



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA INDUSTRIAL
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

**AGENTES INTELIGENTES BASADOS EN LARGE LANGUAGE MODELS
PARA LA ATENCIÓN AUTOMATIZADA EN CANAL CONVERSACIONAL
ONLINE DE SERVICIO AL CLIENTE DE UNA EMPRESA FINANCIERA**

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL INDUSTRIAL

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN

FELIPE IGNACIO HERNÁNDEZ MARTÍNEZ

PROFESOR GUÍA:
MARCEL GOIC FIGUEROA

PROFESOR CO-GUÍA:
ANDRÉS ABELIUK KIMELMAN

MIEMBROS DE LA COMISIÓN:
CAROLINA SEGOVIA RIQUELME
FELIPE BRAVO MÁRQUEZ

SANTIAGO DE CHILE

2024

RESUMEN DE LA MEMORIA PARA OPTAR
A LOS TÍTULOS DE INGENIERO CIVIL INDUSTRIAL
E INGENIERO CIVIL EN COMPUTACIÓN
POR: FELIPE IGNACIO HERNÁNDEZ MARTÍNEZ
AÑO: 2024
PROF. GUÍA: MARCEL GOIC FIGUEROA

AGENTES INTELIGENTES BASADOS EN LARGE LANGUAGE MODELS PARA LA ATENCIÓN AUTOMATIZADA EN CANAL CONVERSACIONAL ONLINE DE SERVICIO AL CLIENTE DE UNA EMPRESA FINANCIERA

Los Modelos de Lenguaje Grandes (*LLM*, por sus siglas en inglés) han revolucionado el campo del procesamiento del lenguaje natural producto de sus capacidades de resolución de problemas y a las habilidades emergentes que presentan. Sumado a su excelente capacidad para comunicarse con los seres humanos, los LLMs han generado oportunidades al área de la Inteligencia Artificial para poder alcanzar la construcción de *agentes inteligentes*, los cuales pueden aplicarse en diversas áreas de la industria para la automatización de procesos que requieran la interacción de máquinas con humanos. En particular, estas capacidades sugieren que los LLMs pueden ser una herramienta poderosa para automatizar el servicio al cliente. No obstante, estos modelos han sido criticados por sus potenciales alucinaciones y la generación de contenido peligroso para los humanos. En el contexto de la atención de clientes, este es un riesgo mayor que puede afectar la calidad del servicio y la imagen de las firmas.

En este trabajo se estudia el desempeño de diferentes estrategias de *prompting* en la aplicación de agentes basados en LLMs para la atención de clientes bancarios que buscan soluciones en el contexto de transacciones desconocidas en sus productos de crédito. Para la evaluación de los agentes, se comparan las percepciones de los usuarios en métricas como Resolutividad, Confianza, Satisfacción y Eficacia de comunicación, contrastándolas con mediciones objetivas como porcentajes de errores en las acciones realizadas y tiempos de respuesta.

Para ello, se diseña e implementa una plataforma experimental donde se puede interactuar con un agente inteligente en escenarios ficticios de desconocimiento de transacciones. Luego, los datos obtenidos son sometidos a análisis de medias y regresión para estudiar el efecto de las estrategias de *prompting* en base a un modelo conceptual propuesto.

A partir de los resultados, se obtienen cuatro conclusiones. Primero, aumentar el control sobre el agente, tanto a nivel de detalle de las instrucciones que posee como las acciones que puede realizar, mejora su desempeño medido con métricas objetivas, pero la percepción de resolutividad que el cliente tiene sobre este empeora. Segundo, en el contexto estudiado los humanos no son capaces de percibir cuando el agente comete un error. Más aún, basan su percepción del desempeño de este principalmente en las capacidades de comunicación que demuestra y la confianza que transmite. Tercero, y en base a lo anterior, para maximizar la experiencia del cliente es necesario priorizar un agente con amplias capacidades de comunicación que actúe con seguridad. Finalmente, considerando estos aprendizajes se concluye que los agentes automáticos pueden ser una solución para las fuentes de ineficiencia temporales de los canales de servicio al cliente digitales, dado a que tienen tiempos de respuesta despreciables y pueden ser asignados directamente a una atención.

A mis hermanos, Víctor y Fernanda.

Agradecimientos

Quiero comenzar agradeciendo a mi familia, especialmente a mis padres, Víctor y Viviana. Ambos, cada cual a su estilo, han sido un apoyo fundamental durante este proceso, y por ello a ustedes debo todo lo que soy a día de hoy. También deseo destacar a mi hermano Víctor, quien fue mi principal referente para ingresar a esta escuela, y a mi hermana Fernanda, quien constantemente me inspira a superarme día tras día. Además, quiero agradecer a Katherine por estar ahí con su amor incondicional y por ser mi soporte en los momentos más difíciles de mi desarrollo como ingeniero. Te amo.

También deseo expresar mi profundo agradecimiento al profesor Marcel Goic, quien demostró una excelente disposición a guiar este trabajo de título y para sacarme de los problemas que encontré en este camino, motivandome a superar mis capacidades con cada iteración. Asimismo, quiero mencionar al profesor Andrés Abeliuk, quien aceptó el desafío de co-guiar este proyecto y estuvo siempre dispuesto a brindarme su asistencia cuando la necesité.

Por otro lado, quiero agradecer a todas las personas con las que tuve el placer de compartir en mi paso por la facultad. Primero, al Bas y al Dimitri, por ser eternos amigos y colegas en los diversos proyectos que aparecieron en el camino. También al profesor Juan Velásquez, por depositar su confianza en mí como profesor auxiliar de Ingeniería de la Información y Web Intelligence. Agradezco también a los colegas con los que tuve el honor de compartir en equipos docentes, y los alumnos que asistieron a mis clases; espero que hayan aprendido tanto de mí como yo aprendí de ustedes.

Quiero también reconocer a la Xime y al Gera de la secretaría de estudios, que fueron mi primer acercamiento al mundo laboral y con quienes pude colaborar organizando los procesos de matrículas de la facultad entre los años 2018 a 2023. Agradezco a los *amigos* en ventana, que hicieron de la u un espacio más ameno, y al equipo del WIC, especialmente al Felipe y la Fran, por abrirme sus puertas para permitirme desarrollar mis habilidades profesionales en diversos proyectos.

Además, quiero agradecer a mis amigos, tanto los de toda la vida como los que se sumaron en los últimos años: Guaja, Dankert, Tavy, Pipe, Pablo, Tifa y Fer. Y por último, pero no menos importante, gracias a Pancita por acompañarme durante innumerables noches de estudio.

A todos ustedes, ¡infinitas gracias por estar ahí!

Tabla de Contenido

1. Introducción	1
1.1. El impacto de los Modelos de Lenguaje Grandes	1
1.2. Oportunidad detectada	2
1.3. Objetivos	5
1.3.1. Objetivo General	5
1.3.2. Objetivos Específicos	5
1.4. Estructura del informe	5
2. Marco Conceptual	6
2.1. Agentes	6
2.1.1. Definiciones	6
2.1.2. Marcos de diseño de agentes	7
2.2. Introducción al Modelado del Lenguaje	8
2.2.1. Definiciones	8
2.2.2. Modelos de Lenguaje Estadísticos	9
2.2.3. Modelos de Lenguaje Neuronales y Modelos Preentrenados	10
2.3. Modelos de Lenguaje Grandes	11
2.3.1. Definiciones	11
2.3.2. Características	12
2.4. Prompt Engineering	14
2.4.1. Aspectos básicos	14
2.4.2. Técnicas de razonamiento	15
2.5. Agentes basados en LLMs	16
2.5.1. Modelos de lenguaje como agentes inteligentes	16
2.5.2. Marco propuesto	17
2.5.3. LangChain	19
2.6. Evaluación de interacciones en iNLP	22
2.6.1. Métodos de evaluación	22
3. Descripción del Problema	25
3.1. Descripción del flujo de atención	25
3.2. Problema de investigación	28
3.3. Metodología	30
4. Solución Computacional	33
4.1. Diseño del agente	33
4.1.1. Prompts	33
4.1.2. Herramientas	36

4.2.	Implementación	38
4.2.1.	Arquitectura física	38
4.2.2.	Modelo de datos	39
5.	Evaluación	41
5.1.	Modelo conceptual	41
5.2.	Plataforma experimental	42
5.2.1.	Diseño experimental	42
5.2.2.	Interfaz de usuario	43
6.	Resultados	45
6.1.	Clasificación de instancias	45
6.2.	Descripción de datos obtenidos	47
6.3.	Herramientas de análisis	50
6.4.	Análisis de medias	50
6.5.	Análisis de regresión	54
6.5.1.	Métricas a nivel de interacción	54
6.5.2.	Métricas globales	57
6.6.	Discusión	59
7.	Conclusiones	62
7.1.	Trabajo Futuro	63
	Bibliografía	64
	Anexos	67
A.	Pseudocódigo de flujo de ejecución de AgentExecutor	67
B.	Datos disponibles	67
B.1.	Tabla Agents	67
B.2.	Tabla Customers	68
B.3.	Tabla Messages	68
B.4.	Tabla Conversations	69
C.	Reconstrucción de conversaciones	70
D.	Prompts implementados	74
D.1.	Prompt Simple para GPT-3.5 Turbo	74
D.2.	Prompt Extenso para GPT-3.5 Turbo	74
D.3.	Prompt Extenso con Restricciones para GPT-3.5 Turbo	76
D.4.	Prompt Extenso para Chat Bison	77
E.	Código fuente de herramientas	79
E.1.	Herramientas de respuestas libres	79
E.2.	Herramientas con restricciones	81
F.	Resultados de comparaciones entre GPT-3.5 Turbo y Chat Bison	86
G.	Interfaz de usuario de plataforma experimental	90
G.1.	Interfaz de inicio	90
G.2.	Interfaz de instrucciones	91
G.3.	Interfaz de descripción de escenario	93
G.4.	Interfaz de chat	95
G.5.	Interfaz de encuesta de situación	97

G.6.	Interfaz de encuesta de final	99
H.	Resultados complementarios	101
H.1.	Esquema de bases generadas con datos obtenidos	101
H.2.	Relaciones entre variables	103
H.3.	Test de Shapiro-Wilk sobre métricas objetivo	108
H.4.	Tests de Levene y Bartlett	108
H.5.	Tests T y U de Mann-Whitney para diferencias entre medias	109
H.6.	Regresiones a nivel de interacción	110
H.7.	Regresiones a nivel global	113

