



**UNIVERSIDAD TÉCNICA DEL NORTE**  
**FACULTAD DE INGENIERÍA EN CIENCIAS**  
**APLICADAS**  
**CARRERA DE TECNOLOGÍAS DE LA**  
**INFORMACIÓN**

**TRABAJO DE INTEGRACIÓN CURRICULAR**

**TEMA:**

” DESARROLLO DE UN CHATBOT PARA BANCA ELECTRÓNICA UTILIZANDO  
INTELIGENCIA ARTIFICIAL GENERATIVA”

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

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

**AUTOR:**

Jonathan Wilson Pilatuña Villamarín

**DIRECTOR:**

MSc. Diego Javier Trejo España

**Ibarra – Ecuador 2025**



# UNIVERSIDAD TÉCNICA DEL NORTE

## BIBLIOTECA UNIVERSITARIA

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

#### 1. IDENTIFICACIÓN DE LA OBRA

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

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	1004562508		
APELLIDOS Y NOMBRES:	Pilatuña Villamarín Jonathan Wilson		
DIRECCIÓN:	Cotacachi, Calle Segundo Luis Moreno y Pasaje Lomas del Sur		
EMAIL:	jwpilatunav@utn.edu.ec		
TELÉFONO FIJO:		TELÉFONO MÓVIL:	0998560409

DATOS DE LA OBRA	
TÍTULO:	Desarrollo de un chatbot para banca electrónica utilizando inteligencia artificial generativa
AUTOR (ES):	Pilatuña Villamarín Jonathan Wilson
FECHA: DD/MM/AAAA	04/09/2025
SOLO PARA TRABAJOS DE GRADO	
PROGRAMA:	<input checked="" type="checkbox"/> PREGRADO <input type="checkbox"/> POSGRADO
TÍTULO POR EL QUE OPTA:	Ingeniero en Tecnologías de la Información
ASESOR /DIRECTOR:	PhD. Antonio Quiña/MSc. Diego Trejo

#### 2. CONSTANCIAS

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

Ibarra, a los 04 días del mes de septiembre de 2025

#### EL AUTOR:

Pilatuña Villamarín Jonathan Wilson

**CERTIFICACIÓN DEL DIRECTOR DEL TRABAJO DE  
INTEGRACIÓN CURRICULAR**

04/09/2025

Diego Javier Trejo España

**DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR**

CERTIFICA:

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



---

MSc. Diego Trejo

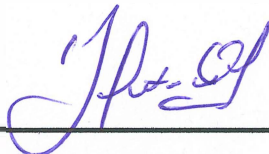
## APROBACIÓN DEL COMITÉ CALIFICADOR

El Comité Calificado del trabajo de Integración Curricular “DESARROLLO DE UN CHATBOT PARA BANCA ELECTRÓNICA UTILIZANDO INTELIGENCIA ARTIFICIAL GENERATIVA” elaborado por Pilatuña Villamarín Jonathan Wilson, previo a la obtención del título del INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN, aprueba el presente informe de investigación en nombre de la Universidad Técnica del Norte:



---

MSc. Diego Trejo



---

PhD. Antonio Quiña Mera

## **DEDICATORIA**

Este trabajo de tesis, se lo dedico a mi esposa Majito y a mis hijos Skarlett e Isaac, quienes estuvieron conmigo en cada momento de este largo proceso, brindándome su apoyo incondicional y sobre todo confiando en lo que yo puedo hacer y conseguir.

Jonathan Wilson Pilatuña Villamarín

## **AGRADECIMIENTO**

Agradezco a Dios por la vida y sabiduría para poder culminar con éxito mis proyectos. A mi familia por darme la fortaleza en momentos de angustia.

A la Industria Tecnología MTC por la oportunidad de realizar este proyecto en su infraestructura tecnológica.

A mi director MSc. Diego Trejo y asesor PhD. Antonio Quiña Mera, por compartir sus conocimientos y el acompañamiento constante durante el desarrollo de este trabajo.

Jonathan Wilson Pilatuña Villamarín

## ÍNDICE DE CONTENIDOS

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UTN.....	II
CERTIFICACIÓN DEL DIRECTOR DEL TRABAJO DE .....	III
INTEGRACIÓN CURRICULAR .....	III
APROBACIÓN DEL COMITÉ CALIFICADOR.....	IV
DEDICATORIA .....	V
AGRADECIMIENTO .....	VI
RESUMEN .....	XIII
ABSTRACT.....	XIV
CAPITULO I .....	1
INTRODUCCIÓN .....	1
1.1 Problema de investigación .....	1
1.2 Objetivos .....	3
1.2.1. Objetivo general .....	3
1.2.2. Objetivos específicos.....	3
1.3 Formulación del problema de investigación .....	3
1.4 Justificación.....	4
1.4.1 Importancia y trascendencia .....	4
1.4.2 Beneficiarios.....	5
1.4.3 Factibilidad.....	6
CAPÍTULO II.....	7
MARCO TEÓRICO.....	7
2.1 Antecedentes .....	7
2.1.1 Evolución de los Canales Electrónicos .....	7
2.1.2 Evolución de los Chatbots.....	8

2.1.3 Chatbots en el sector bancario.....	10
2.2 Bases Teóricas.....	10
2.2.1 Marco Legal Regulatorio en el Ecuador Respecto a los TRX Financieros.....	10
2.2.2 Definición y funcionamiento del chatbot.....	13
2.2.3 Definición y funcionamiento de aplicaciones de mensajería.....	14
2.2.4 Procesamiento de lenguaje natural (NPL).....	16
2.2.5 Modelos de IA aplicados en NPL.....	18
2.2.6 Seguridad en chatbots para banca electrónica.....	21
2.2.7 Modelos de arquitectura de software.....	22
2.2.8 Framework Backend NestJS para Aplicaciones con PLN/IA.....	24
2.2.9 Plataformas APM (Application performonce managment).....	26
2.2.10 Metodologías de desarrollo de software.....	29
CAPÍTULO III.....	33
MATERIALES Y MÉTODOS.....	33
3.1 Metodología.....	33
3.2 Levantamiento de requisitos.....	33
3.2.1 Estimación del esfuerzo en historias de usuario.....	36
3.2.4 Product Backlog.....	45
3.3 Desarrollo.....	46
3.3.1 Sprint 0.....	46
3.3.1 Sprint 1.....	48
3.3.1 Sprint 2.....	52
3.3.1 Sprint 3.....	58
3.3.1 Sprint 4.....	62
3.4 Infraestructura tecnológica y herramientas utilizadas.....	65

3.5 Justificación de Funcionalidades No Implementadas .....	66
CAPITULO IV.....	67
RESULTADOS Y ANÁLISIS .....	67
4.1 Evaluación del Rendimiento y Fiabilidad del Bot Bancario.....	67
4.1.1 Métricas de Rendimiento y Latencia .....	67
4.1.2 Fiabilidad de Operaciones .....	69
4.1.3 Consumo y Gestión de Recursos .....	69
4.1.4 Distribución del Tiempo de Procesamiento por Componente.....	71
4.1.5 Capacidad de Throughput.....	72
4.2 Análisis de Resultados .....	72
CONCLUSIONES .....	74
RECOMENDACIONES .....	75
REFERENCIAS BIBLIOGRÁFICAS .....	76
ANEXOS.....	81

## ÍNDICE DE FIGURAS

<b>Fig. 1.</b> Arquitectura Inicial del Chatbot .....	4
<b>Fig. 2.</b> Arquitectura del Final del Chatbot .....	48
<b>Fig. 3.</b> Proceso de autenticación .....	50
<b>Fig. 4.</b> Diagrama de secuencia del Proceso de autenticación .....	50
<b>Fig. 5.</b> Términos y condiciones.....	51
<b>Fig. 6.</b> Diagrama de secuencia del Proceso de Términos y condiciones .....	51
<b>Fig. 7.</b> Consulta de saldos .....	54
<b>Fig. 8.</b> Diagrama de secuencia del proceso de Consulta de saldos .....	54
<b>Fig. 9.</b> Consulta - depósitos a plazo fijo .....	55
<b>Fig. 10.</b> Diagrama de secuencia del Proceso de Consulta - depósitos a plazo fijo .....	55
<b>Fig. 11.</b> Consulta – últimos movimientos .....	56
<b>Fig. 12.</b> Diagrama de secuencia del Proceso de Consulta – Últimos Movimientos .....	56
<b>Fig. 13.</b> Consulta – créditos .....	57
<b>Fig. 14.</b> Diagrama de secuencia del Proceso de Consulta – créditos.....	57
<b>Fig. 15.</b> Transferencias internas.....	59
<b>Fig. 16.</b> Diagrama de secuencia del Proceso de Trasterferencia interna .....	60
<b>Fig. 17.</b> Transferencia interbancaria .....	60
<b>Fig. 18.</b> Diagrama de secuencia del Proceso de Trasterferencia interbancaria.....	61
<b>Fig. 19.</b> Pago de tarjeta de crédito .....	63
<b>Fig. 20.</b> Diagrama de secuencia del Proceso de Pago de tarjeta de crédito .....	63
<b>Fig. 21.</b> Pago de servicios básicos .....	64
<b>Fig. 22.</b> Diagrama de secuencia del Proceso de Pago de servicios básicos.....	64
<b>Fig. 23.</b> Tiempo Promedio de Transacción.....	67
<b>Fig. 24.</b> Tiempo de Respuesta Web .....	68
<b>Fig. 25.</b> Latencia del Event Loop.....	68
<b>Fig. 26.</b> Resumen de Transacciones .....	69
<b>Fig. 27.</b> Utilización Promedio de CPU .....	70
<b>Fig. 28.</b> Memoria Heap .....	70
<b>Fig. 29.</b> Recolección de Basura .....	71
<b>Fig. 30.</b> Distribución de Segmentos de Transacción .....	71

<b>Fig. 31.</b> Throughput Web.....	72
<b>Fig. 32.</b> Configuración del webhook en el entorno de desarrollo.....	81
<b>Fig. 33.</b> Panel de configuración de la API de WhatsApp en Meta Developer .....	81
<b>Fig. 34.</b> Interfaz de la plataforma OpenAI para la administración de claves de API .....	82
<b>Fig. 35.</b> Panel de control de New Relic .....	82

## ÍNDICE DE TABLAS

<b>Tabla 1</b> Requisitos Funcionales .....	34
<b>Tabla 2</b> Escala basada en una progresión adaptada de Fibonacci.....	36
<b>Tabla 3</b> Participantes Involucrados .....	37
<b>Tabla 4</b> Historia de Usuario RQBI-001 .....	38
<b>Tabla 5</b> Historia de Usuario RQAU-002.....	39
<b>Tabla 6</b> Historia de Usuario RQAU-003.....	39
<b>Tabla 7</b> Historia de Usuario RQTR-001 .....	40
<b>Tabla 8</b> Historia de Usuario RQTR-002 .....	40
<b>Tabla 9</b> Historia de Usuario RQTR-003 .....	41
<b>Tabla 10</b> Historia de Usuario RQTR-004 .....	41
<b>Tabla 11</b> Historia de Usuario RQTR-005 .....	42
<b>Tabla 12</b> Historia de Usuario RQTR-006 .....	42
<b>Tabla 13</b> Historia de Usuario RQTR-007 .....	43
<b>Tabla 14</b> Historia de Usuario RQTR-008 .....	43
<b>Tabla 15</b> Historia de Usuario RQTR-009 .....	44
<b>Tabla 16</b> Historia de Usuario RQTR-010 .....	44
<b>Tabla 17</b> Product Backlog.....	45
<b>Tabla 18</b> Sprint 0.....	46
<b>Tabla 19</b> Sprint 1.....	48
<b>Tabla 20</b> Sprint 2.....	52
<b>Tabla 21</b> Sprint 3.....	58
<b>Tabla 22</b> Sprint 4.....	62
<b>Tabla 23</b> Infraestructura tecnológica y herramientas utilizadas.....	65

## RESUMEN

La presente investigación aborda el diseño y desarrollo de un chatbot bancario conversacional que integra inteligencia artificial generativa (IAG) y procesamiento de lenguaje natural (PLN) directamente en la plataforma WhatsApp Business API.

El propósito central de esta solución es optimizar la experiencia del usuario y la eficiencia operativa de los servicios de la banca electrónica, permitiendo interacciones fluidas y transacciones financieras seguras desde la aplicación de mensajería instantánea.

El proyecto se implementó y validó en un entorno de desarrollo controlado, utilizando un usuario y datos de prueba proporcionados por la Industria Tecnológica MTC. La arquitectura del sistema se fundamentó en tecnologías clave como el framework NestJS, un Modelo de Lenguaje Grande (LLM GPT) vía OpenAI API para las capacidades de PLN, y una API financiera para la gestión transaccional.

La metodología Scrum guio el ciclo de desarrollo, facilitando una gestión ágil de los requisitos funcionales, la formulación de historias de usuario y la ejecución de sprints planificados.

La validación de la aplicación se llevó a cabo utilizando la herramienta APM New Relic, monitoreando el rendimiento y la fiabilidad operativa. Los resultados evidenciaron una eficiencia sobresaliente que registró un tiempo promedio de transacción de 7.63 milisegundos y una tasa de éxito del 100% en 675 transacciones.

Además, la solución demostró su robustez al manejar hasta 59 transacciones por minuto. Estos hallazgos confirman la capacidad del chatbot para ofrecer un servicio bancario digital rápido, confiable y accesible, marcando un avance significativo en la modernización de la interacción entre usuarios y entidades financieras.

**Palabras clave:** Bot bancario, NestJS, Inteligencia Artificial Generativa, Procesamiento de Lenguaje Natural, WhatsApp Business API.

## ABSTRACT

This research addresses the design and development of a conversational banking chatbot that integrates generative artificial intelligence (GAI) and natural language processing (NLP) directly into the WhatsApp Business API platform.

The primary purpose of this solution is to optimize the user experience and operational efficiency of electronic banking services, enabling fluid interactions and secure financial transactions from the instant messaging application.

The project was implemented and validated in a controlled development environment, using user and test data provided by the MTC Technology Industry. The system architecture was based on key technologies such as the NestJS framework, a Large Language Model (LLM GPT) via OpenAI API for NLP capabilities, and a financial API for transactional management.

The Scrum methodology guided the development cycle, facilitating agile management of functional requirements, the formulation of user stories, and the execution of planned sprints.

The application was validated using the New Relic APM tool, monitoring performance and operational reliability. The results showed outstanding efficiency, with an average transaction time of 7.63 milliseconds and a 100% success rate in 675 transactions.

In addition, the solution demonstrated its robustness by handling up to 59 transactions per minute. These findings confirm the chatbot's ability to deliver fast, reliable, and accessible digital banking services, marking a significant advance in the modernization of interaction between users and financial institutions.

**Keywords:** Banking bot, NestJS, Generative Artificial Intelligence, Natural Language Processing, WhatsApp Business API.

# CAPITULO I

## INTRODUCCIÓN

### 1.1 Problema de investigación

La Industria Tecnológica MTC CIA. LTDA, es una empresa que se especializa en el desarrollo de software personalizado para entidades financieras, uno de sus enfoques es el mejorar la accesibilidad a los servicios bancarios mediante canales de banca electrónica, por lo que las instituciones financieras se benefician de los productos desarrollados por esta industria, formando parte de los usuarios finales.

El impacto de la digitalización en las instituciones financieras ecuatorianas ha transformado profundamente sus operaciones, permitiendo reducir costos, optimizar procesos internos y mejorar la experiencia del usuario, esta revolución tecnológica ha posicionado a las instituciones financieras ecuatorianas en un lugar competitivo dentro del mercado global, además de promover la inclusión financiera, especialmente en áreas rurales con acceso limitado a servicios bancarios (Araujo et al., 2024).

La banca electrónica se ha consolidado como una herramienta esencial para realizar transacciones, consultas y otros servicios de manera eficiente, evitando largas filas y pérdida de tiempo. Estas plataformas ofrecen una asistencia más ágil y oportuna, contribuyendo a la satisfacción del cliente. Sin embargo, la implementación inicial de herramientas digitales no basta; es necesario actualizarlas continuamente para garantizar una experiencia más amigable, segura y confiable.

Como señala Petersson et al., factores como la facilidad de uso y la capacidad de los sistemas para manejar errores de comunicación son clave para la percepción y adopción por parte de los clientes (Petersson et al., 2023). Además, Kitsios et al. subraya que la transformación digital exige un enfoque estratégico que combine educación a los usuarios y gestión cultural dentro de las organizaciones (Kitsios et al., 2021).

A pesar de sus beneficios, la banca electrónica en Ecuador enfrenta desafíos relacionados con accesibilidad y eficiencia. Procedimientos complejos para pagos, transferencias o consultas generan frustración en los usuarios, limitando su confianza en las plataformas. En este contexto,

la inteligencia artificial (IA) surge como una solución transformadora. Priya y Sharma destacan que los asistentes virtuales inteligentes habilitados por IA facilitan interacciones naturales y dinámicas, mejorando significativamente la experiencia del cliente (Priya & Sharma, 2023).

La integración de IA con chatbots representa una oportunidad única para transformar la banca electrónica en Ecuador. Según Gupta et al., modelos generativos como ChatGPT pueden redefinir la interacción cliente-banco, ofreciendo personalización y eficiencia en la resolución de consultas (Gupta et al., 2024). Estos sistemas no solo responden preguntas frecuentes, sino que también gestionan tareas complejas como solicitudes de préstamos o asesoramiento financiero.

Además, los chatbots avanzados son capaces de interpretar el lenguaje natural y responder en tiempo real, fortaleciendo la relación con los clientes. Como menciona Petersson et al., la percepción de humanización en los sistemas digitales puede marcar la diferencia entre una experiencia exitosa y una frustrante (Petersson et al., 2023). Esto permite a los bancos optimizar operaciones y ofrecer un servicio más humano y accesible.

La implementación de un chatbot para banca electrónica con inteligencia artificial generativa es una necesidad urgente. Este sistema debe garantizar un servicio personalizado, simplificar procesos complejos y proporcionar asistencia segura y eficiente. Esto no solo mejorará la experiencia del cliente, sino que también fortalecerá la competitividad del sector bancario ecuatoriano en el entorno digital global.

Finalmente, como señalan Baldassarre et al., las iniciativas tecnológicas deben equilibrar la adopción de soluciones digitales con la necesidad de mantener la confianza y acceso para los clientes más tradicionales (Baldassarre et al., 2024). El éxito radica en conectar tecnología y personas, transformando la banca en un servicio verdaderamente accesible y humano.

## 1.2 Objetivos

### 1.2.1. Objetivo general

Desarrollar un chatbot para banca electrónica utilizando Inteligencia Artificial generativa.

### 1.2.2. Objetivos específicos

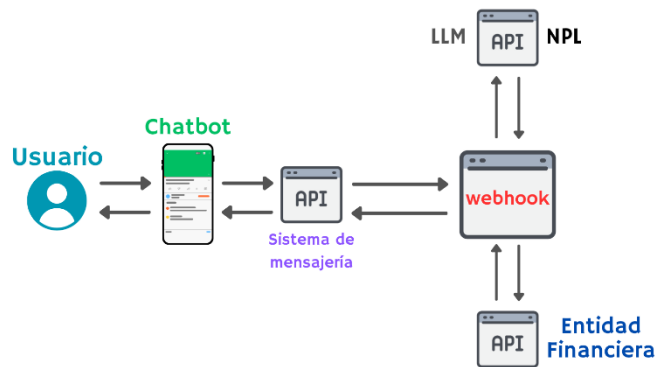
- Generar una base teórica sobre soluciones técnicas y metodológicas para la implementación de un chatbot en la banca electrónica.
- Realizar un diagnóstico de la situación actual para comprender cual es la problemática y diseñar la propuesta que responda a las necesidades de los usuarios.
- Desarrollar un chatbot para banca electrónica con Inteligencia Artificial generativa para automatizar la atención y los trámites financieros mediante un sistema de mensajería.
- Evaluar el chatbot con herramientas de medición de desempeño en un entorno de prueba.

## 1.3 Formulación del problema de investigación

- Desarrollo de un orquestador de chatbot basado en webhook que integra diversas APIs para interactuar con servicios bancarios.
- El usuario por medio del chatbot será capaz de realizar trámites financieros tales como:
  - **Consultas de Saldos:**
    - Cuentas de ahorros
    - Cuentas de certificados de aportación
    - Cuentas de ahorro programado
    - Cuentas de depósitos a plazo fijo
    - Saldos, estado de morosidad y cuotas de créditos
  - **Transferencias:**
    - Internas (entre cuentas del mismo banco)
    - Interbancarias (entre distintos bancos)
  - **Pagos:**
    - Pago de tarjetas de crédito
    - Pago de servicios (agua, luz, teléfono, internet, TV)

- **Consultas de movimientos de cuenta**
- Contará con los esquemas de seguridad proporcionados por la entidad financiera y seguridad a nivel de comunicación entre componentes.
- Estará disponible para el equipo de Stakeholders de la Industria Tecnológica MTC.
- Interacción con herramientas de medición de desempeño.

En la Fig. 1 se puede visualizar la Arquitectura inicial del chatbot.



**Fig. 1.** *Arquitectura Inicial del Chatbot*

## 1.4 Justificación

El desarrollo de un chatbot para banca electrónica utilizando inteligencia artificial generativa es una respuesta concreta a los desafíos actuales del sector financiero. En un entorno donde la digitalización se ha convertido en un pilar clave, este proyecto busca mejorar la accesibilidad y la eficiencia en los servicios bancarios, a la vez que garantiza altos estándares de seguridad y personalización en las interacciones con los usuarios.

