autores *Li, Xu y Hong* lo definieron como un protocolo diseñado específicamente para estandarizar la comunicación entre aplicaciones y los LLMs, permitiéndoles acceder de forma segura y estructurada a herramientas y datos externos [9]. Básicamente, este protocolo actúa como un puente crítico, ya que posibilita que un agente de IA pueda utilizar herramientas especializadas, como por ejemplo un navegador web, a través de una interfaz estandarizada sin necesidad de configuraciones adicionales para cada caso.

La capacidad del MCP para orquestar la colaboración entre múltiples agentes especializados fue demostrada por *Tupe y Thube* en 2025. En su sistema de respuesta a incidentes TI, integraron el MCP con otros protocolos para coordinar agentes con roles distintos (diagnóstico, coordinación, resolución), mostrando cómo este protocolo facilita la creación de sistemas multi-agente cohesivos y eficientes [10].

La viabilidad de esta arquitectura para tareas de extracción y estructuración de datos fue confirmada de manera práctica por *Jin* y su equipo también en 2025. En su framework MedScreenDental, utilizaron el protocolo MCP para orquestar de forma dinámica múltiples LLMs y un módulo de OCR, logrando automatizar con alta precisión la extracción y estructuración de registros médicos a partir de capturas de pantalla [11].

Este caso de uso evidenció la capacidad del MCP para coordinar agentes y herramientas, mitigando las limitaciones de los LLMs aislados, en un ámbito diferente pero igualmente técnico, *Avila, Ilbay y Rivera* aplicaron el MCP para crear una interfaz conversacional entre un LLM y

el software de ingeniería estructural OpenSeesPy, su sistema logró una precisión superior al 99 porciento en análisis complejos, demostrando que el protocolo es capaz de garantizar resultados verificables y de alta confiabilidad en entornos donde el error no es una opción [12].

2.2 Fundamentos del Software como Servicio

El Software como Servicio es un modelo provisional en el cual las aplicaciones se ejecutan en infraestructura gestionada por el proveedor y se consumen remotamente mediante internet, *Según Suárez et al.*, el SaaS surge para responder a problemas propios de la gestión tradicional del software, entre ellos altos costos operativos, baja calidad del servicio y tiempos de respuesta poco eficientes, elementos que afectan directamente la satisfacción del cliente y la sostenibilidad del servicio [13], de acuerdo con este mismo documento, la adopción del SaaS permite mejorar la eficiencia, escalar servicios y simplificar la administración tecnológica dentro de las organizaciones.

Según el artículo *A Comparative Study on Three Selective Cloud Providers (2022)*, SaaS se distingue porque ofrece acceso remoto a aplicaciones que están alojadas en plataformas cloud, lo cual reduce bastante la necesidad de instalar software en infraestructura local y, al mismo tiempo, facilita la estandarización del acceso mediante interfaces web [14]. Esta característica, sin duda, ha impulsado su integración en diversos sectores donde se requiere disponibilidad continua y una escalabilidad que pueda adaptarse de forma dinámica a las necesidades cambiantes.

2.2.1 Definición y Modelo de Negocio SaaS

Según *Suárez et al.*, la arquitectura SaaS se fundamenta en el principio de multitenencia, donde una misma aplicación atiende a múltiples clientes de forma aislada y segura, permitiendo reducir costos y facilitar la administración centralizada del servicio [13], este principio arquitectónico es clave para entender por qué el SaaS es altamente escalable y eficiente en entornos empresariales.

De acuerdo con el artículo *Pricing-driven Development and Operation of SaaS*, la operación de plataformas SaaS incorpora mecanismos continuos de monitoreo, aprovisionamiento automático de recursos y ajuste dinámico según la demanda, lo cual mejora la confiabilidad del sistema y la calidad percibida por el usuario [15]. Estos autores destacan que esta automatización permite mantener niveles de servicio estables aun cuando varían las cargas de trabajo, consolidando al SaaS como un modelo robusto para entornos críticos.

Por su parte, el mismo estudio *A Comparative Study on Three Selective Cloud Providers* señala que la arquitectura SaaS se ve beneficiada por la infraestructura cloud que tiene por debajo, ya que los proveedores garantizan altos índices de disponibilidad y tiempos de respuesta optimizados mediante centros de datos distribuidos en distintas partes del mundo [14]. Esto demuestra que el rendimiento de SaaS depende no solamente del software en sí mismo, sino también del ecosistema tecnológico en el que se ejecuta.

2.2.2 Estrategia de Precios en SaaS

El modelo de negocio del SaaS se basa principalmente en la comercialización mediante suscripciones. Según el artículo *Subscription-Based Pricing Models in SaaS: An Overview (2022)*, los modelos de precios más utilizados incluyen el pago por usuario, el pago por uso, los niveles de servicio y los esquemas freemium [16]. De acuerdo con lo que plantea esta investigación, este tipo de modelos permiten ajustar el costo del servicio al perfil de consumo que tenga cada cliente, lo cual lo vuelve mucho más flexible en comparación con la compra tradicional de licencias perpetuas.

Según el autor *García-Fernández* en el artículo *Automated Analysis of Pricings in SaaS-based Information Systems (2025)*, el análisis de precios se ha convertido en una actividad crítica para los proveedores SaaS debido a la variabilidad del mercado y a la necesidad de mantener competitividad [17]. El autor sostiene que la automatización del análisis tarifario permite identificar patrones en el comportamiento de precios, ajustar las estrategias comerciales y anticipar reacciones del mercado.

Asimismo, de acuerdo con la investigación *Pricing-driven Development and Operation of SaaS*, los precios no solo determinan la rentabilidad del servicio, sino que influyen directamente en decisiones de diseño, operación y escalabilidad del sistema [15], por ello, los modelos modernos integran métricas de consumo real, costos operativos y valor percibido para definir tarifas más precisas y sostenibles

2.2.3 La necesidad del Análisis Comparativo de Precios

El análisis comparativo en los servicios SaaS resulta fundamental principalmente por la gran diversidad de proveedores que existen hoy en día y por la continua modificación de sus precios. De acuerdo con el artículo *A Comparative Study on Three Selective Cloud Providers*, se pueden encontrar diferencias bastante relevantes entre un proveedor y otro en términos de disponibilidad, latencia, escalabilidad y costo. Por esta razón, la comparación sistemática entre servicios termina siendo esencial para poder tomar decisiones informadas y acertadas [14].

Además, según con el articulo *Automated Analysis of Pricings in SaaS-based Information Systems*, la recopilación continua de precios es necesaria para detectar variaciones temporales, cambios de políticas comerciales y tendencias del mercado [17], los autores enfatizan que este análisis temporal permite comprender la evolución del servicio y anticipar fluctuaciones que pueden afectar a los usuarios.

Según *Suárez et al*., la evaluación sistemática del servicio es clave para asegurar que los sistemas de software respondan a las necesidades del cliente y mantengan altos niveles de calidad en el tiempo [13], esto refuerza la idea de que tanto los usuarios como las organizaciones requieren mecanismos automatizados que permitan monitorear, comparar y evaluar la evolución de los precios y características de los servicios SaaS.

En este contexto, la implementación de agentes de inteligencia artificial que sean capaces de

extraer y analizar datos de precios de forma continua se convierte en una herramienta esencial, sobre todo si se considera que se está hablando de mercados digitales altamente competitivos, donde las tarifas suelen cambiar con bastante rapidez y sin previo aviso.

2.3 Extracción de Datos Web y Automatización

La extracción automatizada de datos a partir de fuentes web constituye un pilar fundamental para los sistemas de inteligencia de mercado y también para el análisis comparativo a lo largo del tiempo, tal como se plantea en esta tesis [18]. Sin este tipo de automatización, resultaría muy difícil mantenerse al día con la gran cantidad de información que se genera constantemente.

Este proceso, conocido como web scraping o extracción web, permite la recopilación estructurada de información volátil y actualizada frecuentemente en la Web, tal como los precios, características y términos de servicio del software bajo el modelo SaaS [19].

Esta sección expone los conceptos fundamentales, las técnicas y enfoques avanzados que habilitan la automatización robusta, y los imperativos legales y éticos que enmarcan esta práctica [20],[21].

2.3.1 Conceptos Fundamentales de Web Scraping

El web scraping se define como la técnica automatizada de recuperar y analizar (parsing) contenido de la World Wide Web para transformar información no estructurada o semi-estructurada, predominantemente en formato HTML, en datos estructurados aptos para su almacenamiento y análisis en bases de datos u hojas de cálculo [18]. Tal como se muestra en la Figura 2.



Figura 2.
Web Scraping (Adaptado de: [22])

Como bien lo destaca *Sirisuriya* [18], el objetivo final de este tipo de procesos es poder su-

perar las limitaciones que trae consigo el acceso manual, el cual suele resultar tedioso y, además, es bastante propenso a errores cuando se manejan grandes volúmenes de datos. De ahí la importancia de automatizar la conversión de la información que se encuentra disponible en sitios web hacia formatos mucho más útiles y manejables, como pueden ser CSV, XML o directamente bases de datos relacionales.

El proceso se basa en dos componentes principales interconectados: el rastreador web (web crawler o spider) y el extractor de datos [19]. El crawler es un bot que navega de forma metódica y automatizada por la web, siguiendo hipervínculos de manera recursiva a partir de una lista de URLs semilla, descargando las páginas para su posterior procesamiento [19].

Una vez que se ha obtenido el contenido de la página, el extractor procede a aplicar ciertas técnicas de análisis sintáctico (parsing) con el fin de identificar y luego extraer los elementos que realmente interesan. Todo esto se logra mediante una inspección detallada de la estructura HTML, para lo cual se utilizan selectores como XPath o CSS. Dichos selectores son los que permiten localizar aquellas etiquetas y atributos específicos donde se encuentran los datos que se desean recuperar [20].

Un concepto fundamental derivado de este proceso es el ”wrapper” o envoltorio, los autores *Raheja y Katiyar* [20] definen un wrapper como un programa o plantilla específica desarrollada para extraer información de páginas web con una estructura particular, traduciendo los datos relevantes incrustados en el HTML a un esquema relacional o formato estructurado. Dichos autores [20] señalan que, si bien los wrappers pueden generarse manualmente, los enfoques de inducción (wrapper induction) o semi-automáticos son preferibles para balancear precisión y esfuerzo de desarrollo.

Esta técnica es especialmente relevante cuando no existe una API (Interfaz de Programación de Aplicaciones) pública y unificada para acceder a los datos, convirtiendo al scraping en la única vía viable para obtener información homogénea y comparable, como sucede frecuentemente en la comparación de precios de SaaS [18], [21].

2.3.2 Técnicas y Enfoques Avanzados

Para abordar la complejidad y dinamismo de la web moderna, las técnicas de web scraping han evolucionado significativamente, existiendo un amplio espectro que va desde métodos simples hasta enfoques sofisticados guiados por inteligencia artificial [21].

Una clasificación amplia, presentada por *Sirisuriya* [18], incluye técnicas como la copia manual, la coincidencia de expresiones regulares, la programación HTTP, el parsing de HTML y DOM, y el uso de software especializado. Para sitios web dinámicos donde el contenido se genera en el lado del cliente mediante JavaScript, donde técnicas como el DOM parsing son esenciales.

Esta técnica implica, en la práctica, embeber un navegador web real o utilizar lo que se conocen como navegadores headless, entre los cuales se encuentran herramientas como Puppeteer o Playwright. Dichos navegadores tienen la capacidad de ejecutar scripts, renderizar la página por

completo y también simular diversas interacciones propias de un usuario, tales como hacer clics, desplazarse por la página o llenar formularios, todo ello antes de proceder con la extracción de los datos [18], [19].

Los autores *Vicente y Lera* [19] explican este desafío con un ejemplo bastante ilustrativo: el caso de un número telefónico que solo se llega a mostrar después de que el usuario hace clic en algún botón o elemento de la página. Esto significa que el scraper no solo debe limitarse a leer el HTML inicial, sino que también necesita identificar cuál es la petición HTTP subyacente y luego ejecutarla para poder acceder al contenido que realmente interesa.

La automatización robusta y escalable enfrenta el desafío de los mecanismos anti-scraping, enfoques para mitigar detecciones incluyen la rotación de user agents y direcciones IP, la incorporación de retardos aleatorios (throttling) entre solicitudes, y el respeto al archivo robots.txt [19], [21]. *Krotov, Johnson y Silva* [9] enfatizan que el acceso ético y de bajo impacto no es solo una cuestión técnica, sino una responsabilidad del investigador para evitar sobrecargar los servidores y respetar la propiedad de la fuente.

La arquitectura de un sistema de extracción que sea avanzado suele tener un enfoque modular. En este sentido, *Henríquez y Guzmán* [23] describen un modelo que incluye varios componentes bien diferenciados: uno para la extracción propiamente dicha (al que llaman wrapper), otro para la navegación por categorías (crawler), un módulo para el pre-procesamiento de datos (que se encarga de tareas como limpieza y clasificación) y, finalmente, un componente para construir un resultado ya estructurado y listo para ser utilizado.

De manera similar, el Relational Web Wrapper presentado por *Raheja y Katiyar* [19] demuestra un enfoque de inducción de wrappers para extraer metadatos (título, descripción, palabras clave) de páginas web de manera eficiente, mostrando métricas de desempeño en tiempo y precisión.

La frontera más avanzada la ocupan las plataformas de agregación vertical y los agentes basados en IA, *Sirisuriya* [18] describe las plataformas verticales como sistemas que despliegan múltiples bots especializados en un dominio, como bienes raíces o comparación de productos, para escalar la extracción a miles de sitios web de manera automatizada, abordando la llamada ”larga cola” de fuentes.

Krotov et al. [21] señalan la evolución hacia herramientas de ”point and clickz plataformas en la nube que simplifican el proceso, aunque a menudo requieren supervisión humana y conocimientos técnicos para tareas complejas. La integración de agentes de IA, capaces de interpretar la semántica de una página y adaptarse a cambios menores en su estructura, representa el siguiente paso hacia sistemas de extracción más autónomos y resilientes [21], [23].

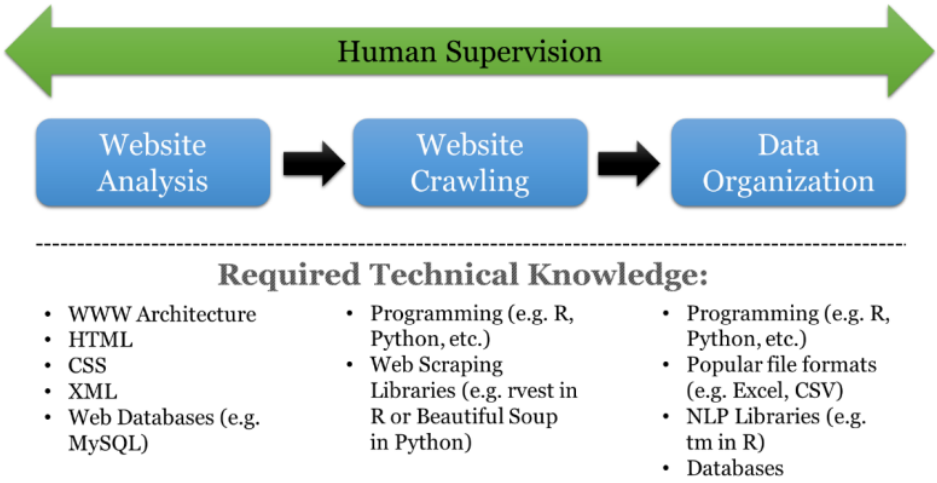


Figura 3.
Diagrama de 3 fases con supervisión humana (Adaptado de [21])

La Figura 3 describe el modelo trifásico de Krotov, Johnson y Silva [21] sobre la extracción de precios SaaS, ilustra el flujo completo desde el análisis inicial de sitios web, donde identifican estructuras HTML y contenido dinámico típico de plataformas SaaS, hasta el rastreo automatizado ético que respeta limitaciones técnicas, culminando en la organización de datos para el análisis temporal.

2.3.3 Desafíos Legales y Éticos

La práctica del web scraping se desarrolla en un terreno legal, ético complejo y en evolución, donde la jurisprudencia varía entre jurisdicciones y los marcos éticos deben ser considerados proactivamente [21], [23].

Desde una perspectiva legal, hay que aclarar que no existe una legislación específica universal para el web scraping. Lo que se aplica en la práctica son marcos legales de carácter más general. *Krotov et al.* [21] identifican los principales problemas legales que pueden surgir, los cuales son los siguientes:

- **Acceso y uso ilegal de datos**: este punto es frecuentemente litigado bajo leyes como la Computer Fraud and Abuse Act (CFAA) en Estados Unidos. Dentro de este marco, la definición de lo que se considera .ªcceso no autorizadoresulta ser clave para determinar si hay o no infracción.
- **Violación de contrato**: esta falta se vincula directamente con los Términos de Servicio (ToS) del sitio web, documentos que normalmente establecen restricciones claras acerca de cómo puede o no emplearse la información publicada en la plataforma.
- **Infracción de derechos de autor**: este tipo de problema se aplica especialmente cuando se copia y se vuelve a publicar contenido que está protegido sin que haya una transformación sustancial del mismo, o cuando no se puede amparar bajo la figura del üso justo.º fair use.

- **Intrusión a la propiedad**: ocurre básicamente si la actividad de extracción llega a causar algún tipo de daño material al servidor del cual se están obteniendo los datos.
- **Revelación de secretos comerciales**: este es otro riesgo que no puede pasarse por alto, en especial cuando la información extraída contiene datos sensibles o estratégicos de una empresa.

El autor *Sirisuriya* [18] señala que casos muy conocidos, como el de hiQ Labs contra LinkedIn, han ayudado a establecer un criterio importante: el scraping de datos que están disponibles de forma pública, siempre que no se estén sorteando barreras de autenticación (como contraseñas o sistemas de inicio de sesión), probablemente no viola la CFAA. Este principio resulta crucial para proyectos orientados a la comparación de precios, ya que otorga un cierto respaldo legal siempre que se actúe dentro de estos límites.

Éticamente, la práctica responsable descansa sobre varios pilares:

Primero, el respeto a las restricciones técnicas y de acceso, aunque su cumplimiento no siempre es obligatorio legalmente, *Krotov et al.* [21] argumentan que seguirlo es una norma social básica y un acto de buena fe hacia el propietario del sitio.

Segundo, la consideración del impacto en la fuente, evitando sobrecargar los servidores y, potencialmente, disminuir el valor del servicio al eludir su interfaz pública y mecanismos de monetización [21], [23].

Tercero, y quizás el punto más crítico de todos, está relacionado con la protección de la privacidad de las personas y los derechos que tienen los sujetos que participan en una investigación. El estudio de *Krotov et al.* [21] advierte que los datos obtenidos mediante scraping, sobre todo cuando provienen de redes sociales o foros públicos, pueden terminar desanonimizando a ciertos individuos o incluso ser utilizados para tomar decisiones que más bien perpetúan sesgos y prácticas discriminatorias. Todo esto, sin duda, va en contra de principios éticos fundamentales como la autonomía de las personas y el principio de no maleficencia (es decir, no hacer daño).

Vicente y Lera [19] coinciden con esta postura, y señalan que, en muchos casos, el problema principal no está tanto en la extracción como tal, sino más bien en el uso que se le llega a dar a los datos una vez que ya han sido obtenidos. Dicho uso, advierten, debe respetar marcos normativos como el Reglamento General de Protección de Datos (GDPR), el cual impone condiciones claras sobre el tratamiento de información personal.

Finalmente, la transparencia y la calidad de los datos son imperativos. *Henríquez y Guzmán* [23] subrayan la necesidad de etapas de pre-procesamiento y limpieza para eliminar ruidoz garantizar la veracidad de la información. *Krotov et al*. [9] concluyen que los investigadores deben reflexionar críticamente sobre el daño potencial de sus acciones, implementar salvaguardas técnicas, como la ofuscación de datos personales y, cuando sea posible, buscar el consentimiento de las fuentes.

En el contexto particular de esta tesis, donde los agentes de IA se utilizan para automatizar la extracción de datos con fines de análisis comparativo a lo largo del tiempo, resulta fundamental que todos estos principios éticos y legales no queden solo como buenas intenciones, sino que

estén integrados desde el propio diseño del sistema. De esta manera, se debe garantizar que tanto el protocolo MCP como los agentes que lo implementan operen siempre dentro de un marco de cumplimiento que sea responsable y, a la larga, sostenible.

2.4 Inteligencia Artificial y Agentes de IA

Los agentes de Inteligencia Artificial se han consolidado como herramientas fundamentales para la transformación digital en diversos sectores, debido a su capacidad para realizar tareas de forma autónoma, aprender de los datos y mejorar procesos [24].

Esta sección tiene como propósito establecer los fundamentos teóricos relacionados con los agentes de IA, abordando desde su conceptualización más básica hasta su aplicación concreta en la automatización de tareas. Se prestará especial atención a su uso en los procesos de extracción de datos, justamente por ser esta un área central dentro de la presente investigación.

2.4.1 Conceptualización de Agentes Inteligentes

Un agente inteligente puede conceptualizarse como una entidad de software que funciona de manera autónoma. Básicamente, lo que hace es percibir su entorno a través de sensores o entradas de datos, y luego actúa sobre dicho entorno mediante actuadores o salidas, todo ello con el fin de dirigir sus acciones hacia el logro de ciertos objetivos específicos, y se espera que lo haga de forma racional [25].

Esta definición, que es ampliamente aceptada dentro de la literatura especializada, pone un énfasis especial en dos características que se consideran distintivas de los agentes: la autonomía y la orientación hacia el logro de objetivos. Tal como lo señalan *Coloma Garofalo et al.*, el agente se desenvuelve en un entorno en el cual es capaz de actuar, y sus acciones pueden llegar a producir cambios dentro de ese entorno; dichos cambios, a su vez, pueden ser detectados por el propio agente [25]. En esencia, esta interacción cíclica que combina percepción, procesamiento y acción constituye el núcleo fundamental de su operación.

La arquitectura y capacidades de un agente pueden variar significativamente. El estudio de *Coloma Garofalo et al*. destaca un conjunto de propiedades clave que los definen: [25]

- **Autonomía**: se refiere a la capacidad que tiene el agente para actuar por sí mismo, sin depender de una intervención constante por parte de un operador humano.
- **Reactividad**: es la capacidad de responder de manera oportuna ante los cambios que se presentan en el entorno donde el agente se desenvuelve.
- **Proactividad**: tiene que ver con la iniciativa propia del agente para generar y luego perseguir metas, sin limitarse únicamente a reaccionar ante estímulos externos.

- **Habilidad social**: consiste en la capacidad de comunicarse e interactuar de forma efectiva, ya sea con otros agentes o directamente con seres humanos.
- **Capacidad de razonamiento y comportamiento adaptativo**: se relaciona con la posibilidad de aprender a partir de la experiencia acumulada, lo cual le permite ajustar su comportamiento según las circunstancias.

No todos los agentes poseen la totalidad de estas propiedades; su diseño y subconjunto de capacidades dependen del dominio de aplicación específico. [25].