Índice de Tablas

4.1.	Resultados de evaluación de modelos con conversaciones reales.	35
6.1.	Tópicos para modelo NMF.	45
6.2.	Tópicos para modelo LDA.	46
6.3.	Métrica de accuracy para las clasificaciones obtenidas con Topic Models y GPT-3.5 Turbo.	46
6.4.	Matriz de confusión para clasificación con LLM.	46
6.5.	Distribución de los participantes dentro de los distintos tipos de prompt. . . .	49
6.6.	Distribución de variables demográficas según tipo de prompt. Las variables categóricas muestran su frecuencia relativa.	49
6.7.	Tabla resumen de coeficientes obtenidos con modelo M4 para regresiones a nivel de interacción.	56
6.8.	Tabla resumen de coeficientes obtenidos con modelo M3g para regresiones a nivel global.	58
6.9.	Regresiones de variables de desempeño subjetivas controladas por porcentaje de errores.	60
B.1.	Esquema de la tabla Agents. Los nombres de los campos han sido censurados y algunos se han omitido para preservar la confidencialidad del banco.	67
B.2.	Esquema de la tabla Customers. Los nombres de los campos han sido censurados y algunos se han omitido para preservar la confidencialidad del banco.	68
B.3.	Esquema de la tabla Messages. Los nombres de los campos han sido censurados y algunos se han omitido para preservar la confidencialidad del banco.	68
B.4.	Esquema de la tabla Conversations. Los nombres de los campos han sido censurados y algunos se han omitido para preservar la confidencialidad del banco. . .	69
C.1.	Mensajes asociados a un mismo identificador de cliente. Los registros se encuentran ordenados temporalmente y todos ocurren en un rango de 33 minutos. Algunos registros se han acertado u omitido para facilitar su visualización. . .	71
F.1.	Resultados para evaluación de agente basado en GPT-3.5 Turbo con prompt Extenso.	86
F.2.	Resultados para evaluación de agente basado en Chat Bison con prompt Extenso.	87
F.3.	Resultados para evaluación de agente basado en GPT-3.5 Turbo con prompt Extenso con Restricciones.	88
F.4.	Resultados para evaluación de agente basado en Chat Bison con prompt Extenso con Restricciones.	89
H.1.	Panel generado con datos por interacción.	101
H.2.	Base con datos globales por experimento.	102
H.3.	Medias por métricas objetivo segmentadas por tipo de prompt.	103
H.4.	Medias por métricas objetivo segmentadas por tipo de escenario.	103
H.5.	Medias por métricas objetivo segmentadas por sexo del participante.	103

H.6.	Medias por métricas objetivo segmentadas por nivel socioeconómico del participante.	104
H.7.	Resultados obtenidos con test de Shapiro-Wilk para métricas por interacción según prompt. Los p-valores indican que se rechaza la hipótesis nula en todos los casos, luego los grupos no pertenecen a una distribución normal.	108
H.8.	Resultados obtenidos con test de Shapiro-Wilk para métricas globales según prompt. Los p-valores indican que se rechaza la hipótesis nula en todos los casos, luego los grupos no pertenecen a una distribución normal.	108
H.9.	p-valores obtenidos con tests de Levene y Bartlett para métricas objetivo.	108
H.10.	p-valores obtenidos para comparaciones entre tipos de prompts con tests T y U de Mann-Whitney para métricas objetivo.	109
H.11.	Resultados de modelos de regresión para la variable de Satisfacción. Se ha incorporado una fila con los p-valores para las diferencias entre los efectos de los prompts E y E-Res en cada modelo.	110
H.12.	Resultados de modelos de regresión para la variable de Resolutividad. Se ha incorporado una fila con los p-valores para las diferencias entre los efectos de los prompts E y E-Res en cada modelo.	111
H.13.	Resultados de modelos de regresión para la variable de Correctitud. Se ha incorporado una fila con los p-valores para las diferencias entre los efectos de los prompts E y E-Res en cada modelo.	112
H.14.	Resultados de modelos de regresión para la variable de Satisfacción global. Se ha incorporado una fila con los p-valores para las diferencias entre los efectos de los prompts E y E-Res en cada modelo.	113
H.15.	Resultados de modelos de regresión para la variable de Resolutividad global. Se ha incorporado una fila con los p-valores para las diferencias entre los efectos de los prompts E y E-Res en cada modelo.	114
H.16.	Resultados de modelos de regresión para la variable de Correctitud global. Se ha incorporado una fila con los p-valores para las diferencias entre los efectos de los prompts E y E-Res en cada modelo.	115

Índice de Ilustraciones

1.1.	Diagrama de flujo de las atenciones en el chat. En rojo se destaca el proceso donde se encuentran las tensiones identificadas.	3
2.1.	Diagrama de la interacción entre un agente y un ambiente.	7
2.2.	Interacción Humano-LLM basada en prompts.	11
2.3.	Interacciones entre los componentes de un agente de LangChain. A modo de ejemplo, se muestra cómo el agente decide utilizar una acción para responder a la pregunta <i>¿Cuál es la capital de Japón?</i>	21
3.1.	Diagrama de flujo de las atenciones de desconocimiento de transacciones. El símbolo <i>API</i> indica cuando los ejecutivos requieren acceder a sistemas fuera del chat.	27
3.2.	Metodología propuesta de tres etapas.	30
4.1.	Arquitectura física de 4 capas para la plataforma experimental.	39
4.2.	Modelo relacional de la plataforma experimental.	40
5.1.	Modelo conceptual de los efectos del tipo de prompt sobre la percepción del cliente.	42
5.2.	Pantalla de chat en su versión oscura. A la izquierda, el menú lateral contiene la información del estado de cuenta ficticio del usuario.	44
6.1.	Distribución de variables de sexo y nivel socioeconómico de los participantes del experimento.	49
6.2.	Distribución de la edad de los participantes del experimento.	50
6.3.	Medias obtenidas por interacción sobre métricas subjetivas, según tipo de prompt. Se muestra además la significancia estadística de las diferencias, obtenidas con el test T.	51
6.4.	Medias globales sobre métricas subjetivas, según tipo de prompt. Se muestra además la significancia estadística de las diferencias, obtenidas con el test T.	52
6.5.	Medias por interacción sobre métricas de largo de interacción, según tipo de prompt. Se muestra además la significancia estadística de las diferencias, obtenidas con el test U de Mann-Whitney.	52
6.6.	Medias por interacción sobre métricas de tiempo de interacción, según tipo de prompt. Se muestra además la significancia estadística de las diferencias, obtenidas con el test U de Mann-Whitney.	53
6.7.	Medias por interacción sobre porcentajes de errores cometidos sobre el total de acciones realizadas, según tipo de prompt. Se muestra además la significancia estadística de las diferencias, obtenidas con el test U de Mann-Whitney.	54
G.1.	Pantalla de inicio de la plataforma experimental, en su versión clara.	90
G.2.	Pantalla de inicio de la plataforma experimental, en su versión oscura.	91
G.3.	Pantalla de instrucciones de la plataforma experimental, en su versión clara.	92
G.4.	Pantalla de instrucciones de la plataforma experimental, en su versión oscura.	93

G.5.	Pantalla de descripción de escenario, en su versión clara.	95
G.6.	Pantalla de descripción de escenario, en su versión oscura.	95
G.7.	Pantalla de chat en su versión oscura, con el menú lateral mostrando el estado de cuenta del participante.	96
G.8.	Pantalla de chat en su versión clara, con el menú lateral mostrando la descripción del escenario.	96
G.9.	Pantalla de chat en su versión oscura, con el menú lateral mostrando la descripción del escenario.	97
G.10.	Pantalla de encuesta de situación, en su versión clara. El menú lateral muestra la conversación que tuvo el participante con el agente virtual.	98
G.11.	Pantalla de encuesta de situación, en su versión oscura. El menú lateral muestra la conversación que tuvo el participante con el agente virtual.	99
G.12.	Pantalla de encuesta final, en su versión clara.	100
G.13.	Pantalla de encuesta final, en su versión oscura.	100
H.1.	Net Promoter Score calculado según tipo de agente.	104
H.2.	Promedio de edad de participante según puntaje asignado a métricas objetivo. La línea punteada indica el promedio de la muestra experimental.	105
H.3.	Promedio de variables de largo de interacción según puntaje asignado a métricas objetivo. La línea punteada indica el promedio de la muestra experimental.	106
H.4.	Promedio de variables de duración de interacción según puntaje asignado a métricas objetivo. La línea punteada indica el promedio de la muestra experimental.	106
H.5.	Promedio de variables de duración de interacción según puntaje asignado a métricas objetivo. La línea punteada indica el promedio de la muestra experimental.	107

Capítulo 1

Introducción

Este capítulo introduce al lector en el tema del trabajo, explorando los agentes inteligentes basados en Modelos de Lenguaje Grandes y su impacto en la industria. Además, se analiza la oportunidad identificada en la firma y se establecen los objetivos en relación con esta. Finalmente, se proporciona una breve descripción de la estructura que seguirá el informe.

1.1. El impacto de los Modelos de Lenguaje Grandes

Durante miles de años, el ser humano ha intentado entender *cómo piensa*, es decir, comprender su inteligencia y cómo esta interactúa con el mundo que lo rodea. El campo de la *Inteligencia Artificial* (AI, por sus siglas en inglés) va aún más allá: no tan solo intenta entender, sino también construir entidades inteligentes (Russell et al., 2010; Bubeck et al., 2023). A partir de esta idea, se desprende el concepto de *Agente*. Desde la perspectiva de la AI, un agente es una entidad que actúa operando de forma autónoma, percibiendo su ambiente y realizando acciones sobre este, adaptándose al cambio, y logrando crear y perseguir metas (Russell et al., 2010). Los agentes son reconocidos como un paso intermedio para alcanzar la *Inteligencia Artificial General* (AGI, por sus siglas en inglés) (Xi et al., 2023; Wang et al., 2024), o en otras palabras, sistemas que demuestran amplias capacidades de inteligencia al mismo nivel o por sobre las que posee el ser humano (Bubeck et al., 2023).

No obstante, los agentes que emplean tecnologías previas a la aparición de los modelos de lenguaje más modernos usualmente se encuentran lejos de replicar un comportamiento similar al humano en la toma de decisiones, debido al uso de entrenamientos basados en políticas simples y ambientes restringidos y aislados, lo que dista mucho de la complejidad de la mente humana y los diversos contextos desde los cuales aprenden (Wang et al., 2024).

Los *Modelos de Lenguaje Grandes* (LLM, por sus siglas en inglés), posiblemente el avance más destacado en la investigación sobre AI del último tiempo, representan una luz de esperanza para el desarrollo de agentes, gracias a que demuestran un potencial prometedor para alcanzar niveles de inteligencia similar a los humanos (Xi et al., 2023; Wang et al., 2024; Zhao et al., 2023). Más aún, algunos autores ya asimilan algunos de estos modelos, específicamente *GPT-4* (OpenAI et al., 2023), como versiones tempranas de AGI (Bubeck et al., 2023).

Los últimos años han dado lugar a una creciente expansión de la investigación y uso de los LLMs, siendo uno de los hitos de mayor impacto el lanzamiento de *ChatGPT* por OpenAI

(2022) a fines del año 2022. Basta con notar que el número promedio de artículos publicados por día en *arXiv* que contienen “*large language model*” en el título o *abstract* pasó de 0.40 a 8.58 después de su aparición (Zhao et al., 2023).

