



**Pontificia Universidad
Católica del Ecuador**
Seréis mis testigos

MANABÍ

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

SEDE MANABÍ

CARRERA DE SOFTWARE

TRABAJO DE TITULACIÓN

**IMPLEMENTACIÓN DE UN SISTEMA DE COMUNICACIÓN Y
PROTOCOLO MCP PARA LA OPTIMIZACIÓN DEL ACCESO A
INFORMACIÓN EN EL SISTEMA BILLEASE**

LÍNEA DE INVESTIGACIÓN

TECNOLOGÍA DE LA INFORMACIÓN Y LA COMUNICACIÓN

SUBLÍNEA DE INVESTIGACIÓN

INGENIERÍA DE SOFTWARE, INNOVACIÓN Y EMPRENDIMIENTO EN TIC

PREVIO AL TÍTULO DE

INGENIERO EN SOFTWARE

AUTORES

BYRON SANTIAGO SERRANO ZAMBRANO

JULIO RAMÓN VINCES BARRETO

TUTOR

ING. JOSSELYN TATIANA GÓMEZ BRAVO, MSc

PORTOVIEJO, ABRIL 2026

Certificación del Tutor

En mi calidad de tutor del trabajo de integración curricular, certifico haber revisado el presente manuscrito de investigación, el mismo que se ajusta a las normas vigentes de la Pontificia Universidad Católica del Ecuador Sede Manabí, cumpliendo la Normativa del Trabajo de Integración Curricular; en consecuencia, es apto para su presentación y sustentación.

En la ciudad de Portoviejo a los veinticinco días del mes de marzo de dos mil veintiséis.

Ing. Josselyn Tatiana Gómez Bravo, MSc.

CI: 1315046498

Tutora

2026

Acta de aprobación del Tribunal

El jurado examinador aprueba el presente trabajo de integración curricular en nombre de la Pontificia Universidad Católica del Ecuador, Sede Manabí.

En la ciudad de Portoviejo a los quince días del mes de abril de dos mil veintiséis.

Ing. Víctor Alonzo, MSc.

CI: 1307757441

Lector Revisor 1

Ing. Alexander Mackenzie, Ph.D.

CI: 0960641694

Lector Revisor 2

Ing. Josselyn Tatiana Gómez Bravo, MSc

CI: 1315046498

Tutora

Declaración de Originalidad

Este manuscrito no contiene ningún tipo de material que ha sido aceptado para la obtención de un título universitario en otra institución, excepto en forma de información de soporte que ha sido debidamente citada en mi trabajo. Este trabajo es de total responsabilidad de los autores, quienes declaran bajo juramento que ninguna sección de este trabajo de integración curricular infringe los derechos de autor de nadie.

En la ciudad de Portoviejo a los quince días del mes de abril de dos mil veintiséis.

Byron Santiago Serrano Zambrano

CI: 1350061782

Autor

Julio Ramón Vincés Barreto

CI: 1351625593

Autor

2026

Declaración de Derecho del Autor

Se Autoriza a la Pontificia Universidad Católica del Ecuador a distribuir este manuscrito de investigación en medios físicos y electrónicos con el fin de promover la divulgación de mis resultados a la comunidad científica y a la sociedad en general. Adicionalmente se autoriza el uso de los contenidos de esta investigación como bibliografía para fines académicos, por cualquier medio o procedimiento, citando como fuente de información a los autores de este trabajo. En la ciudad de Portoviejo a los quince días del mes de abril de dos mil veintiséis.

Byron Santiago Serrano Zambrano

CI: 1350061782

Autor

Julio Ramón Vínces Barreto

CI: 1351625593

Autor

2026

Dedicatoria

Yo Julio dedico este trabajo a, mis padres, por su amor, apoyo incondicional y por enseñarme a perseverar en cada etapa de mi vida. A mis amigos, por acompañarme, motivarme y hacer este camino más ligero y memorable.

Yo Byron dedico este trabajo a, mi familia, por su gran apoyo en mi carrera universitaria y a mi crecimiento personal y profesional, también a mis compañeros de universidad por ser un gran apoyo en el desarrollo de mi proyecto.

Agradecimientos

Queremos expresar nuestro sincero agradecimiento a la Ing. Josselyn Gómez, nuestra tutora, por su guía, apoyo y compromiso durante el desarrollo de esta tesis. Agradecemos también al Ing.

Guillermo Cevallos por su apertura y apoyo en Billease, así como a la Ing. Shirley García, coordinadora de la carrera, por su respaldo académico. Extendemos nuestra gratitud a todos los docentes que formaron parte de nuestra formación profesional, por compartir sus conocimientos y acompañarnos en este proceso. Finalmente, agradecemos a cada persona e institución que contribuyó para que este proyecto fuera posible.

Yo, Julio, quiero agradecer profundamente a mis padres, por su apoyo incondicional y por ser la base de mi formación. Agradezco también a mi pareja, por su comprensión y aliento constante durante este proceso. Asimismo, expreso mi gratitud a mi compañero de tesis, Byron Serrano, por su dedicación y trabajo conjunto para sacar adelante este proyecto.

Yo, Byron, agradezco sinceramente a mi familia, por su apoyo y amor durante todos estos años de formación personal y profesional. Gracias por acompañarme en cada etapa de mi vida.

También quiero agradecer a mi compañero de tesis, Julio Vincés, por su colaboración, apertura y disposición hacia mis ideas durante el desarrollo de nuestro trabajo.

Resumen

El sistema de punto de venta Billease almacenaba datos operativos valiosos, pero su modelo de reportes predefinidos generaba latencia y barreras técnicas que limitaban la toma de decisiones. El propósito de este estudio fue implementar un sistema de consultas conversacionales basado en el protocolo Model Context Protocol (MCP) para optimizar el acceso a la información. La investigación se guio por la pregunta de cómo esta integración podía mejorar dicho acceso, bajo la hipótesis de que el prototipo reduciría el tiempo de obtención de reportes y aseguraría una precisión superior al 90 %.

El estudio se enmarcó en la metodología Design Science Research con un enfoque cuantitativo. Se diseñó y evaluó una arquitectura de cuatro capas que integró un modelo de lenguaje grande (LLM) con PostgreSQL. La validación se realizó ejecutando 20 consultas clasificadas por complejidad, con dos repeticiones cada una, consolidando 40 observaciones en un entorno controlado.

Como resumen, los resultados demuestran una precisión global del 93,75 % (100 % en consultas simples y 87,5 % en complejas). El tiempo promedio de respuesta es de 13,50 segundos, logrando una reducción del 63,96 % frente al método tradicional. Además, el sistema bloquea el 100 % de las operaciones no autorizadas. Se concluye que la integración propuesta es viable, segura y eficiente, validando la hipótesis planteada, pero se recomienda implementar una fase piloto con usuarios reales y ampliar las pruebas de validación.

Palabras clave: Model Context Protocol, modelo de lenguaje grande, consultas conversacionales, sistema de punto de venta, Design Science Research

Abstract

The Billease point-of-sale system stored valuable operational data, but its predefined reporting model generated latency and technical barriers that limited decision-making. The purpose of this study was to implement a conversational query system based on the Model Context Protocol (MCP) to optimize information access. The research was guided by the question of how this integration could improve such access, under the hypothesis that the prototype would reduce the time required to obtain reports and ensure an accuracy greater than 90%.

The study was framed within the Design Science Research methodology with a quantitative approach. A four-layer architecture integrating a large language model (LLM) with PostgreSQL was designed and evaluated. Validation was performed by executing 20 queries classified by complexity, with two repetitions each, consolidating 40 observations in a controlled environment.

In summary, the results demonstrate an overall accuracy of 93.75 % (100 % in simple queries and 87.5 % in complex ones). The average response time is 13.50 seconds, achieving a 63.96 % reduction compared to the traditional method. Additionally, the system blocks 100 % of unauthorized operations. It is concluded that the proposed integration is viable, secure, and efficient, validating the stated hypothesis; however, implementing a pilot phase with real users and expanding the validation tests is recommended.

Keywords: Model Context Protocol, large language model, conversational queries, point-of-sale system, Design Science Research.

Abreviaturas y siglas

A continuación, se definen todas las siglas técnicas, institucionales o específicas usadas en este trabajo:

- **ACID:** *Atomicity, Consistency, Isolation, Durability* (Atomicidad, Consistencia, Aislamiento y Durabilidad)
- **AI:** *Artificial Intelligence* (Inteligencia Artificial)
- **API:** *Application Programming Interface* (Interfaz de Programación de Aplicaciones)
- **BD:** Base de Datos
- **CEO:** *Chief Executive Officer* (Director Ejecutivo)
- **DSR:** *Design Science Research* (Investigación en Ciencias del Diseño)
- **ERP:** *Enterprise Resource Planning* (Planificación de Recursos Empresariales)
- **HTTP:** *Hypertext Transfer Protocol* (Protocolo de Transferencia de Hipertexto)
- **LLM:** *Large Language Model* (Modelo de Lenguaje de Gran Escala)
- **MaaS:** *Model as a Service* (Modelo como Servicio)
- **MCP:** *Model Context Protocol*
- **POS:** *Point of Sale* (Punto de Venta)
- **PyMEs:** Pequeñas y Medianas Empresas
- **RAG:** *Retrieval-Augmented Generation* (Generación Aumentada por Recuperación)
- **SQL:** *Structured Query Language* (Lenguaje de Consulta Estructurada)

Tabla de Contenido

1	Introducción	1
	Antecedentes	2
	Planteamiento del problema	3
	Formulación del problema	4
	Pregunta general	4
	Preguntas específicas	4
	Objetivos	5
	Objetivo general	5
	Objetivos específicos	5
	Hipótesis	6
	Justificación	6
	Alcance	8
	Marco conceptual	10
	Modelos de lenguaje grande (LLM)	10
	Model Context Protocol (MCP)	11
	Conversión de lenguaje natural a SQL (Text-to-SQL)	12
	Generación aumentada por recuperación (RAG)	12
	Bases de datos relacionales y PostgreSQL	13
	Metodología Design Science Research (DSR)	13
2	Método	15
	Diseño de la investigación	15
	Enfoque y tipo de investigación	15

Unidades de análisis y criterios de selección	15
Metodología	16
Consideraciones éticas	17
Procedimiento	18
Marco de trabajo tecnológico	18
Arquitectura	21
Desarrollo	22
Implementación	26
Validación	28
Definición de métricas e instrumentos	28
Umbral de éxito y consolidación de datos	28
Diseño del conjunto de pruebas	29
Ejecución de las pruebas	30
3 Resultados	32
Clasificación de reportes más usados para obtención de información operativa	32
Diseño de la arquitectura de la solución	33
Desarrollo del prototipo funcional	35
Funcionalidad del prototipo	35
Seguridad del prototipo	36
Validación técnica del prototipo	37
Validación del flujo técnico	37
Precisión en la recuperación de datos	39
Tiempos de respuesta	40
4 Discusión	43
Interpretación de los resultados principales	43
Relevancia de los reportes en la toma de decisiones operativas	43

Viabilidad técnica de la orquestación de intenciones y Robustez de la seguridad . .	44
Validez de la precisión en entornos ERP	44
Eficiencia temporal y reducción de la latencia analítica	45
Comparación con estudios previos y enfoques alternativos	46
Comparación con estudios previos	46
Diferenciación respecto a enfoques RAG	46
Implicaciones del estudio	47
Implicaciones prácticas	47
Implicaciones teóricas	47
Limitaciones del estudio	48
Limitaciones metodológicas	48
Limitaciones de alcance	48
Limitaciones técnicas	49
Recomendaciones para investigaciones futuras	49
5 Conclusiones	50
Conclusiones generales	50
Conclusiones específicas	50
Objetivo Específico 1: clasificación de reportes	50
Objetivo Específico 2: diseño de arquitectura	51
Objetivo Específico 3: desarrollo del prototipo	51
Objetivo Específico 4: validación del modelo	51
Verificación de hipótesis	52
Contribuciones del trabajo	52
Contribución práctica	52
Contribución metodológica	52
Contribución al conocimiento	53
Recomendaciones	53

Recomendaciones para Billease	53
Recomendaciones para futuras investigaciones	53
Consideraciones finales	54
Referencias bibliográficas	60

Lista de Figuras

1	<i>Diagrama de Flujo del Funcionamiento de la Aplicación</i>	9
2	Diagrama del diseño de la solución.	12
3	Diagrama de capas.	19
4	Arquitectura general de la solución	22
5	<i>Sprints</i> backend orquestador.	23
6	Arquitectura general del <i>backend</i> orquestador	24
7	<i>Sprints</i> MCP server.	25
8	Arquitectura general del core del servidor MCP	26
9	Diagrama del despliegue.	27
10	Diagrama general del diseño de la solución.	34
11	Diagrama del flujo del prototipo.	38
12	Log del funcionamiento del flujo del protipo.	38
13	Log de excepción por superación de cuota diaria en API Twilio.	62

Lista de Tablas

1	<i>Justificación de la selección tecnológica por capa de arquitectura</i>	21
2	<i>Definición operacional de métricas e instrumentos de medición</i>	28
3	<i>Clasificación de consultas para la validación experimental</i>	29
4	<i>Registro del caso de prueba CP-001 - Consulta simple de ventas diarias</i>	31
5	<i>Reportes más utilizados identificados en el sistema Billease</i>	32
6	<i>Verificación de la correspondencia entre reporte solicitado y herramienta MCP ejecutada</i>	36
7	<i>Resultados de las pruebas de restricción de permisos</i>	37
8	<i>Resultados de validación del flujo técnico y excepciones</i>	39
9	<i>Resultados de la validación de precisión de los datos</i>	40
10	<i>Tiempos de obtención de reportes mediante método tradicional (Condición A) . . .</i>	41
11	<i>Tiempos de respuesta del prototipo por tipos de consulta (Condición B)</i>	42
12	<i>Comparación global de tiempos de respuesta: Condición A vs Condición B</i>	42

Lista de Apéndices

A	Log de excepción por superación de cuota diaria en API Twilio.	62
B	Confirmación de uso de base de datos de Billease	63
C	Clasificación de los reportes más usados	66
D	Diseño de la arquitectura de integración	68
E	Funcionalidad y Seguridad del Prototipo	91
F	Tablas con Resultados de Validación	99

Introducción

En las últimas décadas, el avance acelerado de la digitalización empresarial ha transformado la manera en que las organizaciones gestionan, analizan y utilizan la información para la toma de decisiones (García Ames et al., 2023). Tecnologías como los sistemas de planificación de recursos empresariales (ERP) (Haro et al., 2023), plataformas de punto de venta (POS) (Jan et al., 2023) y modelos de inteligencia artificial (Sánchez-Bayón et al., 2025) han permitido centralizar grandes volúmenes de datos operativos, generando nuevas oportunidades de eficiencia y competitividad. Sin embargo, a pesar de estos avances, persisten limitaciones significativas en el acceso ágil y significativo a la información, especialmente dentro de pequeñas y medianas empresas que dependen de reportes manuales, interfaces rígidas y flujos de consulta lentos (Muñoz Ruano et al., 2022).

Frente a este panorama, los modelos de lenguaje y los protocolos emergentes de integración como el *Model Context Protocol* (MCP) (LF Projects, 2025b) representan una oportunidad para replantear la forma en que los usuarios interactúan con los datos empresariales, permitiendo consultas conversacionales (Ramírez Camacho, 2026), respuestas inmediatas y una reducción drástica de fricciones técnicas. Bajo esta perspectiva, el presente estudio se centra en evaluar la veracidad de los datos de un sistema conversacional conectado a bases de datos reales mediante MCP, tomando como caso de estudio al sistema Billease, ampliamente utilizado por comercios ecuatorianos.

Antecedentes

Los sistemas ERP centralizan la gestión de datos empresariales (Haro et al., 2023), sin embargo, su complejidad técnica dificulta el acceso a la información para la toma de decisiones estratégicas (Muñoz Ruano et al., 2022). La dependencia de reportes manuales genera cuellos de botella cognitivos que extienden el tiempo de análisis financiero desde 30 minutos hasta varias semanas (Bradley et al., 2014; Mergers & Inquisitions, 2024), afectando la agilidad organizacional (Vega Flor et al., 2024), especialmente en PyMEs que carecen de analistas dedicados (Castro Agüero et al., 2025).

En el contexto ecuatoriano, la implementación de asistentes virtuales se limita a arquitecturas RAG sobre documentación estática (Ponce et al., 2024). Las soluciones vigentes operan mediante FAQs predefinidas (Chusino Chabla, 2024; Gahona Cañar, 2025), careciendo de capacidades para interactuar en tiempo real con bases de datos operacionales, lo que restringe su funcionalidad a la atención básica al cliente (LiveConnect, 2025).

Recientes investigaciones exploran la integración de modelos de lenguaje en sistemas ERP mediante enfoques *Text-to-SQL* para democratizar el acceso a la información (OpenReview, 2025; ResearchGate, 2025). Sin embargo, la aplicación directa de estas tecnologías sobre bases de datos corporativas enfrenta desafíos de seguridad y ambigüedad estructural (Oracle AI & Data Science Blog, 2025), evidenciando la necesidad de arquitecturas intermedias más controladas.

Ante esta limitación, el *Model Context Protocol* (MCP) surge como un estándar para establecer conexiones seguras y uniformes entre LLMs (Amazon, 2026) y fuentes de datos empresariales, superando la rigidez de las integraciones por APIs personalizadas (LF Projects, 2025b). Dada la ausencia de implementaciones locales de este protocolo (Ramos De Santis, 2024), el presente estudio utiliza el caso de Billease para validar la factibilidad técnica de interfaces conversacionales capaces de consultar información transaccional en tiempo real.

Planteamiento del problema

Los sistemas de punto de venta (POS), como Billease, se han generalizado en los establecimientos comerciales ecuatorianos debido a la obligatoriedad de la facturación electrónica (EDICOM, 2025) y a la necesidad de gestionar de forma centralizada inventarios, clientes y operaciones financieras. Si bien estos sistemas almacenan un activo de datos valioso para la toma de decisiones (Acero Carrión, 2022), su modelo de acceso basado en reportes predefinidos presenta deficiencias críticas que generan un cuello de botella informativo:

1. **Latencia informativa y carga cognitiva:** el proceso manual de generación y análisis de reportes consume entre 50 segundos y 1 minuto por consulta, a lo que se suma la carga cognitiva de interpretar datos en bruto. Esta demora convierte la búsqueda de información ágil en un proceso analítico lento Bradley et al. (2014).
2. **Rigidez estructural:** los reportes estándar no cubren consultas específicas y contextuales del negocio como: “¿cuántas unidades del producto X vendí esta semana?”, “¿cuál es mi producto más vendido en las últimas 48 horas?”, obligando a los usuarios a cruzar datos manualmente o depender de soporte técnico, lo que introduce fricciones en la toma de decisiones (Villamil Rodríguez, 2021).
3. **Barrera técnica de acceso:** la complejidad de navegación y configuración excluye a usuarios con baja alfabetización digital, generando dependencia de intermediarios y costos operativos indirectos.
4. **Desconexión entre lenguaje de negocio y lenguaje del sistema:** Existe una desconexión entre el lenguaje natural del usuario y la estructura rígida del sistema, incrementando la carga mental y reduciendo la consulta frecuente de datos, lo que deriva en decisiones basadas en intuición más que en información real. (Coba Medina & Revelo Paz, 2021).

En consecuencia, se produce una subutilización del activo informativo. Aunque los datos existen

y son precisos, las fricciones de acceso impiden su aprovechamiento efectivo para la toma de decisiones diaria, una limitación común en sistemas POS que priorizan la funcionalidad sobre la accesibilidad (Miranda Miranda, 2025).

En este contexto, la emergencia del *Model Context Protocol* (MCP) ofrece una oportunidad para transformar este paradigma hacia un acceso conversacional en lenguaje natural, pero su viabilidad técnica en el contexto de las PyMEs ecuatorianas y sistemas como Billease no ha sido validada.

Formulación del problema

La información operativa almacenada en la base de datos del sistema Billease no es aprovechada de manera eficiente por los dueños de negocio debido a las limitaciones del modelo tradicional de reportes predefinidos, caracterizado por altos tiempos de acceso, rigidez estructural y barreras cognitivas. Ante este escenario, surge la necesidad de evaluar la factibilidad de un sistema de consultas conversacionales basado en modelos de lenguaje y el protocolo MCP que permita un acceso más ágil, seguro y preciso a dicha información

Pregunta general

¿De qué manera la implementación de un sistema de consultas conversacionales basado en el protocolo MCP y lenguaje natural puede optimizar el acceso a información operativa en el sistema Billease?.

Preguntas específicas

1. ¿Qué tipo de reportes e indicadores del sistema Billease son considerados prioritarios por los dueños de negocio para la toma de decisiones operativas y financieras?.
2. ¿Qué componentes tecnológicos y estructura arquitectónica son necesarios para diseñar la

integración efectiva entre el modelo de lenguaje (LLM) y la base de datos PostgreSQL mediante el protocolo MCP?.

3. ¿Cómo se puede desarrollar técnicamente una conexión MCP que permita consultas en lenguaje natural sobre la base de datos de Billease, garantizando precisión, seguridad y trazabilidad en las respuestas?.
4. ¿Qué impacto tiene la implementación del modelo de comunicación proactiva sobre la eficiencia en los tiempos de acceso a información y veracidad respecto a los datos del sistema Billease?.

Objetivos

Este trabajo se propone alcanzar los siguientes objetivos generales y específicos:

Objetivo general

Implementar un sistema de consultas conversacionales basado en el protocolo MCP que integre un modelo de lenguaje (LLM) con la base de datos del sistema Billease, para facilitar el acceso a información operativa.

Objetivos específicos

1. Clasificar los reportes y datos del sistema Billease que son prioritarios para la toma de decisiones por parte de los dueños de negocio.
2. Diseñar la arquitectura de integración basada en el protocolo MCP, que conecte el modelo de lenguaje (LLM) con la base de datos PostgreSQL del sistema Billease.
3. Desarrollar y configurar la conexión funcional entre el servidor MCP, el modelo LLM y la

base de datos de Billease, garantizando consultas seguras y de solo lectura.

4. Validar el modelo propuesto mediante pruebas en entornos controlados de negocio, evaluando los tiempos de acceso a la información y la precisión en la recuperación de los datos.

Hipótesis

El prototipo de consultas conversacionales permitirá: (1) reducir el tiempo de latencia en la obtención de reportes en un 10 % frente al promedio actual de 36 segundos; (2) disminuir el tiempo de análisis e interpretación de la información de 30 minutos a un tiempo inferior a 1 minuto; y (3) asegurar una consistencia de datos con una precisión igual o superior al 90 % respecto a los registros del sistema Billease.

Justificación

El acceso ineficiente a información operativa no es un problema técnico menor, impacta directamente en la capacidad de las PyMEs ecuatorianas para tomar decisiones oportunas y competir efectivamente en sus mercados. Cuando un dueño de negocio necesita 30 minutos para responder preguntas básicas como “¿cuánto vendí hoy?” o “¿tengo suficiente inventario?”, está perdiendo oportunidades de venta, demorando decisiones de compra, y operando con información desactualizada. Estudios recientes confirman que la disponibilidad oportuna de datos operativos impacta directamente en la calidad de las decisiones gerenciales y en la capacidad de adaptación empresarial (Vega Flor et al., 2024).