### 1.4.1 Importancia y trascendencia

A nivel local y regional, el avance de la digitalización en el sector bancario ha demostrado ser una herramienta eficaz para reducir costos operativos, agilizar procesos internos y ofrecer mejores servicios a los clientes. Sin embargo, muchos usuarios aún enfrentan dificultades en el uso de plataformas electrónicas, debido a la complejidad de las interfaces o a la falta de personalización en los servicios. Incorporar un chatbot con capacidades de inteligencia artificial

generativa no solo resolverá estas barreras, sino que también potenciará la experiencia del cliente, permitiéndole realizar consultas y transacciones de forma rápida, sencilla y segura.

Los chatbots financieros, al ser sistemas diseñados para interactuar con los usuarios de manera similar a la comunicación humana, han demostrado un impacto positivo en la satisfacción y lealtad del cliente, generando así un valor comercial significativo para las instituciones financieras (Li et al., 2023). Este proyecto no solo se alinea con las tendencias globales en transformación digital, sino que también refuerza la capacidad de las entidades locales para competir en un mercado cada vez más dinámico y exigente.

#### 1.4.2 Beneficiarios

- **Beneficiarios directos**
  - **Usuarios bancarios:** Tendrán acceso a una herramienta accesible y eficiente para realizar consultas, transferencias y pagos sin necesidad de acudir a una sucursal.
  - **Entidades financieras:** Se beneficiarán de la reducción de costos operativos, una atención automatizada al cliente y una mejora en la fidelización de los usuarios.
  - **Industria Tecnológica MTC:** Al liderar este proyecto, consolidará su posición como un referente en el desarrollo de soluciones tecnológicas avanzadas para el sector financiero.
- **Beneficiarios indirectos**
  - **Sociedad ecuatoriana:** Este proyecto fomentará la inclusión financiera, facilitando el acceso a servicios bancarios para personas que enfrentan barreras geográficas o tecnológicas.
  - **Comunidad científica y tecnológica:** La implementación de inteligencia artificial generativa en un entorno financiero abrirá nuevas líneas de investigación y desarrollo, sentando precedentes para futuras aplicaciones tecnológicas (Casheekar et al., 2024).
  - **Estudiantes de TI:** El desarrollo de esta investigación marca una base sólida en futuras investigaciones direccionadas a soluciones tecnológicas dentro del ámbito bancario.

### **1.4.3 Factibilidad**

La implementación de este proyecto es viable gracias a los recursos humanos, tecnológicos y financieros disponibles. La Industria Tecnológica MTC, con su experiencia en el desarrollo de soluciones bancarias, cuenta con el equipo necesario para diseñar, desarrollar y evaluar el chatbot. Además, el uso de tecnologías maduras y accesibles, como modelos de lenguaje generativo y plataformas de mensajería, garantiza la factibilidad técnica del proyecto.

## CAPÍTULO II

### MARCO TEÓRICO

#### 2.1 Antecedentes

##### 2.1.1 Evolución de los Canales Electrónicos

En las últimas décadas, los avances tecnológicos han transformado profundamente la interacción entre las instituciones bancarias y sus clientes, marcando el inicio de un modelo híbrido conocido como "Click-and-Mortar", este modelo combina canales digitales y tradicionales para satisfacer las demandas de los usuarios, optimizar la eficiencia operativa y garantizar la inclusión financiera (Baldassarre et al., 2024; Rodríguez et al., 2022). En Ecuador, este proceso de transformación ha sido particularmente significativo, impulsado por la creciente digitalización y los cambios en el comportamiento de los consumidores, lo que se refleja en el incremento del uso de canales electrónicos para transacciones financieras (Rodríguez et al., 2022).

El uso de canales electrónicos para transacciones bancarias ha mostrado un crecimiento exponencial en el país. En 2021, el número total de transacciones realizadas en el sistema bancario ecuatoriano alcanzó los 695 millones, de las cuales el 41,1 % se efectuaron mediante canales digitales (Rodríguez et al., 2022). Dentro de estos, las aplicaciones móviles se destacaron como la principal herramienta, representando el 24,9 % del total de transacciones, un aumento considerable respecto al 4,6 % registrado en 2019 (Rodríguez et al., 2022). Este incremento refleja tanto la adaptación de los usuarios a las plataformas digitales como los esfuerzos de las instituciones financieras para promover estos canales.

A pesar del auge de los canales digitales, los canales físicos, como las ventanillas y corresponsales no bancarios (CNB), continúan siendo esenciales para muchos usuarios, especialmente en áreas rurales y entre sectores menos digitalizados. En 2021, los canales físicos representaron el 37,1 % de las transacciones, con un monto transaccionado de USD 118.561 millones. Los CNB, en particular, desempeñan un papel crucial en la provisión de servicios básicos, registrando un incremento del 32,1 % en transacciones entre 2019 y 2021 (Rodríguez et al., 2022).

Por su parte, los cajeros automáticos (ATM) también se mantienen relevantes, representando el 21,8 % de las transacciones. Sin embargo, su uso ha disminuido en comparación con años anteriores debido a la migración de los usuarios hacia canales más convenientes, como aplicaciones móviles e internet, que permiten realizar operaciones desde cualquier lugar (Rodríguez et al., 2022).

En términos de inclusión financiera, la digitalización ha facilitado el acceso a servicios bancarios, particularmente para las mujeres y las generaciones más jóvenes. Los millennials lideraron el uso de canales digitales en 2021, con 301 millones de transacciones, lo que representa un aumento del 59,1 % respecto a 2019. No obstante, desafíos como la brecha digital y la preferencia por el uso de efectivo, que prevalece en un 36,5 % de la población ecuatoriana, subrayan la necesidad de estrategias para garantizar una transición inclusiva hacia una economía digital (Rodríguez et al., 2022).

De esta manera, la evolución de los canales electrónicos en Ecuador refleja un equilibrio entre la adopción de nuevas tecnologías y la preservación de los canales tradicionales. Si bien la digitalización ha optimizado la eficiencia operativa y mejorado la experiencia del cliente, los bancos continúan adaptando su infraestructura para atender las necesidades de los diferentes segmentos de la población, contribuyendo así al desarrollo de un sistema financiero más accesible y eficiente.

### **2.1.2 Evolución de los Chatbots**

Los chatbots han atravesado un notable proceso de evolución tecnológica desde sus primeras implementaciones hasta convertirse en asistentes inteligentes que transforman la interacción humano-computadora. Este desarrollo ha sido impulsado por avances en inteligencia artificial, procesamiento del lenguaje natural (NLP) y aprendizaje automático, lo que les permite realizar tareas más complejas y adaptarse a diferentes contextos y necesidades.

- **Primera Generación: Sistemas Basados en Patrones**

El desarrollo de ELIZA en 1966 marcó el inicio de los chatbots como programas que simulaban conversaciones humanas utilizando patrones predefinidos. Diseñado para replicar a un psicoterapeuta, ELIZA analizaba las entradas del usuario y las transformaba en preguntas o afirmaciones sin entender realmente el contenido. Aunque rudimentario, estableció las bases para los chatbots modernos (Adamopoulou & Moussiades, 2020).

En 1972, PARRY avanzó un paso más al integrar "personalidad" en las interacciones, simulando un paciente con esquizofrenia. Este chatbot introdujo respuestas emocionales básicas y un sistema de control que ajustaba su comportamiento según el contexto de la conversación. Sin embargo, tanto ELIZA como PARRY estaban limitados por la falta de capacidad de aprendizaje y comprensión de temas fuera de sus dominios predefinidos (Adamopoulou & Moussiades, 2020).

- **Segunda Generación: Uso de Inteligencia Artificial Básica**

Con la llegada de Jabberwacky en 1988, los chatbots comenzaron a incorporar inteligencia artificial para mejorar la experiencia del usuario. Este sistema utilizaba respuestas basadas en conversaciones previas, marcando el inicio de una transición hacia chatbots más dinámicos. En 1995, ALICE introdujo el lenguaje AIML (Artificial Intelligence Markup Language), lo que permitió una mayor flexibilidad y escalabilidad en el desarrollo de sistemas basados en patrones (Adamopoulou & Moussiades, 2020).

- **Tercera Generación: Asistentes de Voz y Expansión de Funcionalidades**

Entre 2010 y 2016, los asistentes de voz como Siri, Alexa y Google Assistant revolucionaron el uso de los chatbots. Estos sistemas integraron reconocimiento y síntesis de voz, permitiendo interacciones más naturales y complejas. Además, ofrecieron nuevas capacidades, como gestionar dispositivos inteligentes y acceder a información en tiempo real. Esta integración con el Internet de las Cosas (IoT) posicionó a los chatbots como herramientas esenciales para la vida cotidiana (Adamopoulou & Moussiades, 2020).

- **Cuarta Generación: Chatbots en Redes Sociales y Emocionales**

El auge de las plataformas de mensajería como Facebook Messenger en 2016 marcó un punto de inflexión en la popularidad de los chatbots. Miles de bots fueron diseñados para interactuar con usuarios en sectores como comercio, salud y educación, permitiendo una comunicación directa y eficiente. Además, sistemas como XiaoIce de Microsoft integraron cocientes de inteligencia emocional (EQ) y social, estableciendo relaciones emocionales duraderas con los usuarios y demostrando que los chatbots podían ir más allá de tareas funcionales (Adamopoulou & Moussiades, 2020).

### **2.1.3 Chatbots en el sector bancario**

El desarrollo de chatbots en la banca electrónica se sustenta en la inteligencia artificial generativa (IAG), tecnología que emplea modelos avanzados como ChatGPT para interpretar y responder a las necesidades de los usuarios. Según Gupta et al., los modelos generativos permiten personalizar las experiencias al analizar datos y patrones de comportamiento, transformando la interacción entre usuarios y sistemas digitales (Gupta et al., 2024).

La teoría de la respuesta social sostiene que los usuarios tienden a tratar a los chatbots como agentes humanos cuando estos muestran características antropomórficas, como el lenguaje natural y la empatía. Este enfoque, planteado por Nass y Moon, es crucial para entender cómo los usuarios pueden desarrollar confianza y lealtad hacia los chatbots (Huang & Lee, 2022). Asimismo, el marco PACT (Personas, Actividades, Contexto y Tecnología) propuesto por Janssen et al. proporciona una guía estructurada para diseñar chatbots centrados en el usuario, destacando la importancia de un enfoque adaptado al contexto (Janssen et al., 2022).

Por lo tanto, la revisión de la literatura evidencia que el diseño de chatbots para la banca electrónica requiere una integración cuidadosa de tecnologías avanzadas, como la inteligencia artificial generativa, junto con estrategias que prioricen la personalización, la empatía y la seguridad. Aunque se ha investigado ampliamente el impacto de estas tecnologías, aún persisten vacíos en áreas como la medición en tiempo real de la satisfacción del usuario y el diseño de interfaces inclusivas para personas con baja alfabetización digital. Esta investigación busca abordar estos desafíos mediante el desarrollo de un chatbot bancario que humanice y automatice la experiencia del cliente, optimizando tanto los procesos como la satisfacción del usuario.

## **2.2 Bases Teóricas**

### **2.2.1 Marco Legal Regulatorio en el Ecuador Respecto a los TRX Financieros**

El marco legal que regula las transacciones financieras electrónicas en Ecuador está diseñado para garantizar la seguridad, la confianza y la protección de los usuarios en el sistema financiero. Este apartado explora las normativas relevantes, el papel de las instituciones regulatorias y el impacto que estas disposiciones tienen en el diseño y funcionamiento de soluciones tecnológicas, como un chatbot bancario.

Dentro de la normativa legal la Resolución JB-2148, emitida por la Junta Bancaria del Ecuador, establece normas claras para que las instituciones financieras implementen medidas de seguridad avanzadas en sus operaciones electrónicas. Entre los aspectos más relevantes se encuentran:

- **Seguridad en canales electrónicos:** Las entidades financieras deben cumplir con estándares internacionales de seguridad, aplicando controles y monitoreos en hardware, software y redes para prevenir fraudes. Estas medidas son fundamentales para sistemas como los chatbots, que operan en entornos digitales.
- **Autenticación y encriptación:** Es obligatoria la aplicación de autenticación multifactorial (como contraseñas, dispositivos y biometría) y el uso de técnicas avanzadas de encriptación para proteger la información del cliente durante las transacciones financieras.
- **Auditoría y reportes:** Se requiere que las instituciones realicen auditorías anuales de seguridad y comuniquen cualquier incidente relevante a la Superintendencia de Bancos, lo que asegura la confiabilidad y la transparencia en sus operaciones (Junta Bancaria del Ecuador, 2012).

La resolución SEPS IGT IR ISF ITIC IGJ 2017-103 y Reforma 2017-113 emitidas por la Superintendencia de Economía Popular y Solidaria (SEPS) regulan las operaciones de cooperativas de ahorro y crédito, cajas centrales y asociaciones mutualistas (Mejía, 2017, 2017). Aspectos clave incluyen:

- **Plataformas seguras:** Se exige que las instituciones implementen tecnologías con capacidades de encriptación robusta y autenticación basada en factores múltiples, como contraseñas dinámicas y controles biométricos.
- **Registro de operaciones:** Los sistemas deben conservar un historial detallado de cada transacción, asegurando trazabilidad y permitiendo auditorías. Esto es crítico para chatbots que gestionan transacciones electrónicas.
- **Notificaciones en tiempo real:** Las entidades deben informar a los clientes sobre transacciones realizadas y accesos a sus cuentas, promoviendo la transparencia y la comunicación efectiva (Mejía, 2017, 23/112017).

- **Código Orgánico Monetario y Financiero**

Constituye el marco regulatorio principal del sistema financiero en Ecuador. Este documento define principios de seguridad, estabilidad y transparencia, estableciendo obligaciones específicas para las instituciones financieras respecto al uso de tecnologías y la protección de los usuarios (Mejía, 2017).

- La Superintendencia de Bancos y Seguros supervisa el cumplimiento de las normativas en el sistema financiero tradicional. Entre sus responsabilidades destacan:
  - Garantizar que las instituciones financieras adopten medidas para proteger la información de los usuarios.
  - Supervisar las auditorías y exigir el cumplimiento de los estándares de seguridad (Junta Bancaria del Ecuador, 2012).
- La Superintendencia de Economía Popular y Solidaria es el organismo que regula y supervisa a las cooperativas y entidades del sector solidario. Su acción se centra en:
  - Emitir normativas específicas para la seguridad de las transferencias electrónicas.
  - Velar por la aplicación de estándares internacionales en las plataformas tecnológicas utilizadas por estas entidades (Mejía, 2017, 23/112017).

El marco regulatorio tiene un impacto directo en la concepción y funcionamiento de un chatbot bancario. Las siguientes consideraciones son esenciales:

- **Diseño seguro:** El chatbot debe incorporar autenticación multifactorial, encriptación avanzada y monitoreo continuo para garantizar la protección de los datos del usuario.
- **Gestión de riesgos:** Es necesario implementar mecanismos que identifiquen y prevengan fraudes en tiempo real, bloqueando automáticamente transacciones sospechosas.
- **Transparencia:** El sistema debe notificar al usuario sobre cada acción realizada en su cuenta, asegurando que exista una comunicación clara y eficaz.

- **Cumplimiento normativo:** El chatbot debe registrar todas las actividades realizadas, lo que facilita las auditorías requeridas por las entidades regulatorias y respalda la confianza de los usuarios (Junta Bancaria del Ecuador, 2012; Mejía, 2017, 23/112017).

Integrar estas normativas en el diseño de un chatbot bancario no solo asegura el cumplimiento regulatorio, sino que también fortalece la confianza de los usuarios y la competitividad de las instituciones financieras. Esta visión permite el desarrollo de soluciones tecnológicas seguras y alineadas con los más altos estándares de calidad.

### **2.2.2 Definición y funcionamiento del chatbot**

Un chatbot es un agente conversacional diseñado para interactuar con usuarios mediante lenguaje natural, ya sea escrito o hablado, y responder de manera automática a las solicitudes de los usuarios (Rapp et al., 2021). Según el contexto, los chatbots se clasifican en dos grandes categorías: basados en reglas y basados en inteligencia artificial (IA) (Adamopoulou & Moussiades, 2020).

Los chatbots basados en reglas funcionan siguiendo flujos predefinidos que limitan las respuestas a opciones específicas. En cambio, los chatbots basados en IA, a menudo potenciados por aprendizaje automático, tienen capacidades para interpretar consultas más complejas y aprender de las interacciones previas (Agnihotri & Bhattacharya, 2024).

El diseño y uso de chatbots han evolucionado significativamente en la última década, con aplicaciones que abarcan desde la atención al cliente hasta la educación y la salud. Estos sistemas han demostrado ser efectivos para manejar tareas repetitivas, mejorar la disponibilidad de servicios y reducir costos operativos, aunque presentan limitaciones en escenarios que requieren empatía o entendimiento complejo (Lee & Li, 2023).

El funcionamiento de un chatbot sigue un proceso básico que comienza con la recepción de un mensaje del usuario. Este mensaje es procesado mediante técnicas de procesamiento de lenguaje natural (NLP, por sus siglas en inglés) para identificar la intención del usuario y extraer entidades relevantes, posteriormente, el sistema utiliza motores de decisión o modelos entrenados para generar una respuesta apropiada, que se entrega al usuario en tiempo real (Agnihotri & Bhattacharya, 2024).

En los chatbots basados en IA, el aprendizaje automático desempeña un rol crucial al permitir que el sistema mejore su rendimiento con el tiempo. Al analizar las interacciones previas, los modelos ajustan sus predicciones para ofrecer respuestas más precisas y relevantes (Rapp et al., 2021). Además, estos sistemas integran tecnologías avanzadas como análisis semántico y redes neuronales profundas para comprender mejor el contexto de las conversaciones (Lee & Li, 2023).

El éxito de los chatbots depende no solo de su capacidad técnica, sino también de la percepción del usuario en términos de confianza y satisfacción. Características como la capacidad de personalización, la empatía percibida y la consistencia en las respuestas contribuyen significativamente a la aceptación de esta tecnología (Agnihotri & Bhattacharya, 2024; Lee & Li, 2023).

### **2.2.3 Definición y funcionamiento de aplicaciones de mensajería**

Las aplicaciones de mensajería instantánea, como WhatsApp y Telegram, han evolucionado desde simples plataformas de comunicación personal hasta convertirse en herramientas esenciales en múltiples sectores, incluyendo la educación, el comercio y de manera destacada, la banca electrónica. Estas aplicaciones permiten la interacción en tiempo real mediante mensajes de texto, voz, imágenes y otros formatos multimedia, lo que las convierte en un pilar fundamental de la comunicación digital actual.

Entre estas herramientas destaca WhatsApp Business, una versión especializada diseñada para facilitar la interacción entre empresas y clientes. WhatsApp Business ofrece funcionalidades como respuestas automáticas, mensajes de bienvenida y la posibilidad de clasificar conversaciones mediante etiquetas, permitiendo a las organizaciones gestionar consultas de manera ágil y eficiente. Estas capacidades no solo mejoran la experiencia del cliente, sino que también optimizan las operaciones de las empresas al permitirles interactuar directamente con sus usuarios en sus canales preferidos (Sixto-García et al., 2021).

En el contexto de la banca electrónica, estas aplicaciones se han consolidado como herramientas clave para promover la inclusión financiera. En mercados emergentes, donde el acceso a los servicios financieros puede ser limitado debido a barreras de infraestructura, las aplicaciones de mensajería desempeñan un papel crucial al ofrecer una comunicación accesible, confiable y personalizada (Mogaji et al., 2021).

El funcionamiento de las aplicaciones de mensajería se basa en una arquitectura cliente-servidor, donde los mensajes se procesan y almacenan en servidores centrales. Esta arquitectura permite la sincronización en múltiples dispositivos, asegurando que los usuarios puedan acceder a sus mensajes desde cualquier lugar. En el caso de WhatsApp, se utiliza un cifrado de extremo a extremo que garantiza la privacidad de las comunicaciones, una característica especialmente importante para sectores como el bancario, donde la seguridad de los datos es primordial (Sixto-García et al., 2021).

Una de las principales fortalezas de estas aplicaciones es su capacidad para integrarse con sistemas externos mediante interfaces de programación de aplicaciones (API). Esto facilita la incorporación de chatbots, que permiten automatizar tareas como responder consultas frecuentes, realizar transacciones bancarias y ofrecer información personalizada. Un estudio realizado en mercados emergentes destaca que estas integraciones no solo incrementan la eficiencia operativa, sino que también mejoran la experiencia del usuario al proporcionar respuestas rápidas y confiables (Mogaji et al., 2021).

WhatsApp Business, por ejemplo, incluye herramientas avanzadas que mejoran la gestión empresarial, como:

- **Mensajes automatizados:** Respuestas rápidas y mensajes de ausencia que aseguran la disponibilidad continua.
- **Clasificación de contactos:** Organización de conversaciones mediante etiquetas personalizadas, lo que permite una gestión más estructurada.
- **Estadísticas:** Datos sobre mensajes enviados, entregados y leídos, proporcionando métricas valiosas para el análisis y la toma de decisiones estratégicas (Sixto-García et al., 2021).

Estas funcionalidades no solo aumentan la eficiencia de las interacciones empresariales, sino que también posicionan a WhatsApp como un canal preferido para la comunicación directa en entornos financieros.

En la banca electrónica, las aplicaciones de mensajería como WhatsApp se han consolidado como herramientas esenciales para optimizar la interacción entre las entidades financieras y sus clientes. En Ecuador, el Banco Bolivariano ha implementado "Avi24", un asistente digital

automatizado que resuelve el 98 % de las consultas sin necesidad de intervención humana, mostrando cómo estas plataformas mejoran la eficiencia y satisfacción del usuario (*Banco Bolivariano*, s. f.).

Estas herramientas también permiten diseñar experiencias centradas en el cliente al integrar chatbots que automatizan transacciones y generan confianza en los servicios digitales. Con el 70 % de los clientes del Banco Bolivariano prefiriendo WhatsApp como su principal canal de comunicación, se demuestra su potencial para fomentar la inclusión financiera en comunidades con infraestructura bancaria limitada (*Banco Bolivariano*, s. f.).