Este desarrollo vigoroso no es de sorprender, dadas las impresionantes habilidades que presentan estos modelos en resolver tareas complejas basadas en lenguaje natural. Estas habilidades de carácter impredecible, llamadas *habilidades emergentes*, se definen como propias de los LLMs y no se encuentran presentes en modelos más pequeños (Wei et al., 2022). Algunas de las más estudiadas en la literatura asociada a estos modelos son la capacidad de demostrar aprendizajes a partir de entradas de texto, el poder seguir instrucciones y el razonamiento paso a paso para resolver tareas complejas (Zhao et al., 2023). En particular, ChatGPT es un modelo conversacional con capacidades de diálogo sin precedentes al momento de su lanzamiento, dentro de las que destacan una vasta base de conocimiento, habilidades de razonamiento en problemas matemáticos, seguimiento correcto del contexto en diálogos de múltiples turnos y el alineamiento con los valores humanos para otorgar un uso seguro en las interacciones con estos (Zhao et al., 2023).

De esta manera, los modelos de lenguaje grandes han resultado en un impulso para el desarrollo de agentes por parte de la comunidad investigativa, haciéndolos altamente deseables en la construcción de *agentes inteligentes*, dado que demuestran poderosas capacidades para la adquisición de conocimiento, comprensión de instrucciones, generalización, planificación y razonamiento, a la vez que pueden comunicarse efectivamente con los seres humanos (Xi et al., 2023). Así, los agentes basados en LLMs pueden verse aplicados en áreas muy diversas con resultados prometedores, tales como *chatbots* de apoyo a la educación (Dan et al., 2023), el desarrollo de habilidades sociales en pacientes con Trastorno del espectro autista (Ali et al., 2020), el diseño de juegos interactivos para medición psicológica (Yang et al., 2024), la asistencia de ventas en telemarketing (Gao et al., 2023) y el servicio al cliente en canales digitales (Cui et al., 2017; Pandya y Holia, 2023).

1.2. Oportunidad detectada

La oportunidad detectada en la que se centra esta investigación surge dentro de una empresa financiera chilena líder de la industria bancaria, con una fuerte orientación estratégica de cara a la digitalización de la experiencia ofrecida a sus clientes. Esta empresa cuenta con un canal digital de servicio al cliente a través de un *chat*, que consiste de una interfaz de conversaciones en línea donde se llevan a cabo diversos tipos de atenciones. El chat dispone de un bot conversacional (*chatbot*, del inglés) para la atención automática de clientes, así como la participación de ejecutivos humanos para aquellos casos donde la automatización actual no es suficiente. Su acceso está disponible a través de aplicaciones de mensajería, permitiendo comenzar una conversación en distintos dispositivos y desde diversos orígenes, tales como redirecciones desde el sitio web, aplicación móvil, call center o sucursales, o bien, simplemente escribiendo directamente al número del banco. En menor medida, es posible acceder a una atención a través de una interfaz disponible en el sitio web del banco, pero su uso va en declive.

En términos prácticos, las interacciones del chatbot consisten en una secuencia de menús de autoatención conformados por un conjunto de opciones que el usuario puede seleccionar, ya sea para navegar hasta alguna respuesta automatizada, o bien, ser derivado con un eje-

cutivo que continúe su atención, de esta manera posibilitando el acceso a distintos tipos de servicios según sus necesidades. Por ejemplo, un cliente puede utilizar únicamente las opciones del menú de autoatención para solicitar que se le envíe su estado de cuenta bancario a su correo electrónico. Por otro lado, un segundo cliente podría navegar por estos menús de autoatención para solicitar una atención con un ejecutivo por una emergencia bancaria. La figura 1.1 presenta un diagrama de flujo de la forma en que operan las atenciones en el canal.

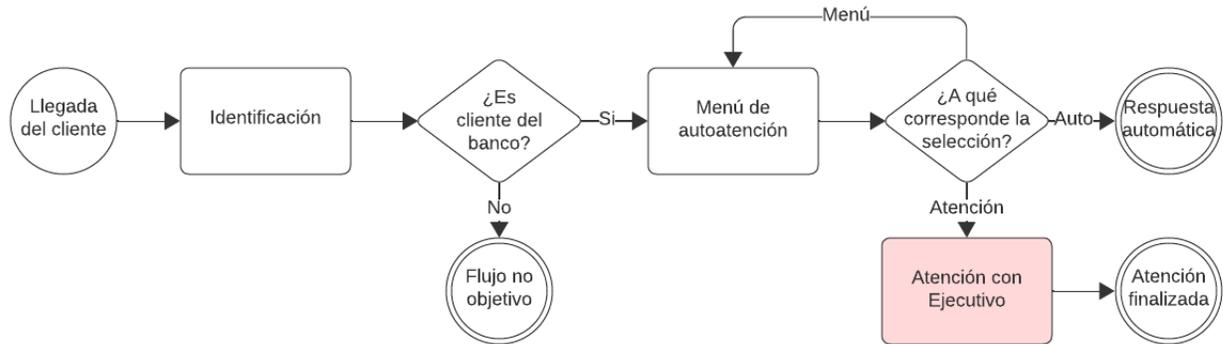


Figura 1.1: Diagrama de flujo de las atenciones en el chat. En rojo se destaca el proceso donde se encuentran las tensiones identificadas. Fuente: Fuente: Elaboración propia

Al día de hoy, este canal corresponde a uno de las infraestructuras digitales más importantes de la empresa, pues se transformó en su principal vía de atención durante el año 2022, participando entre el 40% y 50% de sus interacciones con los usuarios. Luego, surge la necesidad de buscar constantemente mejoras en torno a su eficacia en la atención exitosa de clientes. Específicamente, se han detectado oportunidades de mejora en aquellas atenciones que requieren de la intervención de un ejecutivo humano.

Existen diversos motivos por los cuales puede generarse una atención con un ejecutivo, siendo los principales tipos:

- **Desconocimiento de transacciones:** El cliente manifiesta que existe una transacción o cargo asociado a alguno de sus productos cuyo origen desconoce.
- **Rechazo de transacción:** El cliente indica que tiene problemas al intentar realizar una transacción, ya sea a través de un medio físico o comercio *online*.
- **Consultas de estado de cuenta:** El cliente tiene problemas para obtener su estado de cuenta, o bien, dudas sobre este.

Las tensiones identificadas en torno a estas atenciones son consecuencia de ser un proceso realizado de forma manual, y afectan directamente la experiencia del cliente y los costos que percibe la empresa. Se pueden resumir en las siguientes dificultades:

- **Tiempos de espera:** Un análisis realizado sobre los casos de desconocimiento de transacciones muestran que, en promedio, los clientes deben esperar en un rango de 10 a 15

minutos en ser asignados a un ejecutivo, mientras que el tiempo de atención en la interacción con este se extiende por otros 200 minutos aproximadamente. En otras palabras el tiempo de atención que toma en resolver el problema del cliente alcanza los 215 minutos aproximadamente a través de este canal. La espera en la asignación es producto de que la cantidad de ejecutivos disponibles para realizar la atención es limitada. Además, en ocasiones los ejecutivos deben atender múltiples casos simultáneamente, lo que puede resultar en tiempos de atención más prolongados de lo habitual y en algunos casos, pueden llevarlos a optar por no recibir la atención deseada. Esto abre la oportunidad de impactar positivamente la experiencia del cliente a través de la disminución de estos tiempos.

- **Atenciones incompletas:** De la mano con el punto anterior, en ocasiones los tiempos de espera generan que los clientes dejen de responder, ya sea porque se tienen otras cosas que hacer u olvidaron continuar la conversación. Como consecuencia, existe la oportunidad de mejorar este factor para lograr más resoluciones exitosas, evitando que los clientes necesiten más de una atención. Esto impacta tanto en la experiencia del cliente como en los costos percibidos por los recursos empleados por el banco.
- **Calidad de los datos almacenados:** Tras cada atención realizada, los ejecutivos deben clasificar manualmente la conversación producida dentro de una categoría, acción denominada como *tipificación*. A partir de una estimación en base a palabras clave que utilizan los ejecutivos durante estas conversaciones, se obtiene que en la categoría de atenciones de “desconocimiento de transacciones” son existe un déficit de al menos un 50 % de los casos clasificados. Es decir, al menos la mitad de las conversaciones de esta categoría no son correctamente tipificadas. Esto ofrece una oportunidad de mejora, pues son datos que se utilizan para la toma de decisiones de negocio, tales como determinar dónde invertir en desarrollos sobre la infraestructura del chat para ofrecer mejoras específicas a un tipo de atención que lo requiera.

Los puntos descritos muestran que existen oportunidades ligadas tiempos de espera para tareas que tienen alcances de acción definidos, luego se abre la posibilidad de mejorarlas a través de automatización. No obstante, la naturaleza de las interacciones que ocurren en las atenciones son altamente no estructuradas, dado que están basadas en texto y no son procesos lineales, por lo que existen fuertes limitaciones para emplear herramientas tradicionales.

En base a lo mencionado en la sección 1.1, el uso de Modelos de Lenguaje Grandes para construir agentes que puedan utilizarse para automatizar estas atenciones surge como una opción natural por el tipo de problema que se enfrenta, dada las poderosas capacidades de estos para procesar texto. Además, tomando en consideración la naturaleza compleja de los problemas bancarios y los datos sensibles que pueden aparecer en las conversaciones, es relevante utilizar tecnologías que minimicen posibles errores y riesgos en las interacciones. En consecuencia, la hipótesis de investigación de este trabajo es que una solución que automatice el proceso de atención con agentes basados en LLMs puede reducir estas cifras, mejorando así la experiencia del cliente y solucionando las fuentes de ineficiencia presentes.

El diseño de este agente requiere de la definición de diversos aspectos en el uso de LLMs, como las instrucciones y capacidades de acción que debe tener, para los cuales existen distintos enfoques de resolución. Esta diversidad de estrategias de diseño puede llevar a conversaciones

distintas en un contexto de atención, y por ello, mejor o peor alineadas con los objetivos del banco.

Dado que es ambicioso y complejo construir una solución que automatice los tres tipos de atenciones mencionados, las cuales poseen sus desafíos individuales, este trabajo tendrá su foco en la automatización sobre un solo caso de servicio. Específicamente, se centrará en los casos de desconocimiento de transacciones, dadas las cifras reportadas previamente, además de ser aquella categoría que posee mayor definición en el flujo de atención que debe llevarse a cabo para completarse con éxito.

1.3. Objetivos

A continuación, se presentan los objetivos general y específicos de este trabajo.

1.3.1. Objetivo General

Diseñar y evaluar un sistema computacional que automatiza atenciones de desconocimiento de transacción a partir la implementación de un agente inteligente basado en modelos de lenguaje grandes en interacciones vía chat, mejorando la experiencia de cliente y la eficiencia que tiene el canal para el banco.