En el contexto de las aplicaciones más avanzadas, los agentes han logrado evolucionar de manera significativa: pasaron desde sistemas relativamente simples, como los tutores inteligentes, hasta convertirse en entidades capaces de interactuar de forma mucho más natural y sofisticada [24]. Esta evolución ha sido posible gracias a los avances en áreas como el aprendizaje automático (machine learning) y el aprendizaje profundo (deep learning). Hay que señalar que ambos son subcampos de la inteligencia artificial que permiten que las máquinas aprendan directamente a partir de datos, identifiquen patrones complejos y tomen decisiones con una mínima intervención humana programada [26].

Como bien lo indica *Rouhiainen*, la inteligencia artificial puede entenderse como ”la habilidad que tienen los ordenadores para realizar actividades que normalmente requerirían de inteligencia humana”, lo cual básicamente los capacita para analizar grandes volúmenes de información y, al mismo tiempo, reducir el margen de error en tareas que son repetitivas [26]. Por lo tanto, un agente inteligente moderno lo que hace es integrar justamente estas capacidades de aprendizaje computacional, con el objetivo de mejorar su propio desempeño y su capacidad de adaptación a diferentes situaciones.

2.4.2 Agentes de IA para Automatización de Tareas

La automatización de tareas rutinarias, administrativas o complejas es una de las aplicaciones más destacadas y transformadoras de los agentes de IA, estos sistemas pueden liberar a los humanos de cargas de trabajo repetitivas, permitiéndoles enfocarse en actividades de mayor valor que requieren creatividad, pensamiento crítico y supervisión estratégica [24], [27].

La investigación de *Suescum Coelho et al.*, centrada en el ámbito educativo, identifica la automatización como un beneficio principal, los autores señalan que los agentes de IA pueden encargarse de tareas mecánicas como la calificación de exámenes, la gestión de horarios o la provisión de respuestas a consultas frecuentes, liberando tiempo docente para interacciones pedagógicas más enriquecedoras y un seguimiento individualizado del estudiante [24].

Este principio es extrapolable a otros dominios, como el empresarial o el de gestión de datos, donde la automatización de flujos de trabajo conduce a ganancias significativas en eficiencia y reducción de costos operativos. La efectividad de un agente en la automatización depende críticamente de su capacidad para aprender y adaptarse, como explica *Rouhiainen*, el aprendizaje automático permite a los dispositivos obtener experiencia a partir de su uso, ofreciendo experiencias personalizadas y mejorando sus predicciones o acciones con el tiempo [26].

Esto se logra mediante algoritmos que aprenden de patrones en los datos, los cuales pueden ser estructurados, valores numéricos, fechas, o no estructurados, texto e imágenes, siendo estos últimos predominantes y de gran valor potencial [26]. Un agente de automatización, por tanto, no solo ejecuta reglas predefinidas, sino que optimiza su comportamiento basándose en el análisis continuo de la información que procesa.

Sin embargo, hay que reconocer que la implementación exitosa de este tipo de agentes no está exenta de enfrentar diversos desafíos. Más allá de las barreras puramente técnicas, el estudio de *Suescum Coelho et al.* advierte también sobre la necesidad de capacitar adecuadamente a los usuarios finales. En el caso de su investigación, estos eran docentes, y se destaca que ellos deben desarrollar ciertas habilidades para poder integrar este tipo de herramientas en sus procesos cotidianos, así como también comprender tanto sus capacidades como sus limitaciones [24].

Asimismo, surgen consideraciones éticas cruciales, como los sesgos algorítmicos cuando los datos de entrenamiento replican desigualdades existentes y los riesgos a la privacidad y seguridad de los datos manejados por los agentes, requiriendo marcos regulatorios sólidos para un uso responsable [24].

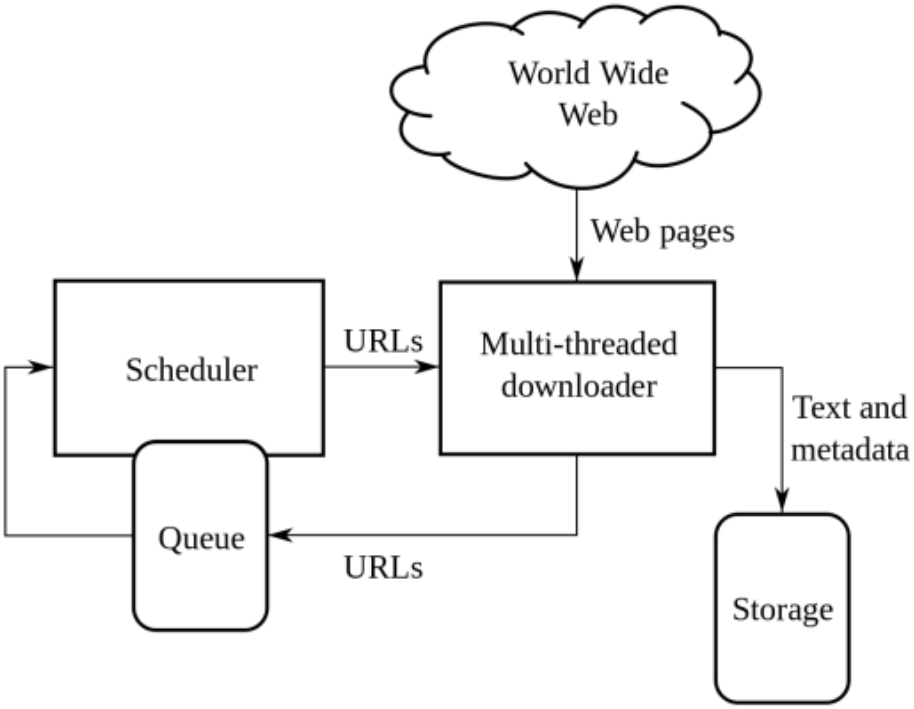


Figura 4.
Arquitectura de un web crawler (Adaptado de [19])

La Figura 4 detalla el flujo operativo del sistema, expandiendo la arquitectura básica de crawler de Vicente Stenhouse y Lera Castro (2018). Muestra cómo un dispatcher coordina mediante MCP múltiples agentes especializados por proveedor SaaS, siguiendo el principio de especialización identificado por Sirisuriya [18]. El flujo integra extracción distribuida, normalización de formatos diversos, alineación temporal precisa y almacenamiento en series históricas, creando un pipeline robusto para el análisis comparativo continuo.

2.4.3 Aplicación en la Extracción de Datos

La extracción de datos, particularmente de fuentes digitales no estructuradas y dinámicas como la web, constituye un dominio de aplicación idóneo para los agentes de IA. Esta idoneidad se fundamenta en las capacidades nucleares de estos a gentes: [25], [26].

- La autonomía con la que cuentan para ejecutar tareas repetitivas, liberando así a los usuarios de esa carga operativa constante.
- La adaptabilidad que demuestran al enfrentarse a entornos que cambian con frecuencia, lo cual les permite seguir funcionando sin necesidad de ajustes manuales continuos.
- La habilidad que poseen para procesar y, además, darle sentido a grandes volúmenes de información que de otra forma serían difíciles de gestionar en tiempos razonables.

El éxito de un agente cuando se enfrenta a esta tarea tan especializada depende de manera crítica de su capacidad para percibir, interpretar y comprender la información que se encuentra disponible en su entorno digital. *Rouhiainen* [26] subraya que las tecnologías de inteligencia artificial contemporáneas han logrado alcanzar hitos bastante significativos en lo que respecta a la emulación de capacidades humanas que resultan clave para esta tarea. Entre ellas se puede mencionar, por un lado, la visión artificial, que permite ver y analizar componentes gráficos e interfaces, y por otro lado, de manera aún más crucial, el procesamiento del lenguaje natural (PLN), que es lo que posibilita entender el verdadero significado semántico del texto.

Un agente que haya sido diseñado específicamente para la extracción de precios de SaaS debe, entonces, integrar todas estas capacidades de las que se ha venido hablando. Porque necesita navegar por estructuras HTML que suelen ser muy variables, identificar y aislar aquellos elementos de datos que sean relevantes ,por ejemplo, tablas de precios, descripciones de los distintos planes o condiciones de licencia y, finalmente, llevar a cabo una extracción que sea precisa y, lo que es más importante, contextualizada. De esta forma, se logra superar la rigidez tan característica de los scripts de scraping tradicionales, que suelen fallar ante el más mínimo cambio en la página.

La viabilidad y eficacia de emplear agentes de IA para tareas de búsqueda, síntesis y estructuración de información a partir de fuentes dispersas queda respaldada empíricamente por estudios aplicados, la investigación de *Berrios Zepeda y Márquez Mora* [27] demostró de manera concluyente que un procedimiento metodológico asistido por un agente de IA generativa basado en modelos de lenguaje grande (LLM) resultaba significativamente más efectivo que los métodos tradicionales en fases críticas de trabajo con información.

El estudio que llevaron a cabo estos autores permitió evidenciar mejoras bastante sustanciales en aspectos clave como la identificación de vacíos dentro de la literatura existente, la localización y posterior selección de referencias bibliográficas que resultaban pertinentes, y también la estructuración coherente del planteamiento de una investigación [27]. Ahora bien, estos hallazgos no se quedan únicamente en el ámbito educativo, sino que más bien ofrecen un precedente sólido y aprovechable para lo que sería la automatización de flujos de trabajo análogos, aplicados esta vez a la extracción de datos de tipo comercial. En ese contexto, el agente actuaría

como una especie de investigador autónomo, con la capacidad de rastrear, filtrar, comparar y, finalmente, consolidar información proveniente de múltiples fuentes en línea.

Cuando se habla de problemas de extracción a gran escala, es decir, aquellos que requieren concurrencia, tolerancia a fallos y cierto grado de especialización, la arquitectura de Sistemas Multi-Agente (SMA) se presenta como la solución más robusta desde el punto de vista arquitectónico. Tal como lo definen *Coloma Garofalo et al.* [25], un SMA no es más que un sistema compuesto por múltiples agentes inteligentes que interactúan entre sí con el propósito de resolver problemas que, de otra forma, serían difíciles o incluso imposibles de abordar si se dependiera de un único agente actuando de manera individual.

En el contexto específico de un monitor de precios de SaaS, un SMA podría estar conformado por todo un ecosistema de agentes, cada uno con sus propias funciones especializadas. Por mencionar algunos ejemplos: habría un agente rastreador, que se encargaría de la gestión de los crawlers y de la descarga de las páginas web; un agente analizador, cuyo trabajo sería la interpretación tanto estructural como semántica del contenido obtenido; un agente normalizador, que se ocuparía de estandarizar los datos ya extraídos para darles un formato consistente; y, finalmente, un agente coordinador, cuya labor sería orquestar toda la tarea a nivel global, gestionar la comunicación entre los distintos subsistemas y, al final del proceso, consolidar la información definitiva.

Este paradigma, como señalan los autores, proporciona una excelente capacidad de integración, reconfiguración dinámica y robustez [25], características esenciales para operar en el entorno impredecible y en constante evolución de la web.

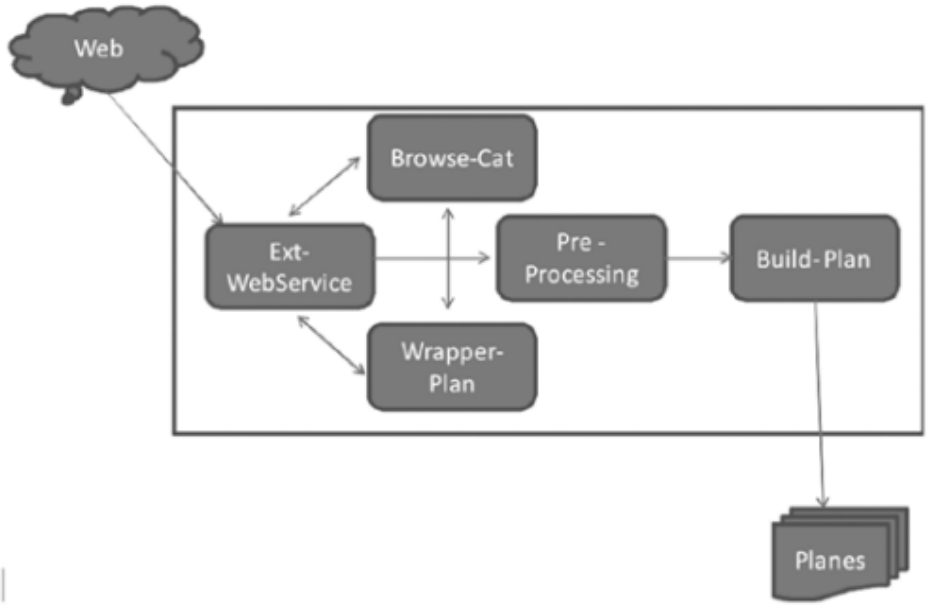


Figura 5.
Modelo Extracción Planes (Adaptado de [23])

La Figura 5 presenta una arquitectura un poco más expandida, que ya incorpora capacidades de inteligencia artificial y, además, el protocolo MCP. Dicha arquitectura se basa en el modelo

modular que propusieron los autores Henríquez Miranda y Guzmán [23]. En el gráfico se puede observar cómo un agente coordinador se encarga de gestionar a múltiples agentes que son altamente especializados: por un lado están los rastreadores inteligentes, que navegan de forma adaptativa; por otro lado, los extractores con wrappers inductivos para cada proveedor de SaaS; y finalmente, los procesadores temporales, cuya función es organizar los datos en series históricas. Gracias a esta estructura, se logra una especialización clara por dominio y, al mismo tiempo, se mantiene la coherencia en todo el procesamiento, algo que resulta esencial cuando se necesita hacer un monitoreo continuo de precios.

2.5 El Protocolo MCP

El MCP, surge como una propuesta orientada a resolver los problemas de interoperabilidad y gestión de contexto en sistemas basados en modelos de lenguaje, *Ray* [28] señala que la ausencia de un protocolo estandarizado obligaba a utilizar integraciones ad hoc, lo que generaba pérdida de contexto, baja escalabilidad y dificultades para el análisis estructurado de la información.

2.5.1 Origen y propósito del MCP

El MCP tiene su origen en la necesidad de dotar a los modelos de lenguaje de mecanismos formales para interactuar con entornos externos de manera estructurada, como lo señala *Ray* [28], a medida que los modelos de IA comenzaron a integrarse con herramientas, APIs y bases de datos, se evidenció la falta de un protocolo estándar que permitiera gestionar el contexto de forma consistente entre diferentes sistemas.

Desde el punto de vista conceptual, el propósito del MCP es establecer una representación común del contexto que pueda ser comprendida tanto por los modelos de lenguaje como por los sistemas que los rodean, *Krishnan* [29] indica que esta representación facilita la persistencia del estado, permitiendo que los agentes de IA mantengan continuidad semántica a lo largo de múltiples interacciones y sesiones de trabajo.

Además, el MCP fue concebido para soportar arquitecturas distribuidas y colaborativas, tal como lo señalan *Hou et al*. [30] que el protocolo responde a la necesidad de coordinar múltiples agentes inteligentes que operan de manera paralela, garantizando que el contexto compartido sea coherente, controlado y trazable en el tiempo.

2.5.2 Arquitectura y Componentes Clave

La arquitectura del MCP se basa en un diseño modular que se estructura a partir de clientes, servidores y recursos, los cuales interactúan mediante mensajes estandarizados que encapsulan el contexto necesario para la ejecución de tareas específicas segun indica *Ray* [28].

En entornos multi-agente, el MCP adquiere mayor relevancia, *Krishnan* [29] destaca que el MCP incorpora mecanismos para la persistencia y versionado del contexto, lo que permite almacenar estados históricos y recuperarlos cuando es necesario, facilitando el análisis comparativo y la ejecución de tareas dependientes del tiempo, permitiendo que múltiples agentes compartan una visión coherente del estado del sistema a lo largo del tiempo.

Desde el punto de vista de la seguridad, *Hou et al.* [30] señalan que la arquitectura del MCP incorpora mecanismos de validación de mensajes y control de acceso, estos componentes permiten proteger la información contextual y garantizan la integridad del contexto compartido entre agentes y sistemas externos.

2.5.3 Casos de Uso y Relevancia para el Proyecto

El Model Context Protocol ha sido aplicado en diversos escenarios donde la gestión estructurada del contexto es un factor crítico, el autor *Ray* [28] destaca su uso en sistemas de análisis de información, automatización inteligente y asistentes basados en IA, donde el mantenimiento de un contexto consistente mejora la precisión y coherencia de los resultados obtenidos.

En sistemas colaborativos, el MCP resulta especialmente adecuado ya que *Krishnan* [29], señala que el protocolo permite coordinar agentes especializados que colaboran en tareas complejas como la recopilación, comparación y evaluación de información proveniente de múltiples fuentes, facilitando el análisis comparativo y el procesamiento de información histórica.

En relación con el presente proyecto, el MCP resulta fundamental para la extracción y el análisis comparativo de precios SaaS. *Hou et al.* [30] indican que el uso de protocolos de contexto permite mantener trazabilidad histórica y coherencia temporal de los datos, lo cual es esencial para analizar la evolución de los precios, detectar tendencias y realizar comparaciones sistemáticas a lo largo de la evolución de servicios digitales.

2.6 Análisis Comparativo Temporal de Datos

El análisis comparativo temporal constituye un enfoque metodológico que resulta esencial cuando se busca estudiar la evolución que tienen ciertos fenómenos cuantificables a medida que avanza el tiempo. Gracias a este tipo de análisis, es posible identificar patrones, tendencias y también divergencias que puedan presentarse entre distintos grupos o entidades a lo largo de un período determinado.

Si nos situamos en el contexto específico de los precios del Software como Servicio (SaaS), este análisis no se limita únicamente a caracterizar la trayectoria que sigue de forma individual cada precio. Más bien, lo que se busca es comparar dichas trayectorias entre diferentes proveedores, planes o incluso mercados, con el objetivo de extraer insights valiosos en temas como la competitividad del sector, las estrategias de fijación de precios (pricing) que adoptan los distintos actores, y la forma en que estos responden ante eventos relevantes que ocurren en el mercado.

Como lo señalan Fitzmaurice y Ravichandran [31], el objetivo principal que persigue cualquier estudio longitudinal es la caracterización del cambio que experimenta la variable de interés a lo largo del tiempo, así como también su relación con otras covariables que puedan resultar relevantes para el análisis. Este tipo de enfoque logra superar varias de las limitaciones que presentan los análisis transversales. Al respecto, *Alaminos* [32] comenta que estos últimos suelen asumir, con frecuencia, un equilibrio que es más bien estático, lo cual puede llevar a cometer inferencias erróneas, sobre todo cuando los fenómenos que se están estudiando se encuentran aún en pleno proceso de cambio.

Para que este tipo de análisis pueda implementarse de manera efectiva, se requiere apoyarse sobre tres pilares fundamentales. El primero de ellos tiene que ver con la correcta conceptualización y el adecuado tratamiento de los datos, los cuales deben ser entendidos y manejados como series temporales. El segundo pilar consiste en la aplicación de técnicas apropiadas, tanto para el análisis propiamente dicho como para la visualización de los resultados. El tercer pilar, finalmente, se refiere al aprovechamiento de herramientas y librerías especializadas, que son las que permiten automatizar y robustecer todo el proceso, tal como lo señalan [33] y [34].

2.6.1 Series Temporales

Una serie temporal se define como una secuencia de observaciones ordenadas de una variable, medida en intervalos regulares a lo largo del tiempo [32]. Esta estructura es la representación natural de datos como los precios de SaaS, que se actualizan periódicamente.

El estudio de estas series trasciende la mera descripción de valores puntuales, ya que, como señala *Alaminos* [32], su orden temporal equiespaciado es crucial para revelar la historia y las cicatrices que los eventos del pasado dejan en el fenómeno observado, permitiendo diagnosticar dinámicas, regularidades y anomalías.

El análisis longitudinal inherente a las series temporales se erige como una alternativa robusta frente a los enfoques transversales, los cuales, al congelar el tiempo, conllevan el riesgo de realizar inferencias causales erróneas bajo presunciones de equilibrio que pueden no ser válidas [32].

Fitzmaurice y Ravichandran [31] destacan que el objetivo central de un estudio longitudinal es precisamente la caracterización del cambio en la respuesta de interés a través del tiempo. Sin embargo, estos datos presentan características intrínsecamente complejas para el análisis estadístico, como la correlación entre mediciones repetidas de una misma entidad y la variabilidad heterogénea, las cuales violan los supuestos de independencia y homogeneidad de varianza de muchas técnicas estadísticas clásicas [31].

Para el análisis de series de datos continuas, se pueden destacar dos enfoques principales en cuanto a su modelado. El primero de ellos es el análisis de perfiles de respuesta, el cual resulta adecuado cuando se tienen diseños balanceados y un número reducido de mediciones. Básicamente, este método compara las secuencias de medias a lo largo del tiempo entre distintos grupos, probando específicamente la interacción que se da entre el grupo y el factor tiempo.

Por otro lado, están los modelos lineales de efectos mixtos, los cuales ofrecen una flexibilidad bastante superior en comparación con el enfoque anterior. Estos modelos permiten modelar tanto los efectos fijos (que vendrían a ser el promedio poblacional) como los efectos aleatorios (es decir, la variación que existe a nivel individual). Gracias a esta característica, resultan ser especialmente idóneos para manejar situaciones con datos desbalanceados, covariables de tipo continuo y, además, para modelar la estructura de covarianza de una manera mucho más parcimoniosa [31].

Para series con una fuerte dependencia temporal, los modelos autorregresivos y de medias móviles (ARMA) y sus variantes (ARIMA, SARIMA) son la base de la metodología Box-Jenkins. Estos modelos, implementados en librerías como statsmodels, capturan la dinámica interna de la serie descomponiéndola en componentes autoregresivos (dependencia de valores pasados) y de media móvil (dependencia de errores pasados) [33].

Finalmente, el modelo Prophet, desarrollado por Facebook, ofrece un enfoque aditivo automatizado que descompone explícitamente la serie en tendencia, estacionalidad y efectos de eventos o ”días festivos”, siendo particularmente robusto ante datos faltantes, valores atípicos y cambios de tendencia [34].

2.6.2 Técnicas de Análisis y Visualización

La exploración y comunicación efectiva de los patrones temporales requiere de técnicas analíticas y de visualización específicas, un primer paso fundamental es el diagnóstico de la serie, que incluye pruebas de estacionariedad (como el test de Dickey-Fuller aumentado) para determinar si sus propiedades estadísticas (media, varianza) en el tiempo [33].

Para poder especificar correctamente los modelos ARIMA, resulta fundamental identificar primero la estructura de autocorrelación (ACF) y la de autocorrelación parcial (PACF) de los datos. Más allá de eso, hay un aspecto que marca una gran diferencia entre un modelo que es meramente extrapolativo y uno que realmente puede considerarse explicativo: la capacidad que tenga dicho modelo para incorporar el impacto de eventos disruptivos de tipo exógeno, como podría ser una pandemia o, por ejemplo, un cambio importante en las regulaciones.

Al respecto, *Saeed et al*. [34] demuestran en su trabajo cómo la clasificación automática de eventos basada en noticias, utilizando para ello técnicas de Procesamiento de Lenguaje Natural (NLP), y su posterior inclusión como regresores dentro del modelo Prophet, permite mejorar de manera significativa la precisión predictiva en el caso concreto de las tasas de flete marítimo.

La visualización juega un papel crítico tanto en la fase exploratoria como en la presentación de resultados. Gráficos de líneas de las series originales y suavizadas, diagramas de autocorrelación y perfiles de medias por grupo son herramientas básicas [32].