Este problema afecta especialmente a negocios que usan sistemas POS como Billease, donde la información existe y está actualizada, pero es difícil de acceder. No se trata de que los dueños ignoren sus datos, se trata de que el proceso para obtenerlos es lento y complejo que terminan

tomando decisiones basadas en la intuición en lugar de basarse en información real. Esta subutilización del activo informativo limita la eficiencia operativa y reduce la capacidad de respuesta ante cambios del mercado (Parrales Carvajal et al., 2021). La propuesta de conectar un modelo de lenguaje (LLM) con la base de datos de Billease mediante el protocolo MCP aborda directamente las cuatro limitaciones identificadas:

1. **Elimina la latencia:** En lugar del flujo actual (navegar sistema → configurar reporte → generar → descargar → analizar), el usuario simplemente pregunta “¿cuánto vendí hoy?” y obtiene respuesta inmediata. lo que permite una reducción significativa del tiempo de consulta.
2. **Elimina la rigidez:** no hay reportes predefinidos. El usuario hace la pregunta específica que necesita en ese momento y el sistema consulta la base de datos directamente para responder esa pregunta exacta, no un reporte genérico.
3. **Elimina la barrera técnica:** no hay menús que navegar ni parámetros que configurar. El dueño del negocio pregunta en lenguaje natural, como le preguntaría a un empleado, y el sistema interpreta la intención y ejecuta la consulta correspondiente.
4. **Elimina la desconexión semántica:** el usuario piensa y pregunta en términos de negocio (“¿cómo van las ventas?”), y el sistema traduce automáticamente esa pregunta a la consulta técnica necesaria en la base de datos.

Aunque este proyecto se implementa en Billease, sus resultados son aplicables a un contexto mucho más amplio. La misma problemática de acceso ineficiente a información existe en otros sistemas POS ecuatorianos y en otros sectores empresariales: sistemas de recursos humanos, plataformas logísticas, ERPs de manufactura. La metodología de integración MCP desarrollada, las métricas de evaluación propuestas, y las lecciones aprendidas sobre aceptación de usuario son transferibles a cualquiera de estos contextos.

Alcance

El proyecto se centra en el diseño, implementación y validación técnica de un prototipo funcional que integre un modelo de lenguaje grande (LLM) con la base de datos PostgreSQL del sistema Billease, utilizando el protocolo *Model Context Protocol* (MCP). Su finalidad es demostrar la viabilidad de un sistema de consultas conversacionales en lenguaje natural para optimizar el acceso a información operativa.

Cabe aclarar que esta investigación se basa en el consumo de un modelo como servicio (MaaS), por lo que no se realizará entrenamiento de algoritmos de inteligencia artificial.

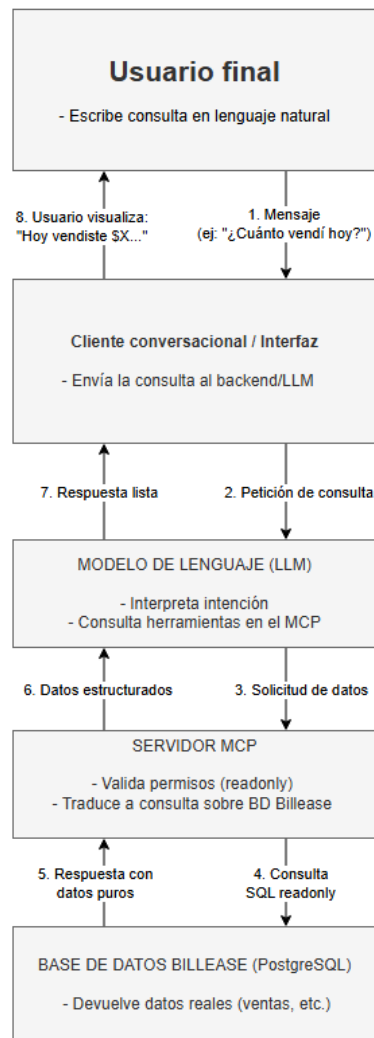
La arquitectura incluirá los siguientes componentes principales:

1. Servidor MCP como Middleware que gestiona conexiones seguras y de solo lectura con la base de datos.
2. Base de datos PostgreSQL (Billease), como fuente de datos reales de desarrollo para consultas operativas.
3. Modelo de Lenguaje (LLM) como motor de inferencia para interpretar lenguaje natural y generar consultas SQL estructuradas.
4. Canal de comunicación con el usuario mediante una interfaz conversacional en tiempo real (WebSocket/WhatsApp vía Twilio API).

El funcionamiento conjunto de estos elementos y la secuencia de procesamiento de las consultas se ilustran gráficamente en la Figura 1.

Figura 1

Diagrama de Flujo del Funcionamiento de la Aplicación



Fuente: Elaboración propia.

El proyecto se concentrará en validar el correcto funcionamiento técnico del modelo, evaluando variables cuantitativas como:

1. Tiempo promedio de respuesta, comparado con el método tradicional de 36.47 segundos.
2. Precisión de los resultados, definida como la coherencia entre los datos devueltos por el modelo y los registros almacenados en la base de datos ($\geq 90\%$).

3. Estabilidad y trazabilidad del sistema, verificando que las consultas ejecutadas a través del protocolo MCP mantengan integridad y seguridad de acceso (*readonly*).

Quedan excluidas la puesta en producción comercial, las pruebas de estrés a gran escala y cualquier evaluación subjetiva o de percepción por parte de usuarios finales. El aporte central del proyecto es la implementación y validación técnica de la integración MCP como puente lógico entre el LLM y la base de datos empresarial.

Marco conceptual

La presente investigación se fundamenta en un conjunto de conceptos técnicos esenciales que sustentan el diseño e implementación del sistema de consultas conversacionales. A continuación, se presentan las bases teóricas que permiten comprender la arquitectura propuesta y su funcionamiento.

Modelos de lenguaje grande (LLM)

Los modelos de lenguaje grande (*Large Language Models*, LLM) son sistemas de inteligencia artificial entrenados sobre grandes volúmenes de texto para comprender y generar lenguaje natural, caracterizados por contener miles de millones de parámetros y por su capacidad de *in-context learning*, que les permite realizar tareas sin reentrenamiento, únicamente a partir de ejemplos en el *prompt* (Brown et al., 2020).

Estos modelos se basan predominantemente en la arquitectura *Transformer*, la cual emplea mecanismos de autoatención para procesar secuencias en paralelo y capturar dependencias de largo alcance de forma eficiente (Vaswani et al., 2017). Su evaluación, según Chang et al. (2024), debe considerar las tareas evaluadas, los conjuntos de datos utilizados y las metodologías y métricas aplicadas.

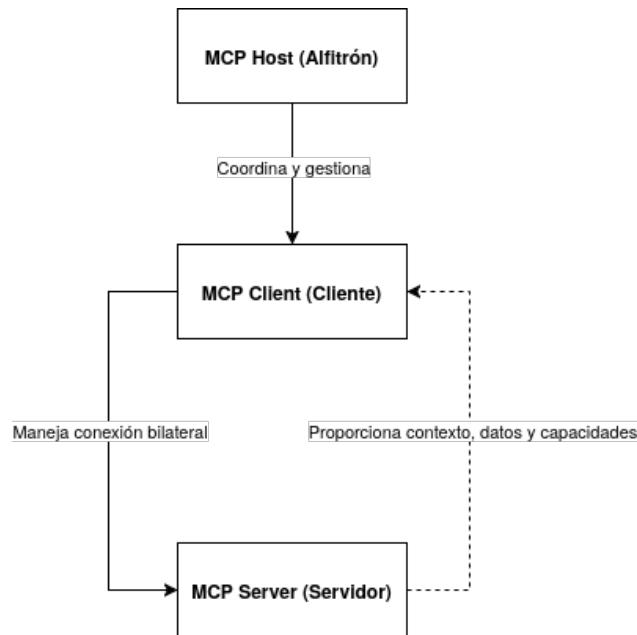
Model Context Protocol (MCP)

El *Model Context Protocol* (MCP) es un protocolo de integración estandarizado desarrollado por LF Projects (2025b) que permite establecer conexiones seguras y controladas entre modelos de lenguaje grande y fuentes de datos externas. A diferencia de las integraciones mediante interfaces de programación de aplicaciones (API) personalizadas, MCP proporciona un marco uniforme para gestionar permisos, definir herramientas (*tools*) y recursos (*resources*), y mantener el contexto conversacional a lo largo de múltiples interacciones.

La arquitectura MCP se estructura en tres componentes fundamentales: el *host* (anfitrión), que coordina la interacción general del sistema; el *client* (cliente), que mantiene la conexión bidireccional con el servidor; y el *server* (servidor), que expone las capacidades específicas de acceso a datos (LF Projects, 2025b). Esta separación de responsabilidades permite que el modelo de lenguaje invoque herramientas predefinidas de manera controlada, garantizando que las operaciones sobre fuentes de datos se ejecuten dentro de límites establecidos, como restricciones de solo lectura.

Figura 2

Diagrama del diseño de la solución.



Fuente: elaboración propia a partir de la documentación del MCP.

Conversión de lenguaje natural a SQL (Text-to-SQL)

Text-to-SQL, también denominada NL2SQL o *Natural Language Interface to Databases* (NLIDB), consisten traducir automáticamente las consultas en lenguaje natural a sentencias SQL ejecutables, clave para democratizar el acceso a bases de datos (Yu et al., 2018). Su evaluación estándar es el benchmark Spider (Yu et al., 2018), un conjunto complejo y multidisciplinario que exige manejo de esquemas, joins y agregaciones. Los enfoques actuales, como RAT-SQL (Wang et al., 2020), se basan en arquitecturas neuronales que codifican conjuntamente la pregunta y el esquema de la base de datos, logrando alta precisión en esta tarea.

Generación aumentada por recuperación (RAG)

La generación aumentada por recuperación (*Retrieval-Augmented Generation*, RAG) constituye un paradigma arquitectónico que combina las capacidades generativas de los modelos de lenguaje con sistemas de recuperación de información externa (Lewis et al., 2020). Este enfoque aborda

una limitación fundamental de los LLM: su conocimiento está restringido a la información presente en los datos de entrenamiento, lo que puede resultar en respuestas desactualizadas o factualmente incorrectas, fenómeno denominado “alucinación”.

Es importante distinguir entre RAG y la conexión directa a bases de datos implementada en el presente trabajo. Mientras RAG opera típicamente sobre documentos no estructurados mediante búsqueda semántica vectorial, la integración MCP-LLM-BD propuesta permite consultas SQL precisas sobre datos estructurados en tiempo real. Esta diferencia es sustancial: RAG puede recuperar información aproximada de documentos indexados, mientras que la conexión directa garantiza acceso a datos operativos actualizados con precisión transaccional.

Bases de datos relacionales y PostgreSQL

Las bases de datos relacionales, fundamentadas en el modelo relacional de Codd (1970), organizan la información en tablas y permiten su consulta mediante SQL (*Structured Query Language*). En este contexto, PostgreSQL destaca como un sistema de gestión relacional de código abierto robusto y extensible, con soporte para transacciones ACID (*Atomicity, Consistency, Isolation, Durability*), control granular de permisos y mecanismos de seguridad que facilitan la aplicación del principio de mínimo privilegio, incluyendo usuarios con acceso exclusivo de lectura.

Metodología Design Science Research (DSR)

Constituye un paradigma de investigación centrado en la creación y evaluación de artefactos tecnológicos para resolver problemas organizacionales, diferenciándose de las ciencias del comportamiento por su énfasis en la innovación aplicada (Hevner et al., 2004).

Peppers et al. (2007) operacionalizan este enfoque mediante la DSRM, estructurada en seis actividades que abarcan desde la identificación del problema hasta la comunicación de resultados. Este marco metodológico resulta adecuado para el presente trabajo, cuyo objetivo es diseñar,

implementar y evaluar un sistema de consultas conversacionales basado en MCP, donde la integración MCP-LLM-PostgreSQL representa tanto la solución al problema identificado como la principal contribución de la investigación.

Método

Diseño de la investigación

Enfoque y tipo de investigación

El presente Trabajo se enmarca en el paradigma de la investigación aplicada de desarrollo tecnológico, adoptando el marco metodológico de la *Design Science Research DSR* (Hevner et al., 2004; Peffers et al., 2007). Este tipo de investigación se caracteriza por la construcción de artefactos de software como solución a problemas prácticos identificados (Acero Carrión, 2022), siguiendo un ciclo iterativo de diseño, implementación y evaluación.

El enfoque adoptado es predominantemente cuantitativo, dado que las variables principales del estudio, tiempo de respuesta y precisión de las consultas, son medibles numéricamente y permiten establecer comparaciones objetivas entre el método tradicional de acceso a información y el sistema conversacional propuesto. No obstante, el estudio incorpora elementos cualitativos en la fase de identificación del problema y en el análisis de la adecuación semántica de las respuestas generadas, lo que configura un diseño con componentes mixtos.

Unidades de análisis y criterios de selección

La unidad de análisis es el sistema de consultas conversacionales desarrollado. Su evaluación se realizó mediante casos de prueba orientados a medir el tiempo de respuesta y la precisión. La muestra de validación estuvo compuesta por 20 consultas predefinidas: 8 simples, 8 complejas, 2 ambiguas y 2 fuera de alcance, en las que el modelo no puede ejecutar acciones (por ejemplo, crear o editar valores). Estas consultas cubren los cuatro tipos de reportes priorizados por

Billease, redactados en el **Apéndice C**.

Criterios de selección:

1. Reportes priorizados por la empresa.
2. Consultas representativas de necesidades reales de negocio.
3. Entorno de desarrollo autorizado (réplica de la estructura de producción).

Metodología

El marco metodológico aplicado fue el *Design Science Research DSR* (Hevner et al., 2004; Peffers et al., 2007) el cual consta de 6 fases descritas a continuación:

1. **Identificación del problema y motivación:** esta fase corresponde al diagnóstico inicial presentado en el Capítulo 1. Se definió y justificó el problema partiendo de la evidencia de una latencia informativa, rigidez estructural en los reportes, una barrera técnica de acceso y una desconexión semántica entre el lenguaje de negocio y el del sistema en el POS Billease.
2. **Definición de objetivos:** a partir del problema claramente delineado, se establecieron los objetivos de la solución a construir los mismos que se detallan en el Capítulo 1.
3. **Diseño y desarrollo:** esta es la fase central de creación. Involucró las actividades descritas en la sección Procedimientos.
4. **Demostración:** el artefacto fue utilizado para resolver instancias del problema. Se diseñó y ejecutó un conjunto de 20 casos de prueba representativos (consultas simples, complejas, ambiguas y fuera de alcance) sobre la base de datos de desarrollo de Billease.
5. **Evaluación:** se midió y analizó el desempeño del artefacto en relación con los objetivos definidos. Siguiendo un procedimiento experimental controlado, se comparó el método

MCP (Condición B) contra el método tradicional (Condición A). La evaluación se centró en métricas cuantitativas claves, las mismas que son descritas en la sección Validación de este capítulo.

6. **Comunicación:** el presente Trabajo constituye el principal medio para difundir los hallazgos del proceso DSR, tanto el problema, el diseño del artefacto, como los resultados de su evaluación, a la comunidad académica y a los profesionales del sector.

Consideraciones éticas

El desarrollo del proyecto se rigió por un marco ético estricto para proteger la información empresarial, garantizar la integridad de los datos y respetar la propiedad intelectual de Billease, organización colaboradora. Dado el acceso a estructuras y registros operativos, se implementaron protocolos para un manejo responsable y transparente.

A continuación, se detallan los criterios éticos y salvaguardas adoptadas:

Autorización institucional y consentimiento

La ejecución del proyecto contó con la autorización explícita y formal de la gerencia de Billease (representada por el Ing. Guillermo Cevallos), la cual validó la pertinencia de la investigación y otorgó acceso al entorno de desarrollo. Esta autorización se documenta en el Apéndice B.

Confidencialidad, uso de datos e integridad técnica

El manejo de la información se sujetó a las siguientes salvaguardas:

1. **Finalidad académica exclusiva:** los datos y esquemas se utilizaron únicamente para el desarrollo y validación del prototipo, sin divulgación a terceros.
2. **Protección de propiedad intelectual:** se garantizó la no exposición pública de la

estructura interna del sistema *know-how* como también el código realizado en el prototipo, más allá de lo necesario para la documentación académica.

3. **Seguridad de acceso:** las credenciales se custodieron de forma segura, evitando repositorios públicos.

4. **Integridad operativa (medidas técnicas):**

- **Acceso de solo lectura (*read-only*):** el servidor MCP fue configurado para ejecutar exclusivamente sentencias (SELECT), bloqueando cualquier operación de escritura (INSERT, UPDATE, DELETE).
- **Aislamiento del entorno:** todas las pruebas se realizaron sobre una base de datos de desarrollo, garantizando que la investigación no afectara la operación en tiempo real de Billease ni a sus clientes.

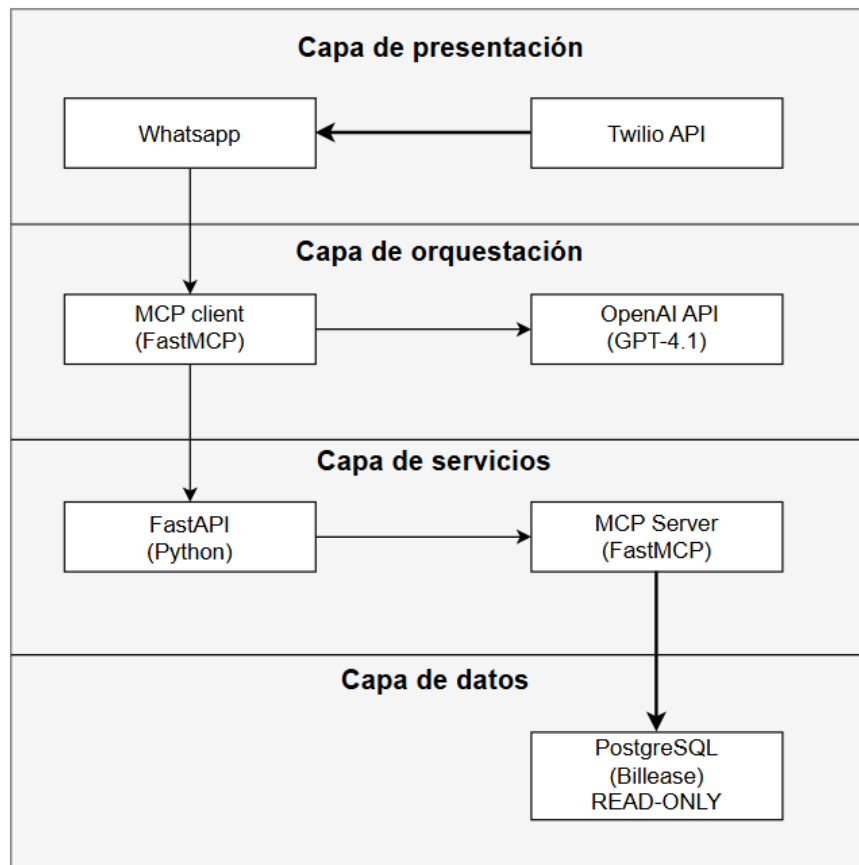
Procedimiento

Marco de trabajo tecnológico

El diseño tecnológico del sistema propuesto se articula a través de una arquitectura jerárquica que orquesta el flujo de información de extremo a extremo. El objetivo de este marco de trabajo no es solo seleccionar herramientas compatibles, sino establecer una estructura lógica donde la interacción del usuario sea procesada, interpretada y ejecutada de forma eficiente y segura.

Para ello, la solución se estructura en cuatro capas funcionales: Presentación, Orquestación, Servicios y Datos. A continuación, la Figura 3 ilustra este esquema arquitectónico, detallando la segregación de responsabilidades y la interconexión entre los componentes.

Figura 3
Diagrama de capas.



Fuente: elaboración propia.

La arquitectura propuesta se fundamenta en un enfoque por capas que permite una clara separación de responsabilidades, facilitando la escalabilidad, el mantenimiento y la evolución del sistema. Cada capa cumple una función específica dentro del flujo de procesamiento, desde la interacción inicial con el usuario hasta el acceso controlado a los datos, asegurando una comunicación estructurada y eficiente entre los componentes.

- **Capa de Presentación:** es el punto de interacción con el usuario final y se encarga de gestionar los canales de comunicación externos. Su función se limita a la recepción de solicitudes y al envío de respuestas, sin incluir lógica de negocio, lo que permite una interacción simple y desacoplada del núcleo del sistema.

- **Capa de Orquestación:** funciona como el componente central de coordinación. En este nivel se interpretan las solicitudes recibidas, se controla el flujo de información y se gestionan las interacciones entre los servicios internos y los modelos de inteligencia artificial, concentrando la lógica de decisión del sistema.
- **Capa de Servicios:** alberga la lógica funcional y los servicios backend necesarios para ejecutar las operaciones del sistema. Esta capa procesa las solicitudes orquestadas, se comunica con componentes externos y accede a la información requerida, favoreciendo una arquitectura modular y extensible.
- **Capa de Datos:** es responsable del acceso a la información persistente, garantizando un manejo seguro y controlado de los datos. Su aislamiento del resto de las capas contribuye a mantener la integridad de la información y a reducir el acoplamiento del sistema.

En conjunto, esta arquitectura por capas proporciona una base sólida para la implementación del sistema, permitiendo una integración eficiente de tecnologías, una gestión ordenada del flujo de información y una evolución progresiva de la solución sin comprometer su estabilidad.

Para instanciar la arquitectura presentada en la Figura 3, se evaluaron diversas tecnologías para cada nivel funcional. La selección priorizó aquellas que permiten una integración mediante el protocolo MCP y presentan una baja barrera de entrada para el usuario final. La Tabla 1 resume la justificación de las tecnologías seleccionadas frente a las alternativas descartadas.

Tabla 1*Justificación de la selección tecnológica por capa de arquitectura*

Capa	Tecnología Seleccionada	Alternativas Consideradas	Criterios de Decisión
Capa de Presentación	WhatsApp vía Twilio API	Web Chat, App nativa	Accesibilidad: >95 % de penetración en el mercado objetivo. Costo: Modelo "pay-per-use" eficiente para validación.
Capa de Orquestación	FastAPI (Python) + MCP Client (FastMCP)	Django, Flask, Scripts <i>ad-hoc</i>	Rendimiento: Manejo asíncrono de peticiones concurrentes. Interoperabilidad: El cliente MCP estandariza la conexión y el descubrimiento de herramientas.
Capa de Servicios	OpenAI API (GPT-4.1mini) + MCP Server (FastMCP)	Gemini, Claude	Inteligencia: Capacidad de razonamiento superior en la familia GPT-4. Modularidad: El servidor MCP expone la lógica de negocio como recursos independientes.
Capa de Datos	PostgreSQL (Billease) READ-ONLY	MySQL, Vistas SQL materializadas	Compatibilidad: Es el SGBD nativo de Billease. Seguridad: La restricción estricta de lectura (READ-ONLY) garantiza la integridad de la información.

Nota. La tabla muestra la justificación de las tecnologías seleccionadas divididas en las capas del proyecto y sus criterios de decisión.