#### **2.2.4 Procesamiento de lenguaje natural (NPL)**

El procesamiento de lenguaje natural (NLP) es un campo fascinante que combina inteligencia artificial, aprendizaje automático y lingüística computacional para permitir que las máquinas comprendan, interpreten y generen lenguaje humano de manera más natural (Liu et al., 2024). Sus inicios se remontan a la década de 1950, cuando se basaba en sistemas rígidos de reglas diseñadas manualmente para modelar la gramática y la sintaxis (Adamopoulou & Moussiades, 2020). Aunque estos primeros enfoques marcaron un avance significativo, tenían muchas limitaciones, ya que no podían adaptarse fácilmente a variaciones lingüísticas ni a nuevos contextos (Ahmad, 2024; Supriyono et al., 2024).

Con el tiempo, durante los años 80 y 90, el NLP evolucionó hacia modelos estadísticos, que utilizaban grandes cantidades de datos textuales para identificar patrones y probabilidades. Métodos como los modelos de n-gramas ayudaron a mejorar el análisis del lenguaje, pero aún tenían dificultades para capturar relaciones semánticas de largo alcance (Ahmad, 2024). Luego, con la llegada del siglo XXI, se introdujeron las redes neuronales recurrentes (RNN), las cuales lograron avances en la comprensión del lenguaje al procesar secuencias de texto de manera más eficiente. Sin embargo, estas redes presentaban problemas como el gradiente desaparecido y requerían una gran capacidad de cómputo, lo que limitaba su uso en tareas más complejas (Supriyono et al., 2024).

La verdadera revolución ocurrió con la aparición de los modelos transformadores, como BERT (Bidirectional Encoder Representations from Transformers) y GPT (Generative Pre-trained Transformer), que marcaron un antes y un después en el procesamiento del lenguaje natural (Supriyono et al., 2024). Estos modelos introdujeron un mecanismo de atención que permite

analizar el texto en su totalidad en lugar de hacerlo de manera secuencial, lo que mejoró significativamente la precisión y la eficiencia en la interpretación del lenguaje. Gracias a estas innovaciones, ahora es posible realizar tareas como análisis de sentimientos, clasificación de texto, generación de lenguaje natural y reconocimiento de entidades con un nivel de comprensión mucho más avanzado (Ahmad, 2024; Supriyono et al., 2024).

Hoy en día, el NLP continúa evolucionando y transformando la forma en que las máquinas interactúan con el lenguaje humano. Su impacto se extiende a múltiples industrias, desde la automatización del servicio al cliente hasta la generación de contenido inteligente. Lo que antes parecía ciencia ficción, ahora es una realidad gracias a la combinación de datos masivos, potencia computacional y algoritmos cada vez más sofisticados (Ahmad, 2024).

Los chatbots son una de las aplicaciones más representativas del NLP y han transformado la forma en que interactuamos con las máquinas. Estas herramientas aprovechan modelos de lenguaje grande (LLM) como ChatGPT, que utilizan técnicas avanzadas para comprender preguntas, interpretar intenciones y generar respuestas coherentes (Ahmad, 2024). En este contexto, ChatGPT se destaca como un sistema capaz de mantener conversaciones naturales, incluso en temas complejos, al procesar preguntas y adaptarse al lenguaje del usuario en tiempo real (Du et al., 2025).

Los chatbots modernos utilizan modelos como GPT, entrenados con miles de millones de parámetros, para analizar patrones lingüísticos complejos y responder de manera precisa y contextual. Según (Supriyono et al., 2024), estos sistemas no solo generan texto de forma coherente, sino que también son capaces de manejar tareas avanzadas, como la traducción automática y el resumen de texto, lo que amplía significativamente su funcionalidad. Además, el aprendizaje por refuerzo con retroalimentación humana ha optimizado la interacción con los usuarios, permitiendo que los chatbots se adapten mejor a las necesidades específicas de cada contexto (Ahmad, 2024).

En sectores como atención al cliente y comercio electrónico, los chatbots utilizan técnicas de análisis de sentimientos y extracción de entidades para comprender las emociones y las intenciones de los usuarios, proporcionando respuestas personalizadas. Según (Liu et al., 2024), en escenarios como la revisión de productos en línea, los chatbots pueden procesar miles de

opiniones de usuarios, identificar tendencias y ofrecer recomendaciones precisas, ahorrando tiempo y mejorando la experiencia del consumidor.

En el sector financiero, el NLP permite a los chatbots procesar preguntas complejas relacionadas con transacciones o informes, mejorando la accesibilidad y eficiencia en servicios bancarios (Du et al., 2025). Esto no solo optimiza las operaciones, sino que también reduce la carga de trabajo en los equipos humanos. Por otro lado, en la educación, los chatbots se han convertido en asistentes de aprendizaje personalizados, proporcionando explicaciones detalladas, evaluaciones y retroalimentación en tiempo real (Supriyono et al., 2024).

A pesar de los avances, los chatbots aún enfrentan desafíos significativos. Entre ellos, la necesidad de procesar lenguajes específicos de dominio o manejar sesgos inherentes a los datos de entrenamiento (Ahmad, 2024), (Liu et al., 2024). Además, la generación de respuestas demasiado generales o irrelevantes sigue siendo un problema en algunos sistemas, lo que subraya la necesidad de una mayor personalización y adaptabilidad.

### **2.2.5 Modelos de IA aplicados en NPL**

El Procesamiento del Lenguaje Natural (NLP) ha evolucionado significativamente gracias a los avances en modelos de Inteligencia Artificial (IA). Tecnologías como las Redes Neuronales Recurrentes (RNN), los Transformers y modelos preentrenados como BERT y GPT han transformado la forma en que las máquinas comprenden y generan lenguaje humano, permitiendo aplicaciones más precisas y contextualmente relevantes. A continuación, se presentan los modelos relevantes:

- **Redes Neuronales Recurrentes (RNN):**

Las RNN sentaron las bases para el procesamiento de secuencias al capturar dependencias entre palabras en un texto. Aunque inicialmente fueron muy útiles para tareas como la traducción automática y el análisis de sentimientos, sus limitaciones, como la desaparición del gradiente, llevaron al desarrollo de variantes más avanzadas, como LSTM (Long Short-Term Memory) y GRU (Gated Recurrent Unit). Estas variantes son capaces de manejar relaciones a largo plazo de forma más efectiva, aunque su desempeño ha sido superado por enfoques más recientes como los Transformers (Liu et al., 2024; Supriyono et al., 2024).

- **Transformers:**

Introducidos en 2017, los Transformers revolucionaron el campo al emplear mecanismos de autoatención, eliminando la necesidad de procesar secuencias de manera lineal. Este enfoque no solo aceleró el procesamiento, sino que también permitió capturar relaciones contextuales complejas en textos largos. Modelos como BERT (Bidirectional Encoder Representations from Transformers) y GPT (Generative Pre-trained Transformer) han destacado en este paradigma (Supriyono et al., 2024). BERT sobresale por su capacidad para comprender el contexto bidireccional de las palabras en una oración, mientras que GPT, particularmente en su versión más reciente (GPT-4), ha demostrado un rendimiento excepcional en la generación de texto coherente y adaptativo (Ahmad, 2024; Supriyono et al., 2024).

- **Modelos Preentrenados:**

Los modelos preentrenados representan un cambio de paradigma en NLP al permitir que una arquitectura entrenada en grandes cantidades de datos sea ajustada para tareas específicas. Por ejemplo, FinBERT, adaptado al sector financiero, es capaz de analizar reportes y predecir tendencias del mercado, mientras que GPT-4 ha mostrado un rendimiento sobresaliente en tareas generales y específicas gracias a su capacidad de adaptarse a diferentes contextos mediante aprendizaje transferido (Ahmad, 2024; Du et al., 2025).

- **Combinación de técnicas avanzadas:**

En ciertas aplicaciones, los modelos de NLP han integrado teorías difusas para manejar la ambigüedad inherente al lenguaje humano. Las teorías difusas, basadas en la idea de que los datos y conceptos no siempre son estrictamente verdaderos o falsos, permiten trabajar con grados de pertenencia y razonamiento intermedio, en lugar de valores absolutos. Esto ha sido clave en problemas complejos, como la clasificación de emociones y la categorización de opiniones en texto (Liu et al., 2024).

Por ejemplo, mediante conjuntos difusos, es posible asignar diferentes grados de pertenencia a palabras relacionadas con un sentimiento específico, como "feliz" o "moderadamente feliz". La lógica difusa se utiliza para construir reglas flexibles tipo "SI-ENTONCES" que permiten razonar con ambigüedad, como ocurre en sistemas de

recomendación o en análisis de sentimientos. Estas capacidades hacen que los modelos de NLP sean más adaptativos y precisos al manejar conceptos vagos o contextos variados en el lenguaje natural (Liu et al., 2024).

La integración de estos modelos ha transformado a los chatbots, permitiéndoles interpretar y responder con mayor precisión. Por ejemplo, GPT-4 ha revolucionado los sistemas conversacionales al manejar interacciones complejas mediante ingeniería de prompts, que permite adaptar las respuestas a las necesidades del usuario (Ahmad, 2024). Asimismo, BERT se ha utilizado para tareas como extracción de información y análisis de sentimientos en tiempo real, mejorando significativamente la experiencia del cliente en plataformas de atención al usuario (Supriyono et al., 2024).

En sectores como las finanzas, modelos como FinBERT han permitido automatizar análisis de riesgos y predecir movimientos de mercado, mientras que en el comercio electrónico, el uso de NLP ha facilitado el análisis de reseñas y la personalización de recomendaciones, mejorando la experiencia del cliente (Du et al., 2025; Naz et al., 2025).

A pesar de sus logros, estos modelos enfrentan desafíos significativos. Requieren grandes volúmenes de datos para el preentrenamiento, lo que puede ser un obstáculo para idiomas con recursos limitados. Además, su desempeño está influenciado por los sesgos presentes en los datos de entrenamiento, lo que plantea riesgos éticos importantes en aplicaciones críticas como la salud y las finanzas (Casheekar et al., 2024). Por ejemplo, en el ámbito de la salud, un modelo sesgado podría priorizar diagnósticos erróneos para ciertos grupos demográficos, mientras que en finanzas, podría discriminar al evaluar la elegibilidad para un préstamo, basándose en patrones injustos aprendidos de los datos históricos (Ahmad, 2024; Supriyono et al., 2024).

Más allá de los chatbots, los modelos de NLP han encontrado un amplio uso en el sector financiero para tareas como análisis de sentimientos, generación de reportes automatizados y predicción de mercado (Du et al., 2025). De igual manera, en el análisis de reseñas de productos, técnicas avanzadas como el reconocimiento de entidades han mejorado la identificación de patrones clave y la experiencia del usuario (Naz et al., 2025).

### **2.2.6 Seguridad en chatbots para banca electrónica**

La transformación digital en el sector bancario ha traído consigo tecnologías innovadoras como los chatbots, los cuales han permitido optimizar procesos operativos, mejorar la atención al cliente y facilitar el acceso a servicios bancarios las 24 horas del día. No obstante, esta implementación también representa un desafío significativo en términos de seguridad, ya que los chatbots manejan información altamente sensible, como datos personales y financieros de los usuarios. Estos factores los convierten en objetivos atractivos para ciberataques, lo que podría poner en riesgo tanto la confianza de los usuarios como la estabilidad de las instituciones financieras (Araujo et al., 2024; Hasan et al., 2023).

Entre los riesgos más destacados se encuentra el robo de datos. Los chatbots pueden ser vulnerables a ataques que explotan fallas en su integración con los sistemas bancarios. Por ejemplo, un ataque de phishing puede disfrazarse de un chatbot legítimo para engañar a los usuarios y extraer sus credenciales de acceso o datos bancarios (Alagarsamy & Mehroliya, 2023). Además, las debilidades en la infraestructura tecnológica podrían facilitar accesos no autorizados, comprometiendo la integridad de la información almacenada o transmitida (Kitsios et al., 2021).

Otro desafío importante es la percepción de riesgo por parte de los usuarios. En la investigación (Petersson et al., 2023), se señala que la falta de transparencia en el manejo de datos genera incertidumbre, lo cual afecta directamente la adopción de estas tecnologías. Este aspecto se agrava en contextos donde no se comunican claramente las medidas de protección implementadas por las instituciones, como ocurre en el uso de chatbots en la banca, donde la confianza en el manejo de la información personal es un factor crítico para su aceptación. En este sentido, garantizar que los sistemas sean robustos y confiables es crucial para fomentar la confianza de los usuarios y su disposición a interactuar con chatbots en escenarios financieros sensibles (Alagarsamy & Mehroliya, 2023; Hasan et al., 2023).

Dado el entorno dinámico y las amenazas en constante evolución, las instituciones financieras deben adoptar un enfoque proactivo para mitigar los riesgos asociados al uso de chatbots. Una de las prácticas más esenciales es la implementación de cifrado de extremo a extremo, que protege la información transmitida entre el usuario y el sistema, asegurando que no pueda ser interceptada por terceros no autorizados. Este mecanismo es especialmente crucial en

transacciones financieras, donde la confidencialidad es prioritaria (Araujo et al., 2024; Kitsios et al., 2021).

Adicionalmente, la autenticación de dos factores (2FA) se ha consolidado como una de las medidas más efectivas para prevenir accesos no autorizados. Este método requiere que los usuarios proporcionen dos elementos de verificación independientes, lo que refuerza significativamente la seguridad. Por otro lado, la protección contra ataques de denegación de servicio (DDoS) es vital para garantizar la disponibilidad continua de los servicios, evitando interrupciones que puedan afectar tanto a usuarios como a operaciones internas (Alagarsamy & Mehroliya, 2023; Hasan et al., 2023).

Las auditorías regulares de seguridad también juegan un papel fundamental. Estas revisiones permiten identificar y corregir vulnerabilidades en el sistema antes de que puedan ser explotadas. Asimismo, implementar garantías estructurales, como políticas claras de privacidad y términos de uso accesibles, no solo mitiga riesgos, sino que también mejora la percepción de confianza por parte de los usuarios (Kitsios et al., 2021; Petersson et al., 2023).

Otro aspecto crítico es la capacitación y educación del cliente. Informar a los usuarios sobre las medidas de seguridad implementadas y los límites del sistema puede reducir la desconfianza y mejorar la aceptación de los chatbots. Es esencial que las instituciones comuniquen de manera clara cómo están protegiendo la información de sus clientes y qué pasos pueden tomar los usuarios para protegerse aún más (Alagarsamy & Mehroliya, 2023; Hasan et al., 2023).

Finalmente, los avances en inteligencia artificial y aprendizaje automático están siendo integrados en chatbots para detectar patrones anómalos que podrían indicar posibles amenazas. Estos sistemas de detección temprana permiten responder de manera más rápida y efectiva ante cualquier intento de ataque, asegurando que los servicios financieros permanezcan seguros y confiables (Alagarsamy & Mehroliya, 2023; Petersson et al., 2023).

### **2.2.7 Modelos de arquitectura de software**

Entre los patrones de diseño arquitectónicos aplicados a chatbot o sistemas de mensajería más destacados se encuentran los microservicios, que dividen las funciones en módulos independientes, permitiendo actualizaciones y escalabilidad sin interrupciones (Adamopoulou & Moussiades, 2020). Por ejemplo, los microservicios son especialmente útiles en aplicaciones

bancarias, donde los sistemas necesitan manejar grandes volúmenes de datos en tiempo real (Supriyono et al., 2024).

Otro enfoque clave son las arquitecturas basadas en APIs, que integran el chatbot con otros sistemas y servicios. Esto facilita la interoperabilidad y el acceso a fuentes de datos externas, mejorando la capacidad del chatbot para responder de manera precisa y personalizada (Ahmad, 2024; Hasan et al., 2023). Además, los modelos preentrenados (PLMs) como ChatGPT han revolucionado la capacidad de los chatbots para manejar tareas complejas de NLP. Estos modelos se destacan por su capacidad para comprender el contexto y generar respuestas coherentes en múltiples idiomas (Baabdullah et al., 2022; Rapp et al., 2021).

En cuanto al diseño, el enfoque en la experiencia del usuario (UX) es esencial. Un diseño bien estructurado no solo mejora la interacción, sino que también fomenta la aceptación del chatbot por parte de los usuarios. El Modelo-Vista-Controlador (MVC) es un patrón popular que organiza el diseño, separando las responsabilidades del sistema y facilitando su mantenimiento (Baabdullah et al., 2022).

Además, características como la personalización, la capacidad de respuesta y la conectividad ubicua son críticas para garantizar interacciones satisfactorias (Mogaji et al., 2021). Por ejemplo, los chatbots diseñados con personalización pueden ajustar sus respuestas a las necesidades específicas del usuario, lo que aumenta la confianza y la satisfacción (Jiang et al., 2022). Este tipo de diseño es especialmente relevante en mercados emergentes, donde las soluciones deben adaptarse a las limitaciones culturales y técnicas (Li et al., 2023).

En el sector bancario, los chatbots han demostrado ser herramientas eficaces para mejorar la accesibilidad y seguridad de los servicios. Al adoptar arquitecturas híbridas, las instituciones financieras han logrado equilibrar la funcionalidad técnica con la facilidad de uso, aumentando la retención de clientes (Petersson et al., 2023).

Por otro lado, la integración de teorías como la teoría difusa ha permitido manejar mejor la ambigüedad inherente al lenguaje humano. Esto ha sido fundamental en tareas como el análisis de sentimientos y el reconocimiento de entidades, donde la precisión es crucial (Liu et al., 2024). Además, los avances en *affordance* y diseño orientado al servicio han optimizado la interacción

del usuario en aplicaciones críticas, como la atención al cliente y el comercio electrónico (Hasan et al., 2023).

Los chatbots ya no son simples asistentes digitales; son herramientas integrales que transforman la manera en que las personas interactúan con las organizaciones. Desde responder preguntas frecuentes hasta realizar transacciones complejas, los chatbots están diseñados para ofrecer soluciones prácticas, rápidas y efectivas.

Sin embargo, el futuro de estas tecnologías también enfrenta desafíos. La privacidad de los datos, la confianza del usuario y la capacidad de adaptarse a contextos multiculturales son áreas que requieren atención continua. Según (Ahmad, 2024), la combinación de patrones arquitectónicos modernos con diseños centrados en el usuario será clave para superar estas barreras y expandir aún más el alcance de los chatbots.

### **2.2.8 Framework Backend NestJS para Aplicaciones con PLN/IA**

En el desarrollo de software, un framework (o marco de trabajo) se refiere a una estructura conceptual y tecnológica de soporte definido, usualmente con artefactos o componentes de código reusables, que busca facilitar y estandarizar la creación de aplicaciones. Proporciona una base sobre la cual los desarrolladores pueden construir software, ofreciendo herramientas para tareas comunes y, a menudo, guiando la arquitectura general del sistema (*Documentation | NestJS - A Progressive Node.js Framework, 2025*).

Dentro de este contexto, NestJS se presenta como un framework progresivo basado en Node.js, diseñado para la creación de aplicaciones del lado del servidor que sean eficientes, confiables y, notablemente, escalables (*Documentation | NestJS - A Progressive Node.js Framework, 2025*). Si bien su propósito principal no es el Procesamiento de Lenguaje Natural (PLN) o la Inteligencia Artificial (IA), su arquitectura modular y el uso de TypeScript proporcionan una base sólida y estructurada para desarrollar aplicaciones complejas que requieren la integración de dichas tecnologías, como es el caso de asistentes virtuales o chatbots avanzados (*Documentation | NestJS - A Progressive Node.js Framework, 2025*).

La filosofía de diseño de NestJS se centra en la organización del código a través de componentes bien definidos como Módulos, Controladores y Proveedores (Servicios), gestionados mediante un potente sistema de Inyección de Dependencias (*Documentation | NestJS - A Progressive Node.js Framework, 2025*). Esta estructura resulta particularmente útil al integrar funcionalidades de PLN:

- **Módulos (Modules)**

Permiten encapsular distintas partes de la aplicación. Por ejemplo, se podría tener un módulo específico para gestionar la lógica del chatbot, otro para la interfaz con el servicio o librería de PLN encargado de interpretar la intención del usuario, y módulos adicionales para interactuar con los sistemas bancarios subyacentes (*Documentation | NestJS - A Progressive Node.js Framework, 2025*). Esta separación fomenta la mantenibilidad y la organización del proyecto.

- **Controladores (Controllers)**

Actúan como la capa de entrada de la aplicación, manejando las solicitudes entrantes como los mensajes enviados por el usuario al chatbot a través de una API REST o WebSockets – y devolviendo las respuestas correspondientes (*Documentation | NestJS - A Progressive Node.js Framework, 2025*).

- **Proveedores/Servicios (Providers/Services)**

Contienen la lógica de negocio principal. Un servicio podría orquestar el flujo de la conversación del chatbot, otro servicio podría ser responsable de comunicarse con el motor de PLN (ya sea una API externa o una librería integrada) para analizar el texto del usuario, y otros servicios podrían ejecutar las transacciones o consultas bancarias una vez interpretada la intención (*Documentation | NestJS - A Progressive Node.js Framework, 2025*).

- **Inyección de Dependencias (Dependency Injection)**

Este patrón de diseño, central en NestJS, facilita la gestión de las relaciones entre los distintos componentes (por ejemplo, un controlador que necesita usar un servicio de chatbot), promoviendo un código más desacoplado, reutilizable y fácil de probar (*Documentation | NestJS - A Progressive Node.js Framework, 2025*).

Además, la arquitectura de NestJS está pensada para la escalabilidad, una consideración importante para aplicaciones como chatbots que pueden necesitar manejar un volumen creciente de interacciones (*Documentation | NestJS - A Progressive Node.js Framework, 2025*). Su soporte nativo para microservicios y su compatibilidad con diversas técnicas de comunicación asíncrona (como colas de mensajes) permiten diseñar sistemas robustos capaces de manejar tareas potencialmente intensivas, como el procesamiento de lenguaje natural, de manera eficiente (*Documentation | NestJS - A Progressive Node.js Framework, 2025*).