No obstante, para un análisis comparativo interactivo y dinámico, se requiere de gramáticas de visualización más avanzadas. *VanderPlas et al.* [35] presentan Altair, una librería declarativa para Python basada en la gramática Vega-Lite.

Este enfoque permite construir visualizaciones estadísticas interactivas complejas, como gráficos conectados (brushing and linking), donde la selección de datos en una gráfica filtra o resalta automáticamente la información en otra, facilitando la exploración comparativa de múltiples series temporales de precios de manera intuitiva y eficiente [35].

2.6.3 Herramientas y Librerías

El ecosistema de Python ofrece un conjunto maduro de librerías que cubren todo el flujo de trabajo del análisis temporal, para el modelado estadístico clásico, statsmodels es una suite integral que proporciona funcionalidades para modelos ARIMA, VAR, filtrado (Hodrick-Prescott, Baxter-King) y pruebas de diagnóstico [33]. Para el modelado de efectos mixtos, aunque statsmodels también incluye capacidades, librerías como Linearmodels o PyMC3 (para enfoques bayesianos) ofrecen alternativas robustas.

Cuando se trata de hacer forecasting de forma automatizada pero que al mismo tiempo sea interpretable, y además pueda incorporar múltiples estacionalidades y eventos, Prophet (la herramienta desarrollada por Facebook) se ha convertido en una opción de alto nivel y muy ampliamente adoptada. Su efectividad, por cierto, ha sido validada en diversos campos como el transporte e incluso la epidemiología, tal como lo evidencian [34].

En lo que respecta a la etapa de visualización, Altair [35] se complementa muy bien con Plotly cuando se necesita interactividad, y también con Seaborn o Matplotlib para aquellos casos donde se requieren gráficos más estáticos. Gracias a esta combinación, es posible crear dashboards analíticos que resultan completos y muy útiles.

Ahora bien, la integración de todas estas herramientas dentro de una aplicación web, siempre y cuando esté alimentada por agentes de IA que se encarguen de automatizar la extracción y el preprocesamiento de los datos, permite construir un sistema de análisis comparativo temporal que resulta a la vez poderoso y accesible para los usuarios.

En síntesis, la combinación de modelos estadísticos sólidos (como ARIMA, Prophet y los modelos de efectos mixtos), junto con técnicas de inteligencia artificial para el procesamiento de eventos (NLP) y visualizaciones interactivas de tipo declarativo (como Altair o Vega-Lite), conforma un stack tecnológico que es coherente y, sin duda, muy efectivo para poder abordar el desafío de analizar y comparar la evolución temporal de los precios en un mercado tan dinámico como el del SaaS.

2.7 Metodologías Ágiles de Desarrollo

En el contexto del desarrollo moderno de software y aplicaciones web, caracterizado por entornos volátiles, de alta incertidumbre, complejidad y ambigüedad, las metodologías ágiles surgen como un paradigma fundamental para responder con flexibilidad y rapidez a los cambiantes requerimientos del cliente y del mercado [36].

Estas metodologías, articuladas formalmente en el Manifiesto Ágil de 2001, se fundamentan en valores como la priorización de los individuos y sus interacciones sobre los procesos y herramientas, la entrega de software funcional sobre la documentación exhaustiva, la colaboración con el cliente sobre la negociación contractual, y la respuesta al cambio sobre el seguimiento estricto a un plan [37].

El estudio de *García Urbano et al*. (2024) [36] destaca que este enfoque permite a las organizaciones aumentar su capacidad de adaptación, productividad y competitividad, siendo particularmente relevante en sectores dinámicos como el de Tecnologías de la Información (TI).

Entre la variedad de marcos ágiles existentes, como Extreme Programming (XP), Kanban o Dynamic System Development Method (DSDM), Scrum se ha consolidado como uno de los más adoptados a nivel global y regional, debido a su estructura clara, orientación a resultados y capacidad para gestionar la complejidad mediante iteraciones cortas y entregas incrementales de valor [38].

2.7.1 La Metodología Scrum

Scrum es un marco de trabajo (framework) ágil, iterativo e incremental, diseñado para gestionar el desarrollo de productos complejos, donde los requisitos pueden cambiar con frecuencia [39]. Sus orígenes conceptuales se remontan al trabajo de Takeuchi y Nonaka (1986) en gestión de nuevos productos, siendo posteriormente adaptado y formalizado para el desarrollo de software por Ken Schwaber y Jeff Sutherland, considerados sus creadores [36]. El marco se sustenta en tres pilares fundamentales: [36], [39].

- **Transparencia**: este principio asegura que todos los aspectos que conforman el proceso se mantengan visibles para quienes son responsables del resultado final, evitando así que queden ocultos o sin supervisión.
- **Inspección**: consiste en monitorear los avances de forma frecuente, con el propósito de detectar a tiempo cualquier desviación respecto a lo planeado y poder corregirla cuanto antes.
- **Adaptación**: facilita que se puedan realizar ajustes, ya sea sobre el proceso mismo o sobre el producto que se está desarrollando, en aquellos casos donde se identifica que el resultado obtenido no es el que se esperaba o deseaba.

Además, Scrum se rige por seis principios: [38]

- control empírico de procesos
- auto-organización
- colaboración
- priorización basada en el valor

- trabajo en intervalos de tiempo fijos
- desarrollo iterativo

La estructura de Scrum define roles, eventos y artefactos específicos [38]. Los roles centrales son:

- El Product Owner o Dueño del Producto es el responsable de maximizar el valor del producto y gestionar
- El Product Backlog quien controla la lista priorizada de requerimientos
- El Scrum Master, quien actúa como facilitador y coach, asegurando que el equipo comprenda y siga la teoría, prácticas y reglas de Scrum
- El Equipo de Desarrollo, un grupo multifuncional y auto-organizado que realiza el trabajo de convertir los requerimientos en incrementos de producto funcional.

Por otra parte, los eventos o ceremonias estructuran el ciclo de trabajo en Sprints que generalmente duran de 1 a 4 semanas, estos se observan en la Figura 6.

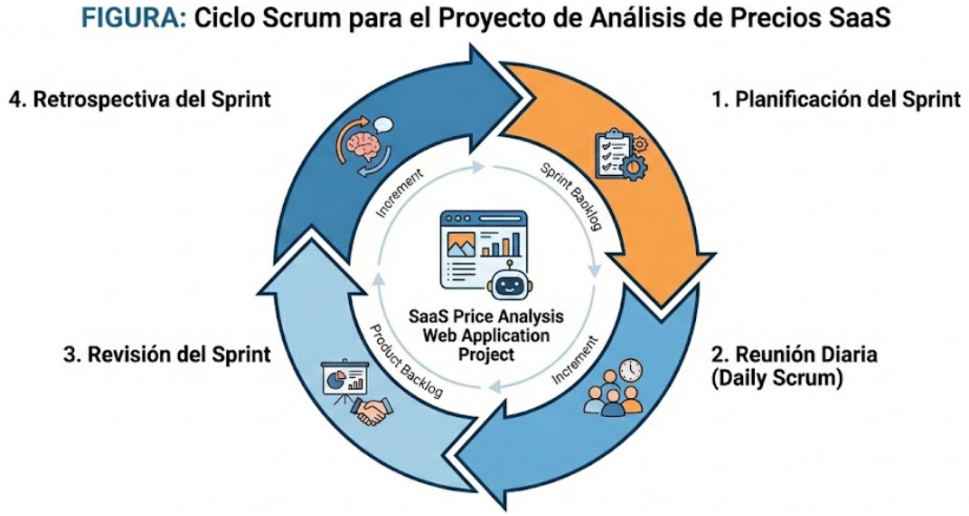


Figura 6.
Ciclo Scrum (Adaptado de [36])

- En “Planificación del Sprint”se seleccionan las tareas del Product Backlog.
- En “Reunión Diaria”se coordina al equipo de desarrollo sobre los avances y los impedimentos.
- En “Revisión del Sprint”se realiza la demostración del incremento de software al Product Owner.
- En “Retrospectiva del Sprint”se realiza un análisis interno del equipo para mejorar el proceso de desarrollo.

Al momento de hacer una comparación con otras metodologías que también se inscriben dentro del enfoque ágil, el estudio llevado a cabo por Merchán-Narváez [38] señala que Scrum se distingue principalmente por varias características. En primer lugar, por su flexibilidad y su buena capacidad para adaptarse a los cambios que puedan surgir en los requisitos del proyecto. En segundo lugar, por su clara orientación hacia la entrega iterativa y la satisfacción del cliente, lo cual se logra gracias a que el cliente se mantiene integrado de forma continua a lo largo de todo el proceso. Y en tercer lugar, por el énfasis que pone en la transparencia y en la colaboración activa dentro del equipo de trabajo.

Si bien metodologías como XP ponen mayor énfasis en las prácticas técnicas de programación (como programación en parejas y pruebas automatizadas) y Kanban se centra en la visualización y mejora del flujo de trabajo continuo, Scrum ofrece un equilibrio estructurado entre gestión de proyectos y desarrollo técnico, haciéndolo especialmente popular y versátil. [37]

2.7.2 Aplicación de Scrum en Proyectos de TI

La aplicación de Scrum en el sector de Tecnologías de la Información ha demostrado ser altamente efectiva, particularmente en el desarrollo de software y aplicaciones web. El estudio sistemático de *Estrada-Velasco et al. (2021)* [39] concluye que Scrum permite un entorno de desarrollo controlado con actividades vigiladas y validadas, características ágiles y flexibles que se ajustan a las necesidades de los requerimientos, y una capacidad de resolución de problemas en equipos de trabajo cohesionados.

En el contexto específico del desarrollo web, Bautista-Villegas [37] indica que Scrum es la metodología más idónea para la gestión de cambios en los requisitos durante el proyecto, siendo capaz de adaptarse de manera óptima y eficiente a un entorno de avances tecnológicos que demandan especialización y constante evolución.

La implementación de Scrum en PYMES del sector TI, como se analiza en el caso colombiano, reporta ventajas significativas. Entre ellas, *García Urbano et al* identifican: mayor eficiencia en la gestión de tiempo, costo y alcance; entregas tempranas que permiten retroalimentación rápida del cliente; aumento de la calidad del producto mediante estándares definidos; fortalecimiento del trabajo colaborativo y habilidades de equipo; y una comunicación más satisfactoria y constante con el cliente, lo que conduce a una mayor satisfacción del mismo [36]. El estudio señala que la metodología posibilita a las pequeñas y medianas empresas aumentar su productividad, innovación y capacidad competitiva en un mercado dinámico, sin embargo, la aplicación exitosa de Scrum enfrenta obstáculos que deben ser gestionados. La investigación de *Estrada-Velasco* et al. (2021) advierte que es crucial dimensionar correctamente el proyecto, ya que Scrum es ideal para proyectos con volúmenes de información manejables y puede presentar desafíos en sistemas de gran escala con interacciones complejas [39].

Por su parte, *García Urbano et al*. (2024) enumeran retos comunes en PYMES, como: la resistencia al cambio en culturas organizacionales tradicionales; la falta de conocimiento y experiencia en metodologías ágiles; la dependencia de un Scrum Master competente; la falta de compromiso del cliente en la definición oportuna de requerimientos; y la carencia de herramientas tecnológicas apropiadas o recursos para capacitación [36].

Para superar estos desafíos, se recomienda un proceso de implementación gradual, iniciando con pilotos en proyectos y equipos dispuestos, acompañado de capacitación especializada, sensibilización sobre el valor agregado, y el uso de herramientas de gestión de proyectos que faciliten el seguimiento por ejemplo Jira o Trello [39]

Scrum se erige como una metodología ágil robusta y ampliamente aplicable en proyectos de TI, especialmente en desarrollo web, su estructura basada en roles, eventos y artefactos, combinada con sus principios de adaptación y entrega continua de valor, la hacen una elección estratégica para organizaciones que buscan alinear su capacidad de desarrollo con las demandas de un mercado cambiante y competitivo.

Sin embargo, conviene no perder de vista que el éxito de todo lo anterior depende en gran medida de varios factores. En primer lugar, se requiere una comprensión profunda de los fundamentos teóricos y prácticos involucrados. En segundo lugar, es necesaria una implementación cuidadosa, que tenga en cuenta el contexto organizacional específico donde se vaya a aplicar. Y por último, pero no menos importante, se necesita un compromiso continuo con la mejora tanto de los procesos como de la propia dinámica del equipo de trabajo.

2.8 Evaluación de Sistemas de IA y Calidad de Datos

La evaluación integral de sistemas de inteligencia artificial, particularmente aquellos basados en agentes autónomos, constituye un pilar fundamental para garantizar su robustez, confiabilidad y utilidad práctica en entornos reales [40].

Este proceso de evaluación es bidimensional: por un lado, requiere métricas específicas para cuantificar el desempeño y las capacidades de los agentes de IA; por otro, exige métricas rigurosas para evaluar la calidad de los datos que estos agentes ingieren, procesan y generan.

La sinergia que se da entre estos dos aspectos es realmente crítica, por una sencilla razón: la calidad que tengan los datos de entrada condiciona de manera directa el desempeño que pueda llegar a tener el agente, mientras que, a su vez, la efectividad del propio agente es la que determina cuán confiables resultan ser tanto los datos extraídos como los analizados posteriormente.

Si nos centramos en el contexto específico de aquellas aplicaciones que hacen uso del Model Context Protocol (MCP) para la extracción y el análisis de precios de Software como Servicio (SaaS), esta evaluación que combina ambos frentes se vuelve todavía más relevante. ¿La razón? Básicamente porque los datos involucrados tienen un carácter bastante dinámico, son muy heterogéneos y, además, resultan ser comercialmente sensibles.

2.8.1 Métricas para Evaluar Agentes de IA

La evaluación de agentes de IA ha evolucionado desde benchmarks estáticos hacia marcos que miden capacidades interactivas y de uso de herramientas en entornos complejos, el estudio

de Liu et al. [41] destaca esta evolución y propone MCPEval, un marco de evaluación automática y profunda para agentes basados en LLM que utilizan el MCP.

Este marco metodológico introduce un conjunto de métricas que podrían calificarse como multidimensionales, ya que van más allá de la simple y tradicional tasa de éxito o fracaso. Según lo que señala el autor, la evaluación se organiza en dos dimensiones que resultan complementarias entre sí: por un lado, el Análisis de Llamadas a Herramientas, y por otro, la Evaluación mediante Jueces LLM.

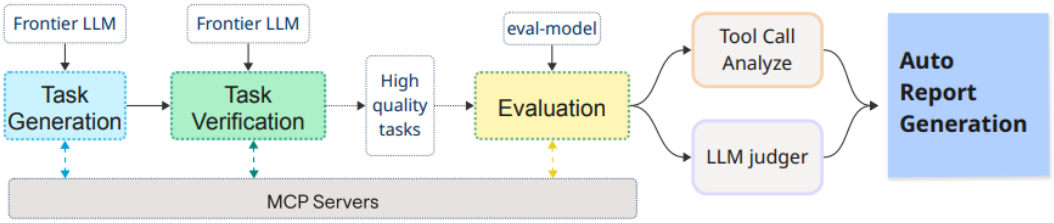


Figura 7.
Flujo de trabajo de evaluación de MCPEval (Adaptado de [41])

El Análisis de Llamadas a Herramientas mide la precisión operativa del agente al interactuar con APIs externas a través del MCP. El flujo de evaluación de este proceso, detallado en la Figura 7 , muestra la interacción entre el cliente MCP (el agente evaluado) y el servidor, seguida de la verificación de la corrección de las llamadas a herramientas y la evaluación mediante un juez LLM, culminando en un reporte automático.

Como indica el estudio de Liu [41], este análisis se compone de métricas granulares como:

- Coincidencia de Nombre (precisión en seleccionar la herramienta correcta)
- Coincidencia de Parámetros (exactitud en especificar los argumentos requeridos)
- Coincidencia de Orden (fidelidad a la secuencia de ejecución de referencia).

Partiendo de estas métricas, se procede a calcular una puntuación general, la cual puede ser de dos tipos dependiendo del nivel de exigencia que se requiera: una puntuación estricta, que exige una coincidencia exacta en los resultados, o una puntuación flexible, que resulta más tolerante ante ciertas variaciones de tipo semántico. Hay que señalar que este enfoque adquiere una relevancia crucial en aplicaciones como la extracción de precios, donde el hecho de realizar llamadas API incorrectas (por ejemplo, usando parámetros de filtro erróneos) puede terminar generando datos que resultan incompletos o, directamente, inválidos para el análisis posterior.

La segunda dimensión, la Evaluación mediante Jueces LLM, utiliza un LLM ”juez”para evaluar aspectos cualitativos del comportamiento del agente que las métricas automáticas pueden pasar por alto.

El estudio [41] detalla que esta evaluación cubre, por un lado, aspectos relacionados con la trayectoria del agente (tales como la Planificación, el Flujo de Ejecución, la Adaptabilidad y la

Conciencia Contextual) y, por otro lado, aspectos vinculados a la finalización de la tarea (como la Cobertura de Requisitos, la Precisión, la Completitud y la Utilidad). El autor subraya que métricas como la Completitud y la Utilidad resultan especialmente relevantes cuando se trata de tareas analíticas, ya que en este tipo de tareas no basta con ejecutar los pasos de manera correcta, sino que además se debe ser capaz de sintetizar una salida que sea informativa y que realmente pueda llevar a una acción concreta a partir de los datos que se han extraído.

Por otro lado, y de manera paralela, la evaluación de agentes en contextos donde se da una colaboración entre humanos e IA introduce también métricas de carácter subjetivo, las cuales se centran principalmente en la experiencia que tiene el usuario durante la interacción. El estudio de Siu et al. [42] evalúa, por ejemplo, equipos conformados por humanos e IA en el juego cooperativo Hanabi, y para ello no solo considera el desempeño objetivo (es decir, la puntuación obtenida en el juego), sino también las reacciones subjetivas que manifiestan las personas involucradas.

El autor destaca, en este sentido, la importancia que tienen métricas como la confianza que inspira el agente, su legibilidad (entendida como la capacidad de expresar claramente su intención) y su predictibilidad. Aunque el estudio en cuestión se enfoca en un juego de cartas, sus hallazgos pueden ser perfectamente extrapolables a otros ámbitos. En concreto, los agentes de IA que sean diseñados para operar en entornos empresariales (como sería el caso del análisis de precios) deben ser capaces de inspirar confianza y, además, su comportamiento debe resultar interpretable para los usuarios finales.

El mismo estudio [42] también señala un punto interesante: incluso cuando el desempeño objetivo era similar entre distintos tipos de agentes, los participantes mostraron una clara preferencia subjetiva por aquellos agentes que estaban basados en reglas (los cuales resultan ser más predecibles) en lugar de los agentes basados en aprendizaje profundo. Esto pone de manifiesto la brecha que puede existir entre la competencia técnica de un agente y el nivel de aceptación que llega a tener por parte de los seres humanos.

Por todo lo anterior, se puede concluir que una evaluación robusta de los agentes de IA orientados a la extracción y el análisis de datos debería integrar, de manera equilibrada, tanto métricas objetivas de ejecución (como las que proporcionan marcos de referencia como MCPEval [41]) como métricas subjetivas relacionadas con la experiencia del usuario y el trabajo en equipo [42]. Esto permitiría asegurar que los agentes no solo sean técnicamente competentes, sino también colaboradores efectivos y, sobre todo, confiables desde la perspectiva de quienes los utilizan.

2.8.2 Métricas para Evaluar la Calidad de los Datos Extraídos

La calidad de los datos es un constructo multidimensional que debe ser evaluado mediante métricas con una base metodológica sólida para apoyar la toma de decisiones y la gestión económica de la calidad, *Heinrich et al.* [40] proponen un marco teórico basado en la toma de decisiones bajo incertidumbre y definen cinco requisitos fundamentales que toda métrica de calidad de datos debe cumplir:

- Existencia de Mínimo y Máximo (R1)

- Escala de Intervalo (R2)
- Calidad de los Parámetros de Configuración y Determinación de los Valores de la Métrica (R3)
- Agregación Sólida de los Valores de la Métrica (R4)
- Eficiencia Económica de la Métrica (R5)

El autor, *Heinrich B.* [40], justifica cada uno de estos requisitos argumentando que, si las métricas no logran cumplirlos, se corre el riesgo de terminar tomando decisiones equivocadas o poco fundamentadas. Para ilustrar este punto, pongamos como ejemplo el primer requisito (R1): este exige que los valores que arroja la métrica estén debidamente acotados, es decir, que puedan alcanzar tanto un valor mínimo (que representaría una calidad perfectamente mala) como un valor máximo (que representaría una calidad perfectamente buena). ¿Por qué es esto importante? Básicamente porque permite que las métricas sean comparables entre sí y también facilita la validación de los resultados contra puntos de referencia claramente definidos.

R2 requiere que la métrica esté en escala de intervalo, donde las diferencias entre valores sean significativas, permitiendo así cuantificar y comparar cambios en la calidad (ej., mejorar de 0.6 a 0.8 representa la misma mejora que de 0.3 a 0.5). Este aspecto es vital para evaluar el impacto de medidas de mejora en un sistema de monitorización de precios.

Ahora bien, el requisito R3 enfatiza la necesidad de que tanto los parámetros de configuración como los valores que arroja la métrica se determinen con criterios de objetividad, fiabilidad y validez. El estudio [40] señala que, cuando se violan estos criterios (algo que ocurre con frecuencia cuando se depende de estimaciones subjetivas hechas por expertos o de definiciones que resultan imprecisas), se terminan generando resultados inconsistentes, lo cual termina socavando por completo la evaluación que se está llevando a cabo.

Si trasladamos esto al contexto específico de la extracción de precios de SaaS, las métricas que se utilicen para dimensiones como la exactitud, la completitud, la consistencia y la actualidad (también conocida como çurrency") deben ser definidas de manera operativa y sin ambigüedades. Además, sus parámetros (por ejemplo, cuál sería el valor real de referencia para medir la exactitud) tienen que poder obtenerse de forma objetiva y fiable, idealmente a través de consultas automatizadas o mediante el uso de fuentes de datos que tengan un carácter de autoridad.

Por su parte, el requisito R4 aborda la necesidad de una agregación consistente de las métricas cuando se consideran diferentes niveles de vista de los datos (por ejemplo, a nivel de un valor individual, de una tupla o de un conjunto completo de datos). El autor [40] argumenta que, si no se dispone de una función de agregación que sea sólida y esté bien definida, el valor de calidad que se obtenga para un conjunto de datos (digamos, todos los precios correspondientes a un mismo producto) podría terminar dependiendo de la forma en que se descomponga dicho conjunto, lo cual abriría la puerta a manipulaciones y, desde luego, a evaluaciones inconsistentes.

Finalmente, el requisito R5 exige lo que se podría llamar eficiencia económica: el beneficio esperado que se obtenga al aplicar la métrica (esto es, la mejora en la toma de decisiones) debe superar claramente el costo que implica su determinación. Este requisito tiene un carácter eminentemente pragmático, ya que permite descartar aquellos diseños de métricas que, aunque

pudieran ser teóricamente sólidos, resulten prohibitivamente costosos de implementar en un escenario real de negocio.