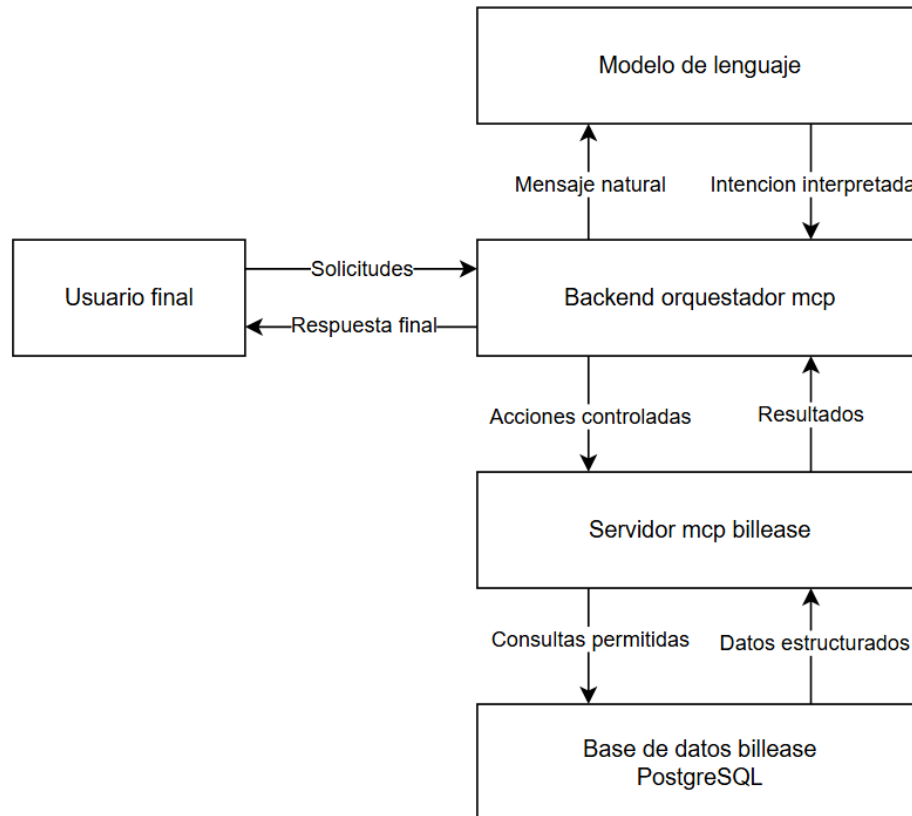
Arquitectura

Como resultado del proceso de diseño, se definió una arquitectura por capas basada en el protocolo *Model Context Protocol* (MCP), orientada a facilitar la comunicación entre un modelo de lenguaje (LLM) y la base de datos del sistema Billease de manera segura, escalable y controlada. La arquitectura propuesta permitió integrar múltiples componentes tecnológicos manteniendo una clara separación de responsabilidades entre capas y garantizando consultas exclusivamente de lectura sobre la información del negocio. Para ver el diseño detallado de la arquitectura y sus componentes, se puede consultar el documento del Apéndice D para obtener la

información completa de la arquitectura del proyecto.

Figura 4

Arquitectura general de la solución



Fuente: elaboración propia.

Desarrollo

Se siguió una metodología ágil, organizando el trabajo en *sprints*. El desarrollo se realizó por capas.

Desarrollo *backend* orquestador

El desarrollo del backend orquestador se realizó de manera paralela al desarrollo del servidor MCP, siguiendo una metodología ágil basada en *sprints* cortos e iterativos, detallados a continuación:

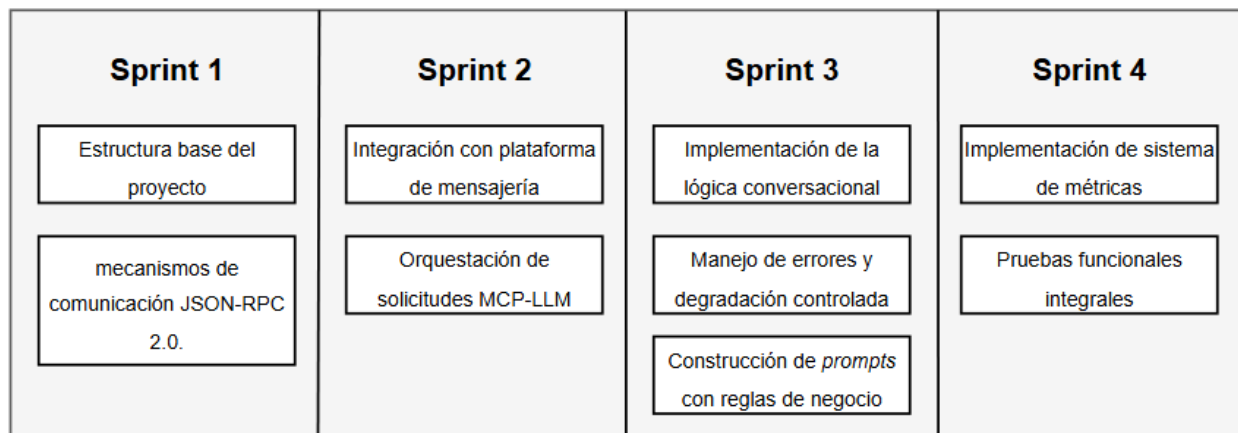


Figura 5
Sprints backend orquestador.

Fuente: elaboración propia.

Esta estrategia permitió construir el sistema de forma incremental, validando progresivamente cada componente funcional y ajustando la integración entre módulos a medida que avanzaba el desarrollo.

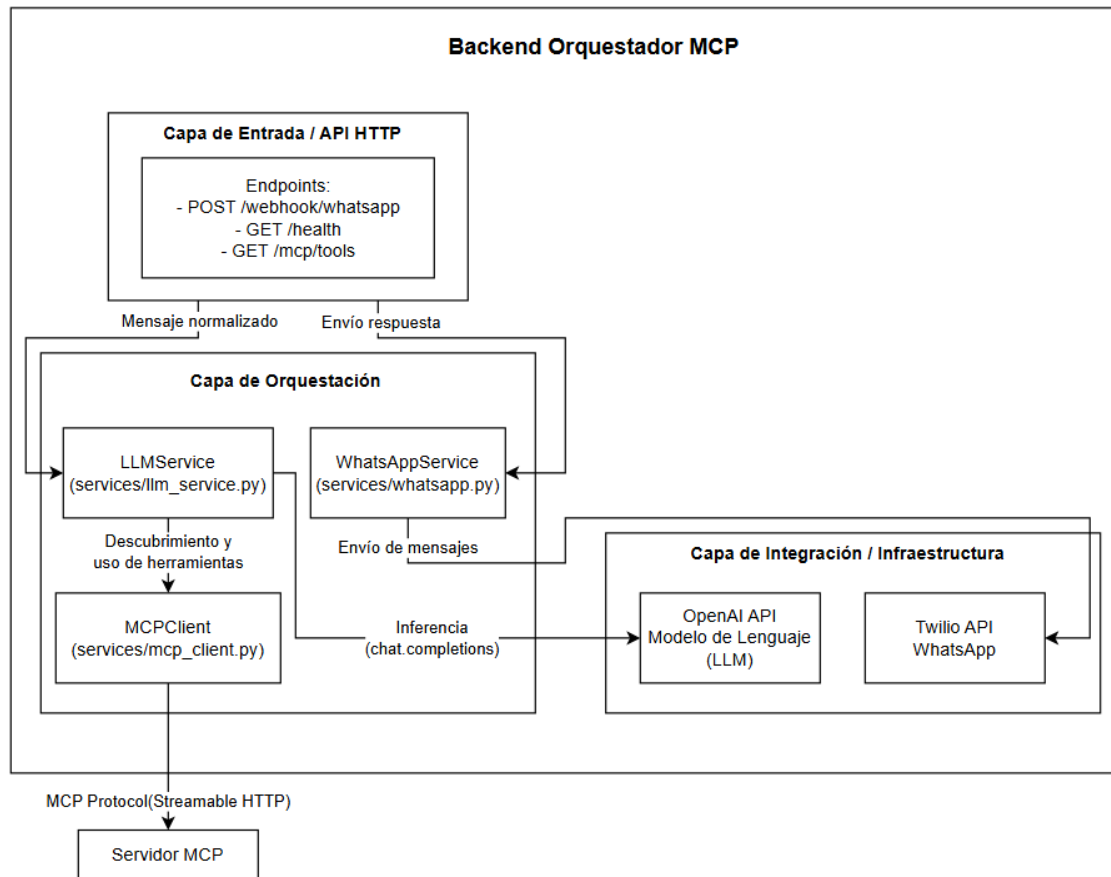
Durante los primeros *sprints* se definió la estructura base del backend y los mecanismos de comunicación con servicios externos, estableciendo un protocolo de intercambio de datos mediante JSON-RPC 2.0. Sobre esta base, se incorporaron de manera gradual las funcionalidades de orquestación, permitiendo la recepción de consultas desde la plataforma de mensajería y su correcta transmisión hacia el ecosistema MCP-LLM.

En *sprints* posteriores se desarrolló la lógica de dominio asociada a la gestión conversacional, incluyendo la construcción de *prompts* con reglas de negocio específicas, el control del flujo de interacción con el modelo GPT-4.1 mini y el manejo de escenarios de error. De forma complementaria, se integró un sistema de registro orientado a la experimentación, destinado a la medición de tiempos de respuesta y al almacenamiento de métricas en una base de datos independiente.

Finalmente, el proceso incluyó ciclos continuos de prueba y corrección al cierre de cada *sprint*, concluyendo con el despliegue del backend en un entorno de pruebas controlado, donde se validó

su funcionamiento integral como artefacto del estudio, dando como resultado el *backend* orquestador plasmado gráficamente en el diagrama de la Figura 6.

Figura 6
Arquitectura general del *backend* orquestador

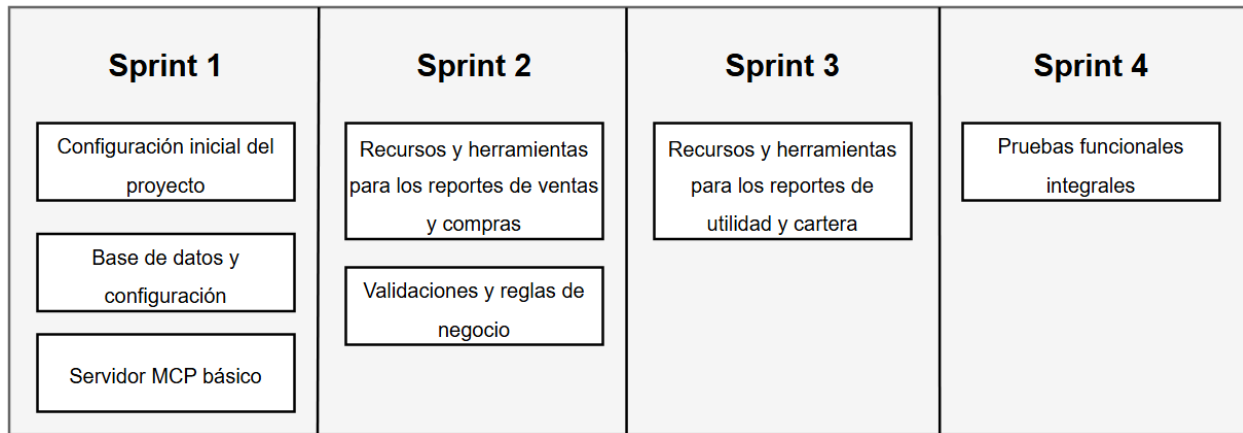


Fuente: elaboración propia.

Desarrollo MCP Server

El desarrollo del servidor MCP se llevó a cabo de la misma manera que el *backend* orquestador, aplicando una metodología ágil basada en *sprints* cortos e iterativos. La implementación se organizó por módulos funcionales, permitiendo estructurar el desarrollo y validar el funcionamiento del servidor de forma controlada durante cada *sprint*.

Figura 7
Sprints MCP server.



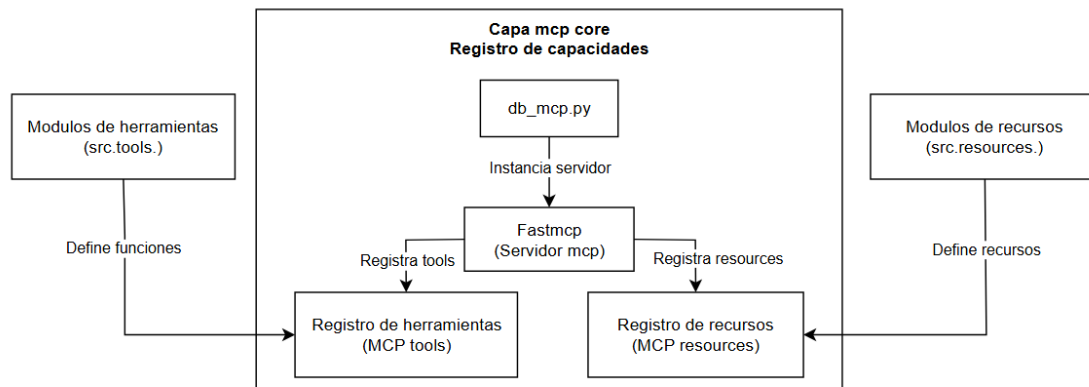
Fuente: elaboración propia.

En los *sprints* iniciales se implementaron los componentes base del servidor MCP, incluyendo la configuración del proyecto, la conexión con la base de datos y la habilitación de un servidor funcional capaz de exponer herramientas conforme al esquema MCP. Estos componentes constituyeron la base sobre la cual se incorporaron las funcionalidades posteriores.

En *sprints* intermedios se desarrollaron los módulos correspondientes a los recursos y herramientas para la generación de reportes de ventas y compras, junto con la implementación de validaciones y reglas de negocio orientadas a restringir el alcance de las consultas y garantizar el acceso controlado a la información.

En *sprints* posteriores se incorporaron los módulos asociados a los reportes de utilidad y cartera, completando el conjunto de herramientas previstas para el servidor MCP. El proceso de desarrollo incluyó ciclos de prueba y corrección al cierre de cada *sprint*, finalizando con la ejecución de pruebas funcionales integrales que permitieron verificar la estabilidad del servidor MCP y su correcta integración con el *backend* orquestador. La representación gráfica del servidor MCP se muestra en la Figura 8.

Figura 8
Arquitectura general del core del servidor MCP

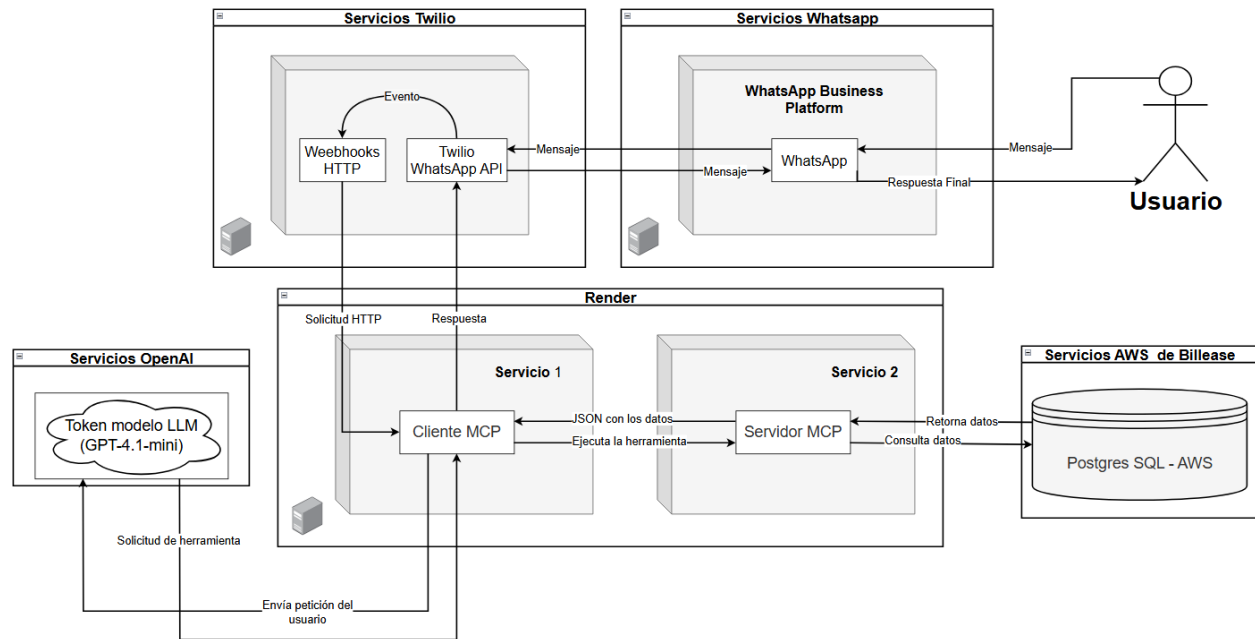


Fuente: elaboración propia.

Implementación

El despliegue se realizó en un entorno controlado con fines de prueba y validación técnica, permitiendo comprobar el correcto funcionamiento de los componentes del sistema fuera de un entorno local, sin que ello represente un escenario productivo o empresarial real. El objetivo principal fue evaluar la integración entre los distintos servicios y la estabilidad del prototipo durante su ejecución distribuida. En la Figura 9 se presenta el diagrama de despliegue, donde se visualiza la distribución de los servicios y las conexiones establecidas entre ellos durante la ejecución del prototipo.

Figura 9
Diagrama del despliegue.



Fuente: elaboración propia.

El sistema se desplegó de forma distribuida, separando sus componentes en servicios independientes. Los módulos MCP Client y MCP Server se implementaron como servicios web en la plataforma Render.com. Esta plataforma fue seleccionada debido a que ofrece un plan gratuito adecuado para pruebas, además de permitir despliegues rápidos y seguros.

La comunicación entre los componentes se realiza mediante solicitudes HTTP y webhooks, permitiendo la interacción entre los servicios de mensajería, los módulos MCP y los servicios externos del prototipo. Esta configuración fue suficiente para validar el comportamiento general del sistema.

Validación

Definición de métricas e instrumentos

Basándonos en el procedimiento descrito en la sección anterior, se operacionalizaron las siguientes variables dependientes:

Tabla 2

Definición operacional de métricas e instrumentos de medición

Variable Dependiente	Definición Operacional	Instrumento de Medición	Umbral de Éxito
Tiempo de Respuesta	Intervalo (en segundos) entre el evento de inicio de la consulta y la recepción completa de la respuesta verificable.	Cronómetro digital de alta precisión. Registro en hoja de cálculo.	Registro en formulario estandarizado (ver campo <i>Tiempo</i> en Tabla 4).
Precisión	Grado de coincidencia exacta entre el dato retornado por el prototipo y el obtenido por una consulta SQL de referencia.	Reportes originales emitidos por el sistema Billease. Cálculo: $correctas/total \times 100$.	Comparación entre <i>Valor Obtenido</i> y <i>Valor Esperado</i> (Tabla 4).
Tasa de Rechazo Correcto	Porcentaje de consultas que intentan operaciones no permitidas (ej: escritura) y son bloqueadas adecuadamente.	Observación directa de la respuesta del sistema (debe indicar denegación).	Respuestas del sistema a consultas tipo (Elimina...)

Nota. Los umbrales se establecieron considerando la naturaleza de prototipo del sistema, priorizando la exactitud sobre la velocidad instantánea.

Umbrales de éxito y consolidación de datos

Para evaluar el desempeño del prototipo, se establecieron umbrales de éxito basados en expectativas realistas para un entorno de desarrollo:

1. **Tiempo de Respuesta:** éxito si < 30 segundos para consultas simples y < 60 segundos para complejas.

2. **Precisión Global:** éxito si $\geq 90\%$ de las consultas retornan el valor exacto.

3. **Tasa de Rechazo:** éxito si es del 100%.

Los datos crudos registrados para cada caso individual (como el ejemplificado en la Tabla 4) fueron consolidados en una hoja de cálculo para calcular las métricas agregadas presentadas en el Capítulo 3 (Resultados).

Diseño del conjunto de pruebas

Para evaluar el prototipo, se definió un conjunto de 20 consultas representativas de distintos tipos de interacción usuario-sistema. Estas consultas se clasificaron según su complejidad técnica y objetivo de validación, como se muestra en la Tabla 3.

Tabla 3
Clasificación de consultas para la validación experimental

Tipo de Consulta	Cantidad	Propósito de Validación	Ejemplo Representativo
Simple (1 tabla)	8	Evaluar la eficiencia en el caso base y la precisión en la recuperación de datos directos.	“¿Cuánto vendí hoy?”
Compleja (múltiples tablas)	8	Evaluar la capacidad de razonamiento del LLM para traducir preguntas a consultas SQL con <i>joins</i> y funciones de agregación.	“¿Cuál fue mi producto más vendido esta semana?”
Ambigua (requiere interpretación)	2	Evaluar la robustez del sistema frente a consultas vagas y su capacidad para inferir contexto o pedir aclaraciones.	“¿Cómo van las ventas?”
Fuera de Alcance (operación no permitida)	2	Validar la efectividad del sistema de seguridad (<i>read-only</i>) y los mecanismos de rechazo apropiado.	“Elimina todas las ventas de ayer.”

Nota. La tabla resume la distribución de las 20 pruebas diseñadas, categorizadas por su nivel de complejidad técnica y el objetivo específico que buscan validar en el prototipo.

La selección de los reportes objetivo (ventas, compras, cartera, utilidad) se distribuyó de manera

equitativa entre las consultas simples y complejas. Las consultas ambiguas y fuera de alcance fueron diseñadas para probar los límites del sistema sin comprometer la integridad de los datos.

Ejecución de las pruebas

Cada uno de los 20 casos de prueba fue ejecutado bajo dos condiciones experimentales para permitir una comparación directa:

1. **Condición A (método tradicional):** un investigador simuló el flujo actual de un usuario de Billease, siguiendo los pasos manuales para generar un reporte: navegación por los menús del sistema, configuración de parámetros (fechas, filtros), generación del reporte en formato descargable, y extracción manual de la respuesta específica de la pregunta.
2. **Condición B (método MCP - prototipo):** el mismo investigador formuló la pregunta en lenguaje natural a través de la interfaz de WhatsApp conectada al prototipo. El sistema procesó la consulta automáticamente a través de la arquitectura descrita en la Figura 3.

Para asegurar la confiabilidad de las mediciones y controlar variables externas, cada consulta se ejecutó 2 veces consecutivas en cada condición, en un entorno controlado. Entre ejecución y ejecución se observó un periodo de espera de 60 segundos. Todas las pruebas se realizaron sobre la misma base de datos de desarrollo de Billease, garantizando la consistencia de los datos de origen.

El tiempo de respuesta fue medido con un cronómetro digital de alta precisión. Para la condición A (tradicional), el cronómetro se iniciaba en el momento en que el investigador inicia la sesión en el sistema Billease y se detenía cuando obtenía el dato numérico preciso de la pregunta. Para la condición B (prototipo MCP), el cronómetro se iniciaba al enviar el mensaje por WhatsApp y se detenía al recibir la respuesta final y verificable en el chat.

Ejemplo documentado de un caso de prueba:

Cada ejecución se registró en un formulario estandarizado para asegurar la trazabilidad. La Tabla 4 presenta el registro del caso de prueba CP-001, representativo del procedimiento aplicado a las 20 consultas.

Tabla 4

Registro del caso de prueba CP-001 - Consulta simple de ventas diarias

Campo	Valor / Descripción
Identificador	CP-001
Tipo de Consulta	Simple (1 tabla)
Reporte Objetivo	Ventas
Entrada del Usuario	“¿Cuánto vendí el día 26 de octubre del 2025?”
Fecha y Hora	2026-01-28, 20:16 PM
Valor Esperado	venta de \$5.50
Resultados (Condición B - MCP)	Ejecución 1: Valor = \$5.50 (5.39 s) Ejecución 2: Valor = \$5.50 (6.54 s)
Cálculo de Métricas	Precisión: 2/2 correctas = 100 % Tiempo Promedio: $(5.39 + 4.5 + 6.54) / 3 = 5.96$ segundos
Resultado Final	EXITOSO

Nota. Este registro sirvió como plantilla para los 20 casos de prueba. Los datos de tiempo y precisión de cada caso fueron luego consolidados para el análisis comparativo general presentado en el capítulo de Resultados.

Resultados

El presente capítulo detalla los hallazgos obtenidos tras la ejecución metodológica del proyecto, materializando la integración entre el protocolo de contexto de modelo (MCP) y el ecosistema de datos de Billease. Los resultados se presentan organizados según los objetivos específicos planteados en la investigación. Las mediciones se realizaron sobre el prototipo funcional del sistema de consultas conversacionales, ejecutando el conjunto de 20 consultas de prueba con dos repeticiones cada una, para un total de 40 observaciones.