### 2.2.9 Plataformas APM (Application performance management)

Las plataformas de Application Performance Management (APM) se han convertido en herramientas esenciales en la gestión tecnológica moderna. Estas plataformas, diseñadas para monitorear, analizar y optimizar el rendimiento de aplicaciones, son fundamentales para garantizar que los sistemas cumplan con los niveles de servicio esperados. Un APM no solo proporciona información sobre métricas clave, como tiempos de respuesta, consumo de recursos o uso de memoria, sino que también ofrece insights detallados que permiten a los equipos técnicos abordar proactivamente problemas antes de que afecten a los usuarios (*¿Qué es la gestión del rendimiento de las aplicaciones (APM)?, 2024*).

El propósito central de una herramienta APM es ofrecer una visibilidad integral del ecosistema de aplicaciones. En un mundo donde las aplicaciones operan en entornos híbridos y distribuidos, como la nube, SaaS y arquitecturas de microservicios, la capacidad de supervisión y diagnóstico que brindan estas plataformas es crucial. Esto no solo mejora la disponibilidad y el rendimiento, sino que también asegura una experiencia satisfactoria para los usuarios finales (Sarumo López, 2020).

Las herramientas APM pueden clasificarse en función de sus capacidades principales y su enfoque:

- **Monitoreo de rendimiento basado en usuarios reales (RUM):** Se centra en recopilar datos directamente de las interacciones de los usuarios finales con las aplicaciones. Permite medir tiempos de carga, errores y la experiencia general del usuario.
- **Monitoreo sintético:** Utiliza pruebas automatizadas para simular la interacción del usuario con la aplicación, evaluando su rendimiento y disponibilidad en diferentes condiciones.

- **Supervisión de transacciones:** Rastrear las transacciones completas desde el inicio hasta el backend para identificar cuellos de botella en la infraestructura.
- **Análisis de dependencias:** Examina cómo interactúan los diferentes componentes de la aplicación, como bases de datos, APIs y servicios externos.
- **Monitoreo de infraestructura:** Evalúa el rendimiento del hardware y software subyacente que soporta la aplicación, como servidores, contenedores y sistemas operativos.

Entre las herramientas APM más utilizadas se encuentran:

- **Dynatrace:** Reconocida por su inteligencia artificial avanzada para automatizar la detección de problemas y su capacidad de monitorear infraestructuras complejas en tiempo real.
- **New Relic:** Proporciona un conjunto integral de herramientas para monitorear aplicaciones, infraestructura y experiencia del usuario.
- **AppDynamics:** Ofrece un enfoque integral que abarca desde el monitoreo del rendimiento hasta el análisis de negocios.
- **Datadog:** Ideal para monitorear sistemas distribuidos y basados en la nube, con capacidades avanzadas de visualización y alertas.
- **CA APM Introscope:** Destacada en grandes empresas por su capacidad para rastrear y optimizar aplicaciones críticas (Ahmed et al., 2016).

Por lo tanto, las herramientas APM son particularmente valiosas debido a su enfoque en la detección temprana de problemas y la optimización continua del rendimiento. Estas plataformas emplean técnicas avanzadas, como la instrumentación de código, que les permite capturar métricas en tiempo real sin interrumpir el funcionamiento de la aplicación. Además, las alertas configurables basadas en umbrales definidos facilitan la identificación de anomalías, mientras que los análisis de correlación ayudan a entender cómo los diferentes componentes del sistema interactúan entre sí (Ahmed et al., 2016).

Un aspecto clave del APM es su capacidad para rastrear transacciones completas, desde la interacción del usuario hasta los sistemas backend. Esto permite identificar cuellos de botella específicos, como consultas ineficientes a bases de datos o problemas en APIs externas, y tomar medidas correctivas de manera oportuna. Por ejemplo, la tesis de Sarumo López destaca cómo la

implementación de CA APM Introscope mejoró significativamente la gestión de incidencias, reduciendo tiempos de respuesta y aumentando la satisfacción de los usuarios (Sarumo López, 2020).

En el contexto de los chatbots, las plataformas APM juegan un rol fundamental para garantizar que estas soluciones, cada vez más populares, funcionen de manera eficiente y sin interrupciones. Los chatbots suelen operar en entornos complejos donde interactúan con múltiples componentes, como servicios en la nube, bases de datos y APIs de terceros. A continuación, sus principales contribuciones:

- **Monitoreo en tiempo real:** Los chatbots procesan miles de interacciones por minuto, lo que puede generar sobrecargas en el sistema. Las herramientas APM detectan estas sobrecargas y alertan a los administradores, permitiendo respuestas rápidas.
- **Identificación de problemas específicos:** Un chatbot puede experimentar fallos debido a problemas en su backend o en las APIs externas con las que interactúa. Las APM ayudan a localizar exactamente dónde se encuentra el problema.
- **Optimización de tiempos de respuesta:** Analizando patrones históricos y en tiempo real, las herramientas APM permiten reducir tiempos de latencia, mejorando así la experiencia del usuario (Ahmed et al., 2016).

Por ejemplo, si un chatbot experimenta demoras al consultar una base de datos, la APM puede señalar que el cuello de botella proviene de una consulta SQL específica, lo que permite a los desarrolladores optimizarla de inmediato.

Un caso real de la eficacia de las herramientas APM es la implementación del software CA APM Introscope en una empresa de telecomunicaciones. Según Sarumo López, este proyecto permitió monitorear aplicaciones críticas con una disponibilidad del 99.9%, reduciendo el tiempo de detección y solución de incidencias en un 59% y aumentando la satisfacción del usuario en un 66% (Sarumo López, 2020).

Asimismo, Ahmed et al. estudiaron cómo herramientas como AppDynamics y New Relic pueden identificar regresiones de rendimiento en aplicaciones web. Estas plataformas demostraron ser especialmente útiles en la detección de problemas como el uso excesivo de memoria y consultas

ineficientes a bases de datos, mejorando significativamente el rendimiento general del sistema (Ahmed et al., 2016).

Entre los beneficios más destacados de las plataformas APM se encuentran:

- **Visibilidad completa:** Permiten supervisar todas las capas de una aplicación, desde el frontend hasta el backend.
- **Detección temprana de problemas:** Identifican anomalías antes de que se conviertan en incidentes graves.
- **Mejora continua:** Facilitan la evaluación de métricas clave, promoviendo optimizaciones constantes.

Sin embargo, estas herramientas también presentan desafíos, la configuración inicial puede ser costosa y compleja, especialmente en organizaciones con infraestructuras TI heterogéneas. Además, algunas plataformas carecen de extensibilidad, lo que limita su capacidad para adaptarse a necesidades específicas (Sarumo López, 2020).

### 2.2.10 Metodologías de desarrollo de software

El desarrollo de software ha evolucionado significativamente con la adopción de metodologías ágiles, que han demostrado ser altamente efectivas en proyectos que requieren flexibilidad, iteración y retroalimentación continua. Estas metodologías, como Scrum y Extreme Programming (XP), se centran en la entrega rápida de valor, la colaboración entre equipos y la adaptación a cambios constantes, características esenciales en el desarrollo de sistemas como chatbots. A continuación, se analizan los principios fundamentales de las metodologías ágiles y su relevancia en la construcción de chatbots, con énfasis en su capacidad de adaptarse a los requerimientos cambiantes del usuario y el entorno.

- **Scrum**

Scrum es una de las metodologías ágiles más populares y se centra en la transparencia, la inspección y la adaptación (Estrada Velasco et al., 2021). Divide el trabajo en iteraciones cortas llamadas sprints, que permiten entregas incrementales y frecuentes. Un equipo Scrum se compone de roles específicos como el Scrum Master, el Product Owner y los desarrolladores. Estas iteraciones garantizan que el producto pueda adaptarse rápidamente a los cambios en los requisitos o el entorno (Estrada Velasco et al., 2021).

- **Extreme Programming (XP)**

XP se enfoca en mejorar la calidad del software y la capacidad de respuesta a los requisitos cambiantes mediante prácticas como el desarrollo guiado por pruebas (TDD), la programación en pareja y la integración continua (Kuhrmann et al., 2022). Estas prácticas fomentan la colaboración y reducen los errores en el desarrollo. Su aplicación en chatbots asegura que los componentes críticos sean probados de manera exhaustiva y que el código sea fácil de mantener (Kuhrmann et al., 2022).

- **Kanban**

Kanban utiliza un sistema visual para gestionar el flujo de trabajo y optimizar los procesos. A través de tableros, se priorizan las tareas, lo que permite identificar cuellos de botella y mejorar la eficiencia. En el desarrollo de chatbots, Kanban ayuda a gestionar tareas complejas y garantiza la entrega fluida de funcionalidades (Kuhrmann et al., 2022).

- **Lean Development**

El desarrollo Lean busca minimizar el desperdicio y maximizar el valor entregado al cliente. Esto se logra mediante la eliminación de actividades innecesarias y la entrega rápida de mínimos productos viables (MVP). En chatbots, esta metodología permite experimentar con versiones iniciales para obtener retroalimentación temprana de los usuarios (Kuhrmann et al., 2022).

- **Scaled Agile Framework (SAFe)**

SAFe se utiliza para proyectos grandes y distribuidos, combinando prácticas ágiles y principios de escalabilidad. Proporciona una estructura para coordinar equipos múltiples y garantizar que los objetivos organizacionales se alineen con las prioridades de desarrollo. En chatbots, SAFe es útil cuando se integran múltiples sistemas o equipos (Kuhrmann et al., 2022).

El desarrollo de chatbots presenta desafíos particulares debido a la necesidad de integrar tecnologías avanzadas, como el procesamiento del lenguaje natural (PLN), y de garantizar una experiencia de usuario intuitiva y adaptativa. En este contexto, las metodologías ágiles son herramientas clave para abordar estas complejidades.

- **Entregas Iterativas y Mejoras Continuas:** La iteración permite a los equipos probar y refinar funcionalidades críticas del chatbot, como la comprensión del lenguaje y las respuestas automatizadas, basándose en datos reales de interacción con los usuarios. Según Strode et al. (2022), la retroalimentación continua entre los miembros del equipo y los usuarios finales fomenta un ciclo de mejora constante que asegura que el producto final se alinee con las expectativas y necesidades del cliente (Strode et al., 2022).
- **Adaptabilidad a Cambios:** La capacidad de las metodologías ágiles para adaptarse a cambios es crucial en proyectos de chatbots, donde las tendencias tecnológicas y los patrones de interacción de los usuarios pueden cambiar rápidamente. Scrum, por ejemplo, ofrece mecanismos como las reuniones diarias (Daily Scrums) y la revisión de sprints, que permiten a los equipos ajustar las prioridades y resolver impedimentos de manera ágil y eficaz (Estrada Velasco et al., 2021).
- **Colaboración y Dinámica de Equipos:** La colaboración entre equipos multidisciplinarios es un pilar fundamental en el desarrollo ágil. Strode et al. (2022) enfatizan que los modelos de efectividad en equipos ágiles, como el Agile Teamwork Effectiveness Model (ATEM), destacan la importancia del liderazgo compartido, la confianza mutua y los modelos mentales compartidos para lograr la cohesión y el éxito en proyectos complejos (Strode et al., 2022).

Los beneficios de las Metodologías Ágiles en Chatbots son los siguientes:

- **Reducción del Riesgo:** Al entregar incrementos funcionales de manera regular, se reduce el riesgo de que el producto final no cumpla con las expectativas del cliente.
- **Incremento de la Calidad:** Las prácticas como la integración continua y las pruebas frecuentes garantizan que los errores sean detectados y corregidos de manera temprana.

- **Mayor Participación del Usuario:** La retroalimentación constante de los usuarios finales permite ajustes precisos en las funcionalidades del chatbot, mejorando la experiencia del usuario. Las metodologías ágiles, con su enfoque en la adaptabilidad, la iteración y la colaboración, son ideales para el desarrollo de chatbots, proporcionando un marco flexible que permite responder rápidamente a los cambios y garantizar que el producto final cumpla con las necesidades del cliente. Documentos como los de Strode et al. (2022), Estrada-Velasco et al. (2021) y Kuhrmann et al. (2021) subrayan la eficacia de estas metodologías en entornos complejos, donde la innovación y la adaptabilidad son cruciales. Esto posiciona a las metodologías ágiles como una elección estratégica para el desarrollo de sistemas modernos e interactivos como los chatbots.

## CAPÍTULO III

### MATERIALES Y MÉTODOS

#### 3.1 Metodología

Para el desarrollo del chatbot bancario se decidió utilizar la metodología Scrum, debido a su enfoque ágil, adaptable y orientado a resultados. Esta metodología permite organizar el trabajo en ciclos cortos llamados sprints, en los cuales se entregan avances parciales que pueden ser revisados y mejorados continuamente.

Scrum promueve una dinámica de colaboración constante entre los integrantes del equipo y facilita la identificación temprana de posibles mejoras o cambios en los requerimientos. Gracias a su estructura iterativa e incremental, se logra mantener un alto nivel de flexibilidad durante el desarrollo, respondiendo de forma efectiva a las necesidades del proyecto.

Esta metodología no solo optimiza la gestión de tiempos y recursos, sino que también fomenta la entrega continua de valor, garantizando que el producto final evolucione de manera acorde a las expectativas del usuario y a los objetivos planteados.

#### 3.2 Levantamiento de requisitos

Durante la fase inicial del desarrollo del chatbot bancario, se ejecutó el proceso de levantamiento de requisitos con el propósito de identificar y estructurar de forma clara las funcionalidades necesarias para satisfacer las necesidades del usuario final. Esta etapa fue esencial para sentar las bases del diseño funcional y técnico del sistema, ya que permitió comprender las expectativas de los stakeholders, así como los flujos conversacionales que el chatbot deberá manejar.

Las funcionalidades identificadas se agruparon en los siguientes módulos clave:

- Levantamiento de requisitos
- Autenticación en el chatbot
- Aceptación de términos y condiciones
- Consulta de saldo - cuenta de ahorros
- Consulta - certificados de aportación
- Consulta - ahorro programado

- Consulta - depósitos a plazo fijo
- Consulta – últimos movimientos
- Consulta - créditos
- Transferencias internas
- Transferencias interbancarias
- Pago de tarjetas de crédito
- Pago de servicios básicos

Estos módulos funcionales fueron definidos a partir del análisis de los tipos de transacciones que el chatbot deberá permitir realizar a los usuarios, entre los cuales se incluyen:

- Consultas de saldo (cuentas, certificados, créditos)
- Transferencias (internas e interbancarias)
- Pagos (tarjetas de crédito)
- Pagos (servicios básicos)

A continuación, se presenta la **Tabla 1** con el detalle de los requisitos funcionales derivados de dichas transacciones:

**Tabla 1**

*Requisitos Funcionales*

<b>ID</b>	<b>Nombre del Requisito</b>	<b>Descripción breve</b>
RQBI-001	Levantamiento de requisitos	Identificar, recolectar y validar los requerimientos funcionales y no funcionales del sistema.
RQAU-002	Autenticación en el chatbot	Implementar un proceso seguro de autenticación de usuarios basado en número de celular y verificación de identidad.
RQAU-003	Aceptación de términos y condiciones	Gestionar la aceptación obligatoria de términos de

ID	Nombre del Requisito	Descripción breve
		<p>uso antes de permitir el acceso a los servicios del chatbot.</p>
RQTR-001	Consulta de saldo - cuenta de ahorros	<p>Permitir a los usuarios consultar el saldo disponible en sus cuentas de ahorros.</p>
RQTR-002	Consulta - certificados de aportación	<p>Facilitar la consulta de certificados de aportación vigentes.</p>
RQTR-003	Consulta - ahorro programado	<p>Brindar acceso a la información de saldo y cumplimiento del ahorro programado.</p>
RQTR-004	Consulta - depósitos a plazo fijo	<p>Mostrar detalles de depósitos a plazo fijo activos, incluyendo fechas de vencimiento e intereses.</p>
RQTR-005	Consulta – últimos movimientos	<p>Mostrar detalles de los últimos movimientos.</p>
RQTR-006	Consulta - créditos	<p>Permitir la visualización del saldo pendiente y cuotas de créditos activos.</p>
RQTR-007	Transferencia interna	<p>Habilitar transferencias de fondos entre cuentas propias dentro de la entidad financiera.</p>
RQTR-008	Transferencia interbancaria	<p>Habilitar transferencias de fondos hacia otras entidades financieras.</p>

<b>ID</b>	<b>Nombre del Requisito</b>	<b>Descripción breve</b>
RQTR-009	Pagos de tarjetas de crédito	Habilitar el pago de tarjetas de crédito.
RQTR-010	Pagos de servicios básicos	Habilitar el pago de servicios básicos como: agua, luz, teléfono, internet y TV.

### 3.2.1 Estimación del esfuerzo en historias de usuario

Para una adecuada planificación del desarrollo, se asignó una estimación de esfuerzo a cada historia de usuario identificada. Esta estimación refleja el nivel de trabajo que implica la implementación de cada funcionalidad, considerando elementos como:

- La integración de APIs externas.
- La complejidad de los flujos conversacionales.
- Los controles de seguridad necesarios.
- Los riesgos técnicos asociados.

Se optó por utilizar una escala basada en una progresión adaptada de Fibonacci, ajustada a intervalos de esfuerzo en días laborales. En la **Tabla 2**, se presenta la escala que sirvió como referencia para valorar las historias de usuario:

**Tabla 2**

*Escala basada en una progresión adaptada de Fibonacci*

<b>Puntos de Estimación</b>	<b>Interpretación</b>
0	Ajustes mínimos o tareas de corrección que pueden resolverse en menos de una hora.
½	Actividades sencillas, estimadas en aproximadamente 4 horas.
1	Funcionalidades de baja complejidad, abordables en una jornada laboral completa (8 horas).

<b>Puntos de Estimación</b>	<b>Interpretación</b>
2	Funcionalidades moderadas, que pueden requerir entre 1 y 2 días de trabajo.
3	Funcionalidades de complejidad media, que demandan entre 2 y 4 días.
5	Funcionalidades más elaboradas, estimadas en un rango de 3 a 5 días de desarrollo efectivo.
8	Funcionalidades de alta complejidad, cuyo desarrollo puede extenderse entre 5 y 8 días hábiles.
13	Funcionalidades amplias que requieren hasta dos semanas de trabajo continuo.
20 o más	Funcionalidades de gran tamaño que, por su magnitud, deben dividirse en tareas más pequeñas y manejables.

### 3.2.2 Participantes del proyecto

En el desarrollo de un chatbot para banca virtual con inteligencia artificial generativa, se contará con un equipo de participantes que asumirán roles definidos dentro del proyecto. Esta estructura organizativa es fundamental para establecer con precisión el alcance del proyecto y asegurar una gestión eficiente del mismo. En la **Tabla 3** se presenta la asignación de roles específicos permitirá una distribución clara de responsabilidades y facilitará la comunicación entre los miembros del equipo:

**Tabla 3**

*Participantes Involucrados*

<b>NOMBRE</b>	<b>DESCRIPCIÓN</b>	<b>ROL</b>
MSc. Diego Trejo	Director del proyecto de titulación.	Propietario del producto (Product Owner) Jefe de proyecto (Scrum Master)

NOMBRE	DESCRIPCIÓN	ROL
Sr. Jonathan Pilatuña	Tesista	Equipo de desarrollo (Development Team)

### 3.2.3 Historias de usuario

Son elementos clave en el desarrollo de un producto, ya que en ellas se describen de manera precisa las acciones y requisitos necesarios para su implementación. A través de ellas, se define la forma en que el software debe operar, asegurando que cumpla con las expectativas y necesidades.

#### Tabla 4

##### *Historia de Usuario RQBI-001*

Historia de Usuario	ID: RQBI-001
<b>Usuario:</b> Equipo de desarrollo	
<b>Nombre de la historia:</b> Levantamiento de Requisitos	
<b>Prioridad:</b> Alta	<b>Riesgo de desarrollo:</b> Medio
<b>Estimación:</b> 5	
<b>Descripción:</b>	
Como equipo de desarrollo, quiero levantar y validar los requisitos del chatbot, para garantizar que el sistema cumpla con las expectativas del cliente.	
<b>Criterios de aceptación:</b>	
Documento consolidado de requisitos, validado y almacenado, incluyendo APIs, flujos, roles y restricciones.	

## Tabla 5

### Historia de Usuario RQAU-002

---

Historia de Usuario	ID: RQAU-002
<b>Usuario:</b> Cliente de la entidad financiera	
<b>Nombre de la historia:</b> Autenticación en el chatbot	
<b>Prioridad:</b> Alta	<b>Riesgo de desarrollo:</b> Alto
<b>Estimación:</b> 5	
<b>Descripción:</b> Como usuario, quiero autenticarme usando mi número de celular y un código de validación, para acceder de forma segura al sistema.	
<b>Criterios de aceptación:</b> El sistema impide el acceso a usuarios no autenticados. La sesión debe expirar en un tiempo configurado.	

---

## Tabla 6

### Historia de Usuario RQAU-003

---

Historia de Usuario	ID: RQAU-003
<b>Usuario:</b> Cliente de la entidad financiera	
<b>Nombre de la historia:</b> Aceptación de términos y condiciones	
<b>Prioridad:</b> Alta	<b>Riesgo de desarrollo:</b> Medio
<b>Estimación:</b> 3	
<b>Descripción:</b> Como usuario, quiero aceptar los términos y condiciones antes de usar el chatbot, para conocer y aceptar las reglas del servicio.	
<b>Criterios de aceptación:</b> El acceso a funcionalidades solo es permitido después de aceptar los términos. El consentimiento queda registrado.	