La aplicación de todos estos principios, en conjunto, permite evaluar dimensiones que resultan críticas para los datos de precios de SaaS. A modo de ejemplo, la completitud podría medirse como la proporción de atributos de precio (tales como el costo base, las características incluidas en cada plan o los límites de uso) que se encuentran presentes en los datos y no son nulos [40].

La actualidad (timeliness) requiere métricas que consideren la volatilidad del mercado de SaaS; una métrica como la propuesta por *Ballou et al.* [40] evalúa la degradación de la vigencia de un dato en función de su edad y vida útil, parámetros que deben ser definidos válidamente (R3) para el dominio específico. La consistencia puede evaluarse verificando reglas de negocio (ej., que el precio de un plan empresarial no sea menor que el de un plan profesional).

Si se diseñan las métricas siguiendo los criterios planteados por *Heinrich et al.* [40], se logra que la evaluación de la calidad de los datos extraídos cumpla con varios objetivos importantes: que sea rigurosa en su enfoque, que sus resultados sean interpretables sin mayor dificultad, que resulte económicamente viable y, lo que es más relevante, que esté en condiciones de fundamentar decisiones de negocio realmente confiables en el marco del análisis comparativo temporal de precios.

CAPÍTULO III

MATERIALES Y MÉTODOS

3.1 Descripción general

El presente capítulo describe el proceso de desarrollo de la aplicación web, una herramienta que automatiza la extracción, almacenamiento y análisis comparativo temporal de precios de servicios Software as a Service (SaaS), el sistema emplea agentes de inteligencia artificial coordinados mediante el Model Context Protocol para navegar páginas web de forma autónoma y extraer información de precios sin necesidad de configuración previa por proveedor.

El capítulo aborda tres dimensiones del proceso de desarrollo: el qué, el con qué y el cómo se llevó a cabo el proceso.

En las secciones siguientes se detalla la arquitectura del sistema, el stack tecnológico empleado, la implementación del Protocolo MCP, el proceso metodológico bajo Scrum y los sprints ejecutados con sus respectivas historias de usuario.

3.2 Arquitectura del sistema

La arquitectura de la aplicación sigue un patrón cliente-servidor multicapa, donde cada componente cumple una responsabilidad específica y se comunica mediante interfaces bien definidas.

- **Capa de Presentación:** Esta capa fue implementada utilizando Streamlit, y desde ella se exponen al usuario un total de cuatro vistas diferentes: una página de inicio, una vista para el análisis de URL, un dashboard de precios y, finalmente, una sección para el historial comparativo temporal. La comunicación entre esta capa y el backend se lleva a cabo mediante peticiones HTTP REST, para lo cual se emplea la librería `requests`.
- **Capa de Negocio:** Por su parte, la capa de negocio se implementó haciendo uso de FastAPI junto con Uvicorn. En ella residen los servicios que se encargan tanto de la extracción como del análisis de la información. Cabe destacar que el `MCPAgent` actúa como un cliente MCP, y su función principal es orquestar la comunicación que se da entre el servidor MCP de Playwright y el modelo de lenguaje Claude 3 Haiku, el cual es accedido a través de la plataforma OpenRouter.

- **Capa de Datos:** Finalmente, la capa de datos está sustentada por PostgreSQL 15, el cual se ejecuta dentro de un contenedor Docker. En esta base de datos se almacena toda la información que se va extrayendo, para lo cual se utilizan dos tablas principales: por un lado, la tabla *saas*, que se encarga de registrar cada proveedor que ha sido analizado, y por otro lado, la tabla *prices*, que almacena los planes de precios junto con su correspondiente historial temporal. El acceso a la base de datos se realiza mediante SQLAlchemy 2.0, el cual se utiliza con soporte asíncrono.

La Figura 8 ilustra la arquitectura general del sistema con sus componentes y flujos de comunicación.

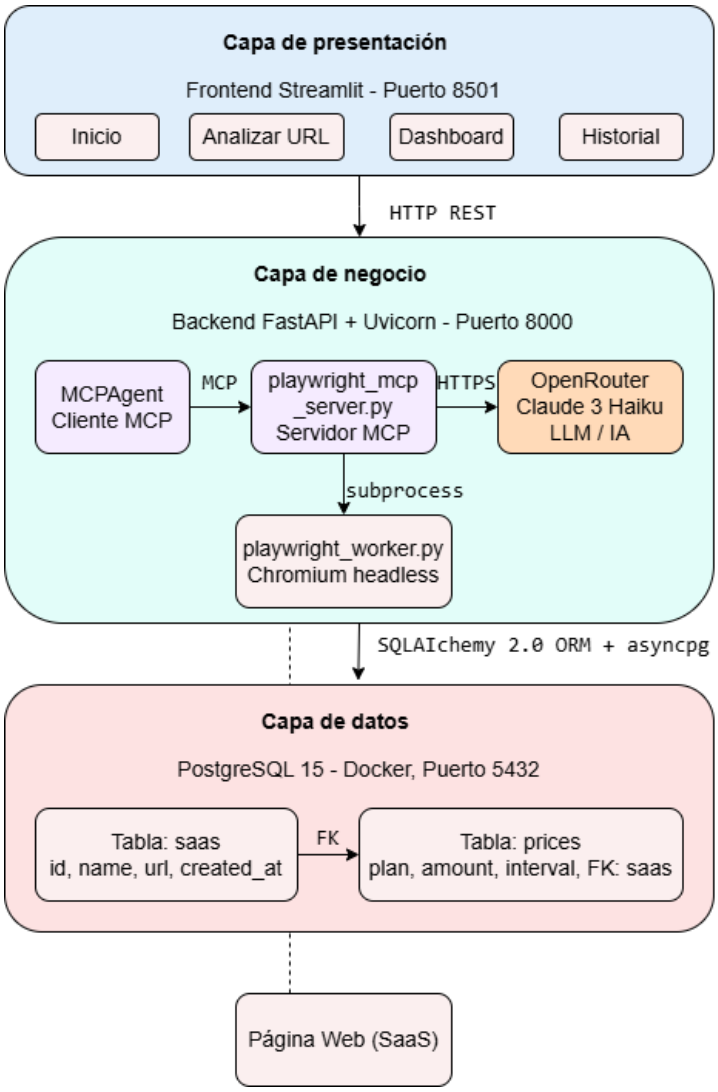


Figura 8.
Arquitectura general de la página web

3.3 Implementación del Protocolo MCP

La implementación del Model Context Protocol constituye el núcleo técnico del proyecto y diferencia esta solución de un sistema de scraping convencional, se utilizó el SDK oficial de MCP de Anthropic (librería mcp versión 1.26) para establecer una comunicación real entre cliente y servidor mediante el protocolo JSON-RPC estandarizado, tal como se describe en la sección 2.5 del Marco Teórico.

3.3.1 Servidor MCP (playwright mcp server)

El servidor MCP expone Playwright como una herramienta estructurada llamada navigate_and_extract. Este servidor se ejecuta como un proceso independiente y se comunica con el cliente mediante el transporte stdio, siguiendo la especificación del protocolo MCP. La herramienta acepta una URL y un tiempo de espera como parámetros de entrada y retorna el contenido textual extraído de la página. La declaración de la herramienta sigue el esquema en formato JSON que exige el protocolo MCP. En esta declaración se especifican tres elementos: el nombre de la herramienta, una descripción de su propósito y el esquema que define los parámetros de entrada que se esperan. Posteriormente, cuando el cliente hace una llamada a la herramienta, el servidor ejecuta Playwright dentro de un subproceso aislado (esto se hace así para evitar posibles conflictos con el event loop principal). Una vez que la ejecución ha terminado, se retorna el resultado obtenido en forma de un objeto `TextContent`, el cual sigue las especificaciones del protocolo MCP.

3.3.2 Cliente MCP (MCPAgent)

El componente `MCPAgent` es el encargado de implementar el cliente MCP, para lo cual hace uso de `ClientSession` y `StdioServerParameters` tomados directamente del SDK oficial que proporciona Anthropic. El flujo de comunicación que sigue este agente se ajusta a los pasos definidos por el propio protocolo MCP, los cuales se detallan a continuación:

- En primer lugar, el cliente lanza el servidor MCP como un subproceso independiente, utilizando para ello `StdioServerParameters`. En este paso se debe especificar cuál es el comando Python que se va a ejecutar y cuál es el script que corresponde al servidor.
- Una vez lanzado el servidor, se procede a establecer una sesión MCP, lo cual se logra mediante `stdio_client` y `ClientSession`.
- Luego, el cliente inicializa formalmente la sesión invocando a `session.initialize()`. Con este paso se completa el ”handshake.º intercambio inicial que exige el protocolo para poder comenzar la comunicación.

- Una vez que la sesión está activa, el cliente procede a listar las herramientas que el servidor tiene disponibles, llamando para ello a `session.list_tools()`. En este caso, entre las herramientas obtenidas se encuentra `navigate_and_extract`.
- Con la herramienta ya identificada, el cliente la invoca haciendo uso de `session.call_tool()`, y en esa llamada le pasa como parámetro la URL que se desea analizar.
- A continuación, el servidor MCP toma el control: navega por la página correspondiente y, una vez que ha extraído el contenido, lo retorna al cliente siguiendo el formato que establece el protocolo MCP.
- Finalmente, el contenido que ha sido retornado es procesado por el modelo Claude Haiku, cuya tarea consiste en extraer de allí los precios ya estructurados, es decir, organizados en un formato que pueda ser utilizado fácilmente por el resto del sistema.

Este flujo cumple con la arquitectura cliente-servidor del MCP: el MCPAgent actúa como cliente, playwright_mcp_server.py como servidor y navigate_and_extract como herramienta expuesta mediante el protocolo JSON-RPC estandarizado

3.4 Flujo Completo de Extracción de Precios

El proceso mediante el cual se lleva a cabo la extracción y el análisis de precios integra, como se ha venido mencionando, el Protocolo MCP junto con un modelo de lenguaje, siguiendo para ello un flujo secuencial que consta de cuatro pasos bien definidos:

- **Paso 1:** El usuario ingresa una URL en el frontend construido con Streamlit. Una vez que la URL ha sido ingresada, el frontend procede a enviar una petición de tipo POST al endpoint `/api/v1/saas/analyze` del backend implementado en FastAPI.
- **Paso 2:** Cuando la petición llega al backend, el `MCPAgent` (que funciona como cliente MCP) se activa. Este cliente lanza el servidor MCP correspondiente a Playwright, establece la sesión correspondiente y procede a llamar a la herramienta `navigate_and_extract`, pasándole como argumento la URL que se desea analizar. A partir de ese momento, el servidor MCP toma el control: navega por la página utilizando Chromium en modo headless, espera a que todo el contenido dinámico se haya cargado (para lo cual utiliza la condición `networkidle`) y, finalmente, retorna el texto que ha sido extraído al cliente, todo ello siguiendo el formato que impone el protocolo MCP.
- **Paso 3:** Una vez que el cliente MCP ha recibido el contenido extraído, procede a enviarlo a Claude 3 Haiku a través de la plataforma OpenRouter. Cabe señalar que, para optimizar los costos asociados al uso del modelo, el contenido se trunca a un máximo de 4.000 palabras. El modelo, entonces, analiza el texto recibido y retorna los planes de precios que ha logrado identificar, pero lo hace en un formato JSON perfectamente estructurado, el cual incluye campos como `plan_name`, `amount`, `currency`, `interval` y `features`.

- **Paso 4:** Finalmente, los precios que han sido extraídos se persisten en una base de datos PostgreSQL. En esta operación de almacenamiento se incluye, además de los datos de precios propiamente dichos, un timestamp que indica el momento exacto en que se realizó la extracción. Gracias a este timestamp, el sistema queda habilitado para realizar análisis comparativos a lo largo del tiempo entre sucesivos análisis realizados para un mismo SaaS.

3.5 Stack tecnológico

En esta sección se describe el conjunto de tecnologías, herramientas y librerías empleadas en el desarrollo de la aplicación web.

3.5.1 Backend

Como se muestra en la Tabla I, el backend fue desarrollado con Python 3.10. Las principales tecnologías utilizadas son FastAPI 0.135 como framework REST con soporte asíncrono nativo, Uvicorn como servidor ASGI, SQLAlchemy 2.0 con asyncpg como ORM asíncrono para PostgreSQL, Playwright 1.58 para automatización de navegadores, el SDK de MCP 1.26 para la implementación del protocolo, httpx para llamadas HTTP asíncronas a OpenRouter, y loguru para logging estructurado.

Tabla I.
Stack tecnológico del backend

Tecnología	**Versión**	**Uso en el proyecto**
Python	3.10	Lenguaje principal del backend
FastAPI	0.135	Framework REST API con soporte asíncrono nativo
Uvicorn	0.43	Servidor ASGI para FastAPI
SQLAlchemy	2.0	ORM asíncrono para mapeo objeto-relacional
asyncpg	0.29+	Driver PostgreSQL asíncrono de alta performance
Playwright	1.58	Automatización de navegadores para web scraping
mcp (SDK)	1.26	SDK oficial MCP de Anthropic — protocolo real
httpx	0.28	Cliente HTTP asíncrono para llamadas a OpenRouter
loguru	0.7+	Sistema de logging estructurado
pydantic	2.12	Validación de datos y serialización

Fuente: Elaboración propia

3.5.2 Frontend

La interfaz de usuario fue desarrollada con Streamlit, esta herramienta es un framework de Python que permite construir aplicaciones web de datos sin requerir HTML, CSS o JavaScript. También, se utilizó Plotly para realizar gráficos interactivos, Pandas para manipulación de datos tabulares y requests para la comunicación con el backend, tal como se observa en la Tabla II.

Tabla II.

Stack tecnológico del frontend

Tecnología	**Versión**	**Uso en el proyecto**
Streamlit	1.40+	Framework principal de la interfaz web
Plotly	5.0+	Gráficos interactivos para dashboard e historial
Pandas	2.0+	Manipulación y transformación de datos tabulares
requests	2.31+	Comunicación HTTP con el backend FastAPI

Fuente: Elaboración propia

3.5.3 Base de datos e infraestructura

En la Tabla III, se presentan los componentes utilizados en el desarrollo de la Infraestructura del proyecto y la base de datos utilizada.

Tabla III.
Infraestructura, IA y herramientas

Componente	Descripción
PostgreSQL 15	Motor relacional en contenedor Docker. Tablas: SaaS y prices con relación FK.
Docker Desktop	Contenerización de PostgreSQL para entorno reproducible y portable.
Claude 3 Haiku	Modelo LLM de Anthropic accedido vía OpenRouter (~$0.001 por análisis).
OpenRouter	Plataforma que provee acceso a modelos Anthropic sin cuenta directa.
Visual Studio Code	Editor de código principal con extensiones Python y FastAPI.
Git	Control de versiones del código fuente.
Postman	Prueba de endpoints REST durante el desarrollo.

Fuente: Elaboración propia

PostgreSQL 15 fue seleccionado como motor de base de datos relacional por su robustez y soporte para tipos JSON. Se ejecuta en un contenedor Docker mediante docker-compose para garantizar la portabilidad del entorno. El esquema contempla dos tablas: saas para el registro de proveedores y prices para el historial de precios con relación de clave foránea.

3.5.4 Inteligencia Artificial y protocolo MCP

El modelo Claude 3 Haiku de Anthropic fue seleccionado por su bajo costo operativo, aproximadamente $0.25 por millón de tokens de entrada (aproximadamente $0.001 por análisis típico), velocidad de respuesta y capacidad para estructurar información no organizada en JSON. Se accede al modelo mediante la plataforma OpenRouter, que permite usar modelos de Anthropic sin necesidad de cuenta directa en Anthropic Console, se utilizó Visual Studio Code como editor principal, Docker Desktop para la contenerización de PostgreSQL, Git para control de versiones y Postman para la prueba de endpoints durante el desarrollo.

3.6 Metodología de Desarrollo: Scrum

El desarrollo de la aplicación web se llevó a cabo bajo la metodología ágil Scrum por su capacidad de adaptarse a los cambios continuos en los requerimientos técnicos durante la integración de tecnologías emergentes como el Protocolo MCP y los modelos de lenguaje grande.

3.6.1 Roles del Equipo

En la Tabla IV se describen los roles del equipo:

Tabla IV.
Roles Scrum del proyecto

Rol	**Responsable**	**Responsabilidades**
Product Owner	MSc. Pablo Landeta	Validación de requerimientos y revisión de incrementos
Scrum Master	MSc. Pablo Landeta	Facilitación del proceso y eliminación de impedimentos
Equipo de Desarrollo	Yoleyda Carvajal	Diseño, implementación, pruebas e integración del sistema

Fuente: Elaboración propia

3.6.2 Product Backlog

El Product Backlog reúne las 13 historias de usuario identificadas para el proyecto, priorizadas según su valor para el sistema, con un total de 52 story points distribuidos en cuatro sprints, detallados en la Tabla V.

Tabla V.
Product Backlog del proyecto

ID	Historia de Usuario	Sprint	Pts	Prioridad
HU-01	Configurar entorno con FastAPI, PostgreSQL en Docker y entorno virtual Python.	Sprint 1	3	Alta
HU-02	Definir modelos de datos SaaS y Price en PostgreSQL con SQLAlchemy.	Sprint 1	3	Alta
HU-03	Crear endpoints REST: POST /analyze, GET /SaaS/, GET /SaaS/id/prices.	Sprint 1	5	Alta
HU-04	Integrar Playwright para navegar páginas dinámicas y extraer contenido textual.	Sprint 2	8	Alta
HU-05	Resolver incompatibilidad Playwright/ProactorEventLoop en Windows con subprocess.	Sprint 2	5	Alta
HU-06	Integrar Claude Haiku vía OpenRouter para extraer precios en JSON.	Sprint 2	5	Alta
HU-07	Analizar cualquier URL de SaaS sin configuración previa del proveedor.	Sprint 3	5	Alta
HU-08	Dashboard con métricas clave filtrables por SaaS seleccionado.	Sprint 3	3	Media
HU-09	Gráficos interactivos de precios por SaaS y distribución por periodicidad.	Sprint 3	3	Media
HU-10	Historial de precios con gráfico de evolución temporal por SaaS.	Sprint 4	5	Alta
HU-11	Extracción automática del nombre del SaaS desde el dominio de la URL.	Sprint 4	2	Media
HU-12	Endpoint DELETE para eliminar registros de SaaS y precios asociados.	Sprint 4	2	Baja
HU-13	Detección automática de cambios de precio entre análisis consecutivos.	Sprint 4	3	Media

Fuente: Elaboración propia

3.7 Planificación de Sprints

El proyecto se organizó en cuatro sprints de dos semanas cada uno, con un total de ocho semanas de desarrollo. La Figura 9 ilustra el proceso de desarrollo seguido durante el proyecto.

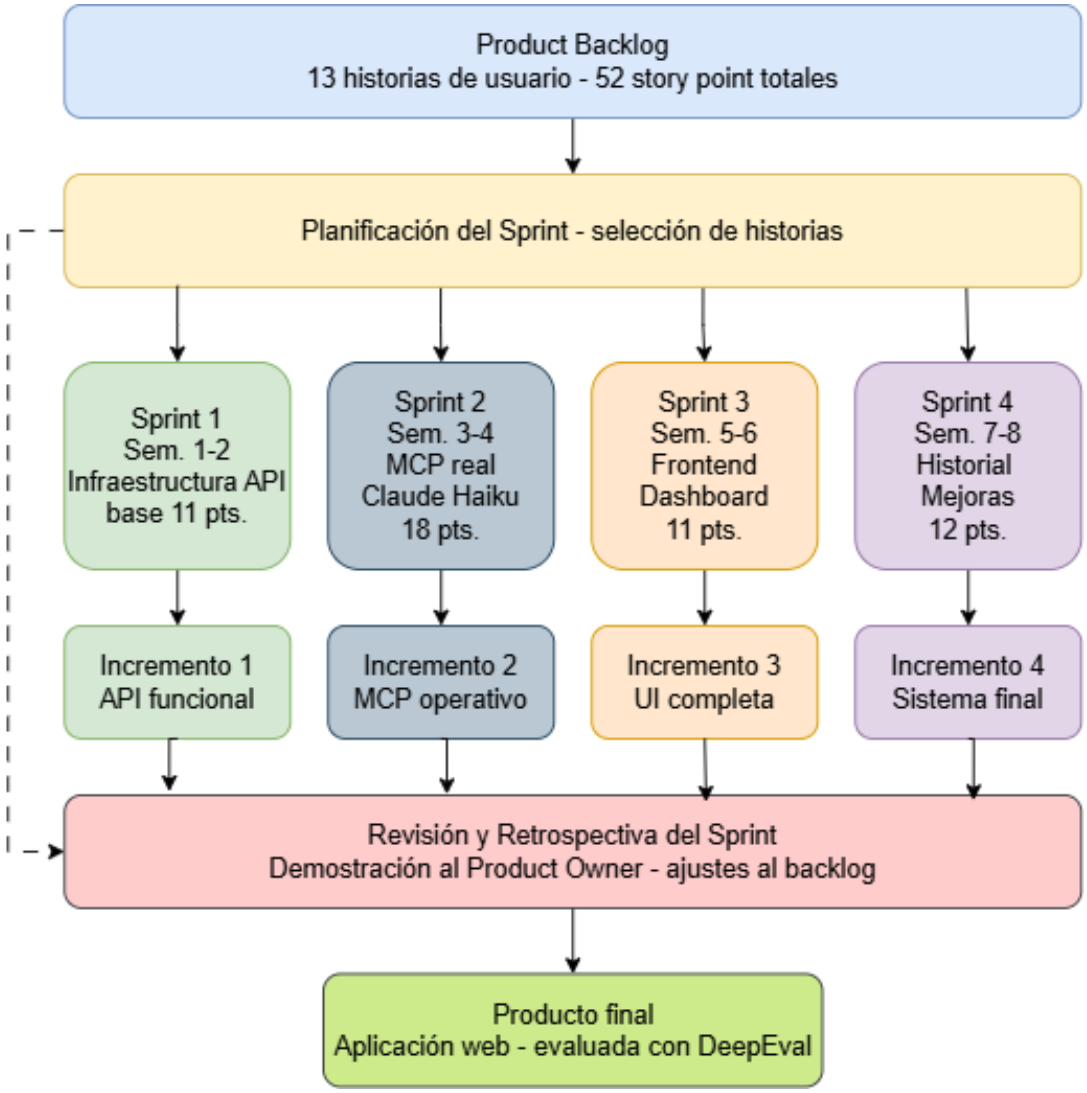


Figura 9.
Proceso de desarrollo basado en Scrum

3.7.1 Sprint 1 — Infraestructura y API Base

Objetivo: Establecer la infraestructura tecnológica completa incluyendo el entorno de desarrollo, la base de datos y los endpoints básicos de la API REST. A continuación, en la Tabla VI se presenta el desarrollo del Sprint 1.