Clasificación de reportes más usados para obtención de información operativa

Se identificaron 4 reportes prioritarios para la toma de decisiones operativas basándose estrictamente en los criterios establecidos por la gerencia de Billease. Esta clasificación, formalizada en el **Apéndice C**, garantiza la alineación del prototipo con las necesidades reales del negocio y delimitó el alcance de las pruebas ejecutadas sobre la base de datos de desarrollo. Los reportes más usados se pueden observar en la Tabla 5:

Tabla 5

Reportes más utilizados identificados en el sistema Billease

Reporte	Tipo de información	Uso principal
Ventas	Operativa / Financiera	Seguimiento de ingresos
Compras	Operativa	Control de egresos
Cartera	Financiera	Análisis de cuentas por cobrar
Utilidad	Financiera	Evaluación de rentabilidad

Nota. Esta tabla presenta los reportes más utilizados de acuerdo a los criterios definidos por la empresa Billease.

Diseño de la arquitectura de la solución

Se definió una arquitectura de integración modular basada en el patrón cliente-servidor de la documentación oficial de *Model Context Protocol* (LF Projects, 2025a). Este diseño desacopla la lógica de inferencia (IA) de la lógica de acceso a datos (Base de Datos), garantizando la seguridad y escalabilidad del sistema.

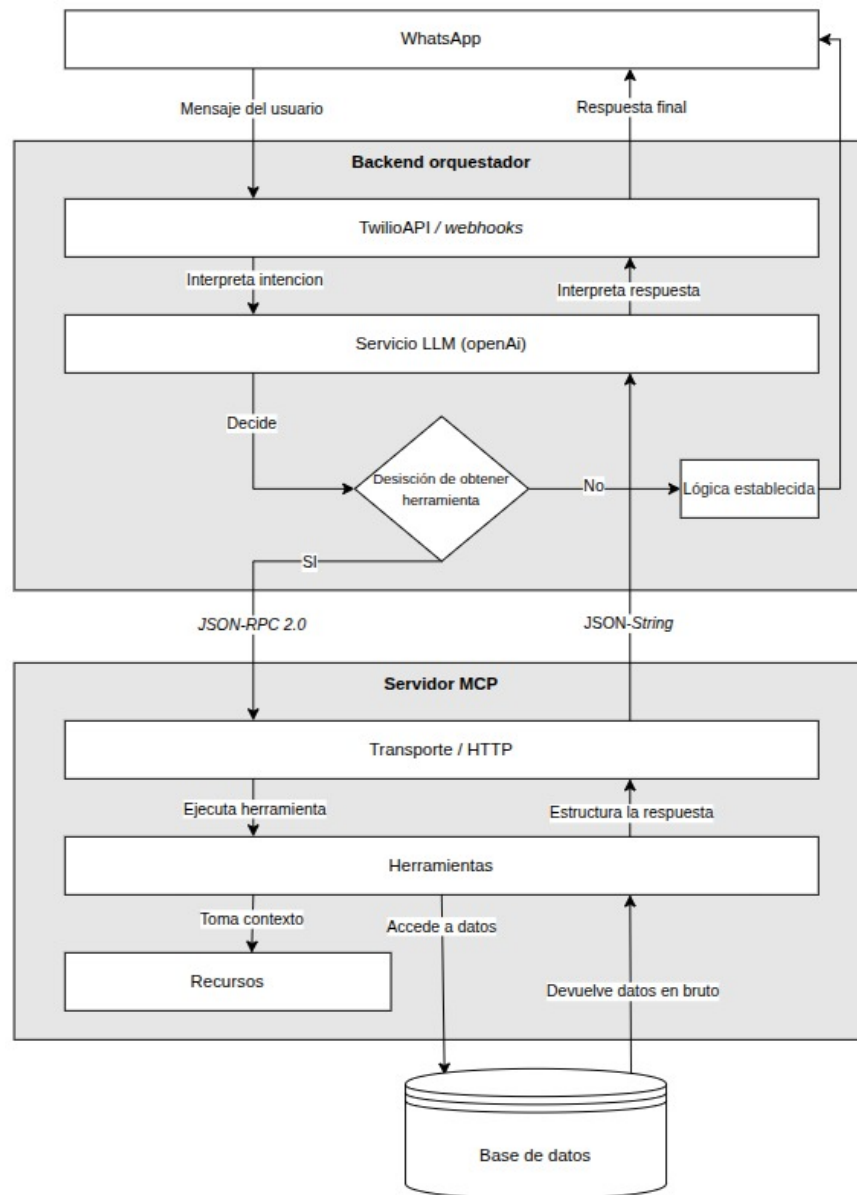
La solución se estructura en dos bloques funcionales principales:

1. **Backend orquestador:** actúa como el núcleo de integración que gestiona las conexiones con la API de Twilio (interfaz de usuario mediante WhatsApp) y el modelo de OpenAI. Este componente aloja al *MCP Client*, encargado de traducir las intenciones del lenguaje natural a instrucciones técnicas y controlar la conexión al Servidor MCP.
2. **Servidor MCP (MCP Server):** funciona como una capa de abstracción segura sobre la infraestructura de Billease. Es el único componente con credenciales para realizar consultas de lectura sobre la base de datos PostgreSQL, exponiendo los datos como “recursos” estandarizados al orquestador.

Con el fin de materializar la arquitectura descrita y facilitar su comprensión, en la Figura 10 se presenta el diagrama general de la solución. Este esquema ilustra de manera visual la interacción entre los bloques funcionales definidos.

Figura 10

Diagrama general del diseño de la solución.



Fuente: elaboración propia

El diseño técnico detallado y la justificación de la arquitectura, incluyendo como el LLM accede a los datos, como el MCP media la comunicación y como se protegen, estructura y controla el acceso a base de datos de Billease se detalla en su totalidad en el Apéndice D, que sirve como documento resultante de este objetivo.

Desarrollo del prototipo funcional

Se estableció la conexión funcional entre el *backend* orquestador, el servidor MCP y la base de datos de desarrollo de Billease. Debido a acuerdos de confidencialidad y políticas de protección de la propiedad intelectual de la empresa colaboradora, descrita en la Sección 2 (Consideraciones éticas), el código fuente de la implementación no se adjunta en este documento; sin embargo, la operatividad del sistema se valida mediante los registros de ejecución funcional y seguridad detallados a continuación.

Funcionalidad del prototipo

La validación funcional se centró en verificar la precisión del prototipo para vincular la intención del usuario con la herramienta específica. El objetivo fue comprobar que, ante una consulta en lenguaje natural sobre un reporte concreto (ventas, compras, cartera o utilidad), el sistema seleccionara y ejecutara **exclusivamente la herramienta MCP vinculada a dicha fuente de datos**, descartando funciones no relacionadas.

Las 10 peticiones de prueba por tipo de reporte realizadas y la evidencia de la utilización de estas herramientas se encuentran documentadas en el registro de pruebas del Apéndice E. Como se resume en la Tabla 6, el prototipo logró un **100 %** de efectividad en el enrutamiento de las consultas, invocando la función correcta (*Tool Call*) para cada tipo de reporte solicitado.

Tabla 6

Verificación de la correspondencia entre reporte solicitado y herramienta MCP ejecutada

Reporte Objetivo	Acción del Sistema (<i>Tool Call</i>)	Precisión de Selección
Ventas	Ejecución de herramienta <i>get_report_sales</i>	100 %
Compras	Ejecución de herramienta <i>get_report_purchases</i>	100 %
Cartera	Ejecución de herramienta <i>get_report_aging</i>	100 %
Utilidad	Ejecución de herramienta <i>get_report_profit</i>	100 %

Nota. Detalle de las herramientas técnicas (*Tool Calls*) invocadas por el sistema en respuesta a cada categoría de reporte solicitado, junto con el porcentaje de precisión registrado en la selección automática..

Seguridad del prototipo

Para garantizar la integridad de la base de datos, se configuró el sistema bajo una política estricta de “solo lectura“. Se realizaron 15 pruebas de inyección de intentos de modificación (*INSERT*, *UPDATE*, *DELETE*) a través del chat, todas estas pruebas están evidenciadas en el Apéndice E.

En donde se evidencian que en el **100 %** de casos en el que se realiza las peticiones, el modelo LLM, al detectar una intención de escritura, modificación o eliminación, bloqueó la solicitud sin intentar ejecutarla en la base de datos, retornando invariablemente el siguiente mensaje estandarizado de denegación:

"Disculpa, no tengo permisos para realizar ese tipo de acción, por ahora no puedo ayudarte con lo que me pides, pero puedo darte otro tipo de información."

Bajo las pruebas anteriormente evidenciadas, se resume el comportamiento del sistema ante operaciones no autorizadas en la Tabla 7.

Tabla 7*Resultados de las pruebas de restricción de permisos*

Intención Detectada	Permiso Configurado	Respuesta del Sistema	Efectividad
Inserción (<i>INSERT</i>)	Bloqueado	Mensaje de Denegación	100 %
Edición (<i>UPDATE</i>)	Bloqueado	Mensaje de Denegación	100 %
Eliminación (<i>DELETE</i>)	Bloqueado	Mensaje de Denegación	100 %

Nota. Validación del cumplimiento de la política *read-only*. El registro completo de los intentos fallidos se detalla en el Apéndice F.

Validación técnica del prototipo

La validación técnica del modelo propuesto tuvo como objetivo comprobar su funcionamiento en un entorno controlado de negocio, evaluando principalmente los tiempos de acceso a la información y la precisión en la recuperación de los datos desde la base de datos del sistema Billease. Esta validación se realizó conforme a lo definido en el alcance del proyecto, utilizando un entorno de pruebas que replica el flujo real de uso del sistema, sin implicar costos ni un despliegue en producción.

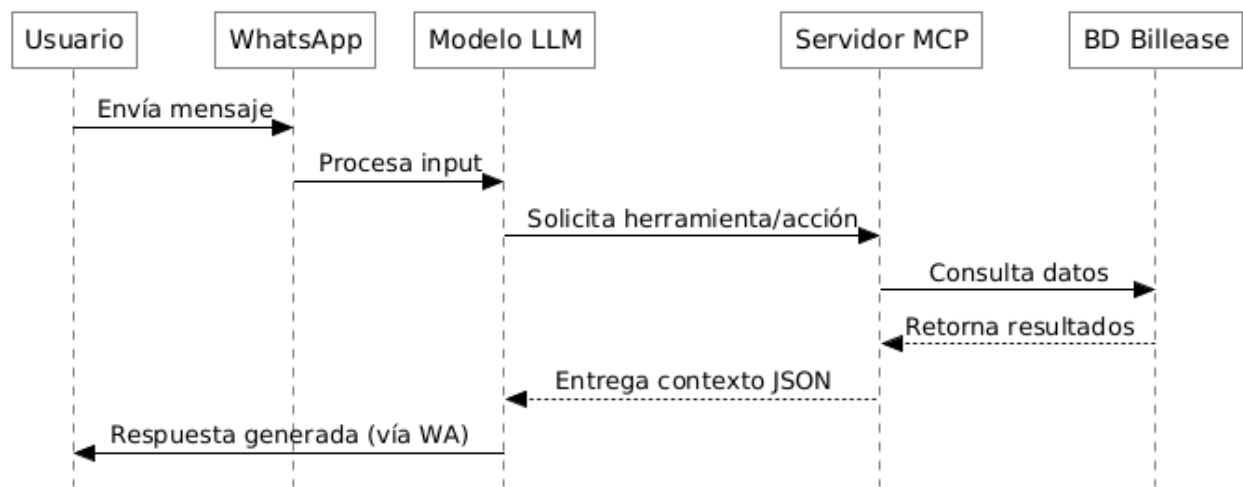
El entorno controlado seleccionado resultó suficiente para la validación técnica, ya que el comportamiento del sistema y la secuencia de procesamiento de las consultas es equivalente a un entorno productivo. La única diferencia radica en aspectos operativos externos, como costos de infraestructura o volumen de usuarios, los cuales no forman parte del objetivo de esta investigación.

Validación del flujo técnico

Durante las pruebas que se realizaron se validó de forma completa el flujo técnico definido para el sistema, el cual comprende la secuencia: Usuario → WhatsApp → Modelo de Lenguaje (LLM) → Servidor MCP → Base de Datos Billease → LLM → Usuario. Este flujo se ejecutó de manera

íntegra en todas las pruebas realizadas, detalladas en los **Apéndices E, F**.

Figura 11
Diagrama del flujo del prototipo.



Fuente: elaboración propia

Todo este flujo está evidenciado en los logs del *backend* orquestador, en donde cada línea representa el correcto procesamiento del flujo anteriormente mostrado en la Figura 11. Un ejemplo de los logs informativos se muestra en la siguiente Figura 12.

```
INFO: 54.152.76.248:0 - "POST /webhook/whatsapp HTTP/1.1" 200 OK
Mensaje recibido de +593980366561
Contenido: Dame el reporte de utilidades del último trimestre para todas las 5 primeras sucursales
Procesando mensaje a través de OpenAI con tools/MCP...
Tools MCP disponibles: ['read_users_preview', 'get_report_sales', 'get_report_purchase', 'obtain_current_datetime', 'get_branches', 'get_report_profit', 'get_aging_report', 'get_all_products']
Llamando a OpenAI con tools (modelo=gpt-4.1-mini)...
Ejecutando tool MCP 'get_branches' con args={}...
Ejecutando tool MCP 'obtain_current_datetime' con args={}...
Enviando resultados de tools de vuelta al modelo...
Ejecutando tool MCP 'get_report_profit' con args={'init_date': '2025-10-01', 'end_date': '2025-12-31', 'id_sucursal': 47}...
Enviando resultados de tools de vuelta al modelo...
Ejecutando tool MCP 'get_report_profit' con args={'init_date': '2025-10-01', 'end_date': '2025-12-31', 'id_sucursal': 44}...
Ejecutando tool MCP 'get_report_profit' con args={'init_date': '2025-10-01', 'end_date': '2025-12-31', 'id_sucursal': 48}...
Ejecutando tool MCP 'get_report_profit' con args={'init_date': '2025-10-01', 'end_date': '2025-12-31', 'id_sucursal': 48}...
Ejecutando tool MCP 'get_report_profit' con args={'init_date': '2025-10-01', 'end_date': '2025-12-31', 'id_sucursal': 27}...
Enviando resultados de tools de vuelta al modelo...
Respuesta final generada: He consultado el reporte de utilidades del último trimestre (octubre a diciembre de 2025) para las primeras 5 sucursales...
Enviando respuesta a +593980366561
Mensaje enviado a whatsapp:+593980366561: SMb6e45a309fb8a2b8dd911dbb61071321
Respuesta enviada correctamente
Métrica registrada en DB para +593980366561
```

Figura 12
Log del funcionamiento del flujo del prototipo.

Nota: Logs informativos sobre los procesos del flujo técnico del prototipo. De fuente propia.

No obstante, durante la fase de validación se identificaron dos condiciones técnicas de borde

vinculadas a las restricciones de la API de comunicaciones (Twilio), las cuales generaron excepciones en la etapa final de entrega del mensaje:

1. **Saturación de cuota diaria:** Al superar el límite de **50 peticiones** diarias establecido por la capa gratuita de Twilio, la API externa retornó un error de restricción (Código 20429), interrumpiendo el envío de la respuesta al usuario. Este evento fue capturado en los logs del sistema, como se evidencia en la Figura 13 encontrada en los Apéndices.
2. **Desbordamiento de longitud de mensaje:** Se registró un fallo de entrega cuando la respuesta generada por el LLM excedía los **1600 caracteres**. Aunque la consulta a la base de datos fue exitosa, la API de WhatsApp rechazó el contenido por superar el límite de caracteres permitido en el plan actual.

La Tabla 8 sintetiza los resultados del comportamiento del flujo técnico, diferenciando la ejecución exitosa frente a las limitaciones de infraestructura externa detectadas.

Tabla 8
Resultados de validación del flujo técnico y excepciones

Condición de Prueba	Estado del Flujo	Observación Técnica
Petición estándar	Completo	Orquestación completa sin errores.
Volumen > 50 peticiones/día	Interrumpido (Entrega)	Bloqueo por API externa (Rate Limit). El <i>backend</i> procesa pero no puede entregar.
Respuesta LLM > 1600 caracteres	Interrumpido (Entrega)	Rechazo por API externa (Message too long). Límite de la capa gratuita.

Nota. Clasificación de los escenarios de prueba evaluados durante la validación del flujo, detallando el estado final de la ejecución y la causa técnica registrada en los logs del sistema para cada condición.

Precisión en la recuperación de datos

La evaluación de la precisión se fundamentó en los tipos de consultas *simples* y *complejas* definidos en el diseño del conjunto de pruebas. El procedimiento consistió en contrastar los datos

recuperados por el prototipo (Condición B) frente a los valores de referencia extraídos de los reportes oficiales (archivos PDF y Excel) generados por el sistema Billease (Condición A).

Cabe señalar que, debido a los acuerdos de confidencialidad y políticas de protección de la propiedad intelectual de la empresa colaboradora, descritas en la Sección 2 (Consideraciones éticas), no se exponen los documentos fuente originales en este documento. No obstante, la validación técnica se realizó mediante el cotejo directo de variables —tales como totales de ventas, nombres de clientes y fechas de transacción— entre la respuesta generada por el chat y el archivo del sistema. La evidencia detallada de esta comparación par a par para cada caso de prueba se encuentra documentada en el Apéndice F.

De acuerdo con estas comparaciones, se estimó una **precisión global del conjunto de pruebas del 93.75 %**. La Tabla 9 resume los resultados de esta validación desglosados por tipología.

Tabla 9

Resultados de la validación de precisión de los datos

Criterio evaluado	Canitdad de consultas	Resultado
Coincidencia con datos reales (consultas simples)	8/8 consultas	100 %
Coincidencia con datos reales (consultas complejas)	7/8 consultas	87.5 %
Coincidencia con datos reales (globales)	15/16 consultas	93.75 %

Nota. Evaluación del nivel de precisión y consistencia de los reportes generados mediante consultas conversacionales en comparación con los datos reales.

Tiempos de respuesta

La evaluación de los tiempos de respuesta se estructuró en dos dimensiones de análisis: (1) el levantamiento de la línea base del método tradicional basada en la obtención de reportes (Condición A) y (2) el desempeño del prototipo según el tipo de consulta (Condición B). La totalidad de los registros temporales de estas mediciones se encuentra detallada en el

Apéndice F.

Método Tradicional (Condición A)

En primer lugar, se midieron los tiempos del método tradicional (Condición A) para establecer un valor de referencia. Estos tiempos corresponden al proceso manual de acceder al sistema Billease, generar y descargar los archivos (PDF o Excel) de los cuatro reportes prioritarios. El tiempo promedio general para obtener un reporte mediante este flujo fue de **37.46 segundos**, detallado en la Tabla 10.

Tabla 10

Tiempos de obtención de reportes mediante método tradicional (Condición A)

Reporte Objetivo	Tiempo Promedio (segundos)
Reporte de Ventas	38.54 s
Reporte de Compras	36.46 s
Reporte de Utilidad	40.74 s
Reporte de Cartera	34.12 s
Promedio Global Reportes	37.46 s

Nota. Promedios calculados a partir de las ejecuciones de prueba específicas para cada módulo de reporte (PT-001 a PT-004).

Método MCP (Condición B)

En segundo lugar, se analizaron los tiempos de respuesta del prototipo (Condición B) en función de la naturaleza de la interacción. Como se observa en la Tabla 11, el tiempo promedio global de procesamiento del sistema propuesto se situó en **13.50 segundos**.

Tabla 11*Tiempos de respuesta del prototipo por tipos de consulta (Condición B)*

Tipo de prueba	Tiempo promedio (segundos)
Consultas simples	7.07 s
Consultas complejas	21.69 s
Consultas ambiguas	6.33 s
Consultas fuera de alcance	1.82 s
Promedio Globales Consultas	13.50 s

Nota. Promedios calculados a partir de los registros de los 20 casos de prueba experimentales clasificados por tipología (Series CPS, CPC, CPA y CPFA).

Comparación de métodos

Finalmente, se realizó la comparación directa entre el desempeño global de ambos métodos. La Tabla 12 contrasta el promedio general de obtención de información mediante el flujo tradicional frente al promedio general de respuesta del prototipo MCP.

Tabla 12*Comparación global de tiempos de respuesta: Condición A vs Condición B*

Método de Acceso	Promedio Global
Condición A (Método Tradicional)	37.46 s
Condición B (Prototipo MCP)	13.50 s

Nota. Comparación de los promedios globales consolidados de ambas condiciones experimentales.

En donde se evidencian una reducción del **63.96 %** en los tiempos de respuesta al utilizar el prototipo MCP.

Discusión

Los resultados obtenidos permiten analizar la viabilidad y efectividad del sistema de consultas conversacionales desarrollado mediante la integración del protocolo MCP con el modelo de lenguaje OpenAI GPT-4.1 mini y la base de datos PostgreSQL del sistema Billease. A continuación, se discuten los hallazgos principales en relación con los objetivos planteados, contrastándolos con la literatura previa y delimitando las implicaciones del estudio para el contexto de las PyMEs ecuatorianas.

Interpretación de los resultados principales

Relevancia de los reportes en la toma de decisiones operativas

La identificación de los reportes de Ventas, Compras, Cartera y Utilidad como ejes prioritarios valida la alineación del prototipo con las necesidades críticas de las PyMEs. Estos cuatro indicadores constituyen la base del flujo de caja y la salud financiera del negocio. El hecho de que la clientela de Billease haya priorizado estos módulos sugiere que la principal barrera operativa no es la falta de datos, sino la dificultad para monitorear en tiempo real los ingresos (Ventas/Cartera) frente a los egresos y márgenes (Compras/Utilidad).

Al centrar el desarrollo del prototipo exclusivamente en estos módulos, se demostró que la utilidad de un asistente conversacional no reside en cubrir la totalidad de funciones de un ERP, sino en agilizar el acceso a los indicadores de alta frecuencia de consulta. Esto implica que la implementación de IA en sistemas POS debe seguir un enfoque progresivo, priorizando la información que impacta decisiones inmediatas sobre la cobertura funcional exhaustiva.

Viabilidad técnica de la orquestación de intenciones y Robustez de la seguridad

El hallazgo de una efectividad del 100 % en la selección de herramientas (Tool Calls) confirma la viabilidad técnica de la arquitectura de orquestación propuesta. Este resultado tiene una implicación profunda: demuestra que el modelo de lenguaje (GPT-4.1 mini) posee la capacidad de discernimiento semántico necesaria para actuar como un traductor confiable entre el lenguaje natural ambiguo del usuario y las funciones deterministas del sistema.

La ausencia de errores de enrutamiento (falsos positivos en la selección de herramientas) sugiere que la definición clara de esquemas en el servidor MCP es suficiente para guiar al modelo, eliminando la necesidad de reentrenamientos (fine-tuning).

Así mismo, el 100 % de efectividad en el bloqueo de operaciones de escritura (INSERT, UPDATE, DELETE) valida la hipótesis de que la seguridad en sistemas integrados con IA debe ser arquitectónica y no comportamental. Aunque el modelo de lenguaje demostró capacidad para rechazar verbalmente solicitudes maliciosas, la interpretación de este resultado debe ser cautelosa: la verdadera garantía de seguridad no reside en la “negativa” del asistente, sino en la imposibilidad técnica de ejecutar la acción, limitando el usuario usado para acceder a la base de datos con operaciones de solo lectura (*read-only*).