1.3.2. Objetivos Específicos

Desde el objetivo general, se desglosan los siguientes objetivos específicos:

1. Caracterizar el proceso que involucra la interacción entre el cliente y el ejecutivo en los casos de desconocimiento de transacciones.
2. Explorar y evaluar diseños del sistema de automatización basado en agentes.
3. Implementar una prueba de concepto con los diseños definidos.
4. Evaluar empíricamente el desempeño del sistema propuesto.

1.4. Estructura del informe

Los siguientes capítulos del informe se estructuran como se indica a continuación. El capítulo 2 presenta el marco conceptual necesario para comprender el problema, la implementación de la solución y su evaluación. El capítulo 3 presenta el problema de investigación, describiendo cada uno de los componentes a resolver. El capítulo 4 presenta la arquitectura de la solución, pasando por el diseño y la implementación de los agentes. El capítulo 5 expone una propuesta de modelo conceptual para explicar el desempeño de los agentes en las interacciones con clientes, además del diseño experimental para implementar el mismo. El capítulo 6 describe los resultados de la evaluación del sistema, explorando los datos recopilados, realizando un análisis con herramientas estadísticas y discutiendo los hallazgos obtenidos. Finalmente, el capítulo 7 presenta las conclusiones del trabajo y una línea de desarrollo futuro basado en este.

Capítulo 2

Marco Conceptual

En este capítulo, se condensa la teoría necesaria para comprender el problema de investigación y su solución. Comienza definiendo qué es un agente inteligente y presenta un marco para su diseño e implementación. Luego, ofrece una introducción al modelado del lenguaje y posteriormente presenta los modelos de lenguaje grandes, con sus características y limitaciones. Más adelante, se presentan técnicas para el uso de estos modelos, conectándolas con su aplicación en la construcción de agentes inteligentes. Finalmente, el capítulo concluye con la discusión de métodos de evaluación de estos agentes.

2.1. Agentes

2.1.1. Definiciones

Si bien, no existe una definición universalmente aceptada en la literatura para este término, se comprenderá por agente a un sistema computacional que se sitúa en un *ambiente*, y tiene la capacidad de actuar de forma autónoma sobre este ambiente con el fin de alcanzar un objetivo específico (Wooldridge, 2002). Matemáticamente, los agentes pueden comprenderse como una función que toma una secuencia de percepciones del ambiente y retorna una acción acorde al objetivo que debe cumplir (Russell et al., 2010). Esta noción permite comprender que los agentes deben basar su elección en las percepciones que han adquirido en el tiempo en su interacción con el ambiente.

Dentro de esta definición es clave comprender el significado de *autonomía*. Al hablar de agentes autónomos, se alude a que estos pueden actuar sin la intervención de humanos u otros sistemas: tienen control sobre su propio estado interno y su comportamiento (Wooldridge, 2002).

Un agente cuenta con un repertorio de acciones disponibles, que en su conjunto representan su *capacidad efectora*, es decir, su habilidad de modificar su ambiente. Por ejemplo, en el caso particular de este trabajo, la capacidad efectora del agente se compondrá por acciones como enviar mensajes al cliente, buscar transacciones desconocidas por este, bloquear su tarjeta bancaria para evitar más cargos inesperados, entre otras. Estas acciones no pueden realizarse en todas las situaciones, pues tienen *pre-condiciones* que definen cuándo pueden utilizarse. Así, el problema clave que resuelve un agente es decidir cuál acción disponible debe realizar para poder satisfacer de mejor manera su objetivo de diseño (Wooldridge, 2002). Para resolver

este problema, un agente debe percibir su ambiente a través de *sensores* y actúa sobre este a través de *actuadores*. Por ejemplo, un agente humano tiene ojos y oídos como sensores, y manos y piernas como actuadores (Russell et al., 2010). La figura 2.1 presenta un diagrama de esta relación.

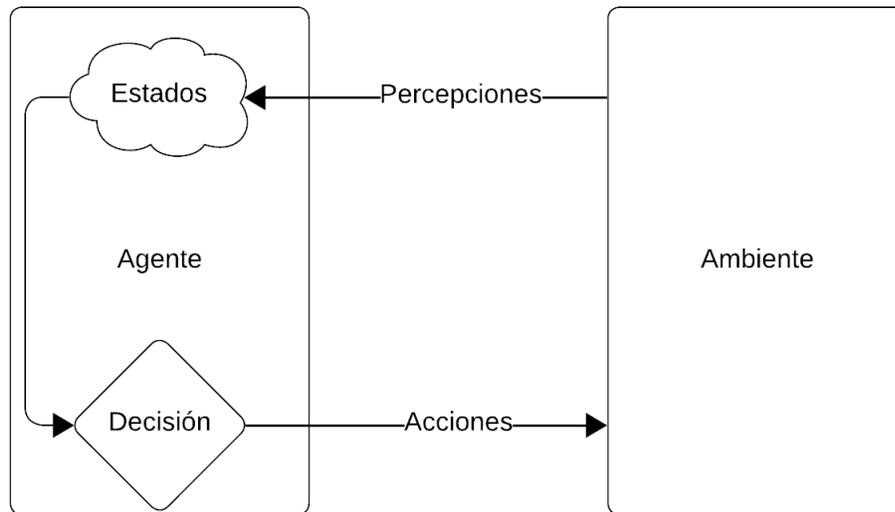


Figura 2.1: Diagrama de la interacción entre un agente y un ambiente. Fuente: Elaboración propia.

No obstante, esta definición de agente y autonomía abarca sistemas que a simple vista podrían no considerarse dentro de este espectro. Por ejemplo, un termostato tiene sensores para percibir la temperatura y puede accionar mecanismos para encender o apagar sistemas de calefacción de forma autónoma. El componente del cual el termostato carece es el de *inteligencia*. ¿Cuándo es posible considerar que un agente posee inteligencia? El concepto en sí no es simple de definir. Se considerará un *agente inteligente* como aquel que tiene una capacidad de acción autónoma *flexible* para poder alcanzar un objetivo definido, donde flexibilidad hace referencia a tres puntos (Wooldridge, 2002):

- **Reactividad:** Tienen la capacidad de percibir su ambiente y responder de forma oportuna a los cambios que ocurren en este para alcanzar sus objetivos.
- **Pro-actividad:** Demuestran un comportamiento adecuado a alcanzar sus objetivos, tomando la iniciativa.
- **Habilidad social:** Son capaces de interactuar con otros agentes (incluyendo humanos) de manera de satisfacer sus objetivos.

2.1.2. Marcos de diseño de agentes

Uno de los marcos clásicos para el diseño de agentes es *PEAS* (*Performance, Environment, Actuators, Sensors*), presentado por Russell et al. (2010). En términos simples, PEAS se basa en la identificación de cuatro elementos fundamentales de un agente en un caso de uso: a métrica de desempeño que indica si el agente cumple su objetivo, el entorno en el que se

desenvuelve, y los actuadores y sensores necesarios. Sin embargo, este marco resulta poco preciso en cuanto a la implementación del agente, especialmente cuando se basa en una tecnología particular como los modelos de lenguaje grandes. Por ejemplo, PEAS no especifica claramente dónde deben ubicarse estos modelos dentro de este sistema de cuatro componentes.

La literatura reciente proporciona marcos más precisos para la aplicación que se diseñará en este trabajo. Específicamente, Xi et al. (2023) proponen un marco de tres componentes: *Cerebro - Percepción - Acción* (abreviado como *C-P-H*). En este, se diseña cómo el agente debe controlar sus decisiones, percibir una interacción a través de texto y realizar acciones a partir del uso de modelos de lenguaje grandes como el núcleo central de su funcionamiento. No obstante, antes de explicar cómo este marco se instancia a la aplicación de este trabajo, es necesario introducir la teoría ligada al uso de estos modelos de lenguaje. Por ello, la sección 2.5 presenta cómo este marco aplica al caso de uso, además de presentar un framework de software que permite su implementación.

2.2. Introducción al Modelado del Lenguaje

2.2.1. Definiciones

El *Procesamiento del Lenguaje Natural* (*NLP*, por sus siglas en inglés) se define como el campo que estudia el diseño de métodos y algoritmos que toman como entrada o producen como salida datos de lenguaje natural no estructurados. Ha ganado una relevancia creciente en los últimos años producto de la cada vez mayor cantidad de datos de texto que produce el ser humano, así como la necesidad de poder interactuar con sistemas computacionales utilizando su propio lenguaje (Goldberg, 2017).

A su vez, el *Modelado del lenguaje* corresponde a la aplicación de NLP para asignar una probabilidad a una secuencia de palabras, u oración, en un determinado lenguaje (Goldberg, 2017). Por ejemplo, esta tarea busca resolver preguntas como: “¿Cuál es la probabilidad de observar la oración *El cielo está soleado?*”.

Además de asignar probabilidades a secuencias de palabras, también es posible obtener la probabilidad de que una palabra sea la continuación de una determinada oración. En base al ejemplo anterior, es posible pensar en la pregunta: “¿Cuál es la probabilidad de observar la palabra *soleado* después de la secuencia *El cielo está?*”.

Aquellos modelos que asignan probabilidades a una secuencia de palabras se conocen como *Modelos de Lenguaje* (abreviados como *LM*, por sus siglas en inglés) (Jurafsky y Martin, 2023). El objetivo de estos es modelar la verosimilitud generativa de secuencias de palabras, para así predecir la probabilidad de *tokens* (o unidades indivisibles de lenguaje) futuros (Zhao et al., 2023).

Zhao et al. (2023) clasifica los LMs en cuatro categorías, que van desde modelos simples basados en métodos estadísticos hasta el uso de arquitecturas complejas, que son la base de las poderosas tecnologías presentes en el estado del arte: los Modelos de Lenguaje Grandes. A continuación, se describen de forma sucinta las características de las primeras tres categorías,

para entregar una contextualización de la historia recorrida en el modelado del lenguaje y comprender la esencia de estos modelos. La cuarta categoría, que es parte del núcleo de este trabajo, será presentada con detalle en la sección 2.3.1

2.2.2. Modelos de Lenguaje Estadísticos

Los Modelos de Lenguaje Estadísticos (*SLM*, por sus siglas en inglés) usualmente utilizan el supuesto de Markov, es decir, que el futuro es independiente del pasado dado el presente (Goldberg, 2017). En términos más concretos, este supuesto se basa en asumir que la probabilidad de ocurrencia de una palabra depende únicamente del *contexto* más reciente, donde el contexto corresponde a la secuencia de palabras anteriores.

Específicamente, se habla de un supuesto de Markov de orden k cuando se asume que la siguiente palabra w_{i+1} en una secuencia solo depende de las últimas k palabras de la misma:

$$P(w_{i+1}|w_1, \dots, w_i) \sim P(w_{i+1}|w_{i-k}, \dots, w_i)$$

El modelo más simple de estos corresponde al caso en que el contexto es de un largo fijo n , también llamados modelos de *n-gramas* (Zhao et al., 2023).