Tabla VI.
Sprint 1 — Infraestructura base

ID	Historia de Usuario	Criterios de Aceptación	Pts.	Prior.
HU-01	Como desarrollador, necesito configurar el entorno con FastAPI, PostgreSQL en Docker y entorno virtual Python.	Docker con PostgreSQL en puerto 5432. FastAPI inicia sin errores. Variables env cargadas correctamente.	3	Alta
HU-02	Como desarrollador, necesito definir los modelos de datos SaaS y Price con SQLAlchemy en PostgreSQL.	Tablas saas y prices creadas. Relación FK entre ambas. SQLAlchemy ORM funcional con asyncpg.	3	Alta
HU-03	Como desarrollador, necesito crear los endpoints REST principales de la API.	POST /analyze, GET /SaaS/, GET /id/prices responden 200 OK. Datos persistidos en PostgreSQL.	5	Alta

Fuente: Elaboración propia

Resultado: Se logró tener una API REST completamente funcional y conectada exitosamente a la base de datos PostgreSQL. Durante este proceso, se identificó y resolvió un problema relacionado con el archivo .env, el cual presentaba codificación UTF-8 con BOM (Byte Order Mark) y estaba causando errores en la lectura de las variables de entorno. Adicionalmente, se dejó definida y establecida la estructura base del proyecto, lo cual facilitó el trabajo posterior en las siguientes etapas del desarrollo.

3.7.2 Sprint 2 — Integración MCP y Claude Haiku

Objetivo: Implementar el servidor MCP real con Playwright y la extracción de precios mediante Claude Haiku. Este fue el sprint más complejo del proyecto y se muestra en la Tabla VII.

Tabla VII.
Sprint 2 — MCP e Inteligencia Artificial

ID	Historia de Usuario	Criterios de Aceptación	Pts.	Prior.
HU-04	Como sistema, necesito integrar Playwright para navegar páginas dinámicas y extraer contenido textual de cualquier URL.	Playwright navega cualquier URL. Extrae texto con document.body.innerText. Espera networkidle.	8	Alta
HU-05	Como sistema, necesito resolver la incompatibilidad entre Playwright y ProactorEventLoop de Windows.	Sin errores NotImplementedError. Playwright corre en proceso hijo aislado vía subprocess.run.	5	Alta
HU-06	Como sistema, necesito integrar Claude Haiku vía OpenRouter para extraer precios en JSON estructurado.	OpenRouter retorna planes en JSON. Timeout 45s. Contenido truncado a 4.000 palabras.	5	Alta

Fuente: Elaboración propia

Resultado: Se logró poner en funcionamiento el pipeline completo basado en MCP. En concreto, el servidor MCP expone correctamente la herramienta denominada navigate-and-extract, mientras que el cliente implementado en MCPAgent es capaz de establecer sesiones MCP reales utilizando para ello el SDK oficial que proporciona Anthropic. Esto implica que la comunicación entre ambos extremos sigue fielmente el estándar definido por el protocolo.

3.7.3 Sprint 3 — Frontend y Dashboard

Objetivo: Desarrollar la interfaz de usuario completa con Streamlit incluyendo análisis de URLs, dashboard filtrable y visualizaciones interactivas, a continuación se muestra en la Tabla VIII.

Tabla VIII.
Sprint 3 — Interfaz y visualización

ID	Historia de Usuario	Criterios de Aceptación	Pts	Prior.
HU-07	Como usuario, quiero analizar cualquier URL de SaaS sin configuración previa del proveedor.	Campo acepta cualquier URL. Resultado en menos de 90s. Nombre extraído del dominio.	5	Alta
HU-08	Como usuario, quiero ver métricas clave en el Dashboard filtrables por SaaS seleccionado.	4 métricas visibles. Selector múltiple funcional. Métricas se actualizan con el filtro.	3	Media
HU-09	Como usuario, quiero gráficos interactivos de precios por SaaS y distribución por periodicidad.	Gráfico de barras agrupadas por plan. Gráfico de pastel. Tabla detallada de precios.	3	Media

Fuente: Elaboración propia

Resultado: Se logró construir una interfaz de usuario completa, la cual cuenta con cuatro páginas que resultan plenamente navegables. Dentro de esta interfaz, el dashboard incluye un selector de tipo múltiple que permite filtrar los SaaS según las necesidades del usuario, así como varios gráficos interactivos generados con Plotly que facilitan la visualización y el análisis de los datos.

3.7.4 Sprint 4 — Historial Comparativo y Mejoras

Objetivo: Implementar el módulo de historial comparativo temporal y realizar ajustes finales al sistema.

Tabla IX.
Sprint 4 — Historial y mejoras finales

ID	Historia de Usuario	Criterios de Aceptación	Pts.	Prior.
HU-10	Como usuario, quiero ver el historial de precios de un SaaS con gráfico de evolución temporal.	Selector de SaaS con precios. Gráfico de línea con evolución. Tabla histórica con registros.	5	Alta
HU-11	Como sistema, necesito extraer automáticamente el nombre del SaaS desde el dominio de la URL.	Nombre extraído con urlparse y capitalizado. Fallback a timestamp si el dominio falla.	2	Media
HU-12	Como administrador, necesito eliminar registros de SaaS y precios desde el endpoint DELETE.	DELETE /saas/{id} funcional. Elimina precios en cascada. Retorna mensaje de confirmación.	2	Baja
HU-13	Como usuario, quiero que el sistema detecte cambios de precio entre análisis consecutivos.	Sección de cambios visible. Muestra variación porcentual. Indica precio anterior y actual.	3	Media

Fuente: Elaboración propia

Resultado: Como se observa en la Tabla IX, se obtiene un módulo de historial funcional con detección automática de cambios de precio y un sistema completo listo para evaluación.

3.8 Resumen de Sprints

La siguiente Tabla X indica a continuación el resumen de los Sprints realizados.

Tabla X.
Resumen de sprints y velocidad del equipo

Sprint	Objetivo Principal	Duración	Puntos	Estado
Sprint 1	Infraestructura y API base	Semanas 1-2	11	Completo
Sprint 2	MCP real + Claude Haiku	Semanas 3-4	18	Completo
Sprint 3	Frontend y Dashboard	Semanas 5-6	11	Completo
Sprint 4	Historial y mejoras finales	Semanas 7-8	12	Completo
Total Story Points:				**52**

Fuente: Elaboración propia

3.9 Estructura del Proyecto

La organización modular del proyecto responde a dos principios arquitectónicos fundamentales: la separación de responsabilidades y la independencia tecnológica entre capas.

La estructura del proyecto se organiza en dos directorios principales. Por un lado, el directorio *backend/* agrupa toda la lógica de negocio, los servicios encargados de la extracción, los modelos de datos y la exposición de la API REST. Por otro lado, el directorio *frontend/* contiene de forma exclusiva la interfaz de usuario, la cual fue construida con Streamlit, junto con la lógica necesaria para la comunicación con el backend mediante peticiones HTTP.

Esta decisión de separar claramente ambos componentes no es arbitraria, sino que responde a la necesidad de que cada uno pueda evolucionar de manera independiente. En términos prácticos, el backend puede escalar o modificar su implementación interna sin que esto afecte al frontend, siempre y cuando se respete y mantenga la interfaz de la API que ambos utilizan para comunicarse. De igual forma, el frontend puede ser rediseñado o pueden agregarse nuevas vistas sin que sea necesario tocar o modificar la lógica de extracción de datos que reside en el backend.

Dentro del backend, la subdivisión en *api/routes/, services/ y models/* sigue el patrón de diseño de capas (presentación, negocio y datos). El directorio *api/routes/* contiene exclusivamente la definición de los endpoints REST, delegando la lógica de negocio a los servicios ubicados en *services/*. Esta separación facilita la prueba unitaria de cada componente, ya que los servicios pueden ser probados de forma aislada sin depender de la capa HTTP. El directorio *models/* define los esquemas de la base de datos utilizando SQLAlchemy, manteniendo la lógica de persistencia desacoplada del resto del sistema. Esta arquitectura limpia permite que el código sea mantenible, testeable y extensible ante futuros requerimientos, como la incorporación de nuevos agentes MCP o la implementación de un sistema de autenticación de usuarios.

La Figura 10 y 11, muestran la organización del directorio del proyecto en Visual Studio

Code.

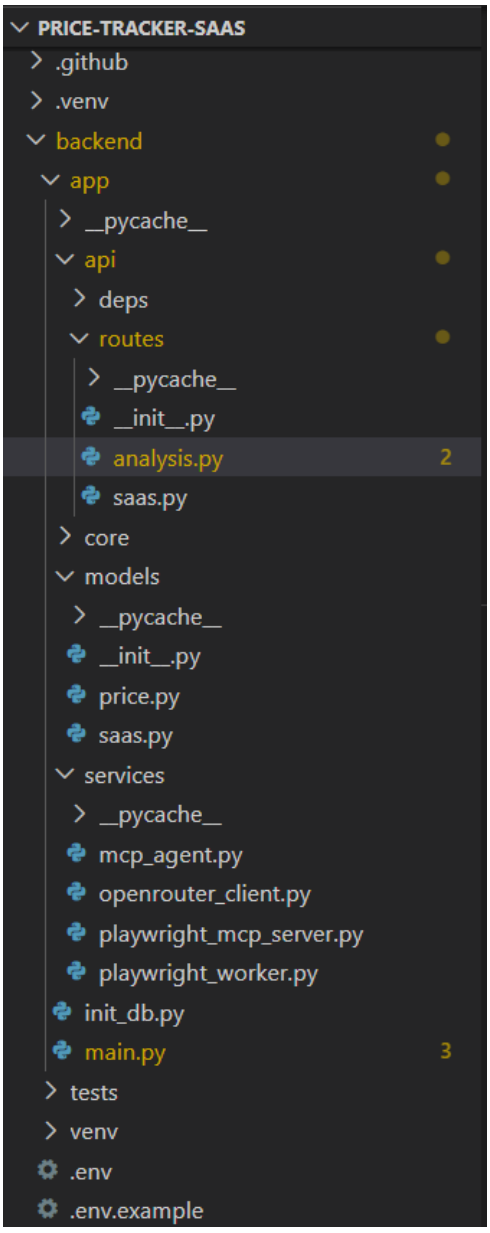


Figura 10.
Organización del directorio

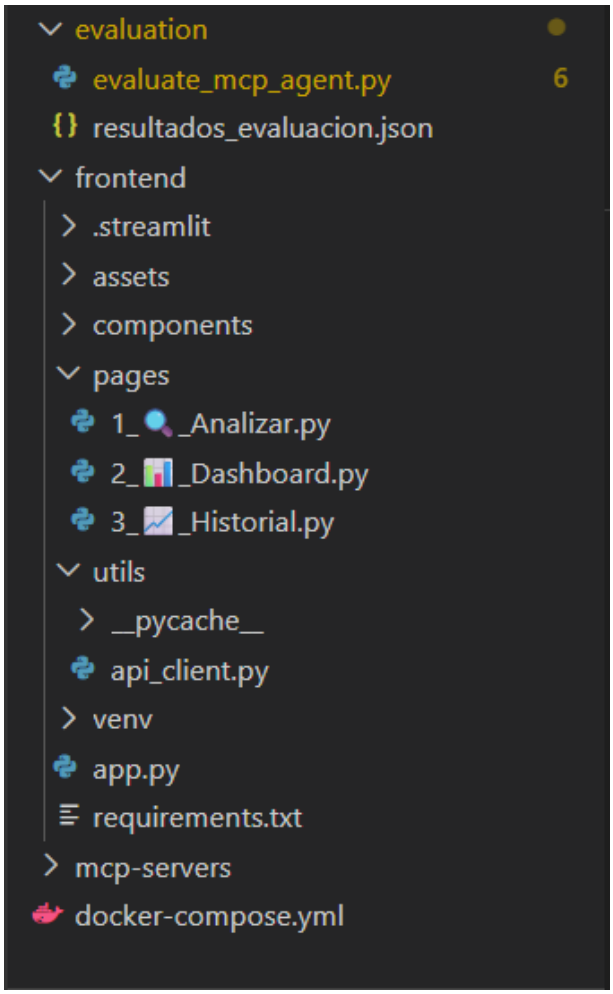


Figura 11.
Organización del directorio

3.10 Desafíos Técnicos y Soluciones Implementadas

Durante el desarrollo surgieron cuatro desafíos técnicos significativos cuya resolución forma parte de la contribución práctica de este trabajo y son descritos a continuación.

3.10.1 Compatibilidad de Playwright con Windows y Uvicorn

Playwright requiere SelectorEventLoop para lanzar subprocesos de Chromium, sin embargo Uvicorn en Windows utiliza ProactorEventLoop, generando NotImplementedError. La solución implementada ejecuta Playwright en un proceso hijo completamente aislado mediante subprocess.run, con resultados transmitidos via stdout en formato JSON. Este patrón se aplica tanto en el cliente directo como dentro del servidor MCP.

3.10.2 Optimización de Costos en el Modelo de Lenguaje

El contenido extraído por Playwright puede superar las 18.000 palabras por página. Para optimizar el uso de tokens, el contenido se trunca a las primeras 4.000 palabras antes de enviarse al modelo. Se configuró un timeout de 45 segundos y se eliminó el parámetro response-format incompatible con OpenRouter. Con estas optimizaciones el costo por análisis es aproximadamente $0.001 por análisis (una décima de centavo de dólar)

3.10.3 Orden de Carga de Variables de Entorno

Se identificó un problema de orden de importación: los módulos de servicios se importaban antes de que load_dotenv() ejecutara, impidiendo que las variables de entorno estuvieran disponibles durante la inicialización. La solución fue mover load_dotenv() al inicio absoluto de main.py, antes de cualquier importación de módulos propios del proyecto

3.10.4 Implementación MCP en Entorno Asíncrono Windows

El SDK de MCP requiere contexto asíncrono para gestionar sesiones cliente-servidor. La integración con FastAPI fue directa mediante async with stdio-client() y async with ClientSession(). Sin embargo, el servidor MCP también necesita lanzar Playwright como subprocess, reintroduciendo el conflicto con ProactorEventLoop. La solución fue aplicar el mismo patrón subprocess.run dentro del servidor MCP, manteniendo la coherencia arquitectónica del sistema.

CAPÍTULO IV

RESULTADOS Y ANÁLISIS

4.1 Introducción

El presente capítulo expone los resultados obtenidos de la evaluación de la página web desarrollada con agentes de inteligencia artificial coordinados mediante el protocolo MCP, la evaluación se estructuró en dos fases complementarias:

- **Evaluación subjetiva (manual):** En esta primera aproximación, se definieron expectativas previas para cada uno de los 15 servicios SaaS que fueron seleccionados como parte del estudio. Las dimensiones consideradas fueron tres: facilidad de extracción, claridad de los precios y estructura de la página. Para cada una de estas dimensiones, se asignaron puntajes en una escala que va del 1 al 5.
- **Evaluación objetiva con DeepEval:** En segundo lugar, se aplicó un conjunto de cinco métricas especializadas, las cuales son *Answer Relevancy*, *Faithfulness*, *Contextual Precision*, *Contextual Recall* y *Hallucination*. Para ello se utilizó el framework DeepEval en su versión 3.9.6, y se empleó a GPT-4o-mini como modelo juez encargado de evaluar las respuestas del agente.

Esta doble aproximación (subjetiva por un lado y objetiva por el otro) permite responder de manera adecuada al tercer objetivo específico planteado en la investigación, el cual consiste en: .[Ev]aluar la aplicación web con métricas específicas para agentes de inteligencia artificial y de precisión de datos, contrastando las expectativas del investigador con los resultados objetivos que arroja el sistema".

4.1.1 Uso de DeepEval

A continuación, se enumeran las principales razones que justifican la elección de DeepEval como framework de evaluación para este proyecto:

- **Especialización en LLM**: a diferencia de otros frameworks de testing que tienen un carácter más genérico y están orientados al software convencional, DeepEval ha sido diseñado específicamente para evaluar sistemas que se basan en grandes modelos de lenguaje, lo cual se ajusta perfectamente a las necesidades del presente trabajo.

- **Métricas estandarizadas**: el framework proporciona un conjunto de métricas estandarizadas, tales como Faithfulness, Hallucination y Contextual Recall, las cuales se alinean de manera directa con los criterios de calidad que resultan relevantes para agentes de inteligencia artificial.
- **Paradigma LLM-as-a-Judge**: DeepEval implementa el enfoque conocido como ”LLM-as-a-Judge”, el cual permite utilizar un modelo de lenguaje como evaluador automático. Esto tiene una ventaja importante: elimina la subjetividad humana que suele estar presente en la evaluación de respuestas abiertas y, además, reduce el posible sesgo que pueda introducir el investigador.
- **Compatibilidad con OpenRouter**: el framework es compatible con cualquier API que siga el estándar de OpenAI, lo cual facilita de manera notable su integración con OpenRouter, que es precisamente la plataforma utilizada en este proyecto para acceder a los modelos de lenguaje.
- **Reproducibilidad de las evaluaciones**: dado que las evaluaciones son completamente automatizadas y no dependen del criterio de un evaluador humano (a diferencia de lo que ocurre con las evaluaciones manuales), DeepEval garantiza que los resultados puedan ser reproducidos en cualquier momento sin que haya variaciones introducidas por factores subjetivos.

4.1.2 Modelo Evaluador (GPT-4o-mini)

La elección de utilizar GPT-4o-mini a través de OpenRouter como modelo juez (bajo el paradigma LLM-as-a-Judge) se fundamenta en seis razones principales, las cuales se detallan a continuación.

- Balance costo-efectividad, GPT-4o-mini es significativamente más económico que GPT-4o, con un costo de aproximadamente 0.15 dólares por millón de tokens de entrada frente a los 5.00 dólares de GPT-4o, lo que permitió evaluar 15 SaaS con cinco métricas cada uno a un costo mínimo.
- Según los benchmarks de la plataforma OpenRouter, GPT-4o-mini alcanza un rendimiento comparable de GPT-4o en tareas de evaluación semántica, con una degradación inferior al cinco por ciento en métricas de precisión.
- La compatibilidad nativa con DeepEval, ya que el framework requiere un modelo compatible con la API dde OpenAI para las métricas LLM-as-a-Judge.
- El acceso simplificado vía OpenRouter, que permitió utilizar GPT-4o-mini sin necesidad de tener una centa directa en OpenAI.
- La consistencia en la evaluación, al utilizar el mismo modelo juaz para todos los casos de prueba, se garantiza que los criterios de evaluación sean uniformes a lo largo de los 15 SaaS analizados.

- Una de las ventajas clave de emplear un modelo juez estandarizado, es que permite eliminar el sesgo que un evaluador humano podría introducir de manera involuntaria al momento de evaluar respuestas que presentan diferentes niveles de complejidad. Este tipo de sesgo humano, difícil de controlar en evaluaciones manuales, queda así prácticamente anulado.

4.1.3 Evaluación Subjetiva y Objetiva

La combunación de una evaluación subjetiva y objetiva responde a la necesidad de contrastar la expectativa de la investigadora con el desempeño real del sistema. La evaluación subjetiva tuvo como propósito establecer las expectativaas de la estudiante investigadora sobre el comportamiento esperado del sistema, basadas en la observación directa de las páginas web de cada SaaS, este método captura el conocimiento experto del dominio, pero tiene la limitación de ser subjetivo y no reproducible.

Por el lado de la evaluación objetiva con DeepEval, lo que se hace es medir el desempeño real del sistema, pero haciéndolo de una manera estandarizada y completamente automatizada. Este enfoque tiene la ventaja de eliminar el sesgo humano que podría introducirse en una evaluación manual y, además, garantiza que los resultados sean reproducibles en cualquier momento. Sin embargo, también hay que reconocer una limitación: este tipo de evaluación automatizada puede no llegar a capturar ciertos matices específicos del dominio, esos que un experto humano, con su conocimiento y experiencia, sí sería capaz de identificar sin mayores dificultades.

La complementariedad de ambos enfoques permite responder preguntas que una evaluación aislada no podría responder, si el sistema se comporta como el investigador esperaba, qué casos superaron las expectativas, y qué casos presentaron dificultades inesperadas.

Por ejemplo, en el caso de Slack, la expectativa era baja (1.67 sobre 5) debido a sus mecanismos de protección anti-bot, pero el Faithfulness obtenido fue perfecto (1.00) porque al no haber contenido de precios en el contexto, no hubo coontradicciones posibles, resultando mejor de lo esperado.

En contraste, servicios como Notion, Trello, Zoom, GitHub y Canva presentaban, según las expectativas definidas previamente, la puntuación máxima (5.00), debido a que mostraban precios de forma claramente estructurada y accesible. Sin embargo, al aplicar la métrica de Faithfulness, estos mismos servicios obtuvieron una puntuación de 0.00. ¿A qué se debió esta diferencia? Básicamente a que el modelo de lenguaje infirió valores distintos a los que aparecían de manera literal en el texto plano que había sido extraído mediante scraping. Este hallazgo es importante porque revela una limitación del scraping basado únicamente en texto: dicha limitación no era evidente cuando se hacía la observación manual de las páginas, pero sí se hizo patente al someter el sistema a una evaluación automatizada y estandarizada.

4.2 Configuración de la Evaluación

La evaluación de la aplicación web se configuró con los parámetros indicados en la Tabla XI:

Tabla XI.
Configuración de la evaluación

Parámetro	**Valor**
Framework de evaluación	DeepEval v3.9.6
Modelo evaluador (juez)	openai/gpt-4o-mini vía OpenRouter
Modelo de extracción	anthropic/claude-3-haiku vía OpenRouter
Protocolo de scraping	MCP real (SDK oficial Anthropic, mcp v1.26)
Navegador	Chromium headless vía Playwright 1.58
Total de SaaS evaluados	15 servicios reales
Umbral de aprobación	0.5 (escala 0.0 – 1.0)
Métricas aplicadas	5 métricas DeepEval
Tiempo total de evaluación	38.37 segundos (fase de métricas)

Fuente: Elaboración propia

4.3 Evaluación Subjetiva (Manual)

Previo a la ejecución de las pruebas automatizadas, se realizó una evaluación manual para establecer expectativas sobre el comportamiento esperado del sistema, se evaluaron tres criterios en escala 1 a 5: facilidad de extracción, claridad de precios y estructura de la página. A partir de estos se calculó un Puntaje Esperado (promedio aritmético), que posteriormente fue contrastado con la métrica Faithfulness obtenida en la evaluación objetiva.

4.3.1 Metodología de Evaluación Manual

Previo a la ejecución de las pruebas automatizadas, la estudiante investigadora definió expectativas para cada uno de los 15 SaaS evaluados, considerando tres criterios en escala de 1 (Muy bajo) a 5 (Muy alto), los criterios están detallados en la Tabla XII.

Tabla XII.
Criterios de evaluación

Criterio	**Descripción**
Facilidad de extracción	Qué tan accesible es la información de precios (¿requiere login? ¿tiene anti-bot?)
Claridad de precios	¿Los precios son explícitos y legibles o están ocultos en texto denso?
Estructura de la página	¿HTML bien estructurado o depende de JavaScript pesado?