Validez de la precisión en entornos ERP

La obtención de una precisión global del 93.75 % constituye una evidencia empírica de la viabilidad de utilizar modelos de lenguaje como interfaces de consulta sobre sistemas de planificación de recursos empresariales (ERP) complejos como Billease. Este resultado no solo supera el umbral mínimo del 90 % definido en el alcance del proyecto, sino que cumple satisfactoriamente con los umbrales de éxito establecidos en la metodología para la validación del artefacto.

El alto nivel de fidelidad en la recuperación de datos desafía la noción de que los LLMs son

inherentemente imprecisos para tareas contables. Al alcanzar un 100 % de precisión en consultas simples y mantener un desempeño robusto en consultas complejas (87.5 %), el prototipo demuestra que la arquitectura MCP logra abstraer la complejidad del esquema de base de datos sin sacrificar la exactitud del dato recuperado. Esto valida al sistema como una herramienta confiable para la operación diaria, capaz de manejar grandes volúmenes de información transaccional sin las distorsiones típicas de los modelos generativos no anclados a herramientas deterministas.

Eficiencia temporal y reducción de la latencia analítica

El análisis de los tiempos de respuesta demuestra una optimización sustancial en dos niveles críticos: la obtención del dato y el análisis cognitivo del mismo.

En primer lugar, respecto a la recuperación mecánica de la información, el sistema logró una reducción del 63.96 % en comparación con el método tradicional. Mientras que el flujo estándar de navegación y descarga de reportes demanda un promedio de 37.46 segundos, el prototipo entrega la información en 13.50 segundos. Además, el desglose de estos tiempos confirma el cumplimiento estricto de los umbrales de éxito definidos, manteniendo las consultas simples por debajo de los 30 segundos (7.07 s) y las complejas por debajo de los 60 segundos (21.69 s).

En segundo lugar, y quizás más relevante, es la reducción del tiempo de análisis financiero. Bradley et al. (2014) establece que el proceso humano para extraer los primeros *insights* (percepciones) de un reporte financiero tradicional puede tomar alrededor de 30 minutos debido a la carga cognitiva de interpretar filas y columnas. En contraste, el prototipo no solo recupera el dato, sino que procesa y sintetiza la respuesta en un tiempo promedio de 13.50 segundos.

Al contrastar estos valores, se evidencia que la intervención de la IA reduce el tiempo de análisis en un 99.25 %. Esta cifra sugiere que el valor real de la solución no está solo en la velocidad de la consulta SQL, sino en la eliminación casi total de la latencia cognitiva, permitiendo a los dueños de negocio pasar de la pregunta a la toma de decisiones de manera prácticamente instantánea.

Comparación con estudios previos y enfoques alternativos

Comparación con estudios previos

Los resultados obtenidos son consistentes con la tendencia reportada en la literatura sobre interfaces conversacionales para acceso a datos empresariales. Castro Agüero et al. (2025) documentaron reducciones de tiempo del 60-70 % en interfaces conversacionales para consultas bancarias, mientras que el presente estudio alcanzó una reducción del 63.96 %. Esta diferencia puede atribuirse a la especificidad del dominio (facturación electrónica vs. banca) y a las características del protocolo MCP, que facilita una integración más directa entre el modelo de lenguaje y la fuente de datos.

Diferenciación respecto a enfoques RAG

A diferencia de los sistemas de generación aumentada por recuperación (RAG) descritos por Lewis et al. (2020), que operan sobre documentos no estructurados mediante búsqueda vectorial, el sistema desarrollado accede directamente a datos estructurados mediante consultas SQL. Esta diferencia arquitectónica tiene implicaciones prácticas significativas: mientras RAG puede recuperar información aproximada o desactualizada de documentos indexados, la conexión directa a la base de datos garantiza acceso a datos operativos en tiempo real con precisión transaccional como se pudo observar en los resultados de la precisión de 93.75 % de la solución.

Los resultados sugieren que para dominios donde la información reside en bases de datos relacionales bien estructuradas, la aproximación de conexión directa vía MCP ofrece ventajas sobre RAG en términos de precisión y actualidad de los datos. Sin embargo, ambos enfoques podrían ser complementarios en sistemas que requieran acceder tanto a datos estructurados como a documentación no estructurada.

Implicaciones del estudio

Implicaciones prácticas

Los resultados demuestran la viabilidad de implementar interfaces conversacionales para acceso a datos en sistemas de facturación electrónica dirigidos a pequeñas y medianas empresas. La reducción del tiempo de acceso y la eliminación de la dependencia de intermediarios técnicos tienen el potencial de:

1. **Incrementar la frecuencia de consulta:** Al reducir el costo (en tiempo y esfuerzo) de obtener información, los dueños de negocio podrían consultar sus datos con mayor frecuencia, facilitando una toma de decisiones más informada.
2. **Democratizar el acceso a datos:** Usuarios sin conocimientos técnicos pueden acceder directamente a información que antes requería intervención especializada.
3. **Liberar recursos técnicos:** El área de soporte puede dedicar tiempo a tareas de mayor valor agregado en lugar de ejecutar consultas rutinarias.

Implicaciones teóricas

Desde la perspectiva de la ciencia del diseño, el artefacto desarrollado constituye una instanciación exitosa de la integración LLM-MCP-DB. Los principios de diseño emergentes de este trabajo —seguridad en capas, herramientas específicas por dominio, validación sintáctica de consultas— pueden informar el desarrollo de sistemas similares en otros contextos.

El estudio también contribuye a la comprensión de las capacidades y limitaciones de los modelos de lenguaje grande en tareas de generación de SQL. Los patrones de error identificados (dificultades con joins complejos, interpretación ambigua de agregaciones) sugieren áreas donde se requiere investigación adicional para mejorar la robustez de estos sistemas.

Limitaciones del estudio

Es necesario reconocer las limitaciones que afectan la interpretabilidad y generalizabilidad de los resultados:

Limitaciones metodológicas

1. **Tamaño de muestra:** El conjunto de 20 consultas, aunque diseñado para cubrir las categorías principales de información, representa una muestra limitada del universo posible de consultas que los usuarios podrían formular.
2. **Evaluador único:** Las mediciones fueron realizadas por el investigador, lo que introduce potencial sesgo del observador. La evaluación de precisión, aunque basada en criterios objetivos (coincidencia con consulta SQL directa), podría beneficiarse de evaluadores independientes.
3. **Ausencia de usuarios finales:** El prototipo no fue evaluado con dueños de negocio reales, lo que limita la validez ecológica de los resultados. La formulación de consultas por parte del investigador puede diferir de cómo los usuarios reales expresarían sus necesidades de información.
4. **Entorno controlado:** Las pruebas se realizaron en un entorno de desarrollo con datos sintéticos, no en el sistema de producción con datos reales y carga operativa.

Limitaciones de alcance

1. **Generalización limitada:** Los resultados son específicos para el esquema de base de datos de Billease y podrían no ser directamente transferibles a otros sistemas de facturación con estructuras diferentes.
2. **Idioma único:** El sistema fue desarrollado y evaluado exclusivamente en español. El

comportamiento con otros idiomas no fue probado.

3. **Modelo de lenguaje específico:** Los resultados corresponden a GPT 4.1 mini; otros modelos podrían presentar desempeños diferentes.

Limitaciones técnicas

1. **Dependencia de servicios externos:** El sistema requiere conexión a los servicios de OpenAI, lo que introduce dependencia de disponibilidad y latencia de servicios en la nube.
2. **Costos operativos:** No se evaluaron los costos asociados al uso del API de GPT en un escenario de producción con múltiples usuarios concurrentes.

Recomendaciones para investigaciones futuras

A partir de las limitaciones identificadas y los hallazgos del estudio, se proponen las siguientes líneas de investigación futura:

1. **Evaluación con usuarios finales:** Conducir estudios de usabilidad con dueños de negocio reales para validar la efectividad del sistema en condiciones naturales de uso.
2. **Ampliación del conjunto de pruebas:** Desarrollar un benchmark más extenso que incluya mayor variedad de consultas y casos límite.
3. **Comparación de modelos:** Evaluar el desempeño del sistema con diferentes modelos de lenguaje (GPT-4, Gemini, modelos de código abierto) para identificar el más adecuado para esta aplicación.
4. **Análisis de costos:** Realizar un estudio de viabilidad económica que considere los costos de API en escenarios de uso realistas.

Conclusiones

Conclusiones generales

La presente investigación tuvo como propósito desarrollar un sistema de consultas conversacionales que permitiera a los dueños de negocio acceder a información operativa de Billease mediante lenguaje natural, utilizando la integración del protocolo MCP con un modelo de lenguaje grande y la base de datos PostgreSQL. Tras el diseño, desarrollo y evaluación del prototipo, se concluye que:

El objetivo general fue alcanzado satisfactoriamente, el sistema desarrollado demuestra la viabilidad técnica y funcional de integrar el protocolo MCP con modelos de lenguaje grande para crear interfaces conversacionales de acceso a bases de datos empresariales. El prototipo permite a usuarios sin conocimientos técnicos en SQL consultar información operativa mediante preguntas en lenguaje natural, recibiendo respuestas precisas en tiempos significativamente menores al proceso tradicional.

Conclusiones específicas

En relación con cada objetivo específico, se concluye lo siguiente:

Objetivo Específico 1: clasificación de reportes

Se identificaron y clasificaron exitosamente **4 tipos de reportes principales** (ventas, compras, utilidad y cartera) que representan las necesidades de información más frecuentes de los dueños de negocio. Esta clasificación sirvió como base para el diseño de las herramientas del servidor

MCP.

Objetivo Específico 2: diseño de arquitectura

Se diseñó una arquitectura con 2 componentes principales (*Backend* Orquestador, Servidor MCP) que implementa los principios del protocolo MCP y garantiza la separación de responsabilidades entre la interfaz de usuario, la lógica de integración y el acceso a datos. El diseño incorpora mecanismos de seguridad en múltiples niveles (comportamiento del modelo, acceso a la información, etc).

Objetivo Específico 3: desarrollo del prototipo

Se desarrolló un prototipo funcional que incluye **ocho herramientas especializadas** para consulta de información operativa. El sistema demostró **100 % de efectividad** en el bloqueo de operaciones de escritura/actualización/eliminación, garantizando la integridad de los datos de producción.

Objetivo Específico 4: validación del modelo

La validación mediante 60 observaciones (20 consultas × 3 repeticiones) arrojó los siguientes resultados:

1. **Tiempo promedio de respuesta: 13.50 segundos**, lo que representa una reducción del 63.96 % respecto al proceso tradicional (37.46 segundos) y cumple ampliamente el umbral de menos de 36 segundos establecido en la hipótesis.
2. **Precisión global: 93.75 %**, superando el umbral del 90 % definido en la hipótesis. La precisión fue mayor en consultas simples (100 %) que en complejas (87.5 %).

Verificación de hipótesis

La hipótesis planteada establecía que *“El prototipo de consultas conversacionales permitirá: (1) reducir el tiempo de latencia en la obtención de reportes en un 10% frente al promedio actual de 36 segundos; (2) disminuir el tiempo de análisis e interpretación de la información de 30 minutos a un tiempo inferior a 1 minuto; y (3) asegurar una consistencia de datos con una precisión igual o superior al 90% respecto a los registros del sistema Billease.”*.

Se concluye que la hipótesis ha sido validada. Los resultados demuestran que:

1. El tiempo se redujo a **13.50 segundos** (< 37.46 segundos) ✓
2. La precisión alcanzó **93.75 %** ($\geq 90\%$) ✓

Contribuciones del trabajo

Contribución práctica

Se desarrolló un artefacto funcional que puede servir como base para la implementación de interfaces conversacionales en sistemas de facturación electrónica. El código del servidor MCP y la documentación técnica quedan disponibles como referencia para desarrollos similares.

Contribución metodológica

Se documentó la aplicación de la metodología Design Science Research (DSRM) al desarrollo de sistemas de integración LLM-base de datos, proporcionando un marco replicable para investigaciones futuras en este campo emergente.

Contribución al conocimiento

Se generó evidencia empírica sobre el desempeño de integraciones MCP-LLM-PostgreSQL en contextos empresariales hispanohablantes, contribuyendo a un área con escasa literatura previa.

Recomendaciones

Recomendaciones para Billease

1. **Normalización de Base de Datos:** Implementar una normalización en la base de datos, muchos nombres de columnas no tienen una correlación directa con el dato que muestran.
2. **Mejora comentarios de las columnas de la Base de Datos:** Añadir comentarios sobre que datos guarda una columna de una tabla, para así el modelo pueda contar con mayor contexto de la tabla y los datos que se guardan ahí.
3. **Implementación piloto:** Se recomienda desplegar el sistema en fase piloto con un grupo reducido de clientes (10-20 usuarios) para validar su funcionamiento en condiciones reales de uso.
4. **Monitoreo y mejora continua:** Implementar registros de uso que permitan identificar patrones de consulta frecuentes y casos de error para refinamiento iterativo del sistema.
5. **Capacitación de usuarios:** Desarrollar material de orientación que ayude a los usuarios a formular consultas efectivas.

Recomendaciones para futuras investigaciones

1. Evaluar el sistema con usuarios finales reales mediante estudios de usabilidad.
2. Ampliar el conjunto de pruebas para incluir mayor variedad de consultas.
3. Comparar el desempeño con diferentes modelos de lenguaje.

4. Investigar técnicas de optimización de prompts para mejorar precisión en consultas complejas.
5. Analizar la viabilidad económica en escenarios de producción.

Consideraciones finales

El desarrollo de interfaces conversacionales para acceso a datos empresariales representa una oportunidad significativa para democratizar el uso de información en pequeñas y medianas empresas. La convergencia de protocolos estandarizados como MCP, modelos de lenguaje cada vez más capaces, y bases de datos relacionales maduras, crea las condiciones para que usuarios sin formación técnica puedan beneficiarse de los datos que sus propios sistemas generan.

El presente trabajo demuestra que esta visión es técnicamente viable y ofrece mejoras sustanciales respecto a los procesos tradicionales de acceso a información. No obstante, la implementación en producción requerirá atención cuidadosa a aspectos de seguridad, costos operativos y experiencia de usuario que exceden el alcance de este prototipo.

La investigación en este campo continuará evolucionando a medida que los modelos de lenguaje mejoren su capacidad de razonamiento sobre esquemas de datos y generación de SQL. Los principios de diseño y lecciones aprendidas documentados en este trabajo pueden servir como referencia para quienes emprendan desarrollos similares en el futuro.

Referencias Bibliográficas

- Acero Carrión, R. F. (2022). *Modelo esquema para el análisis y tratamiento de la información en empresas ecuatorianas de comercio exterior mediante el uso de BIG DATA* [Tesis de grado]. Universidad Politécnica Salesiana (UPS).
<https://dspace.ups.edu.ec/bitstream/123456789/23637/1/UPS-GT003986.pdf>
- Amazon, A. W. S. (2026). *¿Qué es un LLM (modelo de lenguaje de gran tamaño)?* [Consultado el 26 de enero de 2026].
- Bradley, D., Clarke, J., Lee, S., & Ornthanalai, C. (2014). Are Analysts' Recommendations Informative? Intraday Evidence on the Impact of Time Stamp Delays. *The Journal of Finance*, 69(2), 645-673. <https://doi.org/10.1111/jofi.12107>
- Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33, 1877-1901.
- Castro Agüero, F., Cruz Aldave, L., Jiménez Romero, K., & Mercedes Rojas, K. (2025). *Plan de negocio para la creación de una empresa de consultoría especializada en el desarrollo de modelos de inteligencia artificial generativa para la interpretación de datos en sistemas ERP SAP, dirigida a empresas en Perú* [Tesis de Maestría]. Universidad ESAN.
<https://repositorio.esan.edu.pe/items/be440466-ee95-49a3-ac87-a56fe49f9950>

- Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., et al. (2024). A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3), 1-45. <https://doi.org/10.1145/3641289>
- Chusino Chabla, L. E. (2024). *Creación de un prototipo de chatbot que permita interactuar con la historia del Ecuador registrada en periódico antiguos* [Trabajo de titulación]. Universidad de Cuenca. <https://rest-dspace.ucuenca.edu.ec/server/api/core/bitstreams/d815c03b-cf8b-4e4c-acbc-0e74f0301896/content>
- Coba Medina, J. F., & Revelo Paz, E. A. (2021). *Implementación de una solución de inteligencia de negocios para el análisis de datos relacionados con los proyectos de software del sector público en el Ecuador en la última década* [Trabajo de titulación]. Universidad Politécnica Salesiana Sede Quito. <https://dspace.ups.edu.ec/bitstream/123456789/20667/1/UPS%20-%20TTS418.pdf>
- Codd, E. F. (1970). A relational model of data for large shared data banks. *Communications of the ACM*, 13(6), 377-387. <https://doi.org/10.1145/362384.362685>
- EDICOM. (2025, julio). *Factura electrónica en Ecuador: requisitos, normativa y soluciones*. Consultado el 5 de octubre de 2025, desde <https://edicom.co/blog/como-es-la-factura-electronica-en-ecuador>
- Gahona Cañar, J. J. (2025). *Chatbot interactivo basado en voz para responder a preguntas frecuentes acerca de la Universidad Técnica Particular de Loja* [Trabajo de titulación]. Universidad Técnica Particular de Loja. <https://bibliotecautpl.utpl.edu.ec/cgi-bin/abnetclwo/OoDgYKXCdJWhvTo9ZJKCO4LP2vT?MLKOB=101277195656>

García Ames, N., Quequesana Gutiérrez, G., & Sarmiento Espinoza, B. (2023). La digitalización empresarial: un fenómeno de la era post-pandemia. *Revista Lidera*, (18), 62-69.

Haro, A. F., Martínez, E. J., Chango, T. S., Zambrano, T. P., & Zambrano, M. F. (2023). Enterprise Resource Planning (ERP): procesos para una implementación óptima y eficiente. *Prometeo Conocimiento Científico*, 3(1), e21.
<https://doi.org/10.55204/pcc.v3i1.e21>

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS quarterly*, 28(1), 75-105. <https://doi.org/10.2307/25148625>

Jan, S. U., Shah, K., & Mand, N. (2023, febrero). Point of Sale System.
<https://doi.org/10.2139/ssrn.4368998>

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., et al. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, 33, 9459-9474.

LF Projects. (2025a). *Architecture overview*. Consultado el 4 de octubre de 2025, desde <https://modelcontextprotocol.io/docs/learn/architecture>

LF Projects. (2025b). *What is the Model Context Protocol (MCP)?* Consultado el 4 de octubre de 2025, desde <https://modelcontextprotocol.io/docs/getting-started/intro>

LiveConnect. (2025). *Whatsapp Multiagente, Instagram Web y Facebook para Ecuador*. Consultado el 5 de octubre de 2025, desde <https://liveconnect.chat/ec/whatsapp-multiagente-ecuador>

Mergers & Inquisitions. (2024). *Equity Research Analyst Career Path: The Complete Guide*.

Consultado el 27 de enero de 2026, desde

<https://mergersandinquisitions.com/equity-research-analyst/>

Miranda Miranda, G. A. (2025). *La anticipación estratégica y la eficiencia operativa en la empresa Incalza S.A.* [Trabajo de titulación]. Universidad Técnica de Ambato.

<https://repositorio.uta.edu.ec/server/api/core/bitstreams/f68f6f96-1906-46eb-8558-bcce0d586eef/content>

Muñoz Ruano, J., Segura, M., & Mendoza, J. (2022). An ERP System for SMEs in Latin America: Literature Review. *Revista Ingeniería Solidaria*.

<https://revistas.ucc.edu.co/index.php/in/article/view/4128>

OpenReview. (2025). Data-Centric Text-to-SQL with Large Language Models.

<https://openreview.net/>

Oracle AI & Data Science Blog. (2025). *How Text-to-SQL Can Help the Enterprise Engage with Databases*. Consultado el 15 de febrero de 2026, desde

<https://blogs.oracle.com/ai-and-datascience>

Parrales Carvajal, V. M., Aguirre Sanabria, M. E., Gómez Costain, D. J., &

Merchán Jácome, V. A. (2021). Control de inventario para la acertada toma decisiones gerenciales en las pequeñas y medianas empresas de la ciudad de Babahoyo. *Journal of Science and Research*. <https://revistas.utb.edu.ec/index.php/sr/article/view/1185>

Peffer, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of management information systems*, 24(3), 45-77. <https://doi.org/10.2753/MIS0742-1222240302>

- Ponce, M. P., Migo, G., & István, R. M. (2024). Sistemas de recuperación aumentada (RAG): una propuesta de investigación para potenciar las búsquedas semánticas y el contexto interactuando con inteligencia artificial generativa. *Libro de Actas – XXVI Workshop de Investigadores en Ciencias de la Computación (WICC 2024)*, 74-78.
- Ramírez Camacho, M. A. (2026). *Sistema Conversacional Basado en Arquitectura Tool Calling para Optimización de Interacción en Sistemas ERP* [Tesis de maestría, Universidad de los Andes]. <https://hdl.handle.net/1992/77677>
- Ramos De Santis, P. (2024). Satisfacción del cliente en la logística: un análisis de chatbots en las empresas líderes de Colombia, Perú y Ecuador. *Retos: Revista de Ciencias de Administración y Economía*, 115-130.
<https://revistas.ups.edu.ec/index.php/retos/article/view/7995>
- ResearchGate. (2025). AI-Driven ERP Systems: Integrating Large Language Models for Enhanced Customer Interaction and Operational Efficiency. <https://www.researchgate.net/>
- Sánchez-Bayón, A., Yanovskiy, M., Socol, Y., & Sastre, F. J. (2025). Comparación entre modelos de inteligencia artificial: USA-Empresarial vs. UE-Universitario. *Observatorio de las Ciencias Sociales en Iberoamérica*, 6(3), 62-76. <https://doi.org/10.51896/ocsi.v6i3.942>
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 5998-6008.
- Vega Flor, J. G., Carrillo Pulgar, W. G., Huilcapi Peñafiel, S. I., & Zurita Vaca, G. M. (2024). La importancia de las herramientas de la administración financiera en la toma de decisiones: un enfoque en las PYMES del Ecuador. *Ibero-American Journal of Education & Society Research*, 103-109. <https://doi.org/10.56183/iberoeds.v4iS.679>

- Villamil Rodríguez, J. H. (2021). *Diseño e Implementación de un Sistema POS, con Módulo de Gestión de Inventario de Productos para Clientes y Perfiles de Usuario, Aplicando Metodología RUP* [Tesis de grado]. Universidad Nacional Abierta y a Distancia.
<https://repository.unad.edu.co/bitstream/handle/10596/41806/jhvillamilr.pdf?sequence=1&isAllowed=y>
- Wang, B., Shin, R., Liu, X., Polozov, O., & Richardson, M. (2020). RAT-SQL: Relation-aware schema encoding and linking for text-to-SQL parsers. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7567-7578.
<https://doi.org/10.18653/v1/2020.acl-main.677>
- Yu, T., Zhang, R., Yang, K., Yasunaga, M., Wang, D., Li, Z., Ma, J., Li, I., Yao, Q., Roman, S., et al. (2018). Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 3911-3921.
<https://doi.org/10.18653/v1/D18-1425>

Apéndices

Errores de restricciones de la API de comunicaciones de Twilio

```
HTTP Error Your request was:  
  
POST /Accounts/AC6aafa7fcf396a1661dbffec88f85f93f/Messages.json  
  
Twilio returned the following information:  
  
Unable to create record: Account AC6aafa7fcf396a1661dbffec88f85f93f exceeded the 50 daily messages limit  
  
More information may be available here:  
  
https://www.twilio.com/docs/errors/63038
```

Figura 13

Log de excepción por superación de cuota diaria en API Twilio.

Nota: Log de la excepción por superación de la cuota diaria de peticiones de la API Twilio. Fuente de elaboración propia.