Un caso particular de estos modelos corresponde al de *bigramas*, donde se asume que una la probabilidad de una palabra dado un contexto previo se aproxima a la probabilidad condicionada bajo la última palabra de este (Jurafsky y Martin, 2023). A modo de ejemplo, si se desea estimar la probabilidad de que a la oración *La capital de Japón es* le siga la palabra *Tokio*, utilizando un modelo de bigramas, esta probabilidad viene dada por:

$$P(\text{Tokio}|\text{La capital de Japón es}) \sim P(\text{Tokio}|\text{es})$$

Usando la regla de la cadena de probabilidades, la probabilidad de obtener una secuencia $s = w_1, \dots, w_n$, puede ser expresada como:

$$P(w_1, \dots, w_n) = P(w_1) \times P(w_2|w_1) \times P(w_3|w_1, w_2) \times \dots \times P(w_n|w_1, \dots, w_{n-1})$$

Utilizando un modelo de bigramas esta expresión se simplifica, dado que ahora las probabilidades condicionales sólo dependen de la palabra inmediatamente anterior sobre la que se está estimando el cálculo:

$$P(w_1, \dots, w_n) = P(w_1) \times P(w_2|w_1) \times P(w_3|w_2) \times \dots \times P(w_n|w_{n-1})$$

La forma en que se calculan las probabilidades condicionales en un modelo de *n-gramas* viene dada por un Estimador de Máxima Verosimilitud (Jurafsky y Martin, 2023), cuyo cómputo consiste en realizar un conteo sobre un corpus de entrenamiento y luego normalizarlo para obtener valores entre 0 y 1.

A pesar de lo prácticos que puedan resultar para algunas tareas de NLP, estos modelos sufren de algunas limitaciones. Primero, el número de probabilidades a calcular crece exponencialmente con el largo del contexto (lo que se conoce como *maldición de la dimensionalidad*) (Goldberg, 2017; Zhao et al., 2023). Además, el cálculo de probabilidades descrito asigna valores 0 a aquellos n-gramas que no se incluyan en el corpus de entrenamiento, lo

que requiere el uso de técnicas de *smoothing* (Goldberg, 2017). Finalmente, carecen de generalización entre contextos con significados similares. Es decir, el hecho de haber observado el n-grama *Mi camiseta azul* no influencia la estimación del evento *Mi camiseta roja* (Goldberg, 2017), lo que significa que estos modelos no toman en consideración la similitud entre palabras (Bengio et al., 2003).

2.2.3. Modelos de Lenguaje Neuronales y Modelos Preentrenados

Las *Redes Neuronales* son una familia de modelos de aprendizaje inspirados en el funcionamiento del cerebro humano (utilizando una analogía a la neurona humana como unidad básica de cómputo), que consisten en ajustar funciones parametrizadas diferenciables en capas encadenadas (Goldberg, 2017; Jurafsky y Martin, 2023). Esta idea de poder encadenar varias capas le otorga el nombre de *Aprendizaje Profundo* a la rama de *Aprendizaje de Máquinas* que abarca este tipo de modelos.

Los *Modelos de Lenguaje Neuronales* (abreviados como *NLM*, del inglés *Neural Language Model*) utilizan redes neuronales para caracterizar las probabilidades de una secuencia de palabras (Zhao et al., 2023). Gracias a las arquitecturas de redes *feed-forward* y recurrentes, los NLM pueden resolver las limitaciones de los modelos de n-gramas al conseguir procesar contextos de largo variable y obtener un desempeño superior en la tarea de modelado de lenguaje, dado que pueden generalizar mejor en contextos no vistos durante el entrenamiento (Jurafsky y Martin, 2023).

El trabajo de Bengio et al. (2003) introduce el concepto de *representación distribuida de palabras*, que corresponde a utilizar el proceso de entrenamiento de la red neuronal del NLM para encontrar una representación vectorial de las palabras en un espacio numérico continuo. Este avance significó una revolución en el campo de NLP, dado que abre la posibilidad a comparar palabras de forma numérica, alcanzando una noción de similitud y la capacidad de generalizar a contextos no vistos durante el entrenamiento (Goldberg, 2017). Esta noción de vector representativo de una palabra, recibe el nombre de *embedding* (Jurafsky y Martin, 2023; Goldberg, 2017).

El proceso de realizar el entrenamiento de una red neuronal sobre grandes cantidades de texto para aprender una representación de palabras se conoce como *preentrenamiento*. Aquellos modelos de lenguaje que utilizan esta técnica se les conoce como *Modelos de Lenguaje Preentrenados* (*PLM*, por sus siglas en inglés). La arquitectura *Transformer* (Vaswani et al., 2017) es la más utilizada para la construcción de estos modelos, dado que ofrece nuevos mecanismos que permiten representar relaciones temporales y a largas distancias dentro de un contexto de palabras (Jurafsky y Martin, 2023). En particular, *BERT* (*Bidirectional Encoder Representations from Transformers*) (Devlin et al., 2019) es un PLM basado en esta arquitectura que puede ser ajustado en sus parámetros (proceso conocido como *fine-tuning*) para adaptarlo a diversas tareas de NLP sin tener que hacer cambios sustanciales en su arquitectura. Este paradigma de *preentrenamiento y fine-tuning* inspiró un gran número de PLMs desarrollados posteriormente, con el objetivo de ser ajustados a tareas específicas (Zhao et al., 2023).

2.3. Modelos de Lenguaje Grandes

2.3.1. Definiciones

La investigación reciente encontró que escalar los PLMs, tanto en el número de parámetros del modelo como en el tamaño del corpus utilizado en su entrenamiento, generalmente conlleva a una mejora en el desempeño del modelo en la resolución de tareas específicas frente a los de menor tamaño. Estos modelos preentrenados de gran tamaño (generalmente, de billones de parámetros) son denominados en la literatura como *Modelos de Lenguaje Grandes (LLM)*, por sus siglas en inglés) (Zhao et al., 2023).

Estos, dada su cantidad de parámetros, requieren corpus de textos masivos para su entrenamiento como *Common Crawl* (Common Crawl, 2023) o *Wikipedia* (Wikimedia Foundation, 2023). En consecuencia, el desarrollo de LLMs es altamente costoso y por ello hoy en día se encuentra limitado principalmente a empresas privadas (tales como Meta u OpenAI), quienes no revelan varios detalles importantes de su construcción (Zhao et al., 2023). A pesar de ello, es posible utilizar técnicas de fine-tuning sobre estos modelos para entrenarlos en tareas específicas.

Los LLMs han revolucionado la forma en que los seres humanos interactúan con los modelos de lenguaje, pues a diferencia de los PLMs de menor tamaño las interacciones ocurren principalmente a través de *prompts*. Un prompt es una entrada de texto que cuenta con una serie de instrucciones que se proporciona al modelo para guiar su generación de texto en una dirección específica. A diferencia de los datos utilizados para entrenar, este texto es consumido como input del modelo durante su ejecución en una aplicación, lo que los vuelve ideales para proveer interacciones con usuarios en tiempo real. Esto genera que los humanos deban comprender como funciona un LLM para formatear su tarea correctamente a través de la entrada. De esta manera, estos modelos pueden ser utilizados como un *solver* de propósito general en NLP y la investigación está apuntando en torno a las formas de aplicar los LLMs (Zhao et al., 2023). Una representación gráfica de la interacción entre humanos y LLMs se encuentra en la figura 2.2.

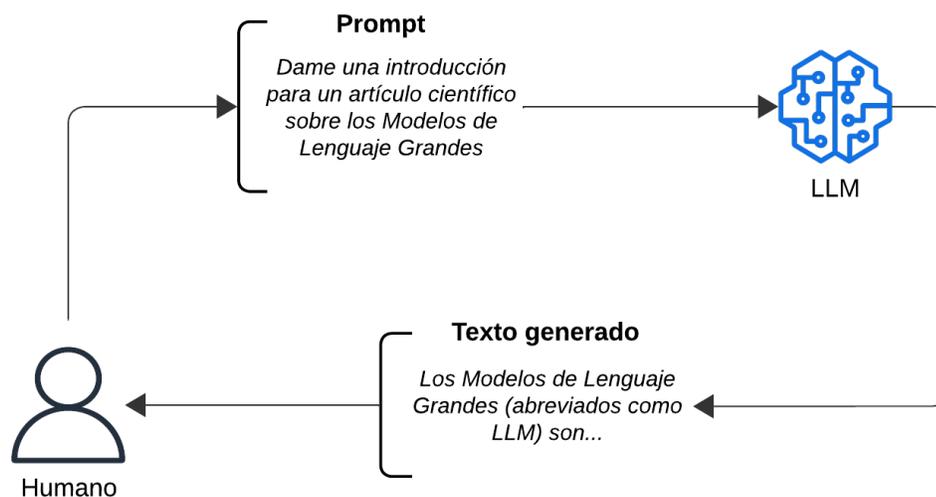


Figura 2.2: Interacción Humano-LLM basada en prompts. Fuente: Elaboración propia.

Uno de los casos de uso exitosos más reciente de este tipo de modelos, debido a su excelente capacidad de comunicarse con humanos, corresponde a *ChatGPT* (OpenAI, 2022), un modelo conversacional lanzado en noviembre de 2022 por OpenAI, basado en los modelos GPT-3.5 y GPT-4. A la fecha, parece ser el chatbot más poderoso en la historia de la inteligencia artificial (Zhao et al., 2023).

2.3.2. Características

Habilidades emergentes

Una de las características principales de los modelos de lenguaje grandes es la aparición de *habilidades emergentes*. Una habilidad se define como emergente si esta no se encuentra presente en modelos más pequeños, pero si en aquellos que son grandes (Wei et al., 2022). A continuación se presentan dos ejemplos de estas, relevantes para este trabajo.

- **In-Context Learning:** Brown et al. (2020) introduce el concepto de *In-Context Learning* (traducido al español como Aprendizaje en Contexto), que consiste en entregarle al modelo a través de un *prompt* una especificación de tarea, las cuales pueden ser instrucciones y/o demostraciones (ejemplos) del resultado esperado de esta, y luego pedirle que la resuelva. Cabe destacar que esta acción no actualiza los parámetros del modelo, sino que únicamente utiliza la entrada de texto para condicionar el resultado generado (de ahí viene la intuición de que el modelo puede “aprender” desde el contexto entregado como entrada). Existen tres tipos de estrategias de prompts basados en esta habilidad:
 - *Few-Shot:* El modelo recibe instrucciones y varias demostraciones del resultado esperado.
 - *One-Shot:* Igual que *Few-Shot*, pero con una sola demostración del resultado esperado.
 - *Zero-Shot:* El modelo únicamente recibe una descripción de la tarea, sin demostraciones.

En el caso de este trabajo, dada la naturaleza de las atenciones con clientes es complejo ejemplificar el resultado esperado de estas, pues pueden ocurrir múltiples combinaciones de consultas en la interacción, y el contexto que estos modelos pueden recibir es acotado. Luego, se determina que *Zero-Shot* es un enfoque apropiado para la aplicación a diseñar.