---

### Tabla 7

#### *Historia de Usuario RQTR-001*

---

<b>Historia de Usuario</b>	<b>ID: RQTR-001</b>
----------------------------	---------------------

---

**Usuario:** Cliente de la entidad financiera

**Nombre de la historia:** Consulta de saldo - cuenta de ahorros

**Prioridad:** Alta **Riesgo de desarrollo:** Medio

**Estimación:** 3

**Descripción:**  
Como cliente, quiero consultar el saldo de mi cuenta de ahorros, para conocer mi disponibilidad de fondos.

**Criterios de aceptación:**  
El chatbot muestra correctamente el saldo disponible tras la autenticación del usuario.

---

### Tabla 8

#### *Historia de Usuario RQTR-002*

---

<b>Historia de Usuario</b>	<b>ID: RQTR-002</b>
----------------------------	---------------------

---

**Usuario:** Cliente de la entidad financiera

**Nombre de la historia:** Consulta - certificados de aportación

**Prioridad:** Media **Riesgo de desarrollo:** Baja

**Estimación:** 2

**Descripción:**  
Como cliente, quiero consultar mis certificados de aportación, para revisar su valor y estado actual.

**Criterios de aceptación:**  
El sistema lista los certificados vigentes, mostrando valores actualizados.

---

## Tabla 9

### *Historia de Usuario RQTR-003*

---

<b>Historia de Usuario</b>	<b>ID: RQTR-003</b>
----------------------------	---------------------

---

**Usuario:** Cliente de la entidad financiera

**Nombre de la historia:** Consulta - ahorro programado

**Prioridad:** Media **Riesgo de desarrollo:** Bajo

**Estimación:** 2

**Descripción:**  
Como cliente, quiero conocer el saldo acumulado de mi ahorro programado, para verificar el cumplimiento de mi meta de ahorro.

**Criterios de aceptación:**  
El chatbot muestra el saldo y el porcentaje de cumplimiento de la meta.

---

## Tabla 10

### *Historia de Usuario RQTR-004*

---

<b>Historia de Usuario</b>	<b>ID: RQTR-004</b>
----------------------------	---------------------

---

**Usuario:** Cliente de la entidad financiera

**Nombre de la historia:** Consulta - depósitos a plazo fijo

**Prioridad:** Media **Riesgo de desarrollo:** Bajo

**Estimación:** 2

**Descripción:**  
Como cliente, quiero visualizar los depósitos a plazo fijo activos, para conocer su vencimiento e intereses generados.

**Criterios de aceptación:**  
Se presenta la información detallada de cada depósito vigente.

---

**Tabla 11***Historia de Usuario RQTR-005*

---

<b>Historia de Usuario</b>	<b>ID: RQTR-005</b>
----------------------------	---------------------

---

**Usuario:** Cliente de la entidad financiera

**Nombre de la historia:** Consulta – movimientos de cuenta

**Prioridad:** Media **Riesgo de desarrollo:** Bajo

**Estimación:** 2

**Descripción:**  
Como cliente, quiero visualizar los últimos movimientos de mis cuentas, para conocer mis transacciones recientes.

**Criterios de aceptación:**  
Se presenta la información detallada de cada movimiento reciente.

---

**Tabla 12***Historia de Usuario RQTR-006*

---

<b>Historia de Usuario</b>	<b>ID: RQTR-006</b>
----------------------------	---------------------

---

**Usuario:** Cliente de la entidad financiera

**Nombre de la historia:** Consulta - créditos

**Prioridad:** Alta **Riesgo de desarrollo:** Medio

**Estimación:** 3

**Descripción:**  
Como cliente, quiero consultar mi crédito activo, para conocer el saldo pendiente y las próximas cuotas a pagar.

**Criterios de aceptación:**  
Se muestra el saldo actual, fechas de pago y estado de morosidad si existiera.

---

### Tabla 13

*Historia de Usuario RQTR-007*

---

<b>Historia de Usuario</b>	<b>ID: RQTR-007</b>
----------------------------	---------------------

---

**Usuario:** Cliente de la entidad financiera

**Nombre de la historia:** Transferencia interna

**Prioridad:** Alta **Riesgo de desarrollo:** Alto

**Estimación:** 5

**Descripción:**  
Como cliente, quiero realizar transferencias entre mis cuentas dentro del mismo banco, para mover fondos de manera rápida.

**Criterios de aceptación:**  
El chatbot valida saldo suficiente, autenticación activa y confirma la transacción entre cuentas propias.

---

### Tabla 14

*Historia de Usuario RQTR-008*

---

<b>Historia de Usuario</b>	<b>ID: RQTR-008</b>
----------------------------	---------------------

---

**Usuario:** Cliente de la entidad financiera

**Nombre de la historia:** Transferencia interbancaria

**Prioridad:** Alta **Riesgo de desarrollo:** Alto

**Estimación:** 8

**Descripción:**  
Como cliente, quiero transferir dinero a otras entidades bancarias, para enviar fondos a terceros de manera segura.

**Criterios de aceptación:**  
El sistema solicita datos del banco destino, valida la información y confirma la transferencia.

---

## Tabla 15

*Historia de Usuario RQTR-009*

---

<b>Historia de Usuario</b>	<b>ID: RQTR-009</b>
----------------------------	---------------------

---

**Usuario:** Cliente de la entidad financiera

**Nombre de la historia:** Pago de tarjeta de crédito

**Prioridad:** Alta **Riesgo de desarrollo:** Medio

**Estimación:** 5

**Descripción:**  
Como cliente, quiero pagar mi tarjeta de crédito a través del chatbot, para mantener mis pagos al día de forma cómoda.

**Criterios de aceptación:**  
Se muestra el saldo pendiente y se permite seleccionar el monto a pagar. Confirmación al finalizar el pago.

---

## Tabla 16

*Historia de Usuario RQTR-010*

---

<b>Historia de Usuario</b>	<b>ID: RQTR-010</b>
----------------------------	---------------------

---

**Usuario:** Cliente de la entidad financiera

**Nombre de la historia:** Pago de servicios básicos

**Prioridad:** Alta **Riesgo de desarrollo:** Medio

**Estimación:** 5

**Descripción:**  
Como cliente, quiero pagar servicios básicos (agua, luz, internet, etc.) mediante el chatbot, para cumplir mis obligaciones fácilmente.

**Criterios de aceptación:**  
Se consulta la deuda pendiente, se solicita autorización de pago y se confirma la transacción exitosa.

---

### 3.2.4 Product Backlog

El Product Backlog agrupa todas las historias de usuario identificadas durante el levantamiento de requisitos. Cada historia está definida en función de su actor principal, prioridad, riesgo de desarrollo y esfuerzo estimado, incluyendo además una descripción detallada y sus respectivos criterios de aceptación.

En la **Tabla 17** se presenta el Product Backlog del proyecto:

**Tabla 17**

*Product Backlog*

<b>ID</b>	<b>Actor Principal</b>	<b>Nombre de la Historia de Usuario</b>	<b>Prioridad</b>	<b>Riesgo</b>	<b>Estimación</b>
RQBI-001	Equipo de desarrollo	Levantamiento de requisitos	Alta	Medio	5
RQAU-002	Cliente	Autenticación en el chatbot	Alta	Alto	5
RQAU-003	Cliente	Aceptación de términos y condiciones	Alta	Medio	3
RQTR-001	Cliente	Consulta de saldo - cuenta de ahorros	Alta	Medio	3
RQTR-002	Cliente	Consulta - certificados de aportación	Media	Bajo	2
RQTR-003	Cliente	Consulta - ahorro programado	Media	Bajo	2
RQTR-004	Cliente	Consulta - depósitos a plazo fijo	Media	Bajo	2

<b>ID</b>	<b>Actor Principal</b>	<b>Nombre de la Historia de Usuario</b>	<b>Prioridad</b>	<b>Riesgo</b>	<b>Estimación</b>
RQTR-005	Cliente	Consulta – últimos movimientos	Media	Bajo	2
RQTR-006	Cliente	Consulta - créditos	Alta	Medio	3
RQTR-007	Cliente	Transferencia interna	Alta	Alto	5
RQTR-008	Cliente	Transferencia interbancaria	Alta	Alto	8
RQTR-009	Cliente	Pago de tarjeta de crédito	Alta	Medio	5
RQTR-010	Cliente	Pago de servicios básicos	Alta	Medio	5

### 3.3 Desarrollo

El desarrollo del software se documentará detalladamente a través de sprints. Cada funcionalidad implementada será descrita en el contexto del sprint en el que se desarrolló, asegurando una comprensión clara y organizada del progreso del proyecto.

#### 3.3.1 Sprint 0

**Tabla 18**

*Sprint 0*

<b>Historia de Usuario</b>	<b>Actividad</b>	<b>Tarea</b>	<b>#Horas</b>
RQBI-001	Planificación inicial	Definir alcance del proyecto y delimitar los objetivos generales	4

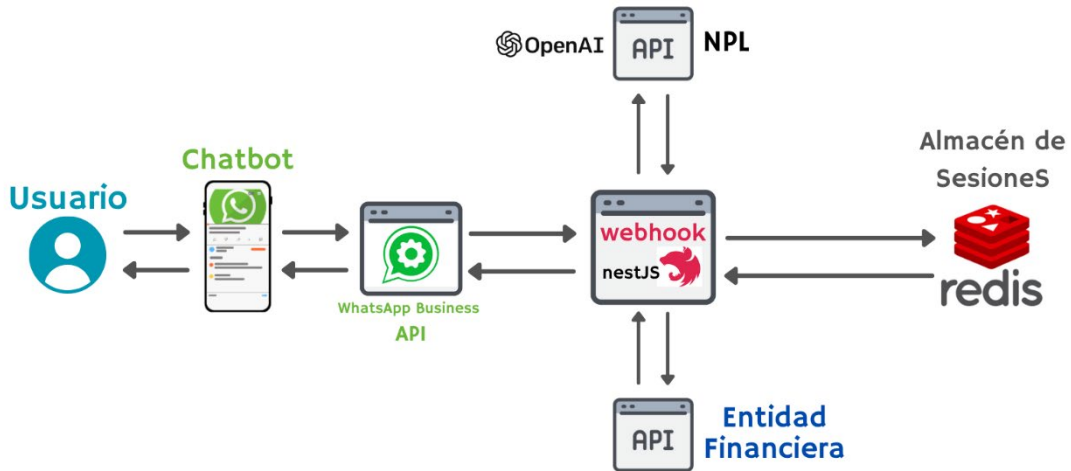
Historia de Usuario	Actividad	Tarea	#Horas
	Levantamiento de requisitos	Reunión y entrevistas con el stakeholder de la Industria Tecnológica MTC	5
		Documentar los requerimientos funcionales y no funcionales	6
	Selección de herramientas	Analizar tecnologías posibles y seleccionar: NestJS, Visual Studio Code, APIs externas	3

## Desarrollo del Sprint

Durante el Sprint 0 se definió el alcance general del proyecto, incluyendo los objetivos principales y las funcionalidades esenciales del chatbot. La colaboración con los stakeholder de la Industria Tecnológica MTC fue directa y fundamental, ya que nos proporcionó información clave sobre las operaciones bancarias a integrar y los flujos conversacionales necesarios. Adicionalmente, ellos facilitaron un usuario de prueba con sus credenciales (usuario y contraseña) para el acceso a la API financiera.

El levantamiento de requisitos permitió identificar las necesidades funcionales prioritarias, tales como consultas de saldos, transferencias internas e interbancarias, y pagos de servicios básicos. Asimismo, se analizaron y seleccionaron las tecnologías que formarían parte del desarrollo (ver **Fig. 2**):

- **Framework de desarrollo:** NestJS para integrar las APIs necesarias.
- **Entorno de desarrollo:** Visual Studio Code.
- **Integraciones:** API de Meta para WhatsApp Business, API de la entidad financiera para conexión de servicios, y la API de OpenAI para procesamiento de lenguaje natural.



**Fig. 2.** *Arquitectura del Final del Chatbot*

Con una comprensión detallada de los requisitos y las herramientas técnicas a utilizar, se estableció una base sólida para iniciar el desarrollo estructurado del chatbot bancario en los siguientes sprints.

### 3.3.1 Sprint 1

**Tabla 19**

*Sprint 1*

Historia de Usuario	Actividad	Tarea	# Horas
RQAU-002	Implementación de autenticación	Integrar framework NestJS para manejar solicitudes entrantes desde WhatsApp	4
		Configurar el webhook para recibir y gestionar mensajes iniciales	3
		Desarrollar flujo de autenticación basado en número de celular y número de cédula	3
		Integrar API de OpenAI (GPT) para interpretar intención de mensajes	3
		Implementar verificación de sesión activa y gestión de sesiones en Redis	3

<b>Historia de Usuario</b>	<b>Actividad</b>	<b>Tarea</b>	<b># Horas</b>
RQAU-003	Gestión de términos y condiciones	Programar la presentación automática de términos y condiciones tras la autenticación exitosa	3
		Registrar el consentimiento del usuario en la base de datos del sistema	2

## Desarrollo del Sprint

Durante el Sprint 1 se establecieron los cimientos para un acceso seguro y regulado a los servicios financieros disponibles en el chatbot.

En primer lugar, se implementó la autenticación de usuarios, desarrollando un flujo que valida tanto el número de teléfono registrado como las credenciales del usuario, garantizando que solo los clientes autorizados puedan acceder a las funcionalidades. Esta verificación se integra con la gestión de sesiones utilizando Redis, asegurando que los usuarios activos mantengan una sesión controlada y temporal.

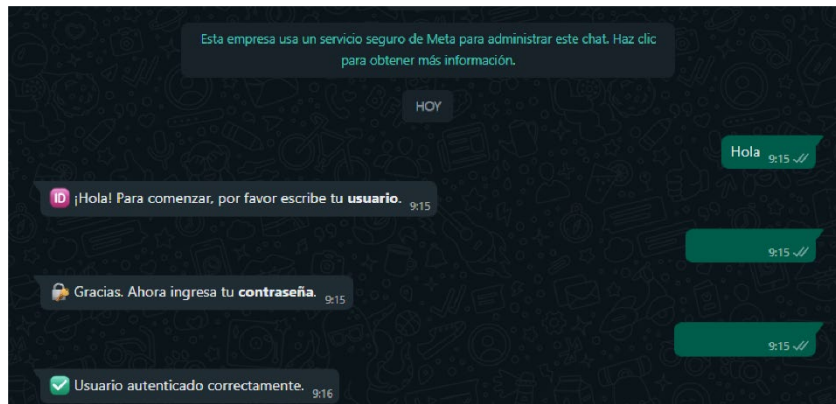
Adicionalmente, se implementó la funcionalidad de aceptación de términos y condiciones, obligando a cada usuario a aceptar explícitamente los términos legales de uso antes de poder interactuar con el sistema.

La arquitectura desarrollada incluyó la configuración del framework NestJS para gestionar las solicitudes entrantes, la integración del webhook de WhatsApp, y el uso de la API de OpenAI para la interpretación inteligente de los mensajes de los usuarios.

Estos avances sentaron la base técnica y de cumplimiento para el despliegue de servicios adicionales en los sprints posteriores.

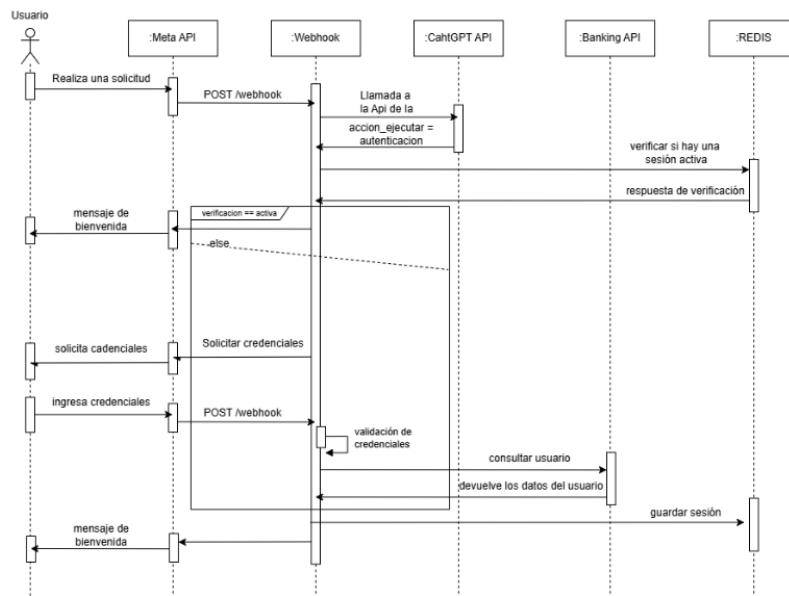
- **RQAU-002** - Implementación de autenticación

En la **Fig. 3** se puede visualizar la interacción del usuario y el chatbot en el proceso de autenticación.



**Fig. 3.** *Proceso de autenticación*

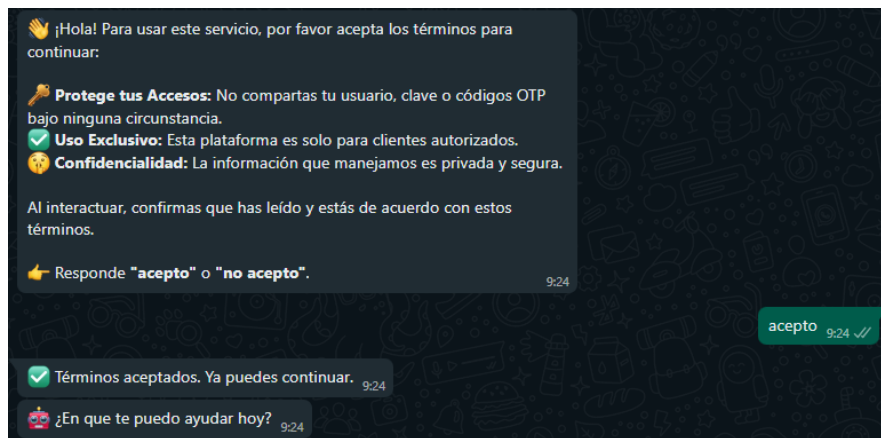
En la **Fig. 4** se ilustra el diagrama de secuencia interno del chatbot en el proceso de autenticación.



**Fig. 4.** *Diagrama de secuencia del Proceso de autenticación*

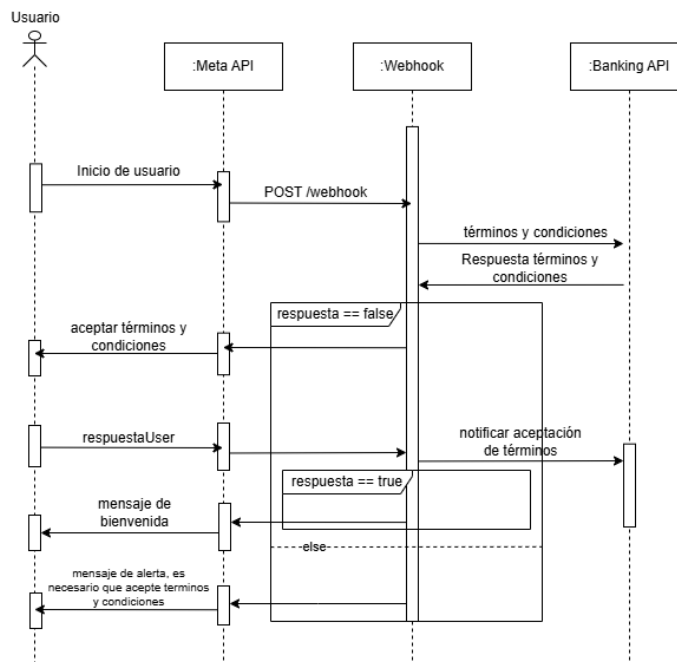
- **RQAU-003** - Gestión de términos y condiciones

En la **Fig. 5** se puede visualizar la interacción del usuario y el chatbot en el proceso de gestión de términos y condiciones.



**Fig. 5.** *Términos y condiciones*

En la **Fig. 6** se ilustra el diagrama de secuencia interno del chatbot en el proceso de aceptación de términos y condiciones.



**Fig. 6.** *Diagrama de secuencia del Proceso de Términos y condiciones*

### 3.3.1 Sprint 2

**Tabla 20**

*Sprint 2*

<b>Historia de Usuario</b>	<b>Actividad</b>	<b>Tarea</b>	<b># Horas</b>
RQTR-001	Consulta de saldo - cuenta de ahorros	Diseñar el flujo conversacional para consulta de saldo de cuenta de ahorros	3
		Integrar la API de la entidad financiera para recuperar saldo de ahorros	3
RQTR-002	Consulta - certificados de aportación	Diseñar flujo para consulta de certificados de aportación	2
		Integrar API de certificados de aportación	2
RQTR-003	Consulta - ahorro programado	Diseñar flujo de consulta de ahorro programado	2
		Integrar API de ahorro programado	2
RQTR-004	Consulta - depósitos a plazo fijo	Diseñar flujo para consulta de depósitos a plazo fijo	2
		Integrar API de depósitos	2
RQTR-005	Consulta – últimos movimientos	Diseñar flujo para consulta de últimos movimientos	2
		Integrar API de movimientos de cuentas de ahorros	2
RQTR-006	Consulta - créditos	Integrar API de depósitos	2
		Integrar API de créditos activos	3

## Desarrollo del Sprint

Durante el Sprint 2 se desarrollaron los flujos conversacionales que permiten a los usuarios del chatbot consultar información financiera relacionada con sus productos bancarios.

El primer conjunto de funcionalidades implementadas correspondió a la consulta del saldo de cuentas de ahorros, a través de la integración con la API de la entidad financiera, asegurando la recuperación de datos actualizados.

Posteriormente, se habilitaron los flujos de:

- Consulta de certificados de aportación, permitiendo visualizar su valor y estado.
- Consulta de ahorro programado, mostrando el saldo acumulado y el porcentaje de cumplimiento de la meta.
- Consulta de depósitos a plazo fijo, con información sobre la fecha de vencimiento y los intereses generados.
- Consulta de últimos movimientos con información sobre la fecha, monto, nombre del beneficiario y descripción corta.
- Consulta del estado de créditos activos, indicando saldo pendiente, fechas de pago y posibles morosidades.