Confirmación de uso de base de datos de Billease

La siguiente documentación corresponde a la confirmación firmada por el Ing. Guillermo Cevallos, CEO de la empresa Billease, para la autorización de acceder a la base de datos de desarrollo del sistema Billease para peticiones de solo lectura.

Solicitud de Acceso a Base de Datos de Desarrollo (Read-only)

Dirigido a: Ing. Guillermo Cevallos

Introducción El presente documento tiene como objetivo formalizar la solicitud de acceso a la base de datos en el entorno de **desarrollo** del sistema BillEase. Esta autorización es un requisito técnico indispensable para la fase de implementación y pruebas del proyecto de tesis, el cual consiste en la creación de un modelo de comunicación proactiva basado en el protocolo MCP (Model Context Protocol).

Con el fin de cumplir con el objetivo de validación del prototipo: “*DESARROLLO DE UN SISTEMA DE COMUNICACIÓN Y PROTOCOLO MCP PARA LA OPTIMIZACIÓN DEL ACCESO A INFORMACIÓN EN EL SISTEMA BILLEASE*”,

solicitamos a la empresa proveedora la concesión de credenciales de acceso limitado a la base de datos de pruebas.

Detalle del Acceso Requerido Para garantizar la seguridad e integridad de la información, el acceso solicitado se rige bajo las siguientes condiciones técnicas estrictas:

1. **Entorno:** Base de datos de Desarrollo (Testing/Staging).
2. **Permisos:** Exclusivamente de **Lectura (Read-Only)** (SELECT). No se requieren permisos de escritura, modificación ni eliminación (INSERT, UPDATE, DELETE).
3. **Finalidad:** Conexión técnica del servidor MCP para pruebas de consulta de información (Reportes, Inventario, etc.) mediante Inteligencia Artificial.

Compromiso de Confidencialidad y Uso Los estudiantes responsables del proyecto se comprometen a:

- Utilizar los datos únicamente con fines académicos y de investigación.
- No divulgar, compartir ni comercializar la estructura o el contenido de la base de datos.
- Mantener las credenciales otorgadas bajo estricta seguridad.

Confirmación Mediante la firma de este documento, la empresa autoriza el acceso a la base de datos de desarrollo bajo las condiciones de "Solo Lectura" descritas anteriormente, permitiendo su uso exclusivo para el desarrollo del proyecto de tesis.

Firma del representante autorizado

Nombre: Ing. Guillermo Cevallos

Cargo: CEO

Fecha:

Clasificación de los reportes más usados del Sistema Billease

La siguiente documentación, firmada por el Ing. Guillermo Cevallos, corresponde con la información de los cuatro reportes más usados por sus clientes en el sistema Billease.

Documento de Clasificación de Reportes Prioritarios – Sistema Billease

Dirigido a:

Ing. Guillermo Cevallos

Introducción

El presente documento tiene como objetivo recopilar y clasificar los reportes del sistema Billease que son considerados prioritarios para la toma de decisiones por parte de los dueños y administradores del negocio. Esta información forma parte del desarrollo de una tesis orientada a la creación de un modelo conversacional basado en MCP, cuyo propósito es reducir el tiempo de obtención y análisis de datos generados por la plataforma Billease.

Con el fin de cumplir con el objetivo específico:

“Clasificar los reportes y datos del sistema Billease que son prioritarios para la toma de decisiones por parte de los dueños de negocio”,
solicitamos a la empresa proveedora del sistema la validación de los reportes más utilizados por los usuarios en su operación diaria.

A continuación, se presenta un listado referencial. La empresa podrá marcar, modificar o complementar según su criterio y políticas de seguridad:

Reportes más utilizados

1. **Reporte de Ventas**
2. **Reporte de Compras**
3. **Reporte de Utilidad**
4. **Reporte de Cartera**

Confirmación

Mediante la firma de este documento, la empresa confirma que los reportes seleccionados representan aquellos que, conforme a la experiencia operativa, son más utilizados para la toma de decisiones dentro del negocio.

Firma del representante autorizado

Nombre: Ing. Guillermo Cevallos

Cargo: CEO

Fecha: 02/12/2025

Diseño de la arquitectura de integración basada en el protocolo MCP

La siguiente documentación se presenta el esquema técnico de la arquitectura de la solución. Este documento desglosa los módulos que componen el servidor MCP, las capas del Backend Orquestador en la cual está incluido el cliente MCP.



**Pontificia Universidad
Católica del Ecuador**
Seréis mis testigos

MANABÍ

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
SEDE MANABÍ**

CARRERA DE SOFTWARE

DESCRIPCIÓN TÉCNICA DEL SISTEMA MCP – BILLIEASE

**DISEÑO DE LA ARQUITECTURA DE INTEGRACIÓN BASADA EN EL
PROTOCOLO MCP**

AUTORES

**BYRON SANTIAGO SERRANO ZAMBRANO
JULIO RAMÓN VINCES BARRETO**

TUTOR

ING. JOSSELYN GOMEZ, MSc.

PORTOVIEJO, FEBRERO 2026

Tabla de Contenido

1. Introducción	1
Visión general de la arquitectura de integración	2
Servidor MCP como componente central de integración	4
Capa de transporte y exposición del MCP	5
Núcleo MCP y registro de capacidades	6
Capa de herramientas	7
Capa de recursos	8
Capa de acceso a datos y lógica de dominio.	9
Cliente MCP y su Integración con el LLM	10
Capa de Entrada y Gestión de Solicitudes	10
Orquestación y Control del Flujo de Integración	11
Integración con el Modelo de Lenguaje	11
Cliente MCP	11
Flujo General de Integración entre LLM, MCP y PostgreSQL	12
Mecanismos de autenticación y seguridad en MCP	15
Protocolo de negociación de conexión (Handshake)	15
Consideraciones de Diseño de la Arquitectura de Integración	17

Lista de Figuras

1.	Diagrama general del diseño.	2
2.	Diagrama de división de responsabilidades.	4
3.	Diagrama de la capa de transporte y exposición.	6
4.	Diagrama del núcleo del servidor MCP.	7
5.	Diagrama de las herramientas del servidor MCP.	8
6.	Diagrama de los recursos del servidor MCP.	9
7.	Diagrama de acceso a datos y lógica de dominio.	10
8.	Diagrama del backend orquestador.	12
9.	Diagrama de secuencia del backend orquestador.	14
10.	Diagrama de secuencia del proceso de autenticación y conexión SSE.	17

Introducción

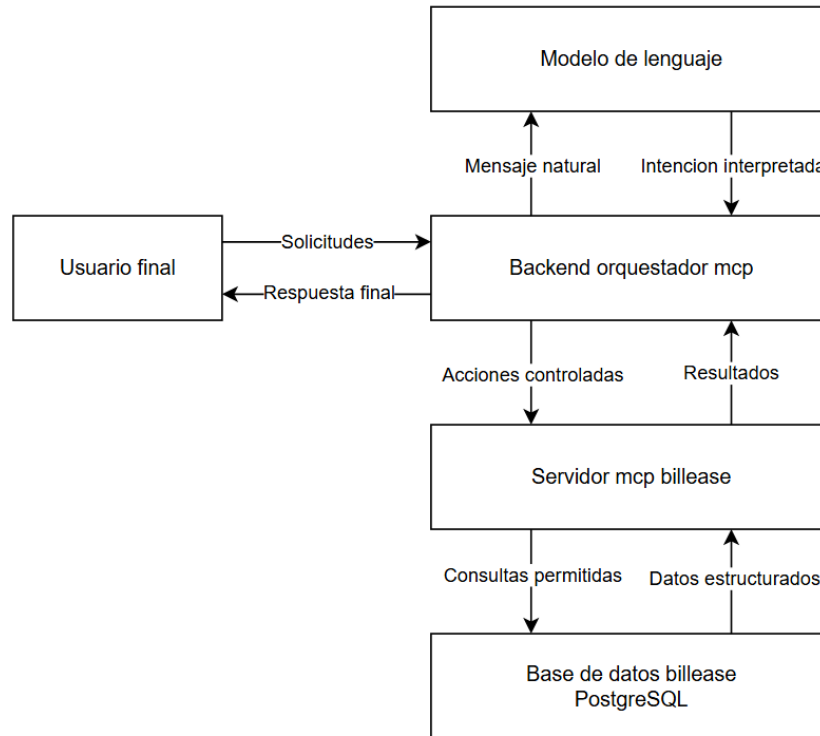
El presente documento desarrolla el diseño de la arquitectura de integración propuesta para el sistema Billease, cuyo propósito es habilitar la comunicación controlada entre un modelo de lenguaje (LLM) y la base de datos PostgreSQL del sistema, utilizando el protocolo Model Context Protocol (MCP) como mecanismo central de interoperabilidad. Este documento responde directamente al objetivo específico de la investigación:

"Diseñar la arquitectura de integración basada en el protocolo MCP, que conecte el modelo de lenguaje (LLM) con la base de datos PostgreSQL del sistema Billease."

Desde una perspectiva de ingeniería de software, el problema abordado no consiste únicamente en permitir que un modelo de lenguaje consulte información, sino en definir una arquitectura que garantice control, trazabilidad, seguridad y mantenibilidad a largo plazo. En este contexto, la arquitectura propuesta evita explícitamente el acceso directo del modelo de lenguaje a la base de datos, introduciendo una capa intermedia especializada que regula las operaciones permitidas y encapsula la lógica de negocio.

La solución se materializa mediante dos proyectos independientes a nivel de despliegue, pero estrechamente acoplados a nivel funcional: un servidor MCP que expone capacidades del sistema Billease, y un orquestador que es el cliente MCP, que integra el modelo de lenguaje con dichas capacidades. En conjunto, ambos conforman una única arquitectura lógica orientada a la integración entre inteligencia artificial y sistemas de información empresariales, como se puede observar en la Figura 1:

Figura 1. Diagrama general del diseño.



Fuente: Elaboración propia a partir de datos del diseño de la solución.

Visión general de la arquitectura de integración

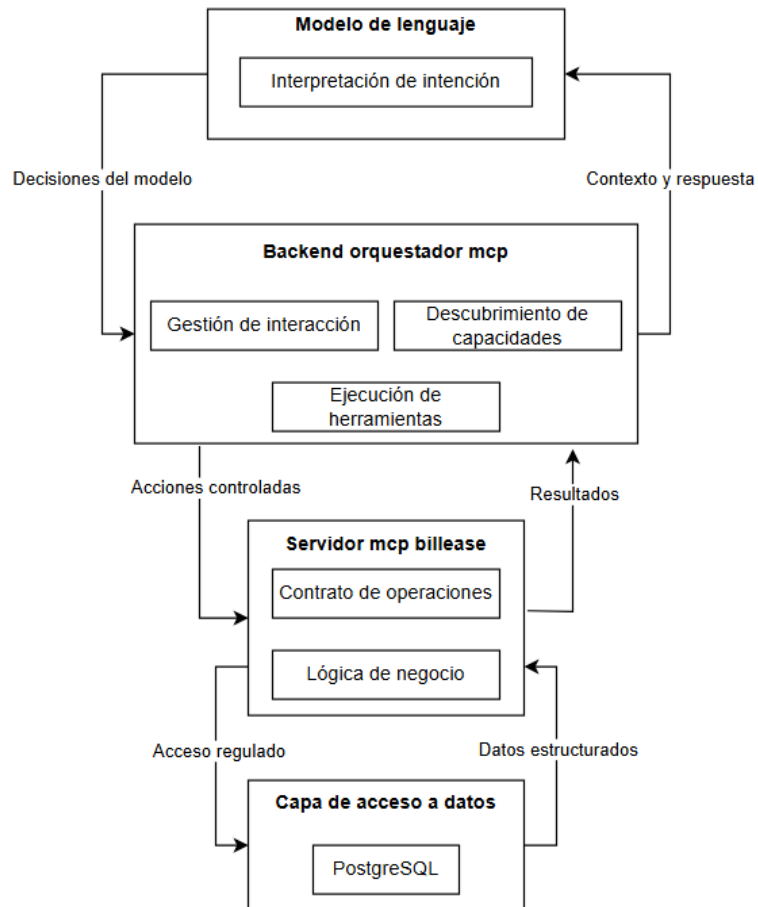
La arquitectura de integración diseñada se apoya en el principio de separación de responsabilidades, estableciendo límites claros entre los componentes que interactúan con el modelo de lenguaje, aquellos que exponen la lógica del negocio y los que acceden a la base de datos. En lugar de permitir que el LLM genere consultas SQL o interactúe directamente con PostgreSQL, el diseño introduce un servidor MCP como intermediario formal. Este servidor define, mediante un contrato explícito, qué operaciones están disponibles, bajo qué parámetros pueden ejecutarse y qué tipo de información puede ser retornada.

El cliente MCP actúa como mediador entre el LLM y el servidor MCP. Su función no se limita a reenviar mensajes, sino que coordina el ciclo completo de interacción: recepción de solicitudes del usuario, inferencia del modelo de lenguaje, descubrimiento dinámico de capacidades MCP, ejecución de herramientas y síntesis de respuestas finales. A nivel conceptual, la arquitectura se estructura en los siguientes bloques principales:

- Modelo de Lenguaje (LLM), responsable de interpretar la intención del usuario y decidir cuándo es necesario acceder a información estructurada.
- Cliente MCP, encargado de traducir las decisiones del modelo en acciones técnicas controladas.
- Servidor MCP del sistema Billease, que expone herramientas y recursos de acceso a datos bajo el protocolo MCP.
- Capa de acceso a datos, responsable de la interacción directa con PostgreSQL.

Para visualizar la interacción y los límites funcionales de estos componentes, se presenta la Figura 2. Este esquema ilustra la segmentación operativa del sistema, evidenciando cómo el servidor MCP actúa como la barrera de contención lógica que desacopla la inteligencia artificial de la capa de persistencia de datos.

Figura 2. Diagrama de división de responsabilidades.



Fuente: Elaboración propia a partir de datos sobre bloques de responsabilidades de la solución.

Servidor MCP como componente central de integración

El servidor MCP constituye el núcleo técnico de la arquitectura de integración. Su rol principal es actuar como intermediario entre el modelo de lenguaje y la base de datos PostgreSQL del sistema Billease, exponiendo un conjunto de capacidades bien definidas que pueden ser consumidas por clientes MCP autorizados. Desde el punto de vista del diseño, el servidor MCP no se li-

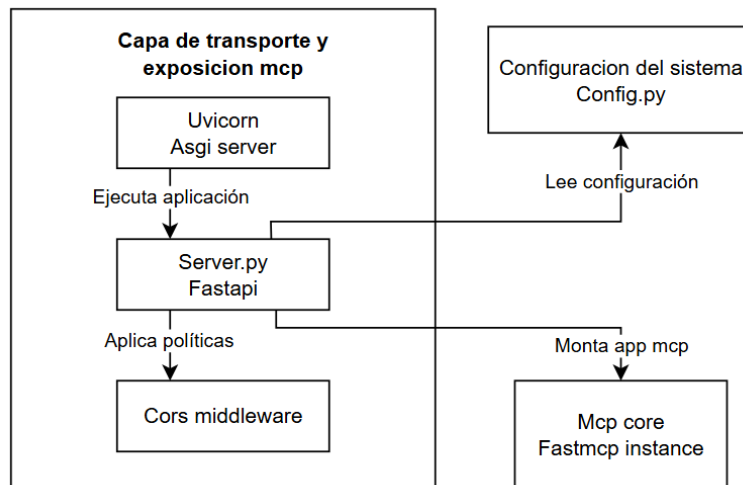
mita a servir datos, sino que define una frontera clara entre la inteligencia artificial y el sistema de información empresarial. Esta frontera permite controlar qué información puede ser consultada, cómo se consulta y bajo qué condiciones, reduciendo los riesgos asociados a consultas arbitrarias o no validadas.

Capa de transporte y exposición del MCP

La capa de transporte es responsable de exponer el servidor MCP como un servicio accesible mediante HTTP. Esta capa se implementa utilizando el framework FastAPI, el cual actúa como contenedor del servidor MCP y gestiona el ciclo de vida de la aplicación. El servidor MCP se monta como una aplicación HTTP de tipo streamable, permitiendo una comunicación eficiente con clientes MCP externos. Además, esta capa gestiona aspectos transversales como la inicialización y cierre controlado de sesiones MCP, así como la configuración de políticas de acceso que permiten la integración con distintos clientes.

Este diseño permite que el servidor MCP sea desplegado como un microservicio independiente dentro de la infraestructura de Billease, favoreciendo la escalabilidad y el aislamiento de responsabilidades.

Figura 3. Diagrama de la capa de transporte y exposición.



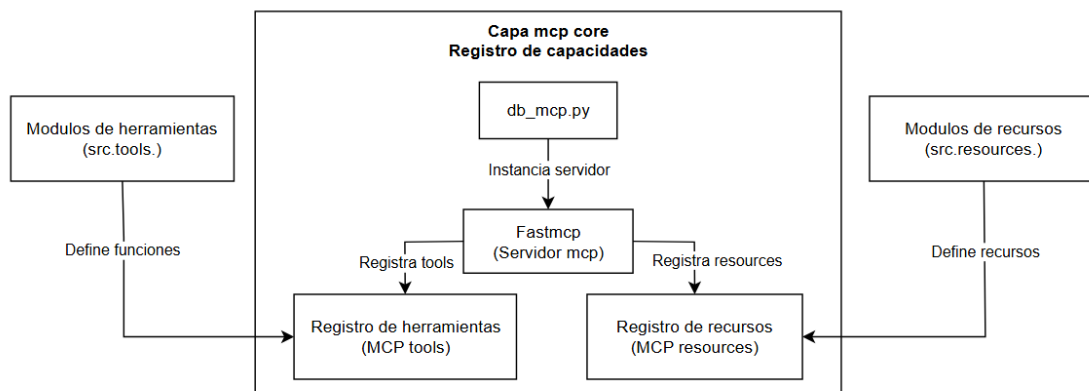
Fuente: Elaboración propia a partir de datos de la capa de transporte y exposición del servidor MCP.

Núcleo MCP y registro de capacidades

El núcleo MCP, implementado a través de la clase FastMCP, define el contrato de comunicación entre el servidor y los clientes. En esta capa se registran de manera declarativa las capacidades que el sistema pone a disposición del modelo de lenguaje, clasificadas en herramientas (tools) y recursos (resources). Un aspecto clave de este diseño es que el cliente MCP puede descubrir dinámicamente las capacidades disponibles, sin necesidad de conocer detalles internos de la implementación ni la estructura de la base de datos.

Esto refuerza el desacoplamiento entre la IA y el sistema Billease, y facilita la evolución del sistema sin afectar a los consumidores. Desde una perspectiva arquitectónica, esta capa concentra las relaciones más relevantes del sistema, ya que conecta al backend orquestador con la lógica de negocio y la capa de acceso a datos.

Figura 4. Diagrama del núcleo del servidor MCP.



Fuente: Elaboración propia a partir de datos del núcleo del servidor MCP.

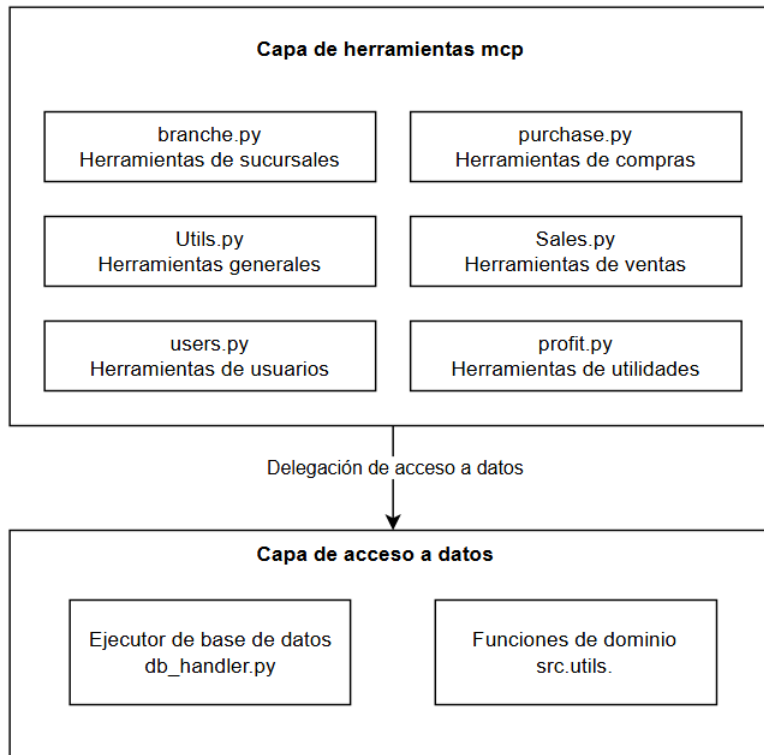
Capa de herramientas

La capa de herramientas representa el principal mecanismo de integración funcional entre el modelo de lenguaje y la información almacenada en PostgreSQL. Cada herramienta encapsula un caso de uso específico del sistema Billease, como la consulta de reportes de ventas, compras, usuarios o utilidades.

Las herramientas se definen como funciones decoradas con mecanismos propios del protocolo MCP, lo que permite su descubrimiento automático por parte del cliente. Este enfoque evita que el modelo de lenguaje deba generar consultas SQL o conocer la estructura interna de la base de datos.

Desde el punto de vista del diseño, esta capa cumple una función crítica: restringe el acceso del LLM a operaciones previamente definidas, actuando como una barrera de seguridad y como un punto de control lógico. Además, facilita la representación de diagramas donde cada herramienta puede modelarse como una operación bien delimitada dentro del sistema.

Figura 5. Diagrama de las herramientas del servidor MCP.



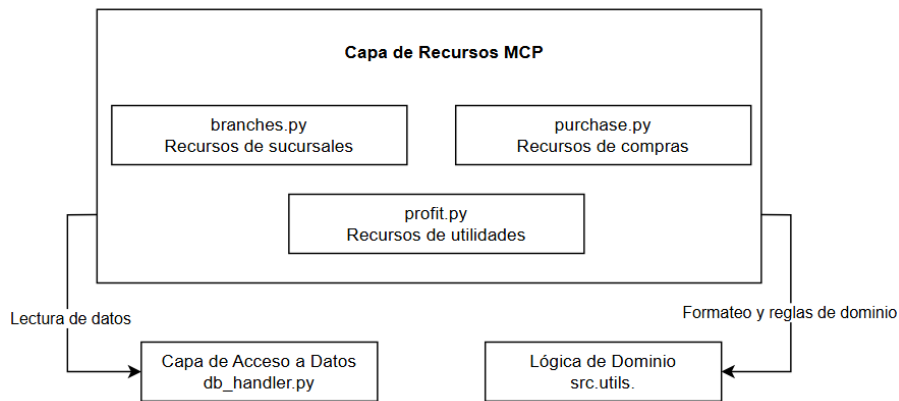
Fuente: Elaboración propia a partir de la estructura de herramientas del servidor MCP.

Capa de recursos

Los recursos del servidor MCP, complementan a las herramientas proporcionando información contextual estructurada que puede ser utilizada por el modelo de lenguaje para enriquecer sus respuestas. A diferencia de las herramientas, los recursos están orientados principalmente a datos de referencia o información relativamente estática.

La separación entre herramientas y recursos permite distinguir claramente entre operaciones activas (acciones a realizar) y contexto informativo, lo cual mejora la claridad del diseño y simplifica el análisis del flujo de datos dentro de la arquitectura.

Figura 6. Diagrama de los recursos del servidor MCP.