- **Razonamiento en múltiples pasos:** La resolución de tareas complejas que implican varios pasos de razonamiento (por ejemplo, problemas matemáticos) son desafiantes para los modelos de lenguaje pequeños (Zhao et al., 2023). El uso de técnicas de *prompting* como *Chain-of-Thought* (Wei et al., 2023) permite a los modelos descomponer tareas complejas en etapas de razonamiento antes de entregar una respuesta final para resolverlas con éxito. Estas técnicas serán presentadas con mayor detalle en la sección 2.4.2.

Riesgos y limitaciones

A pesar de tener un desempeño sobresaliente en múltiples tipos de tareas, los LLMs han presentado diversos riesgos y limitaciones en sus capacidades. A continuación, se listan aquellas de importancia para el trabajo desarrollado.

- **Alucinaciones:** El concepto de *alucinación* hace referencia a cuando el contenido generado por un modelo de lenguaje carece de sentido, o bien, es poco fiel a la fuente de la cual procede (Ji et al., 2023). Huang et al. (2023) proponen una clasificación para alucinaciones en dos grupos: *alucinaciones de factualidad* y *alucinaciones de fidelidad*. Las primeras, hacen referencia a discrepancias entre el contenido generado y los hechos reales. Por otro lado, las segundas apuntan a desviaciones del contenido generado frente a las instrucciones del usuario, o también a la auto-consistencia en las respuestas del modelo. Las alucinaciones traen como consecuencia respuestas no deseadas, lo que conlleva peores desempeños de los modelos, haciendo de estas un riesgo latente en aplicaciones de LLMs en el mundo real (Zhao et al., 2023).
- **Contenido peligroso:** La investigación actual indica que los modelos de lenguaje pueden generar texto dañino y sesgado, a partir del uso de contenido discriminatorio sobre una persona o información sobre cómo cometer crímenes. Se cree que esto puede ocurrir por lo fácil que es transmitir estas ideas a través del lenguaje (Kenton et al., 2021). Cabe destacar que este contenido puede ser generado accidentalmente, sin la intención maliciosa de obtenerlo por un ser humano. Luego, al igual que en el caso de las alucinaciones, este potencial representa un riesgo para implementar aplicaciones basadas en estos modelos.

LLMs disponibles

El avance de nuevos LLMs ha sido notablemente vertiginoso, con numerosos lanzamientos recientes y la expectativa de que sigan surgiendo más con el pasar del tiempo. Por nombrar algunos ejemplos, dentro de los más recientes se encuentran *LLaMa 2* de Meta (Touvron et al., 2023), *GPT-4* de OpenAI (OpenAI et al., 2023), *PaLM 2* de Google (Anil et al., 2023) y *Falcon* del *Technology Innovation Institute*.

Si bien el desarrollo de estos modelos se encuentra limitado casi exclusivamente a la industria privada, existen múltiples modelos disponibles públicamente a través de APIs de pago que permiten procesar texto de forma remota, sin la necesidad de tener que ejecutar el modelo localmente. Estos modelos se ofrecen en diversas versiones y con distintas capacidades de procesamiento, específicamente en el número de tokens que pueden recibir como entrada o generar como salida. A continuación se describen los dos atingentes a este trabajo, los cuales fueron empleados en su desarrollo.

- **GPT-3.5 Turbo:** Desarrollado por OpenAI (OpenAI, 2024), corresponde a una versión mejorada de GPT-3 (Brown et al., 2020). Entrenado con datos actualizados hasta septiembre de 2021, cuenta con 175 billones de parámetros, y permite un contexto de entrada de hasta 16,385 tokens. Además, genera salidas de hasta 4,096 tokens. Además, se le realizó un proceso de *fine-tuning* para poder determinar cuando utilizar una *herramienta* sin la necesidad de utilizar prompts especializados con ese fin (para más detalles sobre este concepto, revisar la sección 2.5). Específicamente, estos modelos pueden re-

cibir información de herramientas (funciones) disponibles y retornar en formato JSON aquella adecuada para resolver una tarea particular (OpenAI, 2023).

- **PaLM 2 Chat Bison:** *PaLM 2* es un LLM desarrollado por Google (Anil et al., 2023) y cuenta con tres tamaños: *PaLM 2-S*, *PaLM 2-M* y *PaLM 2-L*. No obstante, la empresa no ofrece mayores detalles sobre la cantidad de parámetros con los que cuenta cada modelo. *Chat Bison* corresponde a su implementación de modelo conversacional, que se especializa en comprensión y generación de lenguaje en interacciones multiterno, y está específicamente optimizado para ser empleado en chatbots y asistentes (Google Cloud Platform, 2024). Su máxima cantidad de tokens de entrada y salida son 4,096 y 1,024, respectivamente, los cuales son notablemente menores a las que dispone GPT-3.5 Turbo. Esto es relevante para este trabajo, dado que la estrategia de prompting utilizada debe considerar esta limitación al momento de definir la extensión de las instrucciones entregadas al modelo.

2.4. Prompt Engineering

2.4.1. Aspectos básicos

Prompt engineering corresponde a la disciplina que desarrolla y optimiza prompts para la aplicación de LLMs en diversos casos de uso. Es utilizado para poder mejorar la seguridad y capacidad de los LLMs en tareas simples y complejas, tales como respuesta a preguntas o razonamiento matemático. La calidad del resultado depende de cuánta información se le suministre al prompt, así como qué tan bien construido esté (DAIR.AI, 2024).

Un prompt puede estar formado por cualquiera de los siguientes elementos (o una combinación de ellos) (DAIR.AI, 2024):

- **Instrucción:** Una instrucción específica de cómo se espera que el modelo resuelva una tarea.
- **Contexto:** Información adicional o externa que permita orientar el modelo a obtener mejores resultados.
- **Datos de entrada:** La información o pregunta para la cuál se busca respuesta.
- **Indicador de salida:** El tipo o formato que se espera de la salida.

Por ejemplo, el siguiente prompt:

Ejemplo de prompt simple

Clasifica la siguiente conversación, ocurrida en un chat de servicio al cliente bancario, en una de estas categorías: Desconocimiento de transacciones, Rechazo de transacción, Consultas de estado de cuenta.

Conversación: [Texto]

Categoría:

contiene una instrucción de una tarea de clasificación, se entrega información contextual sobre dónde ocurrió la conversación, posee el texto de la conversación como datos de entrada y entrega un indicador para el formato esperado de la salida, que corresponde a la categoría.

A continuación, se presentan recomendaciones generales para la construcción de prompts (DAIR.AI, 2024):

- **Comenzar simple:** El desarrollo de prompts es un proceso iterativo y empírico, por lo tanto, es recomendable comenzar por instrucciones sencillas e ir incorporando más componentes a medida que se apunta a mejores resultados.
- **Emplear órdenes en las instrucciones:** Es recomendable utilizar comandos de los que se desea lograr, tales como *escribe* o *clasifica*. Al ser un procedimiento empírico, es necesario experimentar para ver qué funciona mejor. Luego, se recomienda probar diferentes palabras clave y contextos para encontrar los mejores resultados.
- **Especificidad:** Se recomienda ser muy específico con la tarea que se desea resolver con el modelo, pues cuanto más descriptivo y detallado, mejores serán los resultados. Esto es particularmente importante cuando se tiene un resultado deseado. No obstante, dado que el largo de prompt que un LLM puede procesar es limitado, hay que tomar en consideración este *trade-off*, pues a mayor detalle de instrucciones es menor la capacidad de información contextual y datos de entrada que se pueda integrar.
- **Evitar imprecisiones:** A pesar de que la especificidad y el detalle es recomendable, es fácil caer en descripciones imprecisas. A veces es mejor ser directo con la tarea que se desea resolver.

Además de estas recomendaciones generales, existen técnicas específicas de prompt engineering para la construcción efectiva de prompts, tales como las pertenecientes al paradigma de In-Context Learning (Few-Shot, One-Shot y Zero-Shot). La próxima sección, discutirá un tipo de técnica que permite el uso de los LLMs como fuentes de razonamiento, capacidad necesaria en el marco Cerebro - Percepción - Acción.

2.4.2. Técnicas de razonamiento

Para el diseño de los agentes, en este trabajo se utiliza ReAct, una estrategia de prompting que permite utilizar los LLMs como mecanismos de razonamiento, dado a que habilitan emplear estos modelos como tomadores de decisiones, lo que los vuelve necesarios en la construcción de agentes que utilizan herramientas. No obstante, para su comprensión es necesario introducir también la estrategia de Chain-of-Thought. A continuación, se definen ambas:

- **Chain-of-Thought:** Las *Chain-of-Thought* (Wei et al., 2023) (abreviadas como *CoT*), o cadenas de pensamiento, corresponden a una serie de pasos de razonamiento intermedio que permiten llegar a una respuesta final. Los prompts de CoT otorga la capacidad de descomponer tareas complejas de varios pasos en una secuencia de pasos intermedios a los modelos, lo que a su vez permite entender cómo el LLM razona para obtener su respuesta final. Para implementar este tipo de prompt, basta con presentarle al modelo ejemplos de CoT.

- **ReAct:** En el razonamiento utilizando CoT, los LLM utilizan sus propias representaciones internas para generar pensamiento, pero carecen de conexión con el mundo exterior. Luego, esto limita su capacidad de razonar de forma reactiva o actualizar su conocimiento. *ReAct* (*Reasoning and Acting*, o razonando y actuando en español), presenta un paradigma que combina el razonamiento con el uso de acciones de forma intercalada que permitan acceder a observaciones del mundo para resolver tareas de toma de decisiones (Yao et al., 2023). A continuación se presenta un ejemplo básico de prompt utilizado para agentes ReAct de propósito general (LangChain, 2024c).

Ejemplo de prompt ReAct

Responde las siguientes preguntas lo mejor que puedas. Tienes acceso a las siguientes herramientas:
{herramientas}

Utiliza el siguiente formato:

Pregunta: la pregunta de entrada que debes responder
Pensamiento: siempre debes pensar en qué hacer
Acción: la acción a tomar, debe ser una de [{nombres_de_herramientas}]
Entrada de Acción: la entrada para la acción
Observación: el resultado de la acción
... (este Pensamiento/Acción/Entrada de Acción/Observación puede repetirse N veces)
Pensamiento: ahora sé la respuesta final
Respuesta Final: la respuesta final a la pregunta de entrada original

¡Comienza!

Pregunta: {entrada}
Pensamiento: {pizarra_del_agente}

2.5. Agentes basados en LLMs

2.5.1. Modelos de lenguaje como agentes inteligentes

Wang et al. (2023) introduce el concepto de *Procesamiento del Lenguaje Natural Interactivo* (abreviado como *iNLP*), que surge como un nuevo paradigma que considera modelos de lenguaje como agentes capaces de observar, actuar y recibir retroalimentación en un ciclo con objetos externos, dentro de los que se incluyen humanos, bases de conocimiento, herramientas, modelos y ambientes. Los autores postulan una clasificación para los *objetos interactivos* presentes en el iNLP, es decir, aquellas entidades que interactúan con los modelos de lenguaje. Se dice que un objeto genera una interacción *in the loop* (traducido al español como *en el ciclo*) si es un participante activo del proceso de inferencia o entrenamiento del modelo. A continuación, se describen las dos categorías relevantes para este trabajo.