Fuente: Elaboración propia

4.3.2 Resultados de la Evaluación Subjetiva

La Tabla XIII a continuación muestra el puntaje esperado calculado como el promedio aritmético de los tres criterios:

- *Puntaje Esperado = (Facilidad + claridad + Estructura) / 3*

Tabla XIII.
Expectativa por SaaS

SaaS	Facilidad	Claridad	Estructura	Puntaje Esperado (1-5)
Figma	4	5	4	4.33
Notion	5	5	5	5.00
Linear	4	4	4	4.00
Slack	2	2	1	1.67
Trello	5	5	5	5.00
Basecamp	4	4	4	4.00
Airtable	4	5	4	4.33
Zoom	5	5	5	5.00
Dropbox	5	5	4	4.67
GitHub	5	5	5	5.00
Canva	5	5	5	5.00
HubSpot	2	2	2	2.00
ConvertKit	4	4	4	4.00
Shopify	4	4	4	4.00
Framer	3	3	3	3.00

Fuente: Elaboración propia

4.3.3 Análisis de la Evaluación Subjetiva

SaaS con mayor espectativa (≥ 4.5):

- **Notion, Trello, Zoom, GitHub, Canva (expectativa de 5.00):** estos servicios se caracterizan por presentar precios de manera claramente estructurada y, además, no cuentan con mecanismos evidentes de protección anti-bot que dificulten el acceso a la información.
- **Dropbox (expectativa de 4.67):** en este caso, los planes de precios están bien definidos, aunque se identificó cierta complejidad adicional en lo que respecta a los planes orientados a empresas, lo cual justifica una puntuación ligeramente inferior al máximo.
- **Figma, Airtable, Basecamp (expectativas entre 4.00 y 4.33):** estos servicios presentan una estructura clara en términos generales, aunque se detectaron algunos elementos de

carácter dinámico que añaden un grado de complejidad adicional a la tarea de extracción automatizada.

SaaS con menor expectativa (≤ 2.0):

- **Slack (1.67):** Protección anti-bot conocida, contenido de precios cargado dinámicamente.
- **HubSpot (2.00):** Página compleja con múltiples productos y estructura de navegación pesada.

4.4 Evaluación Objetiva con DeepEval

En cuanto a la evaluación objetiva, esta se llevó a cabo utilizando el framework DeepEval en su versión 3.9.6, y se empleó a GPT-4o-mini como modelo juez, accediendo a él a través de la plataforma OpenRouter. Para la evaluación se aplicaron cinco métricas diferentes, a saber: Answer Relevancy, Faithfulness, Contextual Precision, Contextual Recall y Hallucination. Es importante señalar que la evaluación cubrió exactamente los mismos 15 servicios SaaS que ya habían sido considerados en la fase subjetiva, lo que permitió generar un total de 75 mediciones individuales. Todo este proceso se completó en un tiempo total de 38.37 segundos, correspondientes específicamente a la fase de cálculo de las métricas.

4.4.1 Métricas de Evaluación Aplicadas

Como se observa en la Tabla XIV, se aplicaron cinco métricas del framework DeepEval, seleccionadas en función de su alineación con los criterios de evaluación de agentes de IA descritos en el Marco Teórico.

Tabla XIV.
Métricas de evaluación aplicadas

#	Métrica	Qué mide	Dimensión evaluada
1	Answer Relevancy	Si la respuesta del agente es relevante a la pregunta formulada	Pertinencia de la respuesta
2	Faithfulness	Si los precios extraídos son fieles al contenido real de la página web	Fidelidad al contexto
3	Contextual Precision	Si el contexto recuperado por el agente es preciso y pertinente	Precisión del contexto
4	Contextual Recall	Si el agente recupera toda la información relevante disponible	Cobertura del contexto
5	Hallucination	Si el agente genera información que contradice el contexto real (score alto = más alucinación)	Veracidad factual

Fuente: Elaboración propia

Es importante destacar que la métrica Hallucination tiene una interpretación inversa a las demás: un score de 1.0 indica presencia de alucinación, mientras que 0.0 indica ausencia total. Esta diferencia se toma en cuenta en el análisis de resultados.

4.4.2 Resultados por SaaS

La Tabla XV muestra los scores obtenidos para cada uno de los 15 SaaS evaluados en las cinco métricas, los valores se expresan en escala de 0.0 a 1.0 donde 1.0 representa el desempeño óptimo, excepto en Hallucination, dónde 0.0 es el óptimo.

Tabla XV.
Resultados de evaluación por SaaS.

SaaS	Planes	Ans. Rel.	Faithful.	Ctx. Prec.	Ctx. Recall	Halluc.	Promedio
Figma	3	1.00	0.00	1.00	1.00	1.00	0.60
Notion	3	1.00	0.00	1.00	1.00	1.00	0.60
Linear	3	1.00	0.00	1.00	1.00	1.00	0.60
Slack	4	1.00	1.00	0.00	0.00	1.00	0.40
Trello	4	1.00	0.00	1.00	1.00	1.00	0.60
Basecamp	3	1.00	0.00	1.00	1.00	1.00	0.60
Airtable	4	1.00	0.50	1.00	1.00	1.00	0.70
Zoom	4	1.00	0.25	1.00	1.00	1.00	0.65
Dropbox	4	1.00	0.50	1.00	1.00	1.00	0.70
GitHub	3	1.00	0.00	1.00	1.00	1.00	0.60
Canva	2	1.00	0.50	1.00	1.00	1.00	0.70
HubSpot	3	1.00	0.33	0.00	1.00	1.00	0.47
ConvertKit	3	1.00	0.00	1.00	1.00	1.00	0.60
Shopify	4	1.00	0.25	1.00	1.00	1.00	0.65
Framer	3	0.93	0.00	1.00	1.00	1.00	0.59
PROMEDIO	**3.27**	**0.996**	**0.222**	**0.867**	**0.933**	**1.000**	**—**

Fuente: Elaboración propia

4.4.3 Resumen Estadístico por Métrica

La Tabla XVI presenta el resumen estadístico de cada métrica calculado sobre los 15 SaaS evaluados.

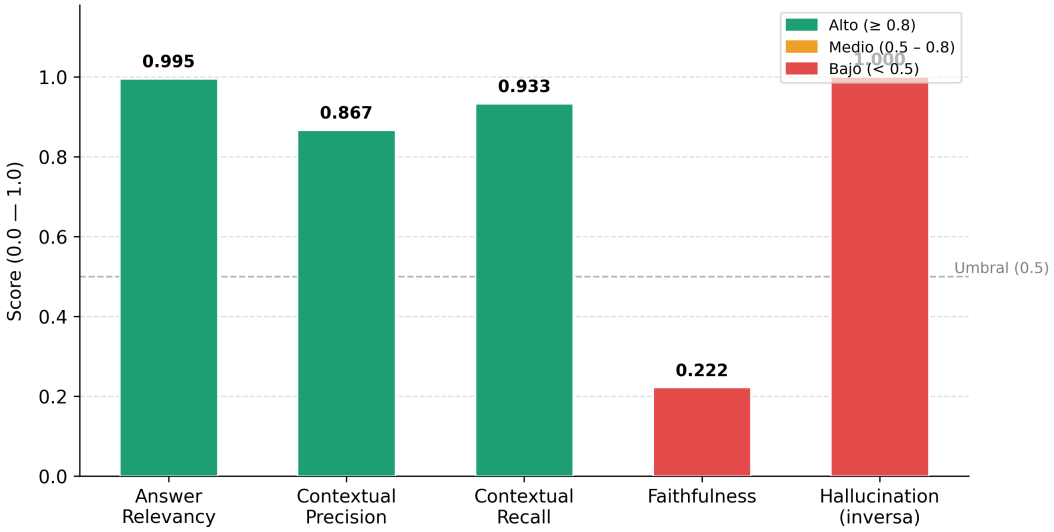
Tabla XVI.

Resumen estadístico por métrica. * Hallucination: score alto indica presencia de contradicciones

Métrica	Promedio	Mínimo	Máximo	Interpretación
Answer Relevancy	0.9956	0.9333	1.0000	Excelente — respuestas muy pertinentes
Contextual Recall	0.9333	0.0000	1.0000	Muy bueno — alta cobertura del contexto
Contextual Precision	0.8667	0.0000	1.0000	Bueno — contexto preciso y pertinente
Faithfulness	0.2222	0.0000	1.0000	Bajo — variaciones respecto al texto literal
Hallucination *	1.0000	1.0000	1.0000	Interpretación especial — ver análisis

Fuente: Elaboración propia

La Figura 12 presenta de forma visual los scores promedio obtenidos por el agente MCP en las cinco métricas de evaluación aplicadas mediante DeepEval sobre los 15 SaaS analizados. Los resultados evidencian un desempeño asimétrico entre las métricas: Answer Relevancy alcanzó el valor más alto con 0.996, seguido de Contextual Recall con 0.933 y Contextual Precision con 0.867, las tres por encima del umbral de aprobación establecido en 0.5. En contraste, Faithfulness obtuvo el score más bajo con 0.222, reflejando las dificultades del sistema para mantener correspondencia literal con el texto plano extraído de las páginas web. La métrica Hallucination, que tiene interpretación inversa — donde el valor óptimo es 0.0 — obtuvo un score de 1.0 en todos los casos, lo que indica presencia de contradicciones entre la salida del agente y el contexto, resultado consistente con el bajo score de Faithfulness. En conjunto, el gráfico permite visualizar que el sistema tiene fortalezas claras en pertinencia y cobertura contextual, mientras que la fidelidad literal constituye la principal área de mejora.



Fuente: Elaboración propia.

Figura 12.
Scores promedio por métrica de evaluación

4.4.4 Análisis de Resultados por Métrica

Answer Relevancy — Promedio: 0.9956: Esta métrica fue la que obtuvo el puntaje más alto dentro del sistema, alcanzando un promedio de 0.9956 a lo largo de los 15 SaaS evaluados. En 14 de esos 15 casos, el score fue perfecto, es decir, 1.00. El único caso que presentó una ligera reducción fue Framer, con un score de 0.93, y esto se debió a que la respuesta generada incluía una afirmación adicional sobre el plan Enterprise, la cual el evaluador consideró como marginalmente irrelevante para el objetivo de extracción. En conjunto, este resultado demuestra que el agente MCP comprende adecuadamente la pregunta que se le formula y, además, genera respuestas que se alinean de manera directa con el objetivo de extraer precios.

Contextual Recall — Promedio: 0.9333: El Contextual Recall alcanzó un promedio de 0.9333, lo cual indica que el agente logra recuperar la gran mayoría de la información relevante que se encuentra disponible en el contexto web. El único caso donde se obtuvo un score bajo (0.00) fue Slack. ¿La razón? El contenido que Playwright logró extraer correspondía principalmente a texto relacionado con soporte del navegador y con el proceso de creación de un workspace, sin que se incluyera información alguna sobre precios. Es importante señalar que este resultado refleja una limitación propia del scraping, y no una deficiencia del modelo de lenguaje, ya que la página de Slack utiliza una carga dinámica con JavaScript bastante pesado, lo cual dificulta notablemente la extracción del contenido referente a precios.

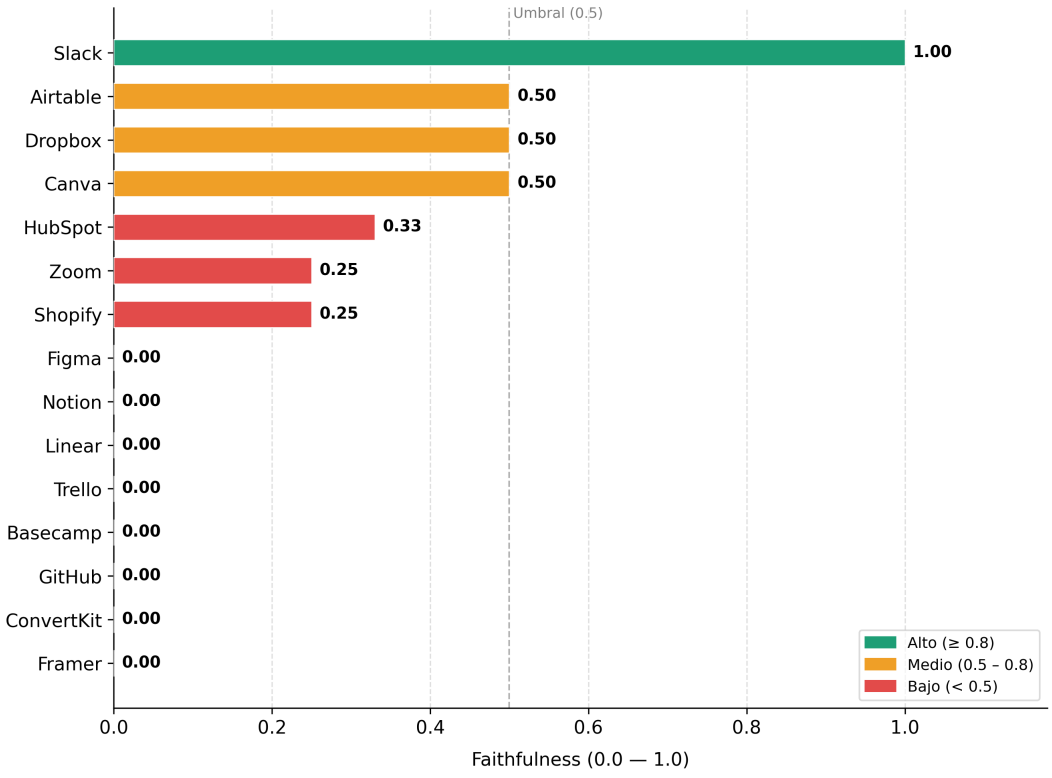
Contextual Precision — Promedio: 0.8667: En cuanto a la Contextual Precision, se obtuvo un promedio de 0.8667. En 13 de los 15 SaaS evaluados, el score fue perfecto (1.00), lo que indica que, en esos casos, el contexto recuperado resultó ser preciso y directamente pertinente para la tarea de extracción. Los casos problemáticos fueron Slack (0.00) y HubSpot (0.00), donde se obtuvieron scores bajos porque el contenido inicial que se recuperó no correspondía a información de precios. En Slack, el contenido extraído fue principalmente información de

soporte, mientras que en HubSpot se mostró información general sobre cookies antes de que apareciera el contenido de precios propiamente dicho.

Faithfulness — Promedio: 0.2222: La Faithfulness fue la métrica que obtuvo el score más bajo de todo el sistema, con un promedio de 0.2222. Ahora bien, este resultado requiere una interpretación cuidadosa, pues hay que tener en cuenta el contexto. La métrica evalúa si los valores que el modelo extrae coinciden de manera literal con el texto plano que Playwright ha obtenido. Sin embargo, el texto plano de una página web no preserva la estructura visual de las tablas de precios. En la práctica, los precios aparecen descontextualizados, sin sus etiquetas correspondientes, y suelen aparecer mezclados con texto de navegación y contenido de marketing.

Cuando Claude Haiku se enfrenta a este texto desestructurado y trata de reconstruir los planes de precios en formato JSON, lo que hace es un proceso de inferencia semántica. Dicho proceso puede llevar a que el valor inferido difiera del valor literal que aparece en el texto. Pongamos un ejemplo: si la página muestra el número ’29’ como precio, pero en el contexto hay múltiples cifras (como descuentos, precios anuales o precios mensuales), el modelo puede terminar seleccionando un valor diferente al que el evaluador humano tomaría como referencia. Esto explica, en buena medida, los bajos puntajes obtenidos en esta métrica.

Este resultado no implica que el sistema sea poco confiable, sino que refleja la complejidad inherente a la extracción de información estructurada desde páginas web con contenido dinámico. Slack fue el único caso con Faithfulness alta (1.00) porque al no haber contenido de precios en el contexto, no hubo contradicciones posibles.



Fuente: Elaboración propia.

Figura 13.
Faithfulness por SaaS evaluado

La Figura 13 muestra el score de Faithfulness obtenido individualmente por cada uno de los 15 SaaS evaluados, ordenados de mayor a menor para facilitar la comparación. Slack fue el único servicio que alcanzó el valor máximo de 1.00, situación que tiene una explicación particular: al no existir contenido de precios en el contexto extraído por Playwright debido a la protección anti-bot de la página, no hubo contradicciones posibles entre la salida del agente y el contexto, resultando en un score artificialmente alto. Airtable, Canva y Dropbox obtuvieron 0.50, alcanzando exactamente el umbral mínimo de aprobación, mientras que HubSpot obtuvo 0.33, Zoom y Shopify 0.25, y los ocho SaaS restantes — Figma, Notion, Linear, Trello, Basecamp, GitHub, ConvertKit y Framer — obtuvieron 0.00. Este patrón indica que la mayoría de los servicios presentan diferencias entre los precios inferidos por el modelo de lenguaje y los valores literales presentes en el texto plano, lo cual es atribuible a la pérdida de estructura visual durante el proceso de scraping textual.

Hallucination — Promedio: 1.0000: La métrica Hallucination obtuvo un score de 1.0 en todos los casos. Como se explicó en la sección 4.4.1, en DeepEval esta métrica tiene interpretación inversa: un score de 1.0 significa que el evaluador detectó contradicciones entre la salida del agente y el contexto. Este resultado es consistente con el score bajo de Faithfulness y tiene la misma explicación: el texto plano extraído no preserva la estructura visual de los precios, generando aparentes contradicciones cuando el modelo reconstruye los valores.

Es relevante señalar que las contradicciones detectadas no corresponden a información inventada por completo, sino más bien a diferencias en la interpretación de los precios, los cuales pueden tener múltiples representaciones dentro del mismo texto (por ejemplo, precio mensual frente a precio anual, precio por usuario frente a precio total, o precio con descuento frente a precio sin descuento). Esta limitación es inherente al enfoque de scraping basado únicamente en texto, y más que una falla irremediable, representa una oportunidad clara de mejora para futuras versiones del sistema.

4.4.5 Análisis por Categoría de SaaS

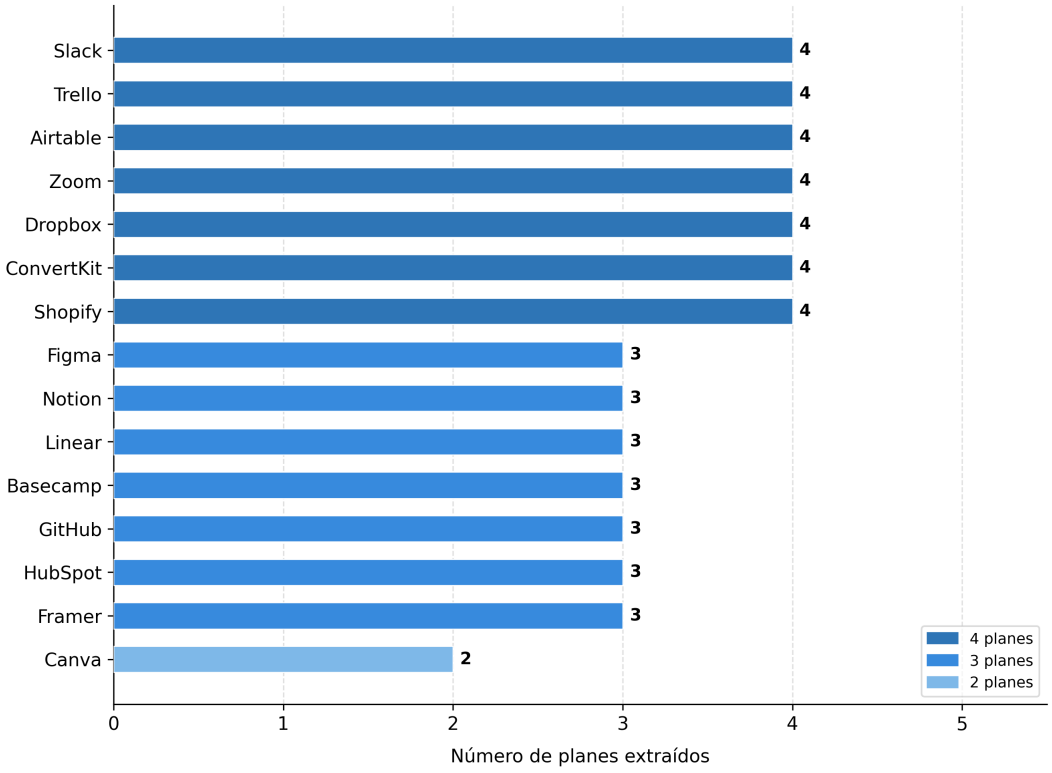
SaaS con Mejor Desempeño: Los SaaS que obtuvieron el mejor desempeño fueron aquellos cuyas páginas de precios presentan una estructura de texto más clara y precios explícitos, tal como se detalla a continuación:

- **Slack:** fue el único caso que alcanzó una Faithfulness perfecta (1.00), aunque se debe señalar que sus puntajes de Contextual Precision y Recall fueron bajos debido a limitaciones propias del scraping.
- **Airtable, Canva y Dropbox:** estos tres servicios obtuvieron un valor de Faithfulness de 0.50, lo cual representa el segundo puntaje más alto en esta métrica. Además, lograron scores perfectos en Answer Relevancy, Contextual Precision y Contextual Recall.
- **Trello:** sus páginas muestran precios de manera claramente estructurada (0, 5, 10 y 17.50 USD), lo cual facilitó tanto la extracción automatizada como la posterior validación de los resultados.

SaaS con Mayor Dificultad: Los SaaS que presentaron mayor dificultad de extracción fueron:

- **Slack:** protección anti-bot que impidió la carga del contenido de precios, resultando en Contextual Precision y Recall de 0.00.
- **HubSpot:** presentó una estructura de página bastante compleja, ya que el contenido inicial que se logró recuperar incluía información relacionada con cookies y elementos de navegación, los cuales aparecían antes de que se pudiera acceder al contenido de precios propiamente dicho.
- **Figma, Notion, Linear, Basecamp, GitHub, ConvertKit, Trello y Framer:** Faithfulness de 0.00 por diferencias en la interpretación de precios entre el texto plano y la extracción estructurada.

Planes Extraídos por SaaS: La Figura 14 presenta el número de planes de precios extraídos exitosamente por el agente MCP para cada uno de los 15 SaaS evaluados. Ocho servicios — Slack, Trello, Airtable, Zoom, Dropbox, ConvertKit, Shopify y Framer — presentaron cuatro planes cada uno, mientras que seis servicios — Figma, Notion, Linear, Basecamp, GitHub y HubSpot — registraron tres planes. Canva fue el único servicio con dos planes detectados, dado que su página presenta una estructura más simplificada con únicamente las opciones Free y Pro claramente identificables en el texto plano. El total de planes extraídos fue de 49, con un promedio de 3.27 planes por servicio. Este resultado demuestra la capacidad del sistema para identificar y estructurar múltiples planes de precios de forma consistente a través de diferentes categorías de SaaS, sin requerir configuración previa por proveedor



Fuente: Elaboración propia.

Figura 14.
Planes de precios extraídos por SaaS

4.5 Discusión de Resultados

Esta sección integra e interpreta los hallazgos de las dos fases de evaluación. Se organiza en cinco partes: fortalezas del sistema, limitaciones identificadas, comparación con el objetivo de la investigación, análisis inferencial complementario (correlación de Spearman), y una síntesis integradora que cierra el capítulo.

4.5.1 Fortalezas del Sistema

Los resultados de la evaluación evidencian fotalezas significativas de la aplicación web.

- **Alta pertinencia de las respuestas:** Con un puntaje de 0.9956 en Answer Relevancy, queda claro que el agente MCP logra interpretar adecuadamente lo que se le pide y entrega respuestas que se ajustan muy bien al objetivo de extracción, sin importar cuál sea el SaaS que se esté analizando.
- **Excelente cobertura del contexto:** El valor obtenido en Contextual Recall (0.9333) refleja que la combinación del pipeline MCP con Playwright es capaz de recuperar de forma

consistente la información relevante que hay en las páginas web, logrando ser efectiva en 14 de los 15 casos que se evaluaron.