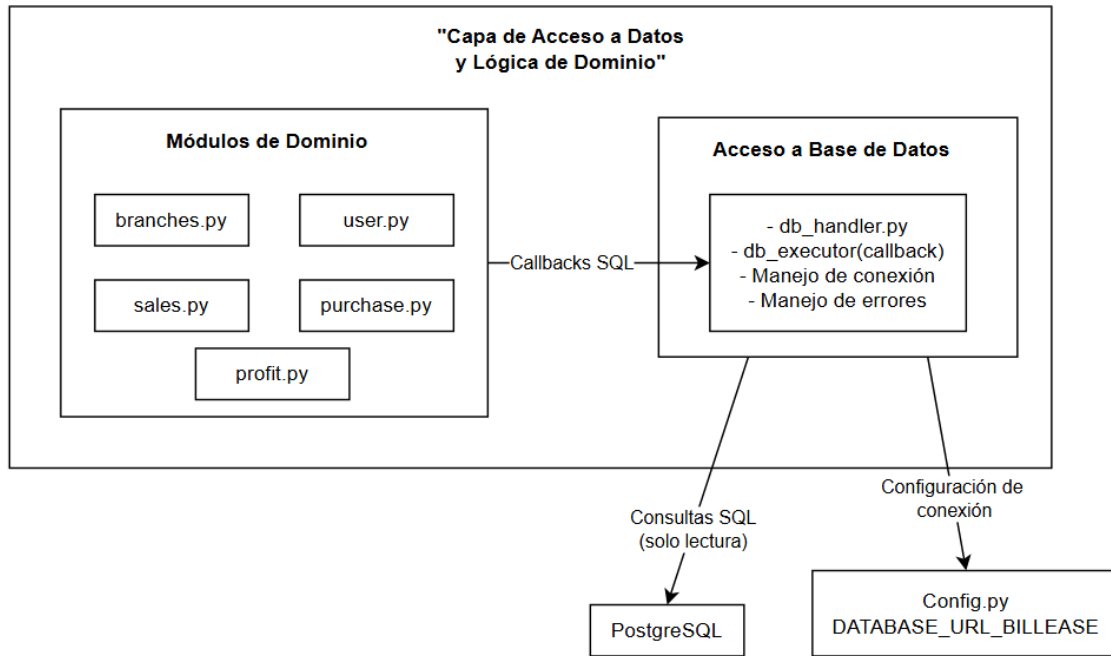
Fuente: Elaboración propia a partir de datos de la capa de recursos.

Capa de acceso a datos y lógica de dominio.

El acceso a la base de datos PostgreSQL se centraliza en una capa específica encargada de gestionar las conexiones, la ejecución de consultas y el manejo de errores. Este diseño evita la duplicación de lógica de acceso a datos y garantiza un comportamiento consistente en todo el sistema.

Las herramientas y recursos delegan la ejecución de consultas a esta capa mediante funciones especializadas, las cuales retornan los resultados en estructuras de datos fácilmente serializables. Este enfoque permite mantener las consultas en modo de solo lectura y refuerza el control sobre la información expuesta al modelo de lenguaje.

Figura 7. Diagrama de acceso a datos y lógica de dominio.



Fuente: Elaboración propia a partir de los datos de la capa de acceso a datos.

Ciente MCP y su Integración con el LLM

El cliente MCP actúa como el componente activo de la arquitectura, responsable de coordinar la interacción entre el usuario final, el modelo de lenguaje y el servidor MCP. A diferencia del servidor, cuyo rol es reactivo, el cliente gestiona el flujo conversacional y la toma de decisiones sobre cuándo es necesario acceder a información estructurada.

Capa de Entrada y Gestión de Solicitudes

La capa de entrada expone endpoints HTTP que permiten recibir mensajes desde interfaces externas, como WhatsApp. Esta capa se encarga de validar las solicitudes entrantes, normalizar los datos y delegar el procesamiento a los componentes internos.

Orquestación y Control del Flujo de Integración

La capa de orquestación constituye uno de los elementos más relevantes del diseño propuesto. En ella se centraliza la lógica que traduce la intención del usuario en acciones técnicas ejecutables, manteniendo la coherencia entre las decisiones del modelo de lenguaje y las operaciones permitidas por el sistema.

Este diseño convierte al cliente MCP en un mediador inteligente, capaz de controlar el flujo de información y garantizar que el acceso a la base de datos se realice únicamente a través de los mecanismos definidos por el protocolo MCP.

Integración con el Modelo de Lenguaje

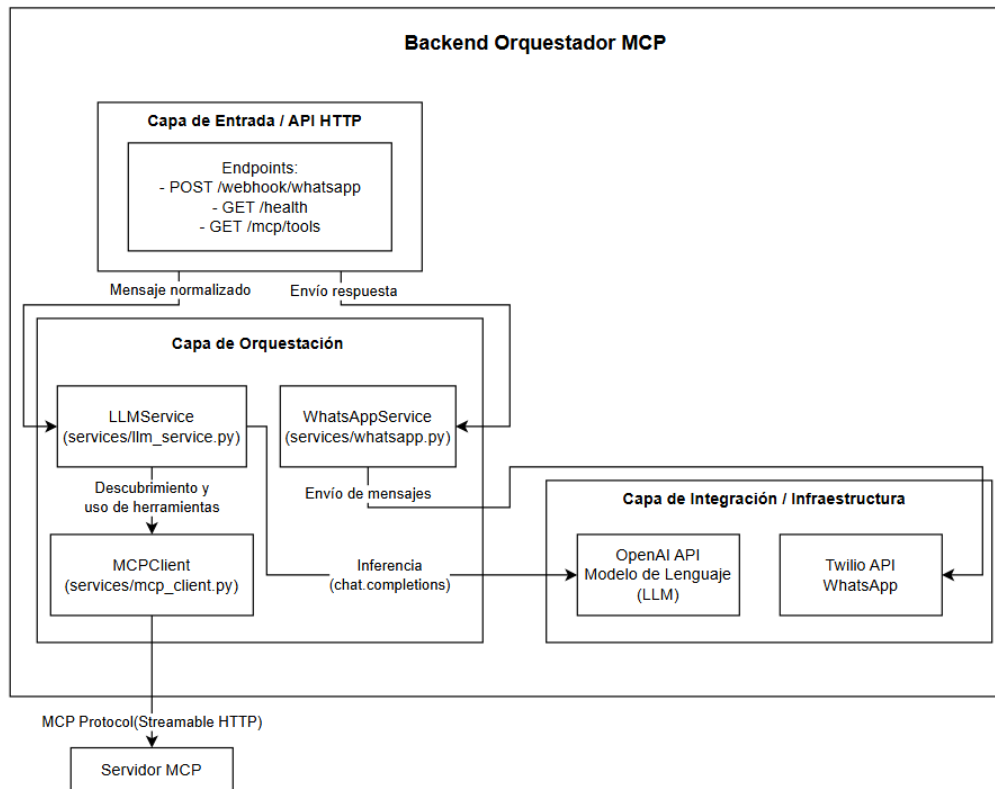
El componente encargado de la integración con el modelo de lenguaje implementa el ciclo completo de interacción entre el LLM y el servidor MCP. Este ciclo incluye la consulta de capacidades disponibles, la ejecución de herramientas cuando el modelo lo solicita y la síntesis de respuestas finales en lenguaje natural.

Desde el punto de vista arquitectónico, este componente es clave para la elaboración de diagramas de secuencia, ya que controla la iteración entre inferencia, ejecución de herramientas y generación de respuestas.

Cliente MCP

El cliente MCP implementa la lógica necesaria para comunicarse con el servidor MCP, manteniendo las sesiones de conexión y ejecutando llamadas a herramientas de forma controlada. Este componente abstrae los detalles del protocolo y simplifica la integración del backend orquestador con el servidor MCP.

Figura 8. Diagrama del backend orquestador.



Fuente: Elaboración propia a partir de datos sobre el diseño del backend orquestador.

Flujo General de Integración entre LLM, MCP y PostgreSQL

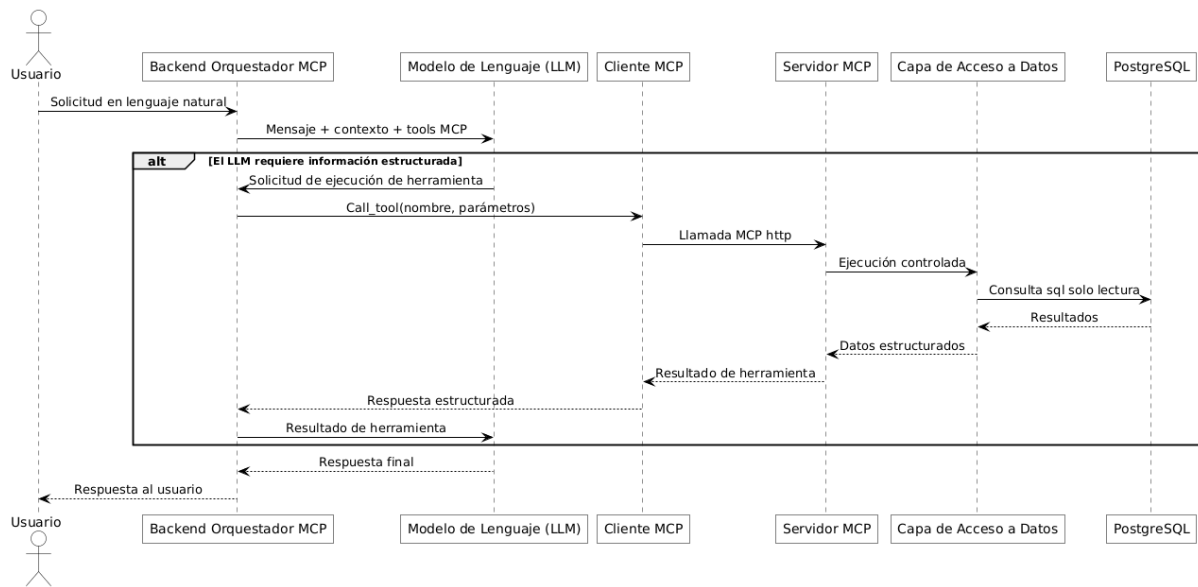
El flujo de integración propuesto se desarrolla de manera secuencial y condicional. Ante una solicitud del usuario, el backend orquestador envía el mensaje al modelo de lenguaje junto con el contexto y la definición de herramientas disponibles. Si el modelo determina que la respuesta requiere información estructurada, solicita la ejecución de una herramienta MCP.

El backend ejecuta dicha herramienta a través del cliente MCP, el servidor procesa la solicitud y accede a la base de datos PostgreSQL mediante la capa de acceso a datos. Los resultados obtenidos se retornan al modelo de lenguaje, el cual genera una respuesta final en lenguaje natural

que es enviada al usuario.

A continuación, la Figura 9 formaliza este procedimiento mediante un diagrama de secuencia. Esta representación gráfica permite validar la trazabilidad de la información paso a paso, desde la intención inicial del usuario hasta la recuperación final de los datos.

Figura 9. Diagrama de secuencia del backend orquestador.



Fuente: Elaboración propia a partir de los datos de la secuencia del backend orquestador.

Mecanismos de autenticación y seguridad en MCP

La naturaleza distribuida de la arquitectura propuesta, donde el MCP Client y el MCP Server operan como entidades independientes comunicadas vía red, exige la implementación de controles de acceso rigurosos. Dado que el protocolo Model Context Protocol opera sobre transportes estándar (como HTTP con Server-Sent Events - SSE), la estrategia de seguridad se centra en la protección de la capa de transporte antes de establecer el canal de comunicación persistente.

El diseño implementa un esquema de autenticación basado en tokens (Token-Based Authentication), el cual garantiza que únicamente instancias autorizadas del Cliente MCP puedan invocar las herramientas y recursos expuestos por el Servidor MCP de Billease.

Protocolo de negociación de conexión (Handshake)

El proceso de autenticación ocurre en la fase de inicialización de la conexión. A diferencia de las arquitecturas REST tradicionales donde cada petición es autenticada individualmente, en el protocolo MCP sobre SSE, la validación crítica se realiza en el momento del establecimiento del túnel de eventos (Handshake).

El flujo técnico se define de la siguiente manera:

1. **Solicitud de conexión:** El MCP Client inicia una petición HTTP GET hacia el endpoint del servidor (por defecto /sse). En esta petición, se inyecta una cabecera de autorización estándar:

Authorization: Bearer <TOKEN_DE_ACCESO>

2. **Intercepción y validación:** Antes de que la solicitud alcance el núcleo del protocolo MCP, un middleware de seguridad en el servidor intercepta el paquete. Este componente verifica la firma criptográfica del token y su vigencia contra el proveedor de identidad (en este caso, la configuración interna o un servicio como ScaleKit).

3. Decisión de acceso:

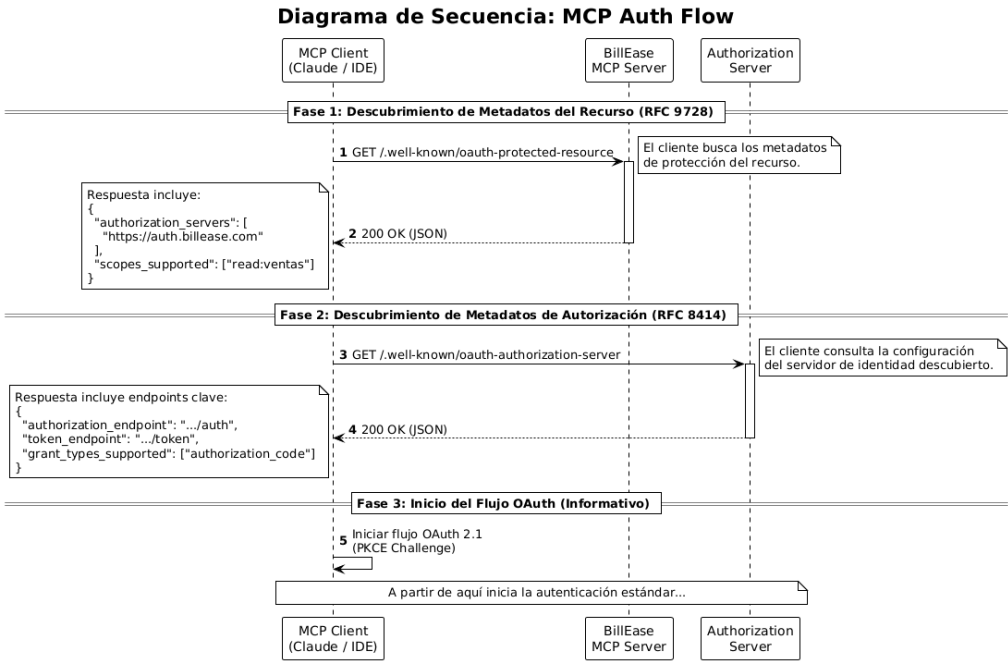
- Si el token es inválido o ha expirado, el servidor rechaza la conexión inmediatamente con un código de estado 401 `Unauthorized` o 403 `Forbidden`, impidiendo cualquier intercambio de datos.
- Si la validación es exitosa, el servidor responde con un código 200 `OK`, establece las cabeceras `Content-Type: text/event-stream` y actualiza el protocolo a una conexión persistente.

4. Intercambio de capacidades:

Una vez asegurado el canal, se procede con el intercambio de mensajes JSON-RPC de inicialización (`initialize`), donde cliente y servidor negocian sus capacidades funcionales (`prompts`, `resources`, `tools`).

Este mecanismo asegura que el descubrimiento de herramientas sensibles (como reportes financieros) sea inaccesible para actores no autenticados, protegiendo la integridad de la base de datos de Billease.

Figura 10. Diagrama de secuencia del proceso de autenticación y conexión SSE.



Fuente: Diagrama traducido al español de (LF Projects, 2026)

Consideraciones de Diseño de la Arquitectura de Integración

La arquitectura propuesta presenta varias ventajas relevantes para el contexto del sistema Billease. En primer lugar, refuerza la seguridad al limitar el acceso a datos mediante herramientas predefinidas y consultas controladas. En segundo lugar, facilita la escalabilidad, ya que nuevas capacidades pueden incorporarse al servidor MCP sin modificar el backend orquestador.

Asimismo, el uso del protocolo MCP mejora la interoperabilidad con distintos modelos de lenguaje y clientes, mientras que la separación por capas contribuye a la mantenibilidad y comprensión del sistema. En conjunto, esta arquitectura constituye una base sólida para la implementación de un sistema de acceso inteligente a información empresarial, alineado con los objetivos de la investigación y con buenas prácticas de ingeniería de software.

Referencias Bibliográficas

LF Projects. (2026). *Autorización - Protocolo de contexto modelo*. Consultado el 26 de enero de 2026, desde <https://modelcontextprotocol.io/specification/draft/basic/authorization>

Funcionalidad y Seguridad del Prototipo

A continuación, se presenta la evidencia técnica y visual que respalda la operatividad del prototipo desarrollado. Este documento compila los registros detallados y capturas de pantalla que validan dos aspectos fundamentales: (1) la correcta ejecución de las herramientas para extraer los datos cuatro reportes prioritarios (ventas, compras, cartera y utilidad), y (2) la efectividad de los mecanismos de seguridad ante intentos de modificación de datos (INSERT, UPDATE, DELETE).



PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
SEDE MANABÍ

CARRERA DE SOFTWARE

PRUEBAS DE FUNCIONALIDAD Y SEGURIDAD DEL PROTOTIPO

AUTORES

BYRON SANTIAGO SERRANO ZAMBRANO
JULIO RAMÓN VINCES BARRETO

TUTOR
ING. JOSSELYN GOMEZ, MSc.
PORTOVIEJO, FEBRERO 2026

En este archivo se presentan las evidencias correspondientes a las pruebas de funcionalidad y seguridad del prototipo MCP. El contenido está compuesto principalmente por capturas de pantalla que permiten visualizar el comportamiento del sistema ante distintas peticiones.

Pruebas de funcionalidad

En esta sección se presentan las pruebas de funcionalidad del prototipo MCP, donde se confirma que el modelo propuesto tiene la capacidad de identificar y seleccionar correctamente la herramienta adecuada según la petición recibida. Las evidencias se muestran a través de los registros generados en los logs del sistema, los cuales son almacenados en la base de datos en la nube (NeonDB). La validación de este comportamiento se respalda mediante la información registrada en la tabla, considerando los siguientes campos:

- **id**: que identifica de forma única cada registro
- **phone_number**: que indica el número desde el cual se envía la solicitud
- **messages_in**: que representa el mensaje recibido
- **messages_out**: que muestra la respuesta generada
- **mcp_tool_used**: que detalla las herramientas utilizadas durante el procesamiento
- **processing_time**: que refleja el tiempo de procesamiento interno
- **total_time**: que indica el tiempo total transcurrido desde el envío del mensaje hasta la entrega de la respuesta al usuario

Herramienta: get_report_sales

ID serial	phone_number varchar(15)	message_in text	message_out text	mcp_tools_used jsonb	processing_time double precision	Total_time double precision
1	+593980366561	¿Cuáles fueron las ven...	Las ventas totales del...	["get_report_sales"]	7.821165323257446	8.451007843017578
2	+593980366561	¿Qué cliente compró má...	No se encontraron regi...	["get_report_sales"]	4.822023868560791	5.043833017349243
3	+593980366561	Muéstrame las ventas d...	No se registraron vent...	["get_report_sales"]	6.308629035949707	6.521631956100464
4	+593980366561	¿Cuál fue el producto ...	El producto más vendid...	["get_report_sales"]	4.463860750190364	4.691803693771362
5	+593980366561	Dame el reporte de ven...	No se encontraron vent...	["get_report_sales"]	3.7706358432769775	3.978952169418335
6	+593980366561	¿Cuánto fue el subtota...	No se registraron vent...	["get_report_sales"]	3.9647774696350098	4.175518989562988
7	+593980366561	Lista los productos ve...	Estos son algunos de l...	["get_report_sales"]	7.357477903366009	7.575470447540283
8	+593980366561	¿Qué clientes compraro...	No se encontraron regi...	["get_report_sales"]	5.425134658813477	5.630405902802549
9	+593980366561	Muéstrame las ventas t...	En el último semestre ...	["obtain_current_datet...	8.510634183883667	8.716410398483276
10	+593980366561	¿Cuál fue el producto ...	No se encontraron regi...	["get_report_sales"]	3.6542704105377197	3.8793697357177734

Herramienta: get_report_purchases

11	+593980366561	¿Cuáles fueron las com...	Las compras totales re...	["get_report_purchase"]	5.254371404647827	5.470771789550781
12	+593980366561	Muéstrame las compras ...	No se registraron comp...	["get_report_purchase"]	4.628293514251709	4.83643913269043
13	+593980366561	¿Qué compras se hicier...	No se registraron comp...	["obtain_current_datet...	8.287305116653442	8.502196786787842
14	+593980366561	Dame el reporte de com...	No se encontraron regi...	["get_report_purchase"]	3.8370182514190674	4.04302716255108
15	+593980366561	¿Cuánto se gastó en co...	No se registraron gast...	["get_report_purchase"]	3.827781915664673	4.0499267578125
16	+593980366561	Lista todas las compra...	No hay registros de co...	["get_report_purchase"]	3.945140838623047	4.158735036849976
17	+593980366561	¿Qué compras se regist...	En el último trimestre...	["obtain_current_datet...	5.755763053894043	6.011415719985962
18	+593980366561	Muéstrame las compras ...	No se registraron comp...	["get_report_purchase"]	4.216265678405762	4.452206134796143
19	+593980366561	¿Cuáles fueron las com...	No se registraron comp...	["get_report_purchase"]	3.89493465423584	4.099677801132202
20	+593980366561	Dame el total de compr...	No se registraron comp...	["get_report_purchase"]	5.317335367202759	5.530741214752197

Herramienta: get_aging_report

23	+593980366561	¿Cuál es el estado de ...	En el mes de noviembre...	["get_aging_report"]	6.198416709899902	6.4235475063323971
24	+593980366561	Muéstrame el reporte d...	No se encontraron dato...	["obtain_current_datet...	5.2288124561309814	5.454603433609009
25	+593980366561	¿Qué cuentas por cobra...	Para poder proporcion...	["obtain_current_datet...	2.4649603366851807	2.6705541610717771
26	+593980366561	Dame el reporte de car...	El cliente con nombre ...	["get_aging_report"]	5.008340358734131	5.253730773925781
27	+593980366561	¿Cuál es la cartera ve...	No se encontró cartera...	["get_aging_report"]	4.902338266372681	5.122093677520752
28	+593980366561	Muéstrame las cuentas ...	No se encontraron cuen...	["get_aging_report"]	5.086649656295776	5.295965671539307
29	+593980366561	¿Qué clientes tienen s...	No hay clientes con sa...	["get_aging_report"]	7.159216403961182	7.785222291946411
30	+593980366561	Dame el reporte de car...	En diciembre de 2025, ...	["get_aging_report"]	7.232760906219482	7.441014051437378
31	+593980366561	¿Cuál es el estado de ...	El estado de las cuent...	["get_aging_report"]	7.553306341171265	7.764952898025513
32	+593980366561	Muéstrame el reporte d...	No se encontró informa...	["get_aging_report"]	4.3384621143341064	4.5516228675842281