Cada flujo fue desarrollado de forma que garantizara una experiencia de usuario intuitiva, rápida y segura, exigiendo que todas las consultas se realicen únicamente si existe una sesión autenticada activa.

Los diagramas y figuras que se presentan posteriormente describen la interacción entre el chatbot, el usuario y las APIs externas durante la ejecución de cada consulta.

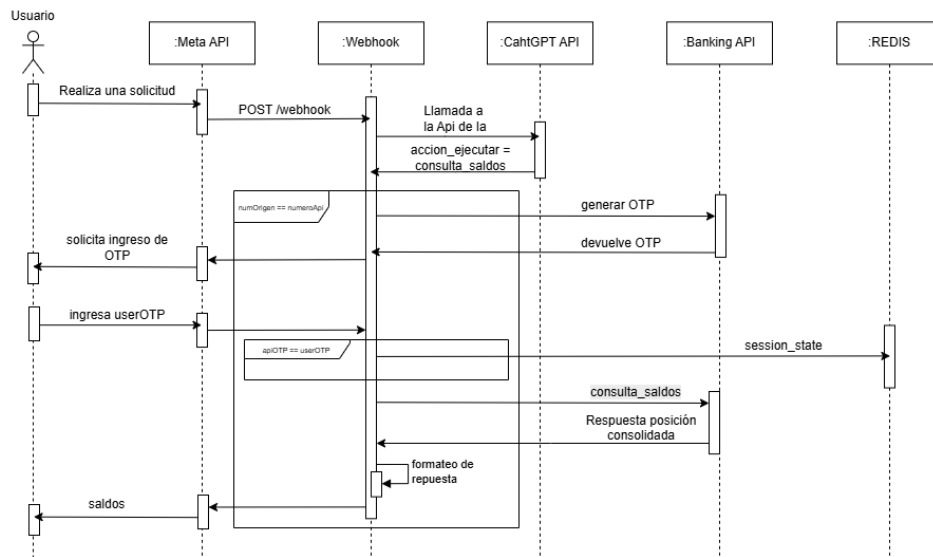
- **RQTR-001** - Consulta de saldo - cuenta de ahorros
- **RQTR-002** - Consulta - certificados de aportación
- **RQTR-003** - Consulta – ahorro programado

En la **Fig. 7** se puede visualizar la interacción del usuario y el chatbot en el proceso de consultas de saldos, certificados de aportación y ahorro programado.



**Fig. 7.** Consulta de saldos

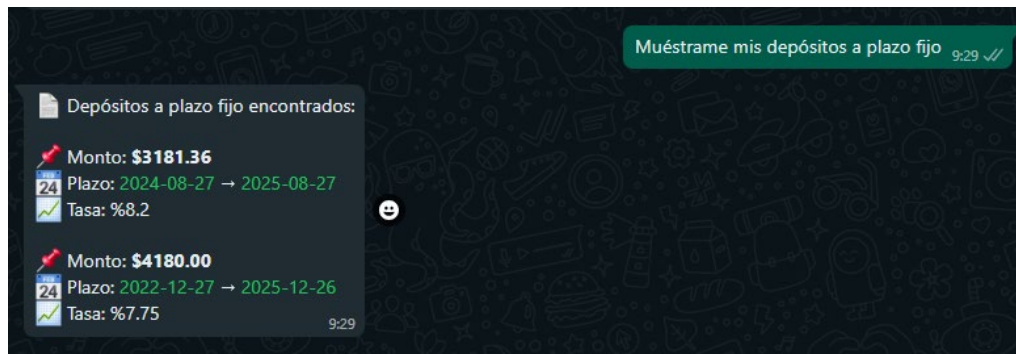
El en la **Fig. 8** se ilustra el diagrama de secuencia interno del chatbot para consultas de saldos.



**Fig. 8.** Diagrama de secuencia del proceso de Consulta de saldos

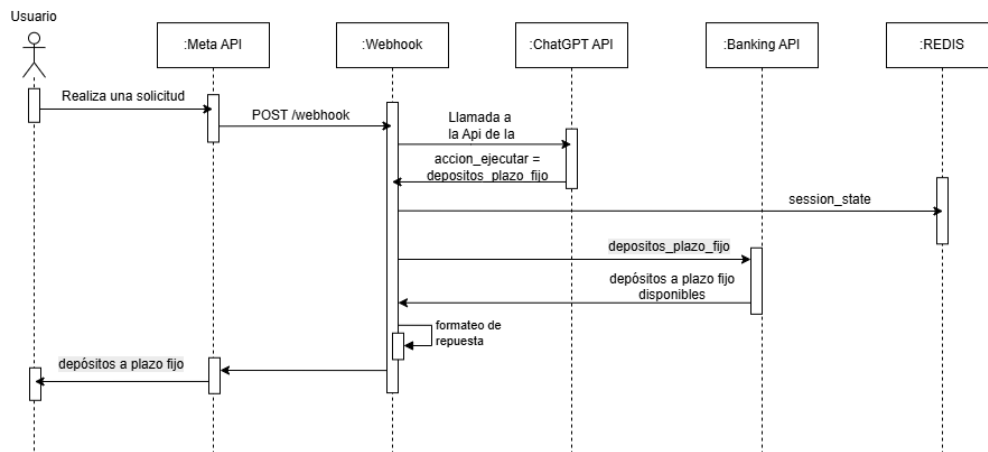
- **RQTR-004** - Consulta - depósitos a plazo fijo

En la **Fig. 9** se puede visualizar la interacción del usuario y el chatbot en el proceso de consulta de depósitos a plazo fijo.



**Fig. 9.** Consulta - depósitos a plazo fijo

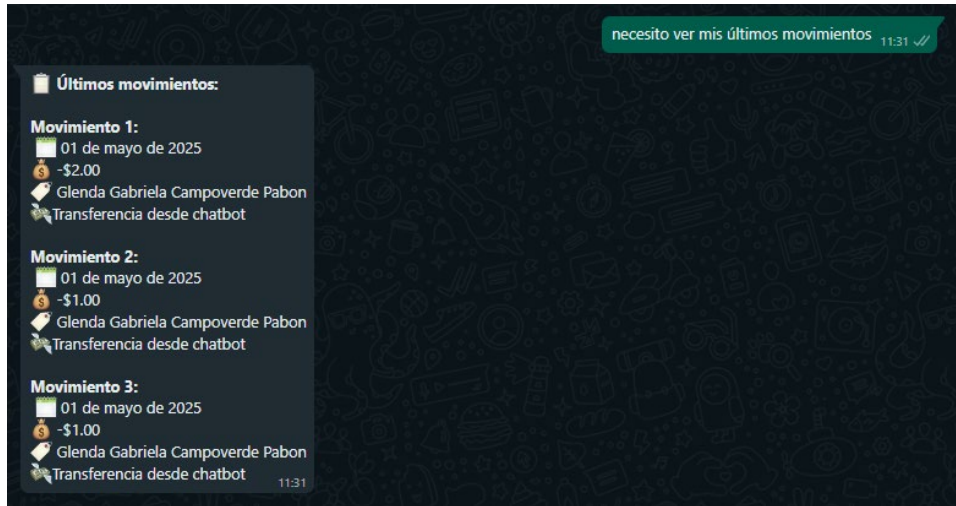
En la **Fig. 10** se ilustra el diagrama de secuencia interno del chatbot para consultas de depósitos a plazo fijo.



**Fig. 10.** Diagrama de secuencia del Proceso de Consulta - depósitos a plazo fijo

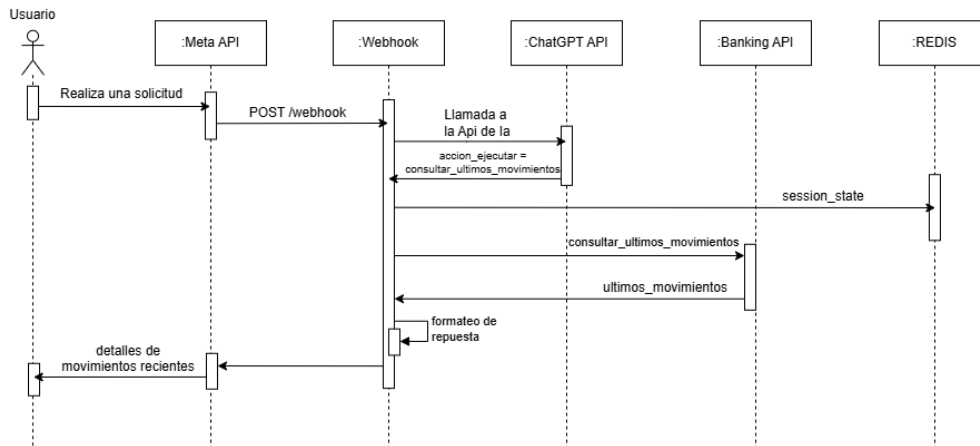
- **RQTR-005** - Consulta – últimos movimientos

En la **Fig. 11** se puede visualizar la interacción del usuario y el chatbot en el proceso de consulta de últimos movimientos.



**Fig. 11.** Consulta – últimos movimientos

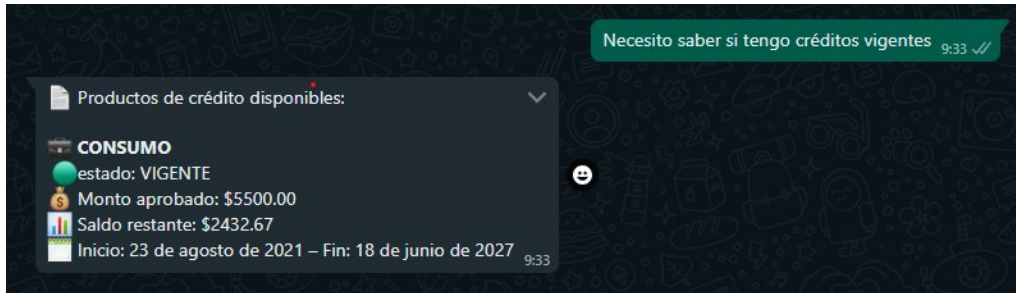
En la **Fig. 12** se ilustra el diagrama de secuencia interno del chatbot para consultas de últimos movimientos.



**Fig. 12.** Diagrama de secuencia del Proceso de Consulta – Últimos Movimientos

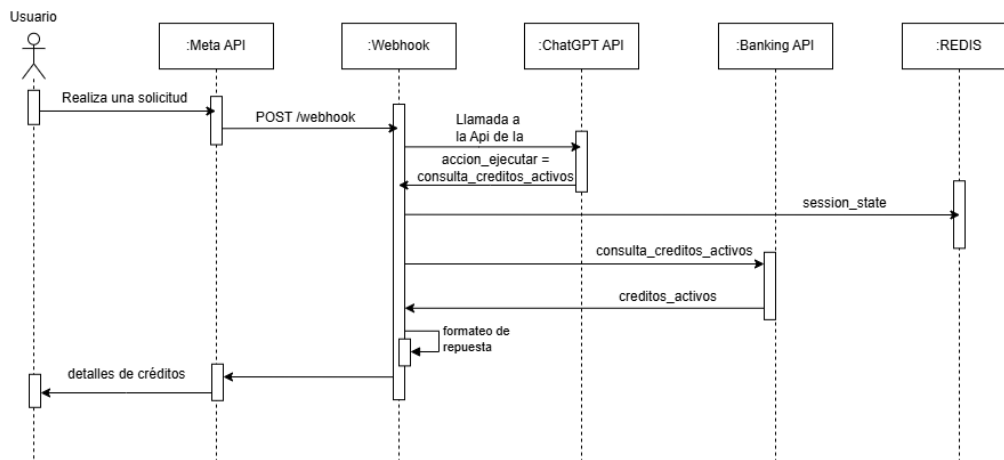
- **RQTR-006** - Consulta - créditos

En la **Fig. 13** se puede visualizar la interacción del usuario y el chatbot en el proceso de consulta de créditos activos.



**Fig. 13.** Consulta – créditos

En la **Fig. 14** se ilustra el diagrama de secuencia interno del chatbot para consultas de últimos movimientos.



**Fig. 14.** Diagrama de secuencia del Proceso de Consulta – créditos

### 3.3.1 Sprint 3

**Tabla 21**

*Sprint 3*

<b>Historia de Usuario</b>	<b>Actividad</b>	<b>Tarea</b>	<b># Horas</b>
RQTR-007	Transferencias internas	Diseñar flujo conversacional para transferencias internas entre cuentas propias	3
		Integrar API de transferencias internas y validar saldo disponible	4
		Programar confirmación de transacción y generación de comprobante	3
RQTR-008	Transferencias interbancarias	Diseñar flujo de transferencia hacia cuentas de otros bancos	3
		Integrar API interbancaria solicitando datos del banco destino y validando la transacción	4
		Implementar verificación mediante código de seguridad (OTP)	3

#### **Desarrollo del Sprint**

Durante el Sprint 3 se desarrollaron las funcionalidades que permiten a los usuarios realizar transferencias de fondos desde el chatbot bancario, tanto entre cuentas propias dentro de la misma entidad como hacia cuentas de terceros en otros bancos.

En primer lugar, se diseñó e implementó el flujo de transferencias internas, asegurando que los usuarios pudieran seleccionar la cuenta origen y destino, validar el saldo disponible, ingresar el monto a transferir y confirmar la operación.

Posteriormente, se desarrolló el flujo de transferencias interbancarias, el cual permite al usuario seleccionar la entidad financiera de destino, ingresar los datos necesarios de la cuenta receptora y confirmar la transacción. Para reforzar la seguridad en este tipo de transferencias, se integró un mecanismo de validación por código de seguridad (OTP) que debe ser ingresado por el usuario antes de completar la operación.

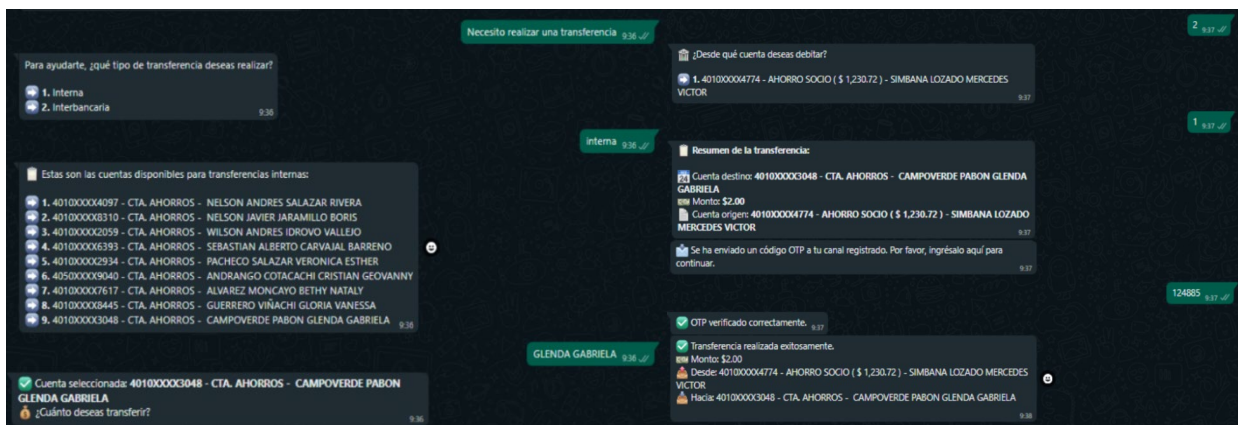
En ambos casos, las transacciones fueron configuradas para generar un comprobante de transferencia exitoso, que puede ser enviado al usuario como confirmación dentro del flujo conversacional.

La autenticación previa del usuario y la aceptación de términos y condiciones continuaron siendo requisitos indispensables para el acceso a estas funcionalidades, garantizando así la seguridad y la integridad de las operaciones bancarias realizadas a través del chatbot.

Las figuras y diagramas de secuencia presentados a continuación describen el flujo de implementación de ambas funcionalidades.

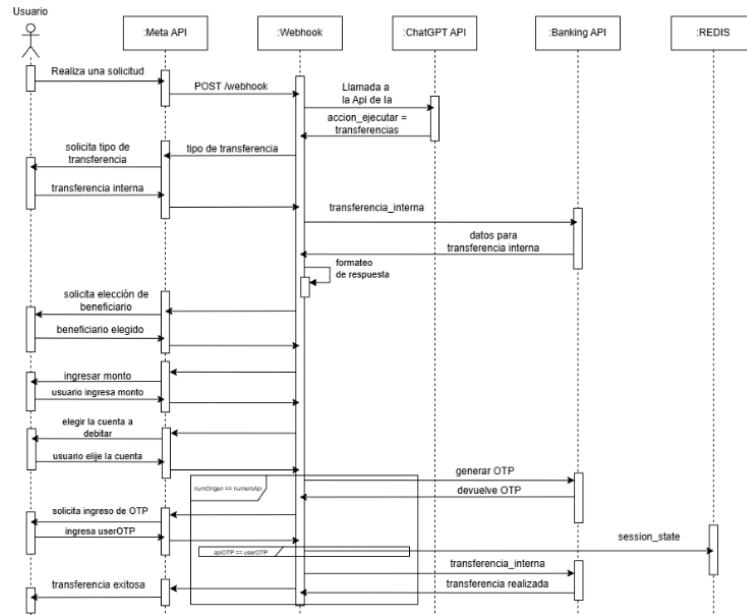
- **RQTR-007 - Transferencia interna**

En la **Fig. 15** se puede visualizar la interacción del usuario y el chatbot para realizar el proceso de una transferencia interna.



**Fig. 15. Transferencias internas**

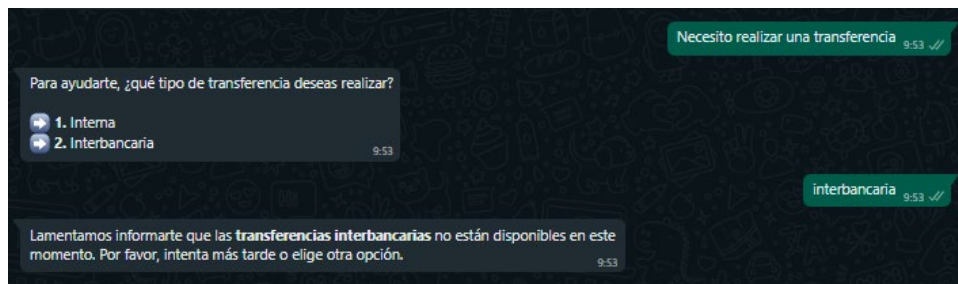
En la **Fig. 16** se ilustra el diagrama de secuencia interno del chatbot para realizar una transferencia interna.



**Fig. 16.** Diagrama de secuencia del Proceso de Trasterferencia interna

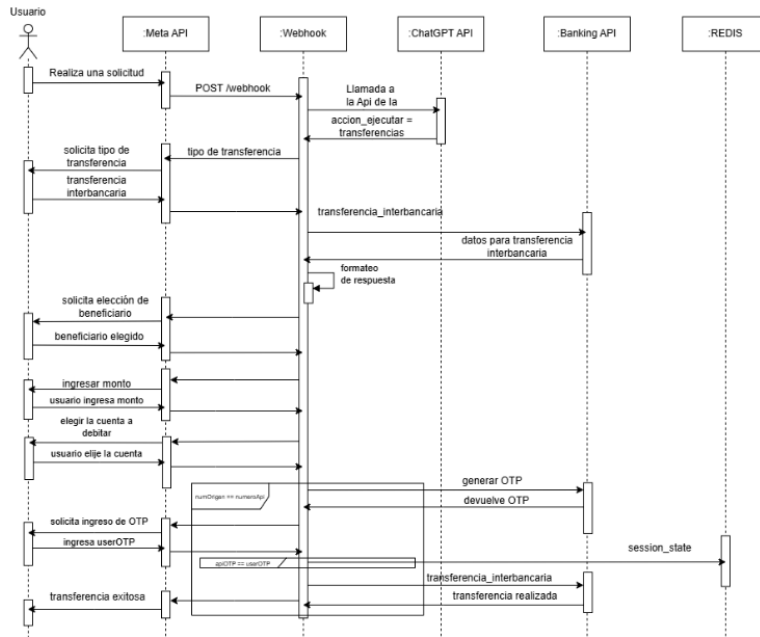
- **RQTR-008** - Transferencia interbancaria

En la **Fig. 17** se puede visualizar la interacción del usuario y el chatbot para realizar el proceso de una transferencia interbancaria.



**Fig. 17.** Transferencia interbancaria

En la **Fig. 18** se ilustra el diagrama de secuencia interno del chatbot para realizar una transferencia interbancaria.



**Fig. 18.** Diagrama de secuencia del Proceso de Tránsito interbancaria

### 3.3.1 Sprint 4

**Tabla 22**

*Sprint 4*

Historia de Usuario	Actividad	Tarea	# Horas
RQTR-009	Pago de tarjeta de crédito	Diseñar flujo conversacional para consultar saldo pendiente de tarjeta	3
		Implementar opción de selección de monto a pagar (total o parcial)	3
		Integrar API para procesar el pago de tarjeta de crédito	3
RQTR-010	Pago de servicios básicos	Diseñar flujo para mostrar facturas pendientes (agua, luz, teléfono, internet, TV)	3
		Implementar selección de facturas y procesamiento de pago	3
		Integrar API de servicios básicos para confirmar pagos realizados	3

#### **Desarrollo del Sprint**

Durante el Sprint 4 se completaron dos funcionalidades esenciales que permiten a los usuarios del chatbot realizar pagos directamente desde la plataforma de mensajería.

La primera funcionalidad implementada fue el pago de tarjetas de crédito, habilitando un flujo conversacional mediante el cual el usuario puede consultar su saldo pendiente, seleccionar si desea realizar un pago total o parcial, e iniciar el proceso de pago de manera segura. El chatbot integra la API bancaria correspondiente para registrar la transacción y confirmar el resultado al usuario.