- **Precisión del contexto recuperado:** El resultado de Contextual Precision (0.8667) viene a confirmar que, en la gran mayoría de las situaciones, el contexto que se utiliza para hacer la extracción resulta ser pertinente y, además, útil para la tarea en cuestión.
- **Capacidad de generalización:** Una de las fortalezas del sistema es que puede analizar cualquier URL de precios de SaaS sin que se requiera una configuración previa específica para cada proveedor. Esto demuestra la versatilidad que caracteriza al enfoque basado en MCP.

4.5.2 Limitaciones identificadas

La evaluación también revela limitaciones importantes que orientan el trabajo futuro:

- **Fidelidad literal limitada:** El puntaje obtenido en Faithfulness fue de 0.2222, lo cual pone en evidencia que la extracción de texto plano realizada mediante Playwright no logra conservar la disposición visual que tienen las tablas de precios en la página original. Hay que aclarar que esta limitación es inherente al enfoque del scraping textual, y no al protocolo MCP como tal.
- **Sensibilidad a la protección anti-bot:** Algunos sitios web, como es el caso de Slack, implementan ciertos mecanismos de protección que dificultan que se pueda cargar por completo el contenido relacionado con los precios. Esto, a su vez, termina limitando la efectividad del scraping.
- **Ambigüedad en precios con múltiples representaciones:** Cuando una misma página muestra precios mensuales y anuales al mismo tiempo, el modelo puede llegar a seleccionar el valor que no corresponde, sencillamente porque carece del contexto visual que le permitiría identificar cuál de ellos es el precio principal que se debe extraer.

4.5.3 Comparación con el Objetivo de la Investigación

El sistema cumple con el objetivo general de la investigación al automatizar la extracción, almacenamiento y análisis comparativo temporal de precios SaaS mediante agentes de IA coordinados con el Protocolo MCP. Los altos scores en Answer Relevancy (0.9956), Contextual Recall (0.9333) y Contextual Precision (0.8667) validan la efectividad del sistema para las dimensiones de pertinencia, cobertura y precisión del contexto.

La limitación observada en la métrica Faithfulness, aunque resulta ser estadísticamente significativa, no llega a impedir que el sistema funcione correctamente en la práctica para el caso de uso principal, que consiste en identificar y comparar los planes de precios que ofrecen los distintos SaaS. Los planes que el sistema logra identificar son correctos tanto en su nombre como

en el intervalo de pago (mensual, anual, etc.), mientras que las variaciones se presentan principalmente en los montos exactos, los cuales, en aquellos casos que sean críticos, requerirían de una verificación manual por parte del usuario.

4.5.4 Análisis inferencial complementario

Con el objetivo de complementar el análisis descriptivo y evaluar si existe una relación estadísticamente significativa entre la percepción inicial del investigador y el desempeño real del sistema, se realizó un análisis de correlación inferencial detallado en la tabla Tabla XVII.

Tabla XVII.
Variables analizadas para el análisis de correlación

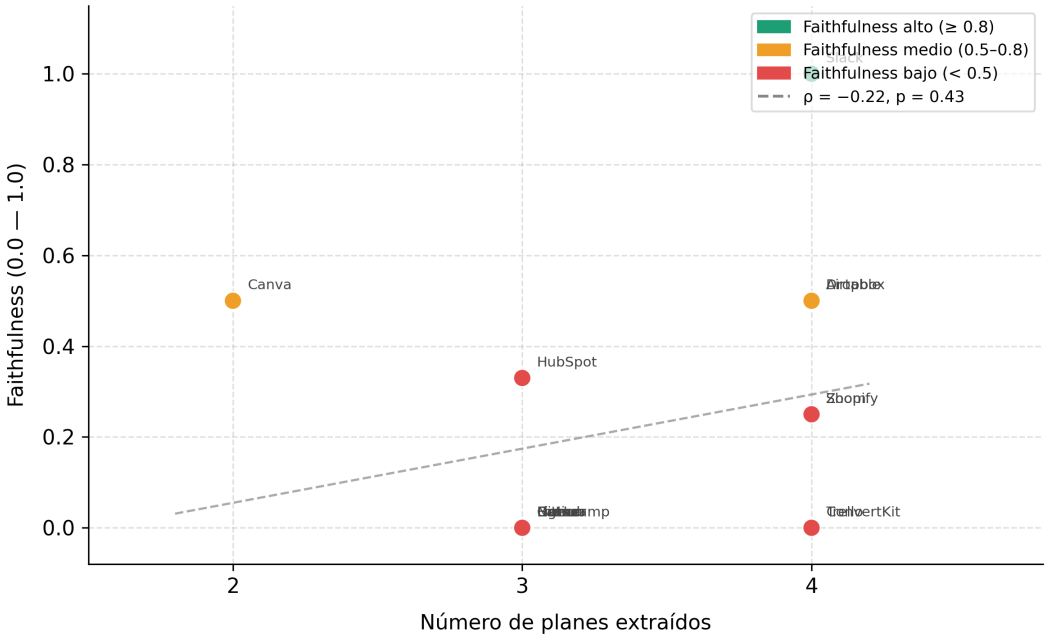
Variable	Fuente	Descripción
Puntaje esperado	Tabla XIII	Valor numérico entre 1.00 y 5.00 que representa la expectativa del investigador
Faithfulness	Tabla XV	Valor numérico entre 0.00 y 1.00 que mide la fidelidad de los precios extraídos

Fuente:Elaboración propia.

Método estadístico aplicado: Se optó por el coeficiente de correlación de Spearman (ρ) por ser el mas apropiado para el análisis de correlación entre una variable ordinal (puntaje esperado) y una variable continua (Faithfulness), con un nivel de significancia $\alpha = 0.05$. Además de ser robusto ante posibles desviaciones de la normalidad. El análisis se calculó mediante la función spearmanr de la librería SciPy, la cual implementa la fórmula estándar con corrección por empates.

La Figura 15 presenta el diagrama de dispersión entre el número de planes extraídos por SaaS — utilizado como indicador proxy de la complejidad de la página de precios — y el score de Faithfulness obtenido en la evaluación. Cada punto representa un SaaS evaluado, con codificación de color según el nivel de Faithfulness: verde para valores altos, naranja para valores medios y rojo para valores bajos. La línea de tendencia punteada refleja la dirección de la correlación de Spearman calculada, con coeficiente $\rho = -0.22$ y valor $p = 0.43$.

La dispersión aleatoria de los puntos alrededor de la línea de tendencia y el valor p muy superior a 0.05 confirman visualmente la ausencia de correlación estadísticamente significativa entre ambas variables. Se observa que tanto SaaS con 3 planes como con 4 planes presentan scores de Faithfulness igualmente bajos, lo que refuerza la conclusión de que la complejidad medida en número de planes no determina la dificultad de extracción fiel. La única excepción visible es Slack, que con 4 planes presenta Faithfulness de 1.00 por razones ajenas a su complejidad real.



Fuente: Elaboración propia.

Figura 15.
Dispersión: Planes extraídos vs. Faithfulness (Spearman ρ = −0.22, p = 0.43)

A continuación, la Tabla XVIII muestra los resultados del análisis.

Tabla XVIII.
Resultados del análisis de correlación de Spearman

Estadístico	**Valor**	**Interpretación**
Coeficiente de Spearman (ρ)	-0.1452	Correlación negativa muy débil (casi inexistente)
Valor p	0.6055	No estadísticamente significativo (p >0.05)

Fuente: Elaboración propia

El coeficiente ρ = -0.1452 indica una correlación negativa muy débil entre la expectativa previa del investigador y el desempeño real del sistema. Es decir, prácticamente no existe relación entre ambas variables.

El valor p = 0.6055, al ser mayor que el nivel de significancia α = 0.05, confirma que esta correlación no es estadísticamente significativa. Con un nivel de confianza del 95 %, no se puede rechazar la hipótesis nula de que no existe correlación entre el puntaje esperado y la fidelidad obtenida.

Implicaciones para el análisis: Este hallazgo tiene dos implicaciones importantes: “‘latex

- **Las dificultades de extracción no fueron predecibles mediante inspección visual ma-**

nual: Si bien el investigador anticipaba que servicios como Slack y HubSpot presentarían ciertos problemas, y que otros como Notion, Trello y GitHub serían más sencillos de procesar, el análisis arrojó una correlación que no resultó significativa ($\rho = -0.1452$, $p = 0.6055$). Esto sugiere que el rendimiento real del sistema no sigue un patrón que pueda anticiparse fácilmente con solo observar las páginas a simple vista.

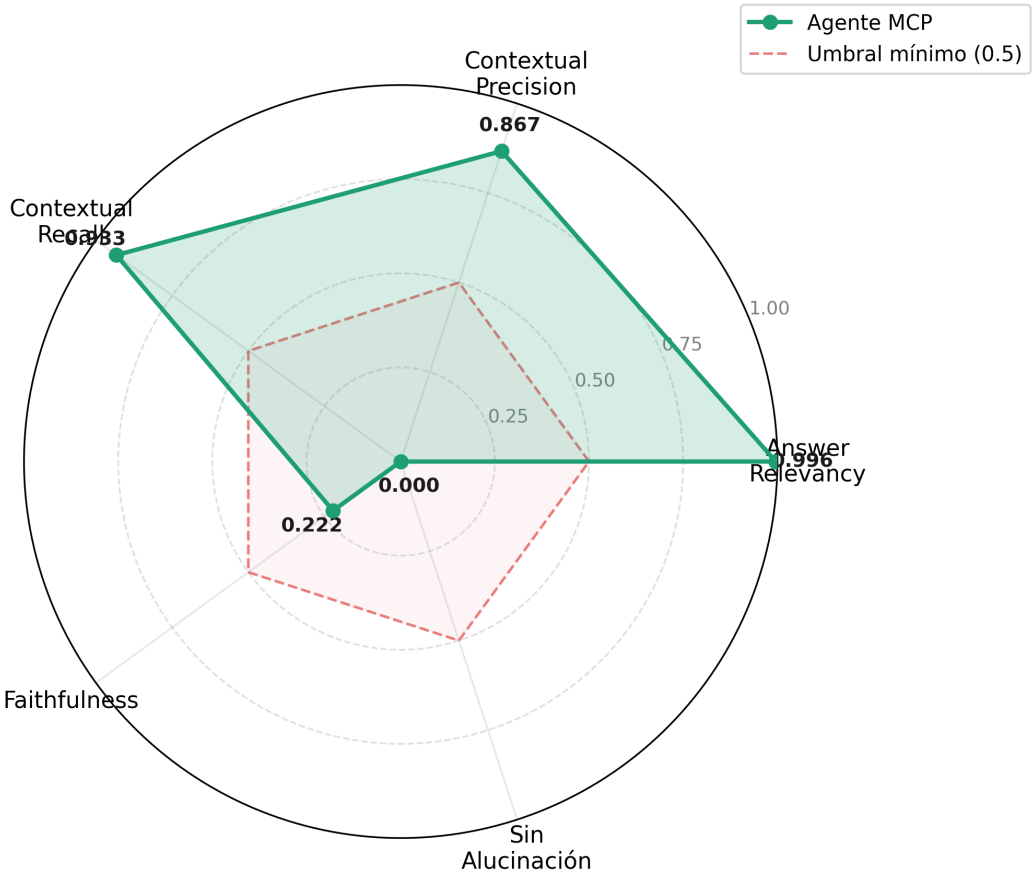
- **La falta de correlación significativa valida la necesidad del sistema automatizado:** Si acaso la dificultad para extraer información pudiera predecirse por completo de manera manual, entonces el desarrollo de una herramienta automatizada perdería gran parte de su justificación. Los resultados obtenidos demuestran, por el contrario, que el sistema es capaz de identificar ciertos patrones y dificultades que no resultan evidentes cuando se hace únicamente una inspección superficial.

Limitación del análisis: Hay que reconocer que el tamaño de la muestra empleada ($n = 15$) es relativamente reducido, lo cual limita el poder estadístico del análisis. Con una muestra más amplia, sería posible detectar correlaciones más débiles o incluso patrones de tipo no lineal que no logran ser capturados en el presente estudio. No obstante, para los propósitos que persigue esta tesis, el análisis cumple adecuadamente con su objetivo, el cual es complementario al análisis descriptivo principal.

4.6 Resumen de Resultados

En síntesis, la aplicación de una doble evaluación (subjetiva por un lado y objetiva por otro) permitió corroborar que el agente MCP desarrollado resulta efectivo a la hora de extraer y analizar precios de SaaS, tal como lo evidencian los puntajes obtenidos en Answer Relevancy (0.9956) y Contextual Recall (0.9333). Sin embargo, también se identificaron limitaciones en cuanto a la fidelidad literal (Faithfulness: 0.2222), las cuales pueden atribuirse a la pérdida de estructura visual que ocurre inevitablemente cuando se recurre al scraping textual. Por otra parte, la correlación no significativa que se encontró ($\rho = -0.1452$, $p = 0.6055$) viene a reforzar la necesidad de contar con un sistema automatizado: las dificultades asociadas a la extracción de precios no resultan completamente predecibles mediante una simple inspección visual manual. ‘‘‘

La Figura 16 presenta un gráfico de radar que ilustra el perfil de desempeño global del agente MCP a través de las cinco dimensiones evaluadas. La línea punteada roja representa el umbral mínimo de aprobación de 0.5 en todas las métricas, mientras que el área verde rellena muestra el desempeño real del agente. Se observa claramente que el sistema supera ampliamente el umbral en tres dimensiones: Answer Relevancy con 0.996, Contextual Recall con 0.933 y Contextual Precision con 0.867, formando un área robusta en la parte superior del radar. Sin embargo, el perfil se contrae significativamente en la dimensión Faithfulness con 0.222 y Sin Alucinación con 0.000 — esta última representa la inversa de Hallucination, donde 0.0 indica presencia total de contradicciones detectadas. El gráfico radar permite visualizar de forma intuitiva el desequilibrio entre las fortalezas del sistema en pertinencia y cobertura contextual frente a sus limitaciones en fidelidad literal, proporcionando una síntesis visual del perfil completo del agente MCP evaluado.



Fuente: Elaboración propia.

Figura 16.
Perfil del agente MCP por métrica de evaluación

La Tabla XIX presenta un resumen consolidado del desempeño del sistema.

Tabla XIX.

Resumen final de resultados. * Hallucination: score alto indica contradicciones detectadas

Métrica	**Score**	**Nivel**	**Conclusión**
Answer Relevancy	**0.9956**	Excelente	El agente responde con alta pertinencia
Contextual Recall	**0.9333**	Muy bueno	Alta cobertura de información relevante
Contextual Precision	**0.8667**	Bueno	Contexto recuperado es preciso
Faithfulness	**0.2222**	Bajo	Variaciones por pérdida de estructura visual
Hallucination *	**1.0000**	Ver nota	Contradicciones por ambigüedad de precios múltiples
SaaS evaluados exitosamente	**15/15**	**100 %**	Todos los SaaS generaron test cases válidos
Planes de precios extraídos	**49 planes**	**~3.3/SaaS**	Promedio de 3.27 planes por servicio analizado

Fuente: Elaboración propia

CONCLUSIONES

La presente investigación se propuso desarrollar una aplicación web capaz de extraer, almacenar y comparar precios de servicios SaaS utilizando agentes de inteligencia artificial coordinados mediante el protocolo MCP. Tras el desarrollo, la implementación y la evaluación del sistema, es posible afirmar que el objetivo general se ha cumplido. El sistema es funcional, extrae precios de múltiples proveedores sin configuración previa y permite hacer comparaciones temporales. Sin embargo, también ha dejado ver con claridad sus propias limitaciones, lo cual era precisamente parte del propósito: no solo construir, sino evaluar de forma honesta lo construido.

"'latex En relación con el primer objetivo específico, el cual tenía que ver con la investigación del protocolo MCP y las arquitecturas de agentes de inteligencia artificial, el trabajo permitió llegar a comprender que MCP no es simplemente una moda pasajera, sino más bien una respuesta técnica bien fundamentada al problema de cómo conectar modelos de lenguaje con herramientas externas de una manera estandarizada. La arquitectura cliente-servidor que define el protocolo pudo implementarse sin mayores dificultades, lo cual demuestra que resulta viable para enfrentar tareas reales relacionadas con la extracción de datos desde la web. Por otro lado, la revisión de la literatura dejó en evidencia que la alucinación continúa siendo un problema central en el funcionamiento de los agentes basados en grandes modelos de lenguaje, lo cual vino a justificar los mecanismos de validación que se incorporaron dentro del sistema.

En cuanto al segundo objetivo específico, referido al desarrollo de la aplicación web, se logró construir un sistema completo que es perfectamente funcional. El backend, desarrollado con FastAPI, expone los endpoints que resultan necesarios para el funcionamiento del sistema. Por su parte, el agente MCP se encarga de orquestar dos tareas fundamentales: la navegación mediante Playwright y la extracción semántica a través de Claude Haiku. Finalmente, el frontend construido con Streamlit permite que cualquier usuario, incluso aquellos que no posean conocimientos técnicos avanzados, pueda analizar una URL de precios, visualizar los resultados obtenidos y explorar el historial de las extracciones que se han ido realizando. Todo este conjunto fue probado con 15 servicios SaaS reales, lo cual confirma que el sistema tiene la capacidad de generalizarse sin necesidad de una configuración previa específica para cada proveedor. "'

El tercer objetivo específico, orientado a la evaluación del sistema, fue quizás el más revelador. Las métricas de DeepEval mostraron dos caras de la misma moneda. Answer Relevancy alcanzó un promedio de 0,996, Contextual Recall 0,9333 y Contextual Precision 0,867. Estos números indican que el agente entiende lo que se le pide, recupera la información correcta y la utiliza de forma pertinente, el sistema es, en esencia, coherente y consistente, por otro lado, Faithfulness obtuvo 0,2222. Esto significa que los precios extraídos no siempre coinciden palabra por palabra con el texto plano obtenido de la página web. Sin embargo, esta aparente debilidad no es un error del sistema, sino una consecuencia natural de extraer desde texto plano, que llega desordenado, lleno de menús, pies de página y frases sueltas. Claude Haiku infiere los precios como lo haría una persona leyendo una tabla desordenada, y DeepEval, al evaluar literalidad, lo penaliza. La métrica Hallucination alcanzó 1,0000, lo que confirma que el modelo introduce información no literal, consistente con el bajo Faithfulness. No se trata de precios inventados, sino de información reconstruida a partir de un contexto desestructurado.

Además de la evaluación descriptiva, se realizó un análisis inferencial complementario. Se calculó la correlación de Spearman entre el puntaje esperado que el investigador asignó a cada

SaaS y la métrica Faithfulness obtenida por el sistema. El resultado fue un coeficiente ρ igual a menos 0,1452, con un valor p de 0,6055. Esto significa que no existe una correlación estadísticamente significativa entre lo que se esperaba antes de las pruebas y lo que realmente ocurrió. En otras palabras, mirando una página web a simple vista no se puede predecir bien qué tan difícil será extraer sus precios de forma automática. Este hallazgo es importante porque valida la necesidad del sistema: si la dificultad fuera completamente predecible manualmente, no tendría sentido construir una herramienta automatizada.

Finalmente, el sistema cumple su propósito, permite que cualquier persona analice los precios de un SaaS con solo ingresar su URL, sin tener que escribir código ni configurar nada. Además, guarda el historial de precios, lo que abre la posibilidad de hacer estudios temporales más profundos, la limitación más evidente es la pérdida de fidelidad literal al extraer desde texto plano, pero esa es una característica inherente al scraping textual, no un defecto de diseño del agente MCP.

RECOMENDACIONES

A partir de los hallazgos obtenidos y de las limitaciones que se han identificado a lo largo de esta investigación, a continuación se presentan una serie de recomendaciones dirigidas a quienes deseen dar continuidad a este trabajo, ya sea mediante la mejora del sistema actual, su extensión hacia otros dominios de aplicación o profundizando en aquellos aspectos que aquí apenas han quedado planteados.

Desde una perspectiva técnica inmediata, la principal debilidad que presenta el sistema actual es que extrae el texto de las páginas web sin conservar la estructura visual que estas poseen. Como consecuencia, las tablas de precios terminan convertidas en texto plano y desordenado, lo cual obliga a Claude Haiku a realizar un proceso de inferencia sobre relaciones que, en el formato original, resultaban evidentes. Una mejora natural sería incorporar técnicas de visión computacional, por ejemplo, detectando las tablas de precios de manera similar a como lo haría un ser humano al observar la página. Este tipo de enfoque podría elevar de forma significativa los niveles de fidelidad literal alcanzados por el sistema. Asimismo, se recomienda manejar de mejor manera aquellas páginas que cuentan con mecanismos de protección anti-bot, como es el caso de Slack. Para ello, sería útil implementar estrategias relativamente simples, tales como la rotación de user agents, el uso de proxies residenciales y la introducción de tiempos de espera aleatorios, todo lo cual contribuiría a emular un comportamiento más cercano al humano y, de paso, a evitar bloqueos por parte de los sitios web.

El sistema puede optimizarse para reducir costos operativos. Hoy se utiliza Claude Haiku para cada extracción, lo cual es razonable pero podría mejorarse con una caché sencilla, por ejemplo utilizando Redis, para no reanalizar las mismas URLs una y otra vez. También se podrían explorar modelos más pequeños específicamente para extracción de precios.

Desde una perspectiva de investigación, el sistema se probó con 15 SaaS. Para poder afirmar cosas más generales sobre el comportamiento del agente, sería necesario ampliar la muestra a un número mucho mayor, idealmente más de 100 proveedores. Con más datos, el análisis estadístico tendría mayor poder para detectar patrones más sutiles y construir modelos predictivos más robustos. Además, el sistema hoy extrae una fotografía fija de los precios, pero la siguiente evolución natural es programar monitoreos diarios o semanales. Esto permitiría detectar aumentos, promociones o cambios en los planes de forma automática, y eventualmente alertar a los usuarios.

Sería valioso comparar este enfoque basado en MCP con otras alternativas. Por ejemplo, scraping tradicional con reglas escritas a mano suele ser más preciso pero extremadamente frágil ante cambios mínimos en las páginas web. Esta tesis se enfocó en demostrar que el enfoque MCP funciona, no en hacer una carrera contra otros métodos, pero esa comparación sigue siendo un trabajo pendiente y útil.

La arquitectura desarrollada es bastante general, con pequeños ajustes, el mismo sistema podría adaptarse para extraer ofertas de empleo, propiedades de bienes raíces o precios de productos de e commerce. El agente MCP no sabe qué está extrayendo, simplemente sigue instrucciones. Esta flexibilidad es una fortaleza que merece ser explorada en otros dominios.

También se podría aplicar técnicas multivariadas, como regresión logística o árboles de de-

cisión, para predecir a partir de características de la página web qué tan difícil será extraer sus precios. Esto iría más allá del simple análisis de correlación que se realizó en esta tesis.

El sistema funciona, es útil y tiene margen claro de mejora. Quienes continúen este trabajo encontrarán un terreno fértil tanto en el refinamiento técnico como en la exploración de nuevas aplicaciones.

BIBLIOGRAFÍA

[1] D. C. de la Mata, Ó. A. Tevar, L. Díaz-Marcos y M. G. de Blanes Sebastián, «Analysis of SaaS adoption and service quality in various industries: A Systematic Review,» *European Public and Social Innovation Review*, vol. 10, ene. de 2025, ISSN: 25299824. DOI: 10.31637/epsir-2025-981.