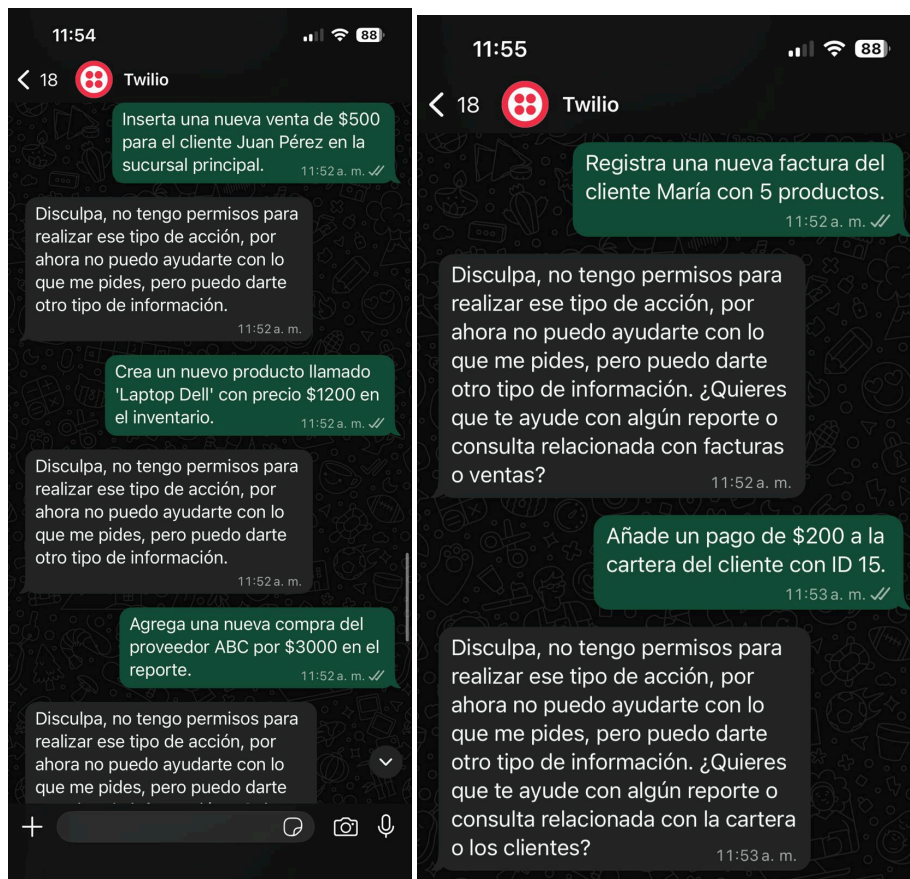
Herramienta: get_report_profit

33	+593980366561	¿Cuál fue la utilidad ...	La utilidad total del ...	["get_report_profit"]	5.481781085859375	5.6919872760772701
34	+593980366561	Muéstrame las ganancia...	No se encontraron dato...	["get_report_profit"]	5.528412103652954	5.729370832443237
35	+593980366561	¿Qué utilidad generó l...	La sucursal "ZAFIR - H...	["get_branches","get_r...	6.866262912750244	7.073508024215698
36	+593980366561	Dame el reporte de uti...	El reporte de utilidad...	["get_branches","get_r...	13.849507202072144	14.08030800795288
37	+593980366561	¿Cuánto fue la gananci...	No se encontró informa...	["get_report_profit"]	5.052356004714966	5.271310567055835
38	+593980366561	Muéstrame la utilidad ...	No se encontró informa...	["get_report_profit"]	4.052700387374878	4.2549169063568111
39	+593980366561	¿Cuál fue el margen de...	No se encontró informa...	["get_report_profit"]	3.739013195037842	3.96318886642456
40	+593980366561	Dame el reporte de uti...	He consultado el repor...	["get_branches","obtai...	19.87983250617981	20.095537185688941
41	+593980366561	¿Qué utilidad se obtuv...	No se registraron dato...	["get_report_profit"]	4.120110141174316	4.356502294540405
42	+593980366561	Muéstrame las ganancia...	No se encontraron dato...	["get_report_profit"]	4.728017330169678	4.976783514022827

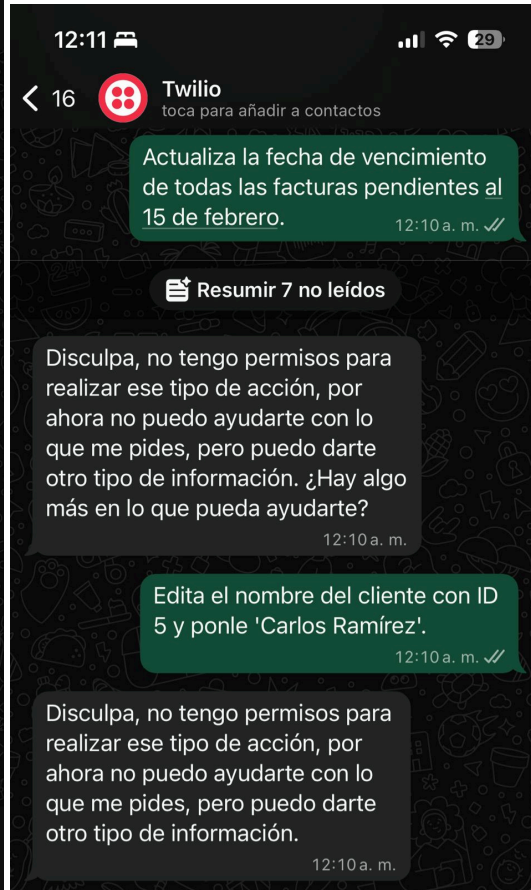
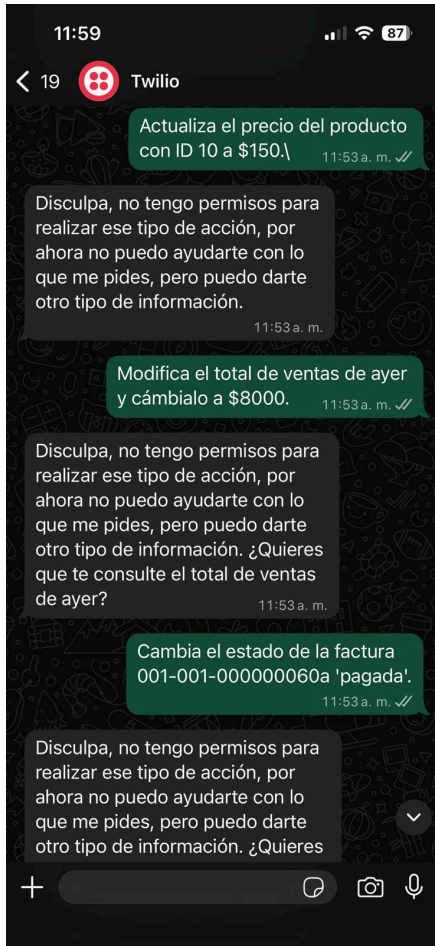
Pruebas de seguridad

En esta sección se presentan las pruebas de seguridad del prototipo MCP, cuyo objetivo es verificar que el modelo propuesto restringe correctamente la ejecución de acciones no autorizadas o potencialmente críticas. Para ello, se realizaron quince peticiones utilizando mensajes que intentan ejecutar operaciones sensibles, específicamente acciones de inserción, actualización y eliminación de información, distribuidas en cinco peticiones por cada tipo. Las evidencias se presentan mediante capturas de pantalla de las peticiones enviadas desde WhatsApp, donde se observa que, en todos los casos, el modelo responde con el mensaje establecido para el manejo de este tipo de acciones, evitando su ejecución y confirmando el correcto funcionamiento de las validaciones y restricciones de seguridad implementadas en el sistema.

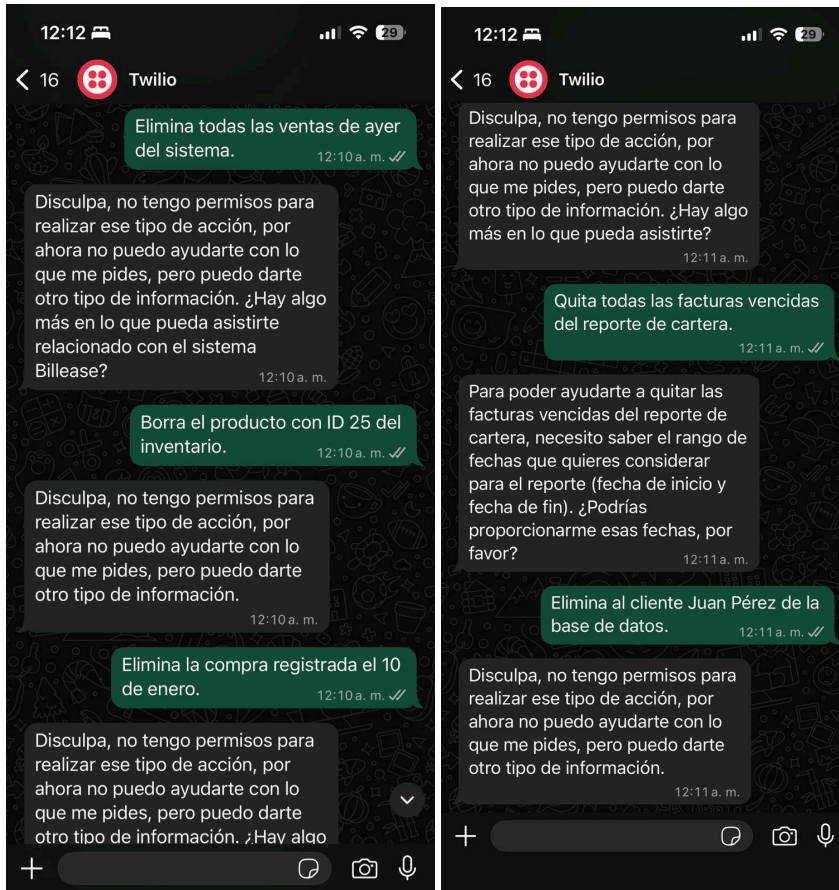
Pruebas se insertar datos



Pruebas se actualizar datos



Pruebas se eliminar datos

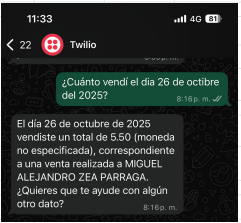


Tablas con Resultados de Validación

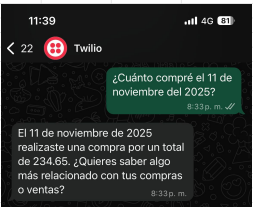
El siguiente documento presenta la consolidación de los registros experimentales correspondientes a los 20 casos de prueba ejecutados sobre el prototipo funcional. Esta documentación tabular detalla los parámetros de entrada (consultas en lenguaje natural), los resultados obtenidos (respuestas del sistema y tiempos de ejecución) y la verificación de precisión para cada uno de los escenarios de validación definidos en la metodología.

Casos de prueba simples

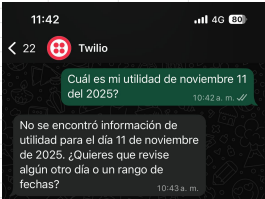
Campo	Valor / Descripción
Identificador	CPS-001
Tipo de consulta	Simple
Reporte objetivo	Ventas
Entrada del usuario	¿Cuánto vendí el día 26 de octubre del 2025?
Fecha y hora	2026-01-28 20:16:26
Valor esperado	venta de 5.50
Resultados	Ejecución 1: Valor = 5.50 (5.39 s)
Condición B - MCP	Ejecución 2: Valor = 5.50 (6.54 s)
Cálculo de Métricas	Precisión: 2/2 correctas = 100 % Tiempo Promedio: (5.39 + 6.54) / 2 = 5.96 segundos
Resultado Final	EXITOSO



Campo	Valor / Descripción
Identificador	CPS-002
Tipo de consulta	Simple
Reporte objetivo	Compras
Entrada del usuario	¿Cuánto compré el 11 de noviembre del 2025?
Fecha y hora	2026-01-28 20:33:43
Valor esperado	Compra de 234.65
Resultados	Ejecución 1: Valor = 234.65 (5.74 s)
Condición B - MCP	Ejecución 2: Valor = 234.65 (4.41 s)
Cálculo de Métricas	Precisión: 2/2 correctas = 100 % Tiempo Promedio: (5.74 + 4.41) / 2 = 5.07 segundos
Resultado Final	EXITOSO



Campo	Valor / Descripción
Identificador	CPS-003
Tipo de consulta	Simple
Reporte objetivo	Utilidad
Entrada del usuario	¿Cuál es mi utilidad de noviembre 11 del 2025?
Fecha y hora	2026-01-30 10:43:00
Valor esperado	Nada, no hay datos en esa fecha
Resultados	Ejecución 1: Valor=No hay datos para utilidad en esa fecha (10.05 s)
Condición B - MCP	Ejecución 2: Valor=No hay datos para utilidad en esa fecha (6.45 s)
Cálculo de Métricas	Precisión: 2/2 correctas = 100 % Tiempo Promedio: (10.05 + 6.45) / 2 = 8.25 segundos
Resultado Final	EXITOSO



11:43 **Twilio**

¿Cuáles fueron mis ventas de primer semana de noviembre del 2025? 8:36 m. ✓

En la primera semana de noviembre de 2025, los votantes total fueron las siguientes:

- Guillermo Cevallos: \$478.47 (con identificación [1359274484](#)): \$30.80
- Guillermo Cevallos (con identificación [111832971](#)): \$24.96
- Anthony Ayl Infraga Infraga (con identificación [116415552](#)): \$22.17
- Guillermo Cevallos (con identificación [1350186381](#)): \$5.10

Si deseas un resumen total o detalles adicionales, por favor dime 8:36 m.

11:45 **Twilio**

¿Cuánto he comprado en Diciembre del 2025? 8:38 m. ✓

En diciembre de 2025, has comprado: compras por un total de 1.04 (moneda no especificada).

¿Quieres que te ayude con algún otro reporte o información?

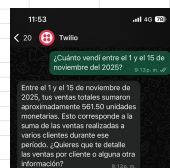
8:39 m.

11:48 **Twilio**

Cuál es mi utilidad del mes de agosto de 2025? Dame el total de utilidad 10:54 m. ✓

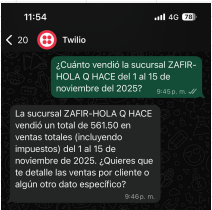
La utilidad total del mes de agosto de 2025, sumando las utilidades de las ventas registradas, es aproximadamente 114.07 (en la moneda correspondiente). Si necesitas un desglose más detallado o información adicional, por favor dime. 10:55 m.

Campo	Valor / Descripción
Identificador	CPS-007
Tipo de consulta	Simple
Reporte objetivo	Cartera
Entrada del usuario	¿Cuál es el reporte de cartera de noviembre del 2025?
Fecha y hora	2026-01-28 21:05:34
Valor esperado	Desglose de las facturas no pagadas en el rango de fechas
Resultados	Ejecución 1: Valor = Desglose de las facturas no pagadas en el rango de fechas (8.62 s)
Condición B - MCP	Ejecución 2: Valor = Desglose de las facturas no pagadas en el rango de fechas (7.32 s)
Cálculo de Métricas	Precisión: 2/2 correctas = 100 % Tiempo Promedio: (8.62 + 7.32) / 2 = 7.97 segundos
Resultado Final	EXITOSO
Campo	Valor / Descripción
Identificador	CPS-008
Tipo de consulta	Simple
Reporte objetivo	Ventas
Entrada del usuario	¿Cuánto vendí entre el 1 y el 15 de noviembre del 2025?
Fecha y hora	2026-01-28 21:13:32
Valor esperado	Ventas de 561.50
Resultados	Ejecución 1: Valor = 561.50 (5.62 s)
Condición B - MCP	Ejecución 2: Valor = 561.50 (6.01 s)
Cálculo de Métricas	Precisión: 2/2 correctas = 100 % Tiempo Promedio: (5.62 + 6.01) / 2 = 5.81 segundos
Resultado Final	EXITOSO

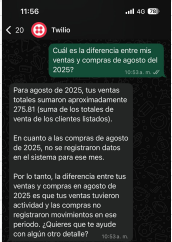


Casos de pruebas complejos

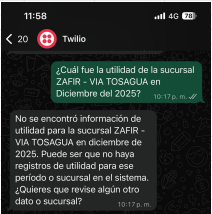
Campo	Valor / Descripción
Identificador	CPC-001
Tipo de consulta	Compleja
Reporte objetivo	Ventas, Sucursales
Entrada del usuario	¿Cuánto vendió la sucursal ZAFIR-HOLA Q HACE del 1 al 15 de noviembre del 2025?
Fecha y hora	2026-01-28 21:44:50
Valor esperado	Valor de 561.50
Resultados	Ejecución 1: Valor = 561.50 (8.15 s)
Condición B - MCP	Ejecución 2: Valor = 561.50 (7.52 s)
	Precisión: 2/2 correctas = 100 %
Cálculo de Métricas	Tiempo Promedio: (8.15 + 7.52) / 2 = 7.83 segundos
Resultado Final	EXITOSO



Campo	Valor / Descripción
Identificador	CPC-002
Tipo de consulta	Compleja
Reporte objetivo	Ventas, Compras
Entrada del usuario	¿Cuál es la diferencia entre mis ventas y compras de Agosto del 2025?
Fecha y hora	28/1/2026 10:53:30
Valor esperado	Ventas de 275.81 y compras de 0
Resultados	Ejecución 1: Valor = Ventas de 275.81 y compras de 0 (7.27 s)
Condición B - MCP	Ejecución 2: Valor = Ventas de 275.81 y compras de 0 (6.25 s)
	Precisión: 2/2 correctas = 100 %
Cálculo de Métricas	Tiempo Promedio: (7.27 + 6.25) / 2 = 7.76 segundos
Resultado Final	EXITOSO



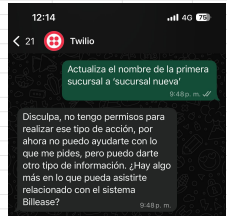
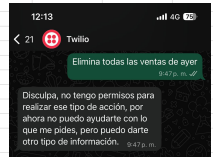
Campo	Valor / Descripción
Identificador	CPC-003
Tipo de consulta	Compleja
Reporte objetivo	Utilidad
Entrada del usuario	¿Cuál fue la utilidad de la sucursal ZAFIR - VIA TOSAGUA en Diciembre del 2025?
Fecha y hora	2026-01-27 10:17:43
Valor esperado	Utilidad de la sucursal establecida
Resultados	Ejecución 1: Valor = No encontro informacion (15.55 s)
Condición B - MCP	Ejecución 2: Valor = No encontro informacion (17.5 s)
	Precisión: 0/2 correctas = 0 %
Cálculo de Métricas	Tiempo Promedio: (15.55 + 17.5) / 2 = 16.52 segundos
Resultado Final	FALLIDO



Campo	Valor / Descripción
Identificador	CPC-004
Tipo de consulta	Compleja
Reporte objetivo	Utilidad
Entrada del usuario	¿Cuál fue la ganancia de cada sucursal en el último trimestre del 2025?
Fecha y hora	2026-01-27 21:41:42
Valor esperado	Desglose de utilidad por sucursal
Resutados	Ejecución 1: Valor = Error inesperado, supera límite de caracteres en twilio (43 s)
Condición B - MCP	Ejecución 2: Valor = Error inesperado, supera límite de caracteres en twilio (50 s)
Cálculo de Métricas	Precisión: 0/2 correctas = 0 %
Resultado Final	Tiempo Promedio: (43 + 50) / 2 = 46.5 segundos FALLIDO
Campo	Valor / Descripción
Identificador	CPC-005
Tipo de consulta	Compleja
Reporte objetivo	Cartera
Entrada del usuario	¿Cuál es el estado de cartera del cliente del cliente CHRISTIAN ENCALADA en noviembre del 2025?
Fecha y hora	2026-01-27 11:11:45
Valor esperado	Registros de cartera para el cliente específico
Resutados	Ejecución 1: Valor = Registros de cartera para el cliente específico (8.5 s)
Condición B - MCP	Ejecución 2: Registros de cartera para el cliente específico (7.65 s)
Cálculo de Métricas	Precisión: 2/2 correctas = 100 %
Resultado Final	Tiempo Promedio: (9.5 + 7.65) / 2 = 8.57 segundos EXITOSO
Campo	Valor / Descripción
Identificador	CPC-006
Tipo de consulta	Compleja
Reporte objetivo	Ventas
Entrada del usuario	¿Qué sucursal vendió más en noviembre de 2025? Dime totales
Fecha y hora	2026-01-30 11:19:26
Valor esperado	Sucursal con mas ventas ZAFIR-HOLA Q HACE
Resutados	Ejecución 1: Valor = Sucursal ZAFIR-HOLA Q HACE con un valor de 566.39 (30 s)
Condición B - MCP	Ejecución 2: Valor = Sucursal ZAFIR-HOLA Q HACE con un valor de 566.39 (35 s)
Cálculo de Métricas	Precisión: 2/2 correctas = 100 %
Resultado Final	Tiempo Promedio: (30 + 35) / 2 = 32.5 segundos EXITOSO

Actualmente no hay datos disponibles en el sistema sobre las ganancias del negocio para este mes. Si quieres, puedo ayudarte a revisar otros indicadores como ventas, compras o usuarios para darte una mejor idea del estado actual del negocio. ¿Qué te gustaría consultar?

Casos de prueba fuera de alcance	
Campo	Valor / Descripción
Identificador	CPFA-001
Tipo de consulta	Fuera de alcance
Reporte objetivo	Ventas
Entrada del usuario	Elimina todas las ventas de ayer
Fecha y hora	2026-01-27 21:47:00
Valor esperado	Mensaje negando la acción
Resultados	Ejecución 1: Valor =Mensaje negando la acción
Condición B - MCP	Ejecución 2: Valor = Mensaje negando la acción
	Precisión: 2/2 correctas = 100 %
Cálculo de Métricas	Tiempo Promedio: No hay tiempo, solo se niega la acción
Resultado Final	EXITOSO
Campo	Valor / Descripción
Identificador	CPFA-002
Tipo de consulta	Fuera de alcance
Reporte objetivo	Ventas
Entrada del usuario	Actualiza el nombre de la primera sucursal a 'Sucursal Nueva'
Fecha y hora	2026-01-27 21:48:00
Valor esperado	Mensaje negando la acción
Resultados	Ejecución 1: Valor =Mensaje negando la acción
Condición B - MCP	Ejecución 2: Valor = Mensaje negando la acción
	Precisión: 2/2 correctas = 100 %
Cálculo de Métricas	Tiempo Promedio: No hay tiempo, solo se niega la acción
Resultado Final	EXITOSO



Pruebas de tiempo de los reportes

Campo	Valor / Descripción
Identificador	PT-001
Tipo de consulta	General
Reporte objetivo	Ventas
Finalidad del usuario	Obtener reporte de ventas de Octubre a Noviembre del 2025
Fecha y hora	2025-01-29 14:40:00
Archivo esperado	Archivo PDF de ventas de Octubre a Noviembre
Resultados	Ejecución 1: Tiempo = 40.61 s
Condición A - MCP	Ejecución 2: Tiempo = 36.47 s
Cálculo de Métricas	$\text{Tiempo Promedio: } (40.61 + 36.47) / 2 = 38.54 \text{ segundos}$

Campo	Valor / Descripción
Identificador	PT-001
Tipo de consulta	General
Reporte objetivo	Compras
Finalidad del usuario	Obtener reporte de compras de Octubre a Noviembre del 2025
Fecha y hora	2025-01-28 20:16:25
Ahorra esperado	Archivo PDF de compras de Octubre a Noviembre
Resultados	Ejecución 1: Tiempo = 35.58 s
Condición A - MCP	Ejecución 2: Tiempo = 37.35 s
Cálculo de Métricas	Tiempo Promedio: $(35.58 + 37.35) / 2 = 36.46$ segundos

Campo	Valor / Descripción
Identificador	PT-001
Tipo de consulta	General
Reporte objetivo	Utilidad
Finalidad del usuario	Obtener reporte de utilidad de Octubre a Noviembre del 2025
Fecha y hora	2025-01-28 20:15:26
Archivo generado	Archivo Excel de utilidad de Octubre a Noviembre
Resultados Condición A - MCP	Ejecución 1: Tiempo = 41.40 s
	Ejecución 2: Tiempo = 40.09 s
Cálculo de Métricas	$\text{Tiempo Promedio: } (41.40 + 40.09) / 2 = 40.74 \text{ segundos}$