La segunda funcionalidad correspondió al pago de servicios básicos, donde se desarrolló un flujo que permite al usuario visualizar las facturas pendientes de servicios como agua, luz, teléfono, internet y televisión. El usuario puede seleccionar cuáles facturas desea pagar y confirmar

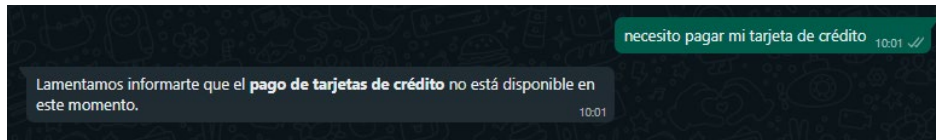
la transacción a través del chatbot. Para ello, se integraron los servicios de consulta y pago disponibles en la API de la entidad financiera.

En ambas funcionalidades, se garantizó que solo usuarios autenticados y que hubieran aceptado los términos y condiciones pudieran realizar transacciones, manteniendo los estándares de seguridad y cumplimiento exigidos.

Las figuras y diagramas de secuencia que se presentan a continuación describen en detalle los flujos conversacionales y la interacción con las APIs externas durante el proceso de pagos.

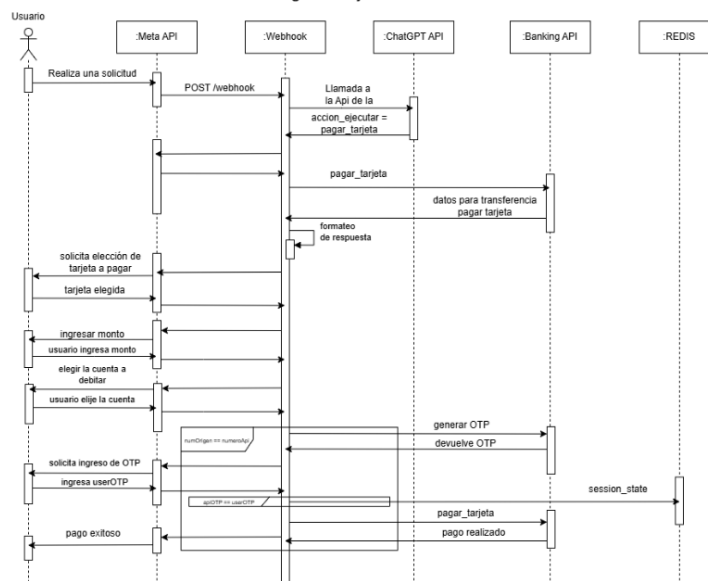
- **RQTR-009** - Pago de tarjeta de crédito

En la **Fig. 19** se puede visualizar la interacción del usuario y el chatbot en el proceso de pago de tarjeta de crédito.



**Fig. 19.** Pago de tarjeta de crédito

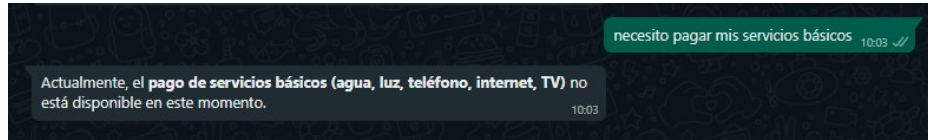
En la **Fig. 20** se ilustra el diagrama de secuencia interno del chatbot para realizar el pago de tarjetas crédito.



**Fig. 20.** Diagrama de secuencia del Proceso de Pago de tarjeta de crédito

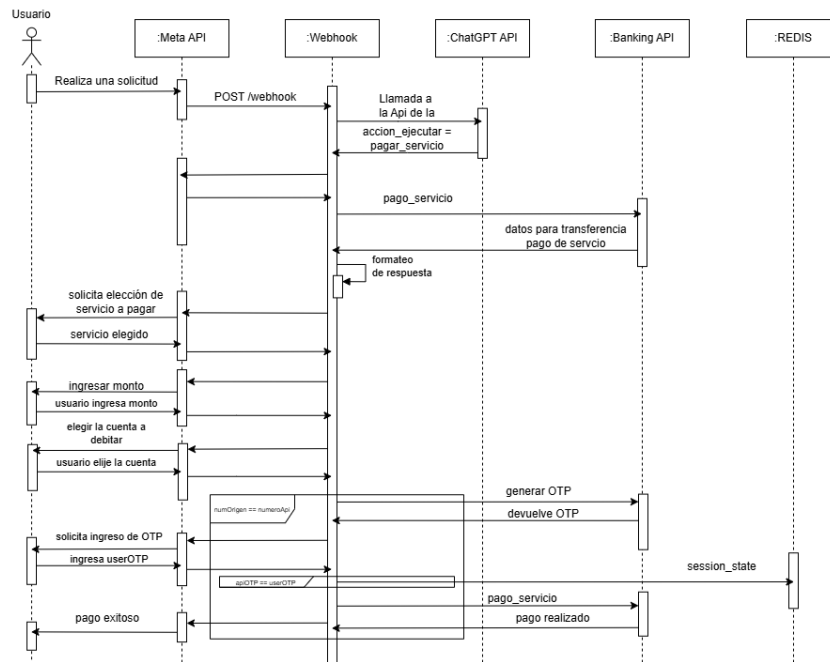
- **RQTR-010** - Pago de servicios básicos

En la **Fig. 21** se puede visualizar la interacción del usuario y el chatbot en el proceso de pago de servicios básicos.



**Fig. 21.** Pago de servicios básicos

En la **Fig. 22** se ilustra el diagrama de secuencia interno del chatbot para realizar el pago de servicios básicos.



**Fig. 22.** Diagrama de secuencia del Proceso de Pago de servicios básicos

### 3.4 Infraestructura tecnológica y herramientas utilizadas

Durante el desarrollo del proyecto se utilizó una arquitectura tecnológica cuidadosamente estructurada, que permite integrar múltiples componentes para el manejo eficiente de información y comunicación entre sistemas. Se han incorporado servidores dedicados tanto para la gestión de datos como para la ejecución de aplicaciones, además de herramientas de desarrollo modernas que facilitan la programación y mantenimiento del sistema.

Asimismo, el uso de tecnologías intermedias como Redis permite gestionar sesiones de forma optimizada, y la conexión con APIs externas amplía la funcionalidad del proyecto, facilitando la interoperabilidad con plataformas de mensajería, entidades financieras y servicios de inteligencia artificial. En la siguiente tabla se describen los principales recursos tecnológicos empleados.

**Tabla 23**

*Infraestructura tecnológica y herramientas utilizadas*

<b>Categoría</b>	<b>Descripción</b>
Servidor de base de datos	Sybase ASE 15 ejecutándose sobre Linux CentOS
Servidor de aplicaciones	Internet Information Services (IIS) con ASP.NET Framework 4.5.2 en Windows Server 2022 (64 bits)
Framework backend	Node.js con el framework NestJS
Entorno de desarrollo	Visual Studio Code
Sistema de caché y sesiones	Redis
APIs externas utilizadas	API de Meta para WhatsApp Business API de la entidad financiera API de OpenAI para ChatGPT

### **3.5 Justificación de Funcionalidades No Implementadas**

Dentro de la planificación inicial del sistema, se contempló el desarrollo e integración de las siguientes funcionalidades: transferencias interbancarias, pagos de tarjetas de crédito y pagos de servicios básicos (agua, luz, teléfono, internet, entre otros). Estas características formaban parte del alcance funcional previsto y fueron consideradas tanto en el diseño de la arquitectura del chatbot como en el levantamiento de requisitos.

Sin embargo, durante el proceso de implementación, no fue posible desarrollarlas debido a limitaciones propias del entorno de pruebas utilizado, el cual opera de forma aislada y controlada para proteger la integridad de los sistemas reales. Estas funcionalidades requieren necesariamente la conexión con APIs externas y el acceso a servicios de producción que manejan información altamente sensible, lo cual fue restringido por políticas de seguridad y confidencialidad establecidas por la institución.

Dado que la infraestructura proporcionada no permite realizar integraciones externas ni pruebas sobre datos reales, se priorizó el cumplimiento de estándares de seguridad, evitando cualquier riesgo de exposición de información. No obstante, se dejó habilitado el soporte técnico y arquitectónico para que estas funcionalidades puedan ser integradas en el futuro, una vez que se disponga de un entorno controlado con acceso a servicios externos autorizados.

## CAPITULO IV

### RESULTADOS Y ANÁLISIS

Este capítulo presenta los resultados obtenidos durante las fases de desarrollo y validación del bot bancario con inteligencia artificial generativa, detallando el proceso de implementación, los hallazgos clave de rendimiento y fiabilidad, y el análisis de la evaluación del sistema en un entorno de pruebas controlado.

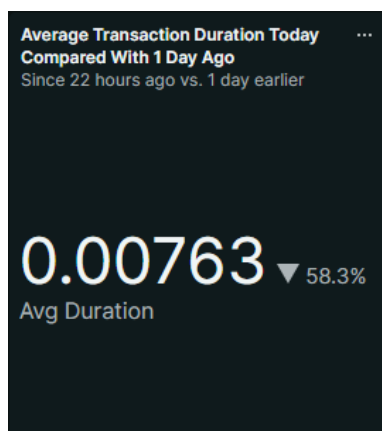
#### 4.1 Evaluación del Rendimiento y Fiabilidad del Bot Bancario

La evaluación del rendimiento y la fiabilidad del bot se realizó a través de la monitorización continua utilizando New Relic APM. Se llevaron a cabo pruebas controladas, donde un único usuario de prueba simuló una variedad de interacciones para generar métricas representativas del comportamiento del sistema bajo condiciones operativas.

##### 4.1.1 Métricas de Rendimiento y Latencia

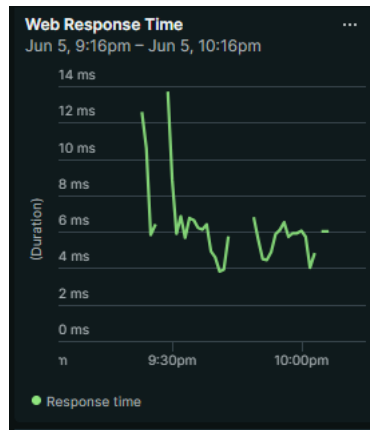
Los resultados de New Relic evidenciaron una alta eficiencia operativa:

- **Tiempo Promedio de Transacción:** El bot exhibió un tiempo promedio de duración de las transacciones de 7.63 milisegundos (ms). Este valor, extremadamente bajo, garantiza una experiencia de usuario fluida y sin demoras perceptibles, lo cual es crucial para la confianza en un servicio bancario digital. Este tiempo representa una mejora significativa del 58.3% en comparación con mediciones iniciales, reflejando optimizaciones en el ciclo de desarrollo (ver Fig. 23).



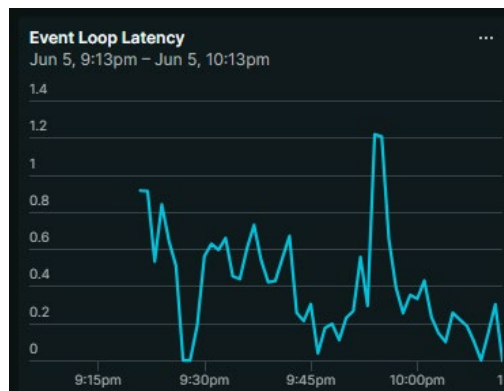
**Fig. 23.** *Tiempo Promedio de Transacción*

- **Tiempo de Respuesta Web:** Los tiempos de respuesta web fluctuaron entre 3 ms y picos máximos de 13 ms, con la mayoría de las interacciones completándose consistentemente por debajo de los 6 ms (ver **Fig. 24**). Estos valores confirman la capacidad del bot para entregar respuestas casi instantáneas a las consultas de los usuarios.



**Fig. 24.** *Tiempo de Respuesta Web*

- **Latencia del Event Loop:** La latencia del Event Loop de Node.js se mantuvo en niveles excepcionalmente bajos, con picos transitorios que no superaron los 1.2 ms (ver **Fig. 25**). Este hallazgo es un indicador crítico de la salud del proceso Node.js, demostrando que el hilo principal de ejecución no se bloquea, permitiendo el procesamiento eficiente de operaciones asíncronas y manteniendo la reactividad del sistema.

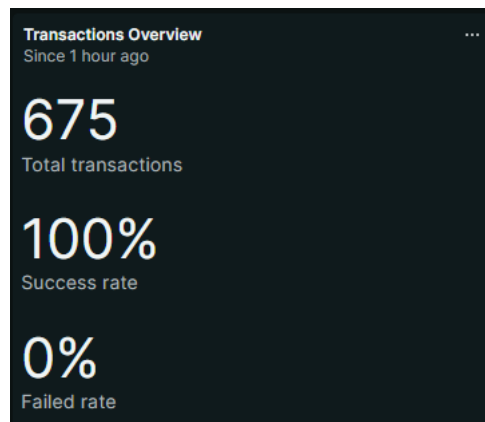


**Fig. 25.** *Latencia del Event Loop*

### 4.1.2 Fiabilidad de Operaciones

La fiabilidad del bot fue uno de los resultados más sobresalientes:

- **Tasa de Éxito del 100%:** Durante el período de monitoreo, el sistema procesó un total de 675 transacciones HTTP con una tasa de éxito del 100% y una tasa de fallos del 0% (ver **Fig. 26**). Este resultado es fundamental y valida la robustez de los flujos de interacción del bot y la correcta comunicación con los servicios bancarios simulados. La ausencia total de errores HTTP (códigos 4xx o 5xx) subraya la estabilidad de la implementación.

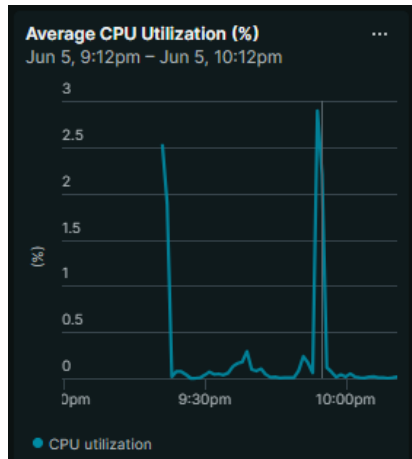


**Fig. 26.** *Resumen de Transacciones*

### 4.1.3 Consumo y Gestión de Recursos

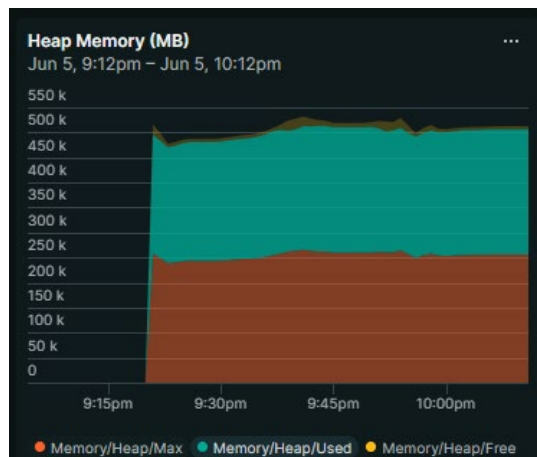
La eficiencia en el uso de los recursos computacionales se demostró a través de las siguientes métricas:

- **Utilización de CPU:** El consumo promedio de CPU se mantuvo en niveles muy bajos (inferiores al 0.5%) durante los períodos de inactividad. Se observaron picos transitorios que alcanzaron entre el 2.5% y el 3% de utilización de CPU, los cuales estuvieron directamente correlacionados con los momentos de interacción activa del usuario de prueba (ver **Fig. 27**). Esto indica una asignación de recursos optimizada bajo demanda.



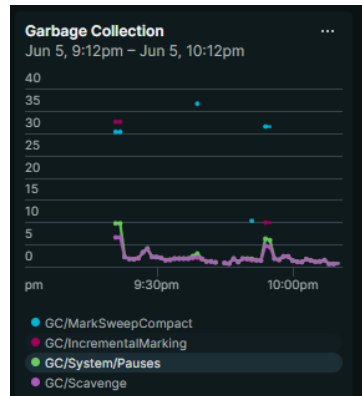
**Fig. 27.** *Utilización Promedio de CPU*

- Uso de Memoria Heap:** La memoria Heap utilizada por el proceso Node.js se estabilizó en aproximadamente 250 MB después del arranque inicial, con un límite máximo operativo entre 450-500 MB (ver **Fig. 28**). La estabilidad observada en el uso de la memoria, sin patrones de crecimiento continuo, sugiere la ausencia de fugas de memoria significativas durante las operaciones de prueba.



**Fig. 28.** *Memoria Heap*

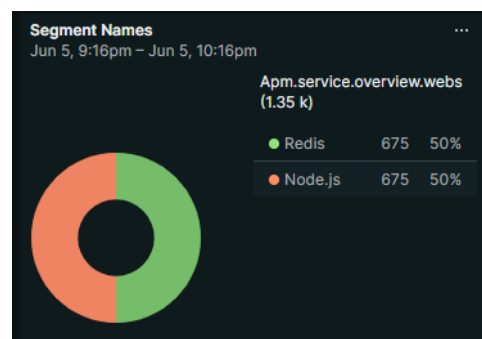
- Recolección de Basura (Garbage Collection):** La actividad de recolección de basura (GC) fue observada, con eventos como GC/MarkSweepCompact y GC/Scavenge activos durante los períodos de interacción. Las pausas del sistema asociadas a GC fueron mínimas y de corta duración (ver **Fig. 29**), lo cual es indicativo de un manejo eficiente de la memoria que no impacta negativamente la reactividad del bot.



**Fig. 29.** *Recolección de Basura*

#### 4.1.4 Distribución del Tiempo de Procesamiento por Componente

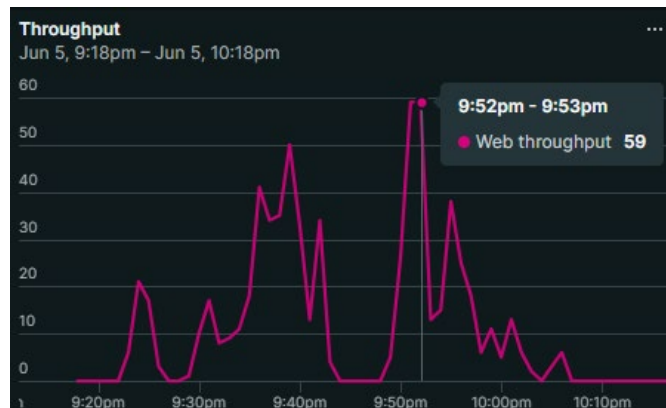
Un análisis granular de los segmentos de transacciones reveló una distribución equitativa del tiempo de procesamiento entre los componentes clave del sistema (ver **Fig. 30**). Específicamente, el 50% del tiempo total de la transacción fue dedicado a la lógica interna y el procesamiento del Webhook Node.js, mientras que el 50% restante correspondió a las interacciones con el servicio Redis. Este hallazgo subraya la interdependencia de los componentes y la criticidad del rendimiento de Redis en la duración total de la transacción, ya que es probable que sea utilizado para la gestión de estado conversacional o caché.



**Fig. 30.** *Distribución de Segmentos de Transacción*

### 4.1.5 Capacidad de Throughput

A pesar de las condiciones de prueba con un solo usuario, el bot demostró una notable capacidad de throughput, alcanzando picos de hasta 59 transacciones por minuto (ver **Fig. 31**). Este alto rendimiento en términos de volumen de transacciones por unidad de tiempo, combinado con los bajos tiempos de respuesta, sugiere una arquitectura escalable y eficiente, capaz de manejar cargas de trabajo futuras significativamente mayores.



**Fig. 31.** *Throughput Web*

## 4.2 Análisis de Resultados

Los resultados presentados en este capítulo demuestran la viabilidad técnica y el alto rendimiento del bot bancario desarrollado. La elección de NestJS y Node.js se valida como una decisión acertada para un sistema que requiere alta reactividad y eficiencia en el manejo de operaciones asíncronas. La integración exitosa con OpenAI para PLN reduce la complejidad de la gestión de intenciones y entidades, permitiendo al bot interpretar y responder a consultas de manera más natural.

La fiabilidad del 100% en las transacciones es un pilar fundamental para cualquier aplicación bancaria y se logró en este entorno de prueba, lo que indica un diseño robusto en el manejo de flujos de negocio y comunicaciones con APIs. La distribución equitativa del tiempo de procesamiento entre la lógica de Node.js y las operaciones de Redis resalta la importancia de monitorizar y optimizar todos los componentes del sistema, no solo el servicio principal.

Si bien las pruebas se realizaron con un único usuario, la generación de un throughput de casi 60 transacciones por minuto es un indicador clave de que el sistema puede manejar un volumen de interacciones mucho mayor del que un solo usuario podría generar manualmente, sugiriendo una buena base para la escalabilidad. La gestión eficiente de recursos (CPU y memoria) también es crucial para el despliegue en entornos de producción, donde la optimización de costos y la estabilidad operativa son prioritarias.

Para futuras fases, será esencial realizar pruebas de carga y estrés para evaluar el comportamiento del bot bajo una concurrencia elevada y volúmenes masivos de transacciones. Además, la implementación de una instrumentación más granular en New Relic permitirá desglosar el rendimiento por tipo de transacción bancaria específica (ej., "tiempo promedio de una transferencia", "tasa de éxito de consulta de saldo"), lo que proporcionará insights aún más valiosos para la mejora continua.