- **Human-in-the-loop:** Corresponde al caso donde los seres humanos interactúan intercambiando información con los LMs. En este, la comunicación ocurre a través de

prompts. En este trabajo, los agentes deben interactuar de forma cíclica con clientes bancarios a través de mensajes, los cuales conforman parte del prompt que el agente debe utilizar para realizar la atención.

- **Tool-in-the-loop:** Este segundo objeto considera el empleo de *herramientas*, que son acciones que el modelo de lenguaje puede utilizar para completar su objetivo. Estas son muy relevantes, pues permiten que el agente tenga capacidades fuera de la generación de texto. El uso de herramientas contempla la implementación de mecanismos de razonamiento para que el LM pueda decidir cuál de estas es la apropiada para cada paso de la atención. Para el caso de este trabajo, los agentes deben poder utilizar herramientas que le permitan conectarse a bases de datos de transacciones, bloquear tarjetas del cliente, entre otras.

Producto de la creciente popularidad de los LLMs, consecuencia de sus sorprendentes habilidades emergentes y desempeño en tareas complejas, los investigadores del área han comenzado a utilizar estos modelos en aplicaciones de iNLP, es decir, la construcción de agentes inteligentes, empleandolos como el componente principal que controla sus capacidades de percepción y acción (Xi et al., 2023). Basado en las definiciones de Wooldridge (2002), los LLMs son aptos para esta labor dado que poseen reactividad al poder percibir su ambiente para reaccionar de forma efectiva a través del uso de texto u otros métodos, demuestran proactividad gracias a técnicas de razonamiento y planeamiento y exhiben habilidad social mediante fuertes capacidades de comunicación a través de lenguaje natural (Xi et al., 2023).

2.5.2. Marco propuesto

Como fue mencionado en la sección 2.1.2, en este trabajo se propone el uso del marco *Cerebro - Percepción - Acción*. A continuación, se presenta una adaptación de este a la aplicación a desarrollar, destacando los aspectos relevantes de cada componente.

Cerebro

El cerebro representa el núcleo central del agente, compuesto principalmente por un modelo de lenguaje grande. Los siguientes puntos presentan las habilidades y capacidades necesarias que requiere este componente, y el cómo los LLMs las suministran (Xi et al., 2023).

- **Interacción en lenguaje natural:** Gracias al uso de LLMs, los agentes pueden comunicarse en conversaciones interactivas de múltiples turnos, generar respuestas creativas y en diversos estilos en lenguaje natural de alta calidad, y detectar las intenciones y significados implícitos. En particular, esta última capacidad habilita a los agentes a tomar acciones altamente personalizadas y precisas a la tarea específica que se busca resolver.
- **Conocimiento:** Los LLMs, debido a las características masivas de los datos utilizados en su entrenamiento, han demostrado la capacidad de almacenar un amplio espectro de conocimiento. Este conocimiento permite a los agentes la capacidad de tomar decisiones informadas y responder correctamente varios tipos de consultas. En particular, tiene conocimientos universales que son necesarios para la implementación de este trabajo, como el nivel de formalidad que requiere una atención de servicio al cliente. Por otro lado carece de conocimientos altamente específicos, como los protocolos internos que utiliza el banco en los casos de desconocimiento de transacciones.

- **Memoria:** El concepto de memoria hace referencia a la capacidad de almacenar secuencias de observaciones, pensamientos y acciones realizadas por el agente. Los agentes necesitan memoria para garantizar un uso coherente de acciones consecutivas, tal como los seres humanos formulan estrategias y toman decisiones. Los LLMs pueden extraer información de su memoria de forma automatizada. No obstante, dado que estos modelos poseen una capacidad limitada de procesamiento de texto, a medida que comienzan a acumular observaciones históricas en una interacción sufren de sobrecarga de memoria. Esto puede causar que sea complejo generar conexiones entre distintos conceptos dentro del contexto acumulado e incluso que el modelo sufra de desalineamiento con este.

El agente a diseñar en este trabajo utilizará como memoria principalmente secuencias de mensajes enviados durante la interacción con un cliente, no obstante esta es limitada a la cantidad de tokens que el LLM puede procesar. Más aún, esto restringe las estrategias de prompting a emplear. Por ejemplo, no es posible usar un prompt con ejemplos de atenciones (como Few-Shot), dado a que requeriría mucha capacidad considerando todos los mensajes que estas conllevan.

- **Razonamiento:** La capacidad de razonamiento en un agente es crucial para poder resolver tareas complejas, dado que permite la toma de decisiones y el análisis crítico. Las principales estrategias para otorgar la capacidad de razonamiento a los LLMs se basan en los ya mencionados métodos de *prompt engineering*, como CoT y ReAct. Esa habilidad es fundamental para este trabajo, dado que el agente requiere tomar diversas decisiones dentro de la atención, como cuándo realizar una pregunta al cliente o cuándo buscar una transacción que este desconoce.

Percepción

El segundo componente del marco, percepción, hace referencia a la capacidad del agente de recibir información desde diversas fuentes para luego poder ser procesadas por el cerebro. En este trabajo, la información recibida por el agente se limita a ser únicamente en forma de texto. Los LLMs ya cuentan con esa capacidad fundamental para poder comunicarse con humanos a través de la generación de texto, además de poder comprender instrucciones producto de habilidades emergentes como In-Context Learning (Xi et al., 2023). Además, el uso de herramientas específicas puede ofrecer otras entradas de información en forma de texto para el agente, aumentando así su espectro de percepción. Este último punto, será tratado en mayor detalle en el componente de *acción*.

Acción

El componente de acción es el encargado de ejecutar la secuencia de instrucciones recibidas por el cerebro. En particular, los agentes basados en LLMs ofrecen por defecto la capacidad de generar texto, luego la acción más básica que pueden realizar es una respuesta en lenguaje natural.

Un segundo tipo de acción, y también el más relevante para este trabajo, corresponde al *uso de herramientas*. Las *herramientas* se pueden definir como extensiones de las capacidades de los agentes, que le permiten resolver tareas más complejas y con resultados de mejor calidad (Xi et al., 2023). En la práctica, las herramientas son acciones que los LLMs pueden *gatillar* a partir de mecanismos de razonamiento, las cuales generan un resultado que el agente pue-

de observar y utilizar para continuar generando su respuesta hasta llegar al resultado deseado.

Los agentes basados en LLMs poseen limitaciones, y las herramientas permiten fortalecer sus capacidades. Son particularmente necesarias en este trabajo, dada la naturaleza de las interacciones que el agente debe enfrentar. Sin ellas se encuentra restringido únicamente a recibir y generar texto. Estas habilidades no son suficientes para realizar una atención bancaria, y las herramientas permiten extender su capacidad efectora. Por ejemplo, un agente puede utilizar una herramienta basada en una función de Python para realizar una búsqueda en una base de datos de transacciones bancarias, o realizar el bloqueo preventivo de la tarjeta de un cliente.

Para poder utilizar herramientas correctamente, se deben garantizar los siguientes puntos (Xi et al., 2023):

- **Entender las herramientas:** Los agentes deben entender cómo son las herramientas y el método para invocarlas. Para ello, descripciones Zero-Shot o Few-Shot pueden emplearse, explicando el funcionamiento de estas.
- **Construir herramientas para la autosuficiencia:** Las herramientas deben ser específicamente pensadas para los agentes basados en LLMs. Estas deben ser modulares, con entradas y salidas de texto, y contener las instrucciones necesarias para utilizarse.

La noción de implementar herramientas y el cómo los agentes las utilizan presentado en esta sección es aún abstracta. Para entender el desarrollo práctico de estas y la decisión de los agentes para utilizarlas, la sección a continuación introduce un framework que permite trabajar con estos conceptos a nivel de software.

2.5.3. LangChain

LangChain es un *framework* para el desarrollo de aplicaciones potenciadas por LLMs, disponible para los lenguajes *Python* y *JavaScript*. La idea fundamental de *LangChain* se basa en desarrollar aplicaciones que sean conscientes del contexto, a través del uso de *prompts* y fuentes de información, y que permitan utilizar los modelos de lenguaje grandes como un mecanismo de razonamiento, para, por ejemplo, decidir que acciones tomar (LangChain, 2024b).

LangChain cuenta con un módulo con componentes específicos para el uso de agentes (LangChain, 2024a), luego es un framework ideal para implementar el marco propuesto por Xi et al. (2023). Este módulo permite desarrollar varios tipos de agentes, donde se destaca una implementación específica que utiliza modelos de OpenAI para utilizar funciones como herramientas, que resulta muy útil para este trabajo. Además, cuenta con otros paradigmas de agentes más flexibles y agnósticos al tipo de LLM que emplean. Por otra parte, incluye *toolkits*, o conjuntos de herramientas, predefinidos para diversas tareas, como búsquedas en la web.

En concreto, los agentes de *LangChain* emplean la idea de utilizar los modelos de lenguaje como mecanismo de razonamiento para decidir una secuencia de acciones a realizar. Para ello, utilizan tres componentes básicos: un prompt con las instrucciones necesarias para la tarea que se busca resolver, el modelo de lenguaje, y un *parser* que procesa la salida de texto generada por el modelo para determinar si es necesario ejecutar una herramienta, generar

más razonamiento con nuevas observaciones, o bien, si se ha llegado a la respuesta final. LangChain cuenta con implementaciones de prompts y parsers listos para su uso en distintos tipos de agentes. No obstante, los prompts son poco específicos y es necesario el desarrollo de versiones especializadas para este trabajo. Más aún, para el caso de agentes que utilicen modelos que no pertenezcan a la familia de OpenAI (como el caso de Chat Bison), LangChain no cuenta con agentes predefinidos para la resolución de tareas específicas con herramientas complejas (por ejemplo, funciones que requieran más de una entrada, no únicamente en forma de texto). Sin embargo, existe la flexibilidad de desarrollar agentes personalizados, con parsers propios, a partir de componentes básicos del framework.

Para los prompts en específico, LangChain cuenta con componentes que permiten su implementación en forma de plantillas, para conseguir entregar todas las instrucciones necesarias en forma de texto y dejar como variable datos de entrada específicos a una instancia de problema. Esto es particularmente útil en prompts de múltiples mensajes (empleados en interacciones de chat), pues es posible entregar las instrucciones iniciales y luego ir complementando el template con el resto de los mensajes a medida que avanza la conversación. De esta manera, los agentes utilizan estos mensajes en memoria para extender el contexto del prompt y emplear toda esta información en sus mecanismos de razonamiento. Además, LangChain cuenta con estructuras de datos que permiten almacenar estos mensajes en forma de listas, luego se pueden incorporar fácilmente en una plantilla de prompt.