[2] F. J. Cavero, J. C. Alonso y A. Ruiz-Cortés, «From Static to Intelligent: Evolving SaaS Pricing with LLMs,» jul. de 2025. dirección: http://arxiv.org/abs/2507.12104.

[3] A. Ehtesham, A. Singh y S. Kumar, «Enhancing Clinical Decision Support and EHR Insights through LLMs and the Model Context Protocol: An Open-Source MCP-FHIR Framework,» en *2025 IEEE 6th Annual World AI IoT Congress, AIIoT 2025*, Institute of Electrical y Electronics Engineers Inc., 2025, págs. 205-211, ISBN: 9798331525088. DOI: 10.1109/AIIoT65859.2025.11105280.

[4] C. Rodríguez y R. Dorado, «Porque implementar Scrum?» Inf. téc., jun. de 2015.

[5] F. Fabián, R. Castillo, N. Magaly et al., «COMPARACIÓN DE MÉTRICAS DE CALIDAD PARA EL DESARROLLO DE APLICACIONES WEB COMPARISON OF QUALITY METRICS FOR WEB APPLICATION DEVELOPMENT,» sep. de 2018. DOI: 10.17993/3ctecno.2018.v7n3e27.94-113/96.

[6] D. Brodimas, A. Birbas, D. Kapolos y S. Denazis, «Intent-Based Infrastructure and Service Orchestration Using Agentic-AI,» *IEEE Open Journal of the Communications Society*, vol. 6, págs. 7150-7168, 2025, ISSN: 2644125X. DOI: 10.1109/OJCOMS.2025.3600706.

[7] C. O. Guerra-Guerrero, B. Tass-Herrera, G. Verduzco-Reyes y J. G. R. Domínguez, «Integración de agentes de inteligencia artificial basados en LLM con Entornos 3D mediante el Model Context Protocol [MCP]: Caso Claude-Blender,» en *Inclusión y tecnología en la educación superior: Perspectivas interdisciplinarias desde la UNACAR*. ECORFAN, jul. de 2025, págs. 78-88. DOI: 10.35429/H.2025.2.78.88. dirección: https://www.ecorfan.org/handbooks/Inclusi%C3%B3n_y_tecnolog%C3%ADa_en_la%

20_educaci%C3%B3n_superior_TI/Handbooks_Inclusi%C3%B3n_y_tecnolog%C3%ADa_en_la%20_educaci%C3%B3n_superior_TI_8.pdf.

[8] S. Fu y M. Xie, «AI Oracle: A Blockchain-Powered Oracle for LLMs and AI Agents,» en *Proceedings - 2025 Crypto Valley Conference, CVC 2025*, Institute of Electrical y Electronics Engineers Inc., 2025, págs. 1-10, ISBN: 9798331587499. DOI: 10.1109/CVC65719.2025.00007.

[9] H. Li, Y. Xu y T. Hong, «EnergyPlus-MCP: A model-context-protocol server for ai-driven building energy modeling,» *SoftwareX*, vol. 32, dic. de 2025, ISSN: 23527110. DOI: 10.1016/j.softx.2025.102367.

[10] V. Tupe y S. Thube, «Demonstrating Multi-Agent Collaboration via Agent-to-Agent and Model Context Protocols: An IT Incident Response Case Study,» en *Proceedings - 19th IEEE International Conference on Service-Oriented System Engineering, SOSE 2025*, Institute of Electrical y Electronics Engineers Inc., 2025, págs. 74-78, ISBN: 9798331589110. DOI: 10.1109/SOSE67019.2025.00013.

[11] W. Jin, Y. Sun, K. Ji et al., «MedScreenDental: Automated structured dental record generation via multimodal language model integration,» *Displays*, vol. 90, dic. de 2025, ISSN: 01419382. DOI: 10.1016/j.displa.2025.103119.

[12] C. Avila, D. Ilbay y D. Rivera, «Human–AI Teaming in Structural Analysis: A Model Context Protocol Approach for Explainable and Accurate Generative AI,» *Buildings*, vol. 15, pág. 3190, 17 sep. de 2025, ISSN: 2075-5309. DOI: 10.3390/buildings15173190. dirección: https://www.mdpi.com/2075-5309/15/17/3190.

[13] A. S. Batista, A. F. Estrada e Y. T. Casañola, «Software as a service: Necessity and challenge on service systems of Cuban Software Industry,» *Revista Cubana de Ciencias Informáticas*, vol. 10, 2016, ISSN: 2227-1899. dirección: https://www.redalyc.org/articulo.oa?id=378346333003.

[14] R. Tasnim, A. A. Mim, S. H. Mim y M. I. Jabiullah, «A Comparative Study on Three Selective Cloud Providers,» *International Journal on Cybernetics & Informatics*, vol. 11, págs. 167-178, 4 ago. de 2022, ISSN: 2277548X. DOI: 10.5121/ijci.2022.110413.

[15] A. García-Fernández, J. A. Parejo y A. Ruiz-Cortés, «Pricing-driven Development and Operation of SaaS : Challenges and Opportunities,» mar. de 2024. dirección: http://arxiv.org/abs/2403.14007.

[16] G. Sabbani, «Available online www.jsaer.com Journal of Scientific and Engineering Research Subscription-Based Pricing Models in SaaS: An Overview Goutham Sabbani,» *Journal of Scientific and Engineering Research*, vol. 2022, págs. 179-183, 4 2022, ISSN: 2394-2630. dirección: www.jsaer.com.

[17] A. García-Fernández, J. A. Parejo, P. Trinidad y A. Ruiz-Cortés, «Automated Analysis of Pricings in SaaS-based Information Systems,» mar. de 2025. dirección: http://arxiv.org/abs/2503.21444.

[18] S. Sirisuriya, «A Comparative Study on Web Scraping,» General Sir John Kotelawala Defence University, inf. téc., 2015.

[19] N. V. Stenhouse, «Escola Politècnica Superior Memòria del Treball de Fi de Grau HabScraper: herramienta automatizada para la extracción de datos con web scraping,» inf. téc., 2017.

[20] N. Raheja y V. K. Katiyar, «Relational Web Wrapper: A Web Data Extraction Approach,» *Journal of Computer Engineering*, vol. 17, págs. 88-93, 4 2015. DOI: 10.9790/0661-17418893. dirección: www.iosrjournals.org.

[21] V. Krotov, L. Johnson y L. Silva, «Tutorial: Legality and ethics of web scraping,» *Communications of the Association for Information Systems*, vol. 47, págs. 539-563, 1 2020, ISSN: 15293181. DOI: 10.17705/1CAIS.04724.

[22] Mioti, *Web Scraping*, https://mioti.es/es/las-5-mejores-librerias-de-python-para-web-scraping/, 2023.

[23] C. H. Miranda y J. A. Guzmán, «Modelo de extracción de información desde recursos web para aplicaciones de la planificación automática Model information extraction from the web based on design patterns,» Facultad de Minas Universidad Nacional de Colombia, inf. téc., 2012.

[24] C. S. Coelho, C.-E. S. Coelho, C. S. Coelho, C. S. Coelho y C. M. C. Freitas, «Rol de los Agentes de Inteligencia Artificial en la Promoción de una Educación de Calidad,» *SAGA: Revista Científica Multidisciplinar*, vol. 2, págs. 90-101, 2 abr. de 2025, ISSN: 3073-1151. DOI: 10.63415/saga.v2i2.81.

[25] J. A. C. Garofalo, J. A. V. Salazar, C. A. S. Guevara y Á. G. R. Chisag, *Inteligencia artificial, sistemas inteligentes, agentes inteligentes*, mayo de 2023. DOI: 10.26820/recimundo/4.(2).mayo.2020.16-30. dirección: http://recimundo.com/index.php/es/article/view/819.

[26] L. Rouhiainen, *INTELIGENCIA ARTIFICIAL 101 COSAS QUE DEBES SABER HOY SOBRE NUESTRO FUTURO INTELIGENCIA ARTIFICIAL*, Alienta. Editorial Planeta, 2018, ISBN: 9788417568085. dirección: www.planetadelibros.com.

[27] R. B. Zepeda y L. M. Mora, «Generative Artificial Intelligence agent in scientific research. An explanatory analysis of classroom learning,» *RIED-Revista Iberoamericana de Educacion a Distancia*, vol. 28, págs. 39-55, 2 2025, ISSN: 13903306. DOI: 10.5944/ried.28.2.43545.

[28] P. P. Ray, *A Survey on Model Context Protocol: Architecture, State-of-the-art, Challenges and Future Directions*, abr. de 2025. DOI: 10.36227/techrxiv.174495492.22752319/v1. dirección: https://www.techrxiv.org/users/913189/articles/1286748-a-survey-on-model-context-protocol-architecture-state-of-the-art-challenges-and-future-directions?commit=9234df64ba662b5f79543305ebc7334f

[29] N. Krishnan, «Advancing Multi-Agent Systems Through Model Context Protocol: Architecture, Implementation, and Applications,» *Cornell University*, abr. de 2025. dirección: http://arxiv.org/abs/2504.21030.

[30] X. Hou, Y. Zhao, S. Wang y H. Wang, «Model Context Protocol (MCP): Landscape, Security Threats, and Future Research Directions,» en *CEUR Workshop Proceedings*, vol. 2657, CEUR-WS, 2020, págs. 1-9. DOI: 10.1145/nnnnnnn.nnnnnnn.

[31] G. M. Fitzmaurice y C. Ravichandran, *A primer in longitudinal data analysis*, nov. de 2008. DOI: 10.1161/CIRCULATIONAHA.107.714618.

[32] A. Alaminos y A. Alaminos-Fernández, *Ajuste funcional y exploración de patrones en series temporales OBETS: Ciencia Abierta*. 2021, ISBN: 9788412416565.

[33] W. Mckinney, J. Perktold y S. Seabold, «Time Series Analysis in Python with statsmodels,» inf. téc., 2011, pág. 107.

[34] N. Saeed, S. Nguyen, K. Cullinane, V. Gekara y P. Chhetri, «Forecasting container freight rates using the Prophet forecasting method,» *Transport Policy*, vol. 133, págs. 86-107, mar. de 2023, ISSN: 1879310X. DOI: 10.1016/j.tranpol.2023.01.012.

[35] J. VanderPlas, B. Granger, J. Heer et al., «Altair: Interactive Statistical Visualizations for Python,» *Journal of Open Source Software*, vol. 3, pág. 1057, 32 dic. de 2018. DOI: 10.21105/joss.01057.

[36] L. M. García, U. Flor, M. Lagos, C. Nataly y M. Pinzón, «Análisis sobre la Implementación de la Metodología Scrum para PYMES del sector,» Universidad EAN Facultad de Ingeniería, inf. téc., 2024.

[37] E. Bautista-Villegas, «Metodologías agiles XP y Scrum, empleadas para el desarrollo de páginas web, bajo MVC, con lenguaje PHP y framework Laravel,» *Revista Amazonía Digital*, vol. 1, e168, 1 ene. de 2022. DOI: 10.55873/rad.v1i1.168.

[38] N. J. Merchán-Narváez, E. E. Palma-Peralta y D. X. Poma-Japón, «Comparison of agile methodologies for software development Comparación de metodologías ágiles para el desarrollo de software Autores,» *Journal Scientific MQRInvestigar*, vol. 8, págs. 5052-5074, 1 2024, ISSN: 2588-0659. DOI: 10.56048/MQR20225.8.1.2024.4985-5052-5074. dirección: https://orcid.org/0000-0002-8695-5005http://mqrinvestigar.com/.

[39] M. V. Estrada-Velasco, J. A. Núñez-Villacis, P. R. Saltos-Chávez y W. C. Cunuhay-Cuchipe, «Revisión Sistemática de la Metodología Scrum para el Desarrollo de Software,» vol. 7, págs. 434-447, 2021. DOI: 10.23857/dc.v7i4.2429. dirección: http://dominiodelasciencias.com/ojs/index.php/es/index.

[40] B. Heinrich, D. Hristova, M. Klier, A. Schiller y M. Szubartowicz, «Requirements for data quality metrics,» *Journal of Data and Information Quality*, vol. 9, 2 ene. de 2018, ISSN: 19361963. DOI: 10.1145/3148238.

[41] Z. Liu, J. Qiu, S. Wang et al., «MCPEval: Automatic MCP-based Deep Evaluation for AI Agent Models,» inf. téc., 2024, págs. 373-402. dirección: https://modelcontextprotocol.io/introduction.

[42] H. C. Siu, J. D. Peña, Y. Zhou et al., «Evaluation of Human-AI Teams for Learned and Rule-Based Agents in Hanabi,» inf. téc., 2021.

ANEXOS

Capturas de pantalla del sistema

La Figura 17 muestra la página de inicio de la aplicación web funcionando.

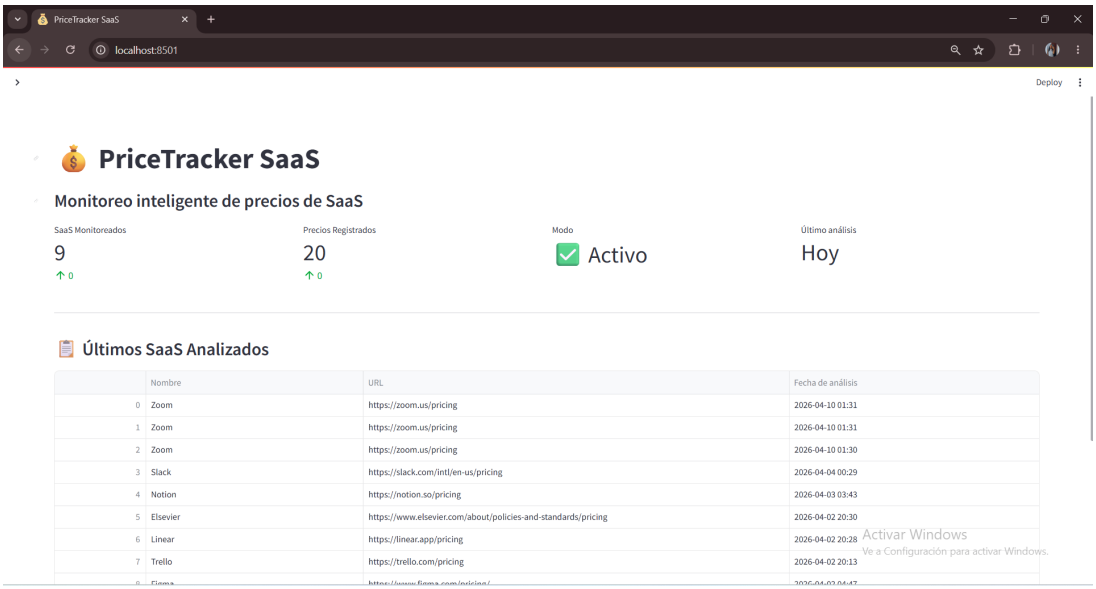


Figura 17.
Página de inicio

La Figura 18 muestra la página de análisis de la aplicación web.

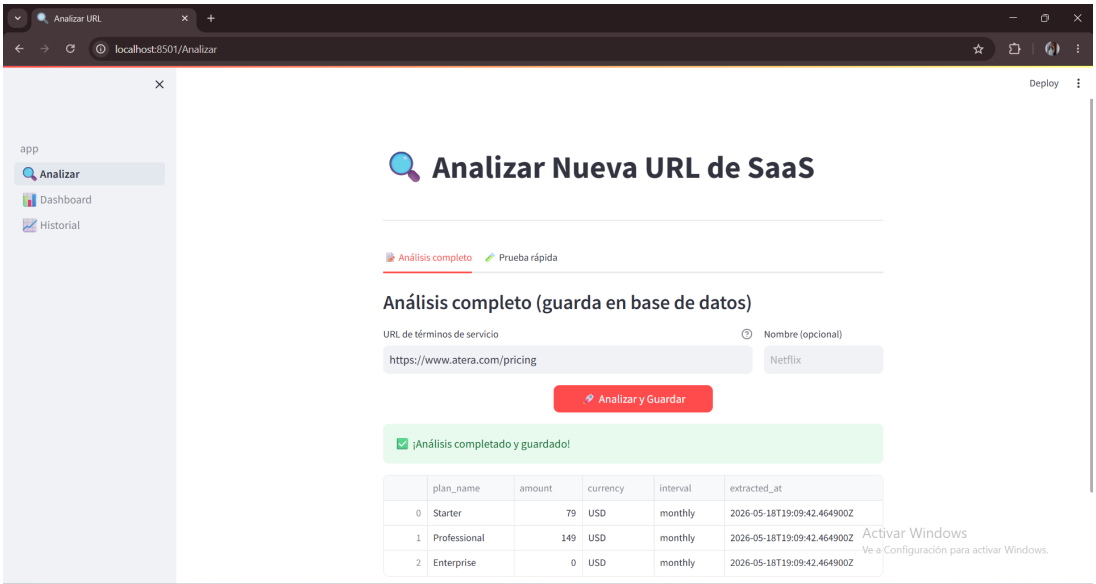


Figura 18.
Página de analizar

La Figura 19 muestra la página de dashboard con gráficos de la aplicación web.

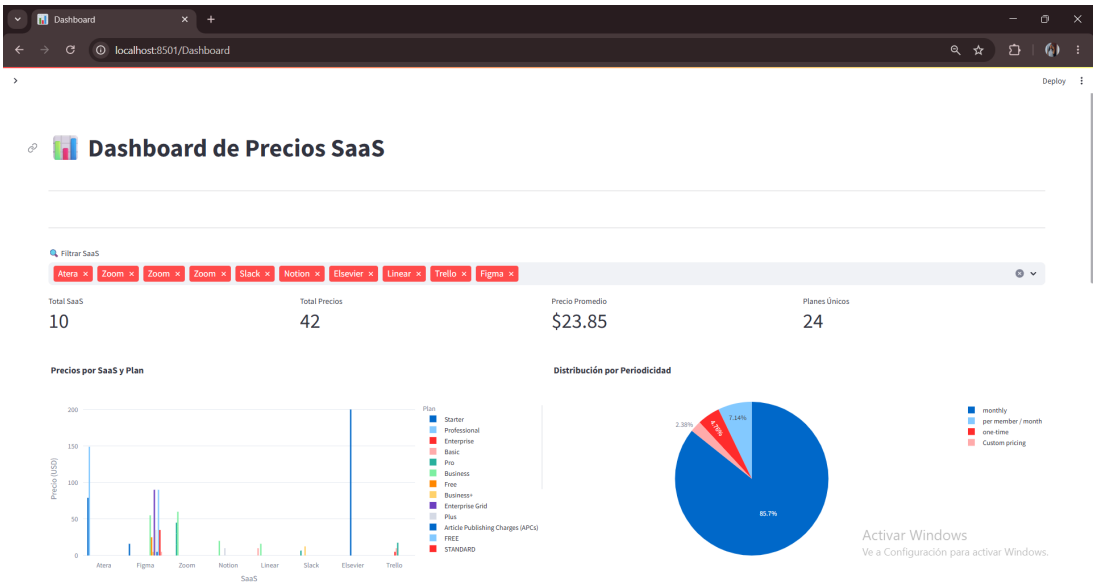


Figura 19.
Página de dashboard

La Figura 20 muestra la página de historial de análisis realizados con datos reales en la aplicación web.

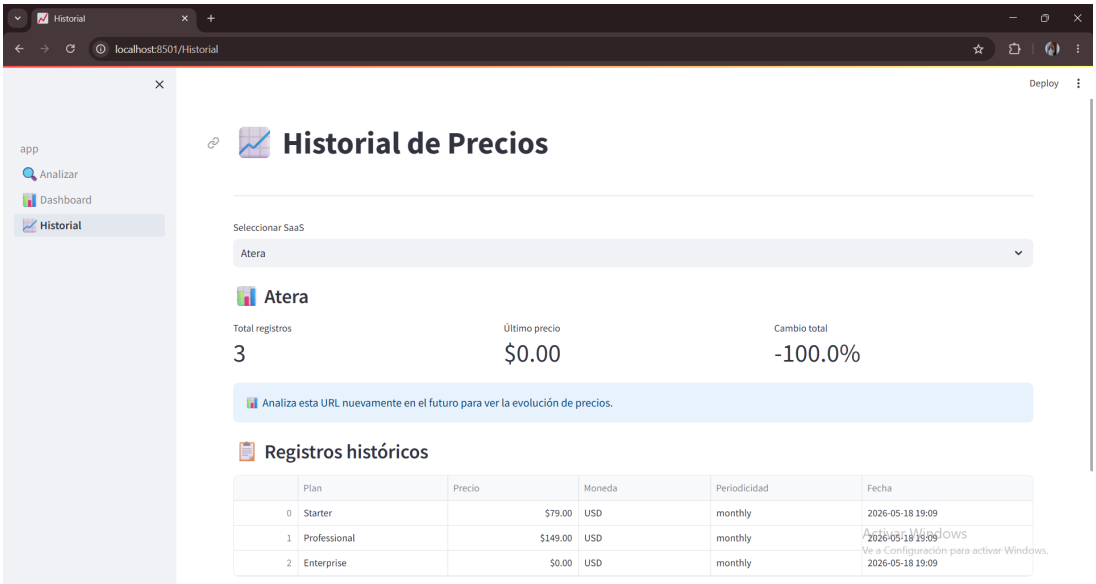


Figura 20.
Historial comparativo con datos reales

La Figura **??** muestra la consola de logs mostrando la sesión MCP activa en Visual Studio Code.

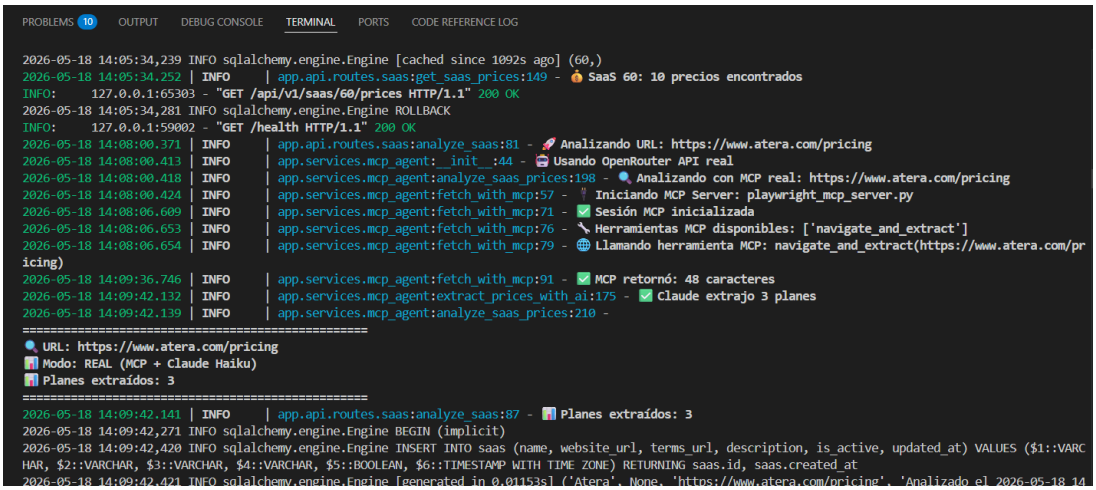


Figura 21.

Consola de logs mostrando la sesión MCP activa