## CONCLUSIONES

- El sistema alcanzó un tiempo promedio de respuesta de 7.63 milisegundos, con una latencia del event loop menor a 1.2 ms y tiempos web por debajo de los 6 ms en la mayoría de las interacciones. Esto evidencia una experiencia fluida y de calidad para el usuario, aspecto clave en aplicaciones bancarias.
- El bot procesó 675 transacciones HTTP con una tasa de éxito del 100%, sin registrar errores del tipo 4xx o 5xx, lo que demuestra la robustez de la arquitectura, el manejo correcto de flujos y la integración adecuada con APIs bancarias simuladas.
- El uso de CPU se mantuvo en promedio por debajo del 0.5%, con picos del 3%, y la memoria Heap fue estable entre 250 y 500 MB, sin signos de fugas. La actividad del recolector de basura (GC) fue mínima y no afectó la reactividad del sistema.
- La distribución equitativa del tiempo de procesamiento entre la lógica Node.js y el servicio Redis evidencia una arquitectura balanceada, preparada para escalar y soportar una mayor carga concurrente sin comprometer el rendimiento.
- El uso de inteligencia artificial generativa (IA-G), procesamiento de lenguaje natural (NLP), frameworks modernos como NestJS, y la integración con OpenAI, permitió automatizar consultas y transacciones con respuestas naturales y contextuales, acercando la banca electrónica a un servicio más humano.
- La implementación del chatbot bancario no solo cumplió con los requerimientos técnicos y de seguridad establecidos por la normativa financiera ecuatoriana, sino que también demostró ser factible a nivel presupuestario, con un costo controlado y el uso de tecnologías maduras.
- Esta solución tecnológica tiene el potencial de acercar servicios bancarios a sectores tradicionalmente excluidos, gracias a su disponibilidad 24/7, su interfaz conversacional amigable y su capacidad de integrarse con plataformas como WhatsApp.

## RECOMENDACIONES

- Para validar la escalabilidad del sistema en producción, se recomienda realizar pruebas de carga y estrés que simulen interacción concurrente, permitiendo detectar posibles cuellos de botella bajo alta demanda.
- Se sugiere desagregar el monitoreo de rendimiento en categorías como consultas, transferencias o pagos, con el fin de obtener indicadores específicos (por ejemplo, tasa de éxito por función), lo cual permitirá una mejora continua más enfocada.
- Implementar respaldo de Redis, balanceo de carga, y monitoreo de salud en tiempo real es fundamental para garantizar continuidad del servicio ante fallos en entornos productivos bancarios.
- Para incrementar el alcance, se recomienda adaptar el chatbot a otras plataformas como Telegram o asistentes de voz, y validar su rendimiento en dispositivos móviles de distinta gama y conexión.
- Es recomendable acompañar el despliegue con campañas de educación digital, manuales interactivos o tutoriales, especialmente para usuarios con baja alfabetización tecnológica.
- La integración de herramientas como New Relic debe mantenerse activa en el entorno real para detectar degradaciones de rendimiento, realizar ajustes dinámicos y asegurar una experiencia estable y confiable.
- Finalmente, se sugiere escalar el chatbot hacia funciones más avanzadas como asesoría financiera, gestión de reclamos, apertura de productos o interacción con otros sistemas empresariales (ERP, CRM).

## REFERENCIAS BIBLIOGRÁFICAS

- Adamopoulou, E., & Moussiades, L. (2020). Chatbots: History, technology, and applications. *Machine Learning with Applications*, 2, 100006. <https://doi.org/10.1016/j.mlwa.2020.100006>
- Agnihotri, A., & Bhattacharya, S. (2024). Chatbots' effectiveness in service recovery. *International Journal of Information Management*, 76, 102679. <https://doi.org/10.1016/j.ijinfomgt.2023.102679>
- Ahmad, E. (2024). Unlocking the Potential: A Comprehensive Systematic Review of ChatGPT in Natural Language Processing Tasks. *CMES - Computer Modeling in Engineering and Sciences*, 141(1), 43-85. <https://doi.org/10.32604/cmes.2024.052256>
- Ahmed, T. M., Bezemer, C.-P., Chen, T.-H., Hassan, A. E., & Shang, W. (2016). Studying the Effectiveness of Application Performance Management (APM) Tools for Detecting Performance Regressions for Web Applications: An Experience Report. *2016 IEEE/ACM 13th Working Conference on Mining Software Repositories (MSR)*, 1-12. <https://ieeexplore.ieee.org/abstract/document/7832882>
- Alagarsamy, S., & Mehroliya, S. (2023). Exploring chatbot trust: Antecedents and behavioural outcomes. *Heliyon*, 9(5), e16074. <https://doi.org/10.1016/j.heliyon.2023.e16074>
- Araujo, J. J. G., Cagua, K. J. A., Burgos, R. T. Y., Muñiz, F. V. C., & Salazar, J. A. G. (2024). Impacto de la digitalización en el sector bancario ecuatoriano: Transformaciones, desafíos y oportunidades para el futuro financiero: Impact of digitalization in the Ecuadorian banking sector: Transformations, challenges and opportunities for the financial future. *LATAM Revista Latinoamericana de Ciencias Sociales y Humanidades*, 5(5), Article 5. <https://doi.org/10.56712/latam.v5i5.2658>
- Baabdullah, A. M., Alalwan, A. A., Algharabat, R. S., Metri, B., & Rana, N. P. (2022). Virtual agents and flow experience: An empirical examination of AI-powered chatbots. *Technological Forecasting and Social Change*, 181, 121772. <https://doi.org/10.1016/j.techfore.2022.121772>
- Baldassarre, S., Bruno, G., Piccolo, C., & Ruiz-Hernández, D. (2024). Multi-channel distribution in the banking sector and the branch network restructuring. *Expert Systems with Applications*, 238, 122294. <https://doi.org/10.1016/j.eswa.2023.122294>

- Banco Bolivariano*. (s. f.). WhatsApp Business. Recuperado 27 de enero de 2025, de [https://business.whatsapp.com/resources/success-stories/banco-bolivariano?lang=es\\_LA](https://business.whatsapp.com/resources/success-stories/banco-bolivariano?lang=es_LA)
- Casheekar, A., Lahiri, A., Rath, K., Prabhakar, K. S., & Srinivasan, K. (2024). A contemporary review on chatbots, AI-powered virtual conversational agents, ChatGPT: Applications, open challenges and future research directions. *Computer Science Review*, 52, 100632. <https://doi.org/10.1016/j.cosrev.2024.100632>
- Documentation | NestJS - A progressive Node.js framework*. (2025). Documentation | NestJS - A Progressive Node.js Framework. <https://docs.nestjs.com>
- Du, K., Zhao, Y., Mao, R., Xing, F., & Cambria, E. (2025). Natural language processing in finance: A survey. *Information Fusion*, 115, 102755. <https://doi.org/10.1016/j.inffus.2024.102755>
- Estrada Velasco, M. V., Núñez Villacis, J. A., Saltos Chávez, P. R., & Cunuhay Cuchiye, W. C. (2021). Revisión Sistemática de la Metodología Scrum para el Desarrollo de Software. *Dominio de las Ciencias*, 7(Extra 4), 54. <https://doi.org/DOI:http://dx.doi.org/10.23857/dc.v7i4.2429>
- Gupta, R., Nair, K., Mishra, M., Ibrahim, B., & Bhardwaj, S. (2024). Adoption and impacts of generative artificial intelligence: Theoretical underpinnings and research agenda. *International Journal of Information Management Data Insights*, 4(1), 100232. <https://doi.org/10.1016/j.jjime.2024.100232>
- Hasan, S., Godhuli, E. R., Rahman, M. S., & Mamun, M. A. A. (2023). The adoption of conversational assistants in the banking industry: Is the perceived risk a moderator? *Heliyon*, 9(9), e20220. <https://doi.org/10.1016/j.heliyon.2023.e20220>
- Huang, S. Y. B., & Lee, C.-J. (2022). Predicting continuance intention to fintech chatbot. *Computers in Human Behavior*, 129, 107027. <https://doi.org/10.1016/j.chb.2021.107027>
- Janssen, A., Rodríguez Cardona, D., Passlick, J., & Breitner, M. H. (2022). How to Make chatbots productive – A user-oriented implementation framework. *International Journal of Human-Computer Studies*, 168, 102921. <https://doi.org/10.1016/j.ijhcs.2022.102921>
- Jiang, H., Cheng, Y., Yang, J., & Gao, S. (2022). AI-powered chatbot communication with customers: Dialogic interactions, satisfaction, engagement, and customer behavior. *Computers in Human Behavior*, 134, 107329. <https://doi.org/10.1016/j.chb.2022.107329>

- Junta Bancaria del Ecuador. (2012). *Resolución JB-2012-2148*. (Resolución normativa No. JB-2012-2148; Codificación de Resoluciones de la Superintendencia de Bancos y Seguros y de la Junta Bancaria, p. 12). Superintendencia de Bancos y Seguros.
- Kitsios, F., Giatsidis, I., & Kamariotou, M. (2021). Digital Transformation and Strategy in the Banking Sector: Evaluating the Acceptance Rate of E-Services. *Journal of Open Innovation: Technology, Market, and Complexity*, 7(3), 204. <https://doi.org/10.3390/joitmc7030204>
- Kuhrmann, M., Tell, P., Hebig, R., Klünder, J., Münch, J., Linssen, O., Pfahl, D., Felderer, M., Prause, C. R., MacDonell, S. G., Nakatumba-Nabende, J., Raffo, D., Beecham, S., Tüzün, E., López, G., Paez, N., Fontdevila, D., Licorish, S. A., Küpper, S., ... Richardson, I. (2022). What Makes Agile Software Development Agile? *IEEE Transactions on Software Engineering*, 48(9), 3523-3539. *IEEE Transactions on Software Engineering*. <https://doi.org/10.1109/TSE.2021.3099532>
- Lee, K.-W., & Li, C.-Y. (2023). It is not merely a chat: Transforming chatbot affordances into dual identification and loyalty. *Journal of Retailing and Consumer Services*, 74, 103447. <https://doi.org/10.1016/j.jretconser.2023.103447>
- Li, C.-Y., Fang, Y.-H., & Chiang, Y.-H. (2023). Can AI chatbots help retain customers? An integrative perspective using affordance theory and service-domain logic. *Technological Forecasting and Social Change*, 197, 122921. <https://doi.org/10.1016/j.techfore.2023.122921>
- Liu, M., Zhang, H., Xu, Z., & Xu, K. (2024). The fusion of fuzzy theories and natural language processing: A state-of-the-art survey. *Applied Soft Computing*, 162, 111818. <https://doi.org/10.1016/j.asoc.2024.111818>
- Mejía, K. (2017). *REFORMA A LA RESOLUCIÓN No. SEPS-IGT-IR-ISF-ITIC-IGJ-2017-103 DE 23 DE NOVIEMBRE DE 201* (No. SEPS-IGT-IR-ISF-ITIC-IGJ-2017-113; p. 2). Superintendencia de Economía Popular y Solidaria (SEPS).
- Mejía, K. (23/112017). *NORMA DE CONTROL DE LAS SEGURIDADES EN EL USO DE TRANSFERENCIAS ELECTRÓNICAS* (No. SEPS-IGT-IR-ISF-ITIC-IGJ-2017-103; p. 12). Superintendencia de Economía Popular y Solidaria (SEPS).

- Mogaji, E., Balakrishnan, J., Nwoba, A. C., & Nguyen, N. P. (2021). Emerging-market consumers' interactions with banking chatbots. *Telematics and Informatics*, *65*, 101711. <https://doi.org/10.1016/j.tele.2021.101711>
- Naz, S., Shafiq, A., Butt, S. A., Tasneem, R., Pamucar, D., & Gonzalez, Z. C. (2025). Decision-making model for selecting products through online product reviews utilizing natural language processing techniques. *Neurocomputing*, *611*, 128593. <https://doi.org/10.1016/j.neucom.2024.128593>
- Petersson, A. H., Pawar, S., & Fagerstrøm, A. (2023). Investigating the factors of customer experiences using real-life text-based banking chatbot: A qualitative study in Norway. *Procedia Computer Science*, *219*, 697-704. <https://doi.org/10.1016/j.procs.2023.01.341>
- Priya, B., & Sharma, V. (2023). Exploring users' adoption intentions of intelligent virtual assistants in financial services: An anthropomorphic perspectives and socio-psychological perspectives. *Computers in Human Behavior*, *148*, 107912. <https://doi.org/10.1016/j.chb.2023.107912>
- ¿Qué es la gestión del rendimiento de las aplicaciones (APM)? | IBM. (2024, noviembre 13). <https://www.ibm.com/es-es/topics/application-performance-management>
- Rapp, A., Curti, L., & Boldi, A. (2021). The human side of human-chatbot interaction: A systematic literature review of ten years of research on text-based chatbots. *International Journal of Human-Computer Studies*, *151*, 102630. <https://doi.org/10.1016/j.ijhcs.2021.102630>
- Rodríguez, M., Villarreal, A., Granizo, D., Vela, J. S., Santillán, A., & Estrella, N. (2022). *El avance de la banca digital en Ecuador Reporte de transacciones efectuadas por canales bancarios 2019 – 2021* (Reporte de transacciones efectuadas por canales bancarios 2019 – 2021, p. 44) [Informe técnico]. Asociación de Bancos Privados del Ecuador (ASOBANCA). <https://asobanca.org.ec/wp-content/uploads/2022/07/Transacciones-digital.pdf>
- Sarumo López, J. P. G. (2020). Implementación del software APM para monitorear eficientemente las aplicaciones en la Empresa América Móvil Perú S.A.C. *Universidad Peruana de Ciencias e Informática*. <http://repositorio.upci.edu.pe/handle/upci/124>
- Sixto-García, J., López-García, X., Gómez de la Fuente, M. del C., Sixto-García, J., López-García, X., & Gómez de la Fuente, M. del C. (2021). La mensajería instantánea como fuente

informativa en la comunicación organizacional: WhatsApp Business en México y España. *Comunicación y sociedad*, 18. <https://doi.org/10.32870/cys.v2021.7679>

Strode, D., Dingsøy, T., & Lindsjorn, Y. (2022). A teamwork effectiveness model for agile software development. *Empirical Software Engineering*, 27(2), 56. <https://doi.org/10.1007/s10664-021-10115-0>

Supriyono, Wibawa, A. P., Suyono, & Kurniawan, F. (2024). Advancements in natural language processing: Implications, challenges, and future directions. *Telematics and Informatics Reports*, 16, 100173. <https://doi.org/10.1016/j.teler.2024.100173>

## ANEXOS

La Fig. 33 se muestra el entorno de desarrollo integrado en Visual Studio Code, mostrando la configuración esencial del webhook y del puerto 3000 utilizado para la ejecución y pruebas de comunicación con el webhook durante la fase de desarrollo.

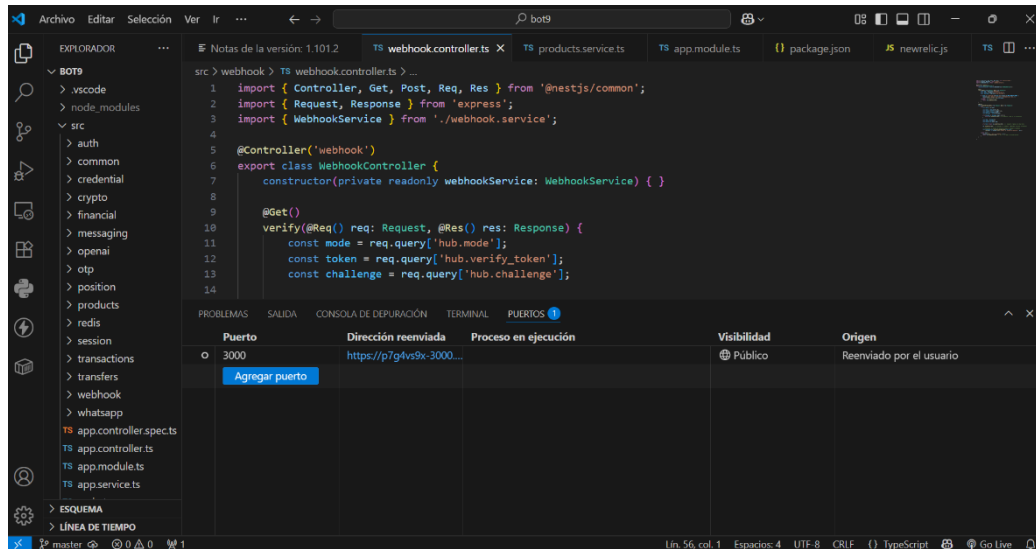


Fig. 32. Configuración del webhook en el entorno de desarrollo

En la Fig. 32 se visualiza el panel de configuración de la API de WhatsApp Business en la plataforma Meta Developers, donde se detalla generación de tokens de acceso temporales y la configuración de los números de teléfono de prueba.

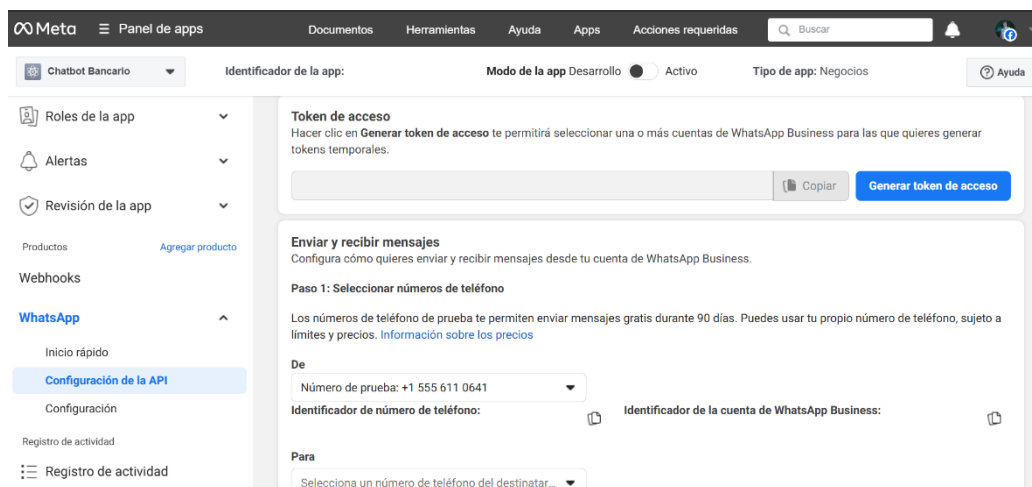
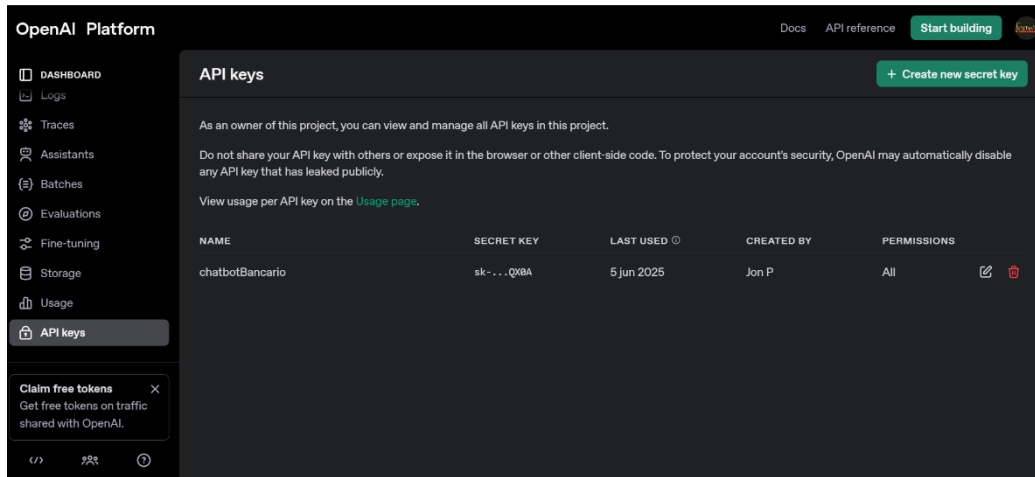


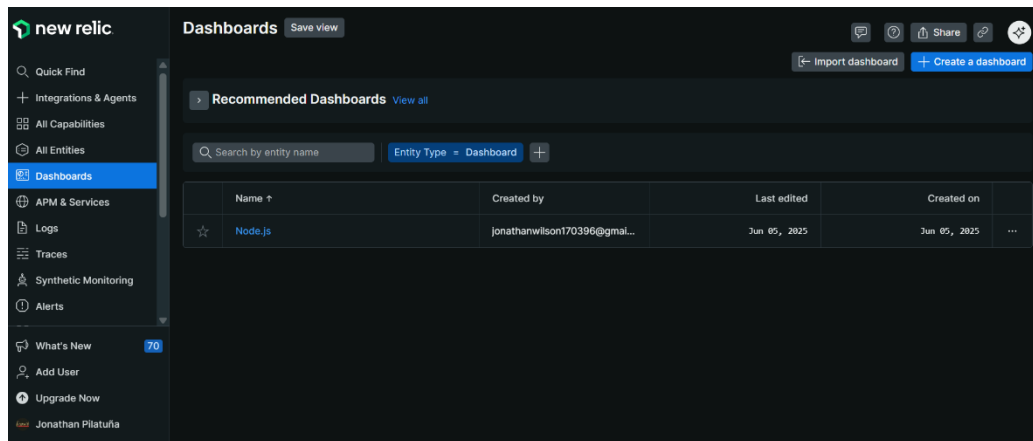
Fig. 33. Panel de configuración de la API de WhatsApp en Meta Developer

La **Fig. 34** presenta la sección de gestión de claves API en la plataforma de OpenAI. Se pueden observar las claves API generadas, su nombre, fecha de último uso y permisos, siendo un componente clave para autenticar las solicitudes realizadas al modelo de lenguaje de OpenAI.



**Fig. 34.** Interfaz de la plataforma OpenAI para la administración de claves de API

En la **Fig. 35** se presenta la interfaz de los dashboards en la plataforma de monitoreo New Relic. Muestra cómo se organizan y visualizan las métricas de rendimiento y los datos operativos del sistema, en este caso, un dashboard específico para Node.js, esencial para la supervisión del chatbot.



**Fig. 35.** Panel de control de New Relic