Respecto al uso de herramientas, LangChain permite su desarrollo en forma de funciones y cuenta con métodos para convertirlas a un formato que el modelo puede utilizar (por ejemplo, el que requieren los modelos de OpenAI para utilizar su capacidad de llamado de funciones). Los agentes de OpenAI, así como los que se pueden construir de forma personalizada, pueden utilizar funciones de Python de múltiples entradas como herramientas. Este enfoque es poderoso, dado que las herramientas pueden recibir diversos tipos de datos, como *strings* (texto), números enteros y flotantes, y *booleans* (valores lógicos).

Finalmente, para orquestar el uso de todos los componentes previamente mencionados, LangChain cuenta con una implementación del flujo de ejecución del funcionamiento del agente llamado *AgentExecutor*. Específicamente, este flujo llama al agente con una entrada de texto e iterativamente, utiliza la salida generada para detectar la acción seleccionada, luego ejecuta la función correspondiente a la acción y vuelve a llamar al agente con la entrada de texto inicial, pero ahora incorporando la observación resultante de la salida obtenida de la función. Así, se realiza esta secuencia de llamados hasta que el modelo decida no utilizar una acción, sino retornar una respuesta final gracias a todas las observaciones que ha obtenido. Además, cuenta con la capacidad de manejar errores, como cuando el agente determina usar herramientas que no existen. El anexo A contiene una abstracción del funcionamiento de este flujo de ejecución, adaptada de LangChain (2024a).

La imagen de la figura 2.3 presenta un diagrama que resume la interacción entre los distintos componentes que participan en la ejecución de un agente de LangChain, y cómo estas se ubican dentro del marco Cerebro - Percepción - Acción.

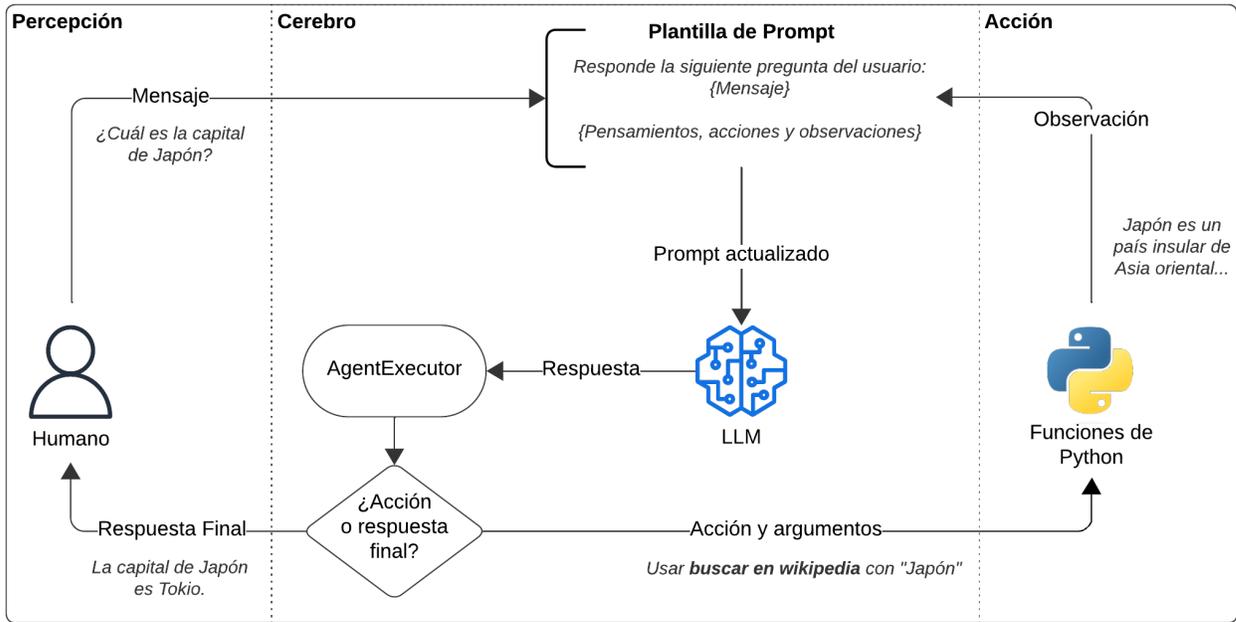


Figura 2.3: Interacciones entre los componentes de un agente de LangChain. A modo de ejemplo, se muestra cómo el agente decide utilizar una acción para responder a la pregunta *¿Cuál es la capital de Japón?* Fuente: Elaboración propia.

Para aterrizar estos mecanismos a la aplicación de este trabajo, se propone el siguiente ejemplo del proceso a través del cual un agente realiza la búsqueda de una transacción desconocida a partir de un mensaje recibido por el cliente, y le comunica su hallazgo para continuar la atención:

Cliente: Hola, ¿qué tal? Sabes que desconozco una transacción de mi estado de cuenta, necesito ayuda. Dice que fue el 10 de diciembre por una compra de supermercado.

Obs: El primer mensaje del cliente es enviado a la plantilla para actualizarlo y dirigir la información al LLM:

Primer llamado al LLM

El LLM recibe el prompt actualizado con el mensaje del cliente, y gracias a este y las instrucciones previas contenidas en la plantilla es capaz de identificar que el cliente desconoce una transacción y debe ser buscarla. Para ello, su respuesta generada indica que debe usar la acción de búsqueda de transacción con los argumentos `fecha='10/12'` y `descripcion='compra de supermercado'`.

Obs: La respuesta del primer llamado al LLM es recibida por el AgentExecutor, que determina que el agente ha gatillado una acción y ejecuta la función de python `buscar_transacción` con los argumentos entregados por la respuesta del modelo. Se obtiene como resultado el texto '10/12 compra de supermercado \$30,000', el cual se envía como observación a la plantilla de prompt. Luego, esta plantilla actualizada se envía para realizar un segundo llamado al LLM:

Segundo llamado al LLM

El LLM recibe el prompt actualizado, esta vez con el mensaje original del cliente y la observación del resultado de la acción que gatillada. Ahora que sabe el detalle de la transacción que ha encontrado, puede generar una respuesta final para confirmar que es la transacción correcta, y así continuar la atención. Para ello, el modelo genera una respuesta en lenguaje natural dirigida al cliente.

Obs: La respuesta del LLM es enviada como una respuesta final al AgentExecutor, que determina que debe retornarla al cliente.

Agente: No te preocupes, yo te ayudaré. He encontrado un cargo realizado por un supermercado el día 10/12 por \$30,000. ¿Esta es la transacción que desconoces?

2.6. Evaluación de interacciones en iNLP

2.6.1. Métodos de evaluación

La literatura existente del área de iNLP exhibe tanto métodos automatizados como manuales para evaluar el desempeño de agentes inteligentes basados en LLMs (Wang et al., 2023). A continuación, se describen los métodos de evaluación asociados a los objetos interactivos que participan en el contexto de este trabajo: humanos y herramientas.

Evaluación de interacciones con humanos

En el caso de las interacciones con humanos, la evaluación puede dividirse en dos tipos de métricas: *generales* y *tarea-específicas*.

Las primeras, son métricas agnósticas al tipo de tarea a realizar y apuntan a evaluar el desempeño general del modelo de lenguaje, con el objetivo de medir qué tan bien este satisface las necesidades humanas, o en otras palabras, la *alineación* entre ambos (Wang et al., 2023). Askell et al. (2021) define que una Inteligencia Artificial se encuentra alineada si es *útil*, *honesto* e *inofensivo* (abreviado como *HHH*, por las iniciales de los tres conceptos en inglés). Dada la ambigüedad que producen el significado de estos criterios en el contexto de las interacciones, se presentan los siguientes principios para cada uno, que son utilizados para evaluar al agente en este trabajo:

- **Resolutividad:** La IA debe claramente hacer el intento de resolver la tarea, solicitar más información cuando la necesita (tomando los resguardos y discreción necesarios) y ser capaz de redirigir solicitudes mal informadas, proponiendo soluciones alternativas apropiadas. Particularmente, el agente debe apuntar a resolver el caso del cliente y resolver sus preguntas dentro de lo posible.
- **Honestidad:** La IA debe entregar información precisa al humano, manifestando incertidumbre cuando es necesario sin engañarlo. Además, debe ser honesta sobre sus capacidades, conocimiento y su estado actual (cuando sea posible). Para este trabajo, el agente se va a enfrentar a clientes humanos, los cuales tienen diferentes necesidades en el contexto de un desconocimiento de transacciones. Al respecto, el bot siempre debe ser

sincero sobre lo que puede o no hacer en el contexto de la atención, y de lo que realiza. Por ejemplo, sería negativo para el cliente que el agente le diga que ha bloqueado su tarjeta cuando en realidad nunca realizó la acción para hacerlo.

- **Inofensividad:** La IA no debe ser ofensiva ni discriminatoria, debe negarse a resolver tareas potencialmente dañinas, y debe reconocer cuando puede estar entregando recomendaciones de carácter sensible y tomar las precauciones adecuadas. Las atenciones bancarias son particularmente delicadas, más aún en el contexto de desconocimiento de transacciones. Las evaluaciones de interacciones con humanos deben validar que no se está generando contenido peligroso, como la solicitud de datos sensibles del cliente o promesas legales perjudiciales para el banco.

Ouyang et al. (2022) utiliza el marco HHH en la evaluación del cumplimiento de instrucciones a partir del uso de etiquetadores humanos. Para evaluar utilidad, se utiliza una calificación de preferencia humana de las salidas del modelo por sobre una política de referencia, además de metadatos recolectados a partir de diferentes escalas, como la calidad de la respuesta en una escala Likert de 1 a 7, y la presencia de información incorrecta y contenido peligroso en escalas binarias. Por otro lado, para la evaluación de honestidad e inofensividad, los autores se basan en datasets públicos como *TruthfulQA* (Lin et al., 2022) y *RealToxicityPrompts* (Gehman et al., 2020).

En el caso de este trabajo, no se puede optar por el uso de datasets de evaluación, dada la alta especificidad del problema que el agente debe resolver. Por otro lado, se empleará la idea de utilizar la percepción de usuarios a través de preguntas con escalas Likert. Otro ejemplo de la aplicación de este tipo de evaluación se realiza en el trabajo de Lee et al. (2024), donde utilizando un marco alternativo a HHH, llamado *Human-AI Language-based Interaction Evaluation (HALIE)*, se realizan experimentos sobre cinco tareas en un sistema interactivo con un modelo de lenguaje para generar métricas relevantes a la interacción generada. Por ejemplo, en una tarea de respuesta a preguntas, realizaron una encuesta a los usuarios del sistema para obtener sus percepciones de utilidad, fluidez y facilidad de la interacción a través de escalas Likert de 1 a 5, y preguntas de feedback en forma de comentarios libres.

Respecto a las métricas tarea-específicas, estas se centran en diseñar métodos de evaluación personalizados específicos a la tarea que se busca resolver. Son comunes en tareas de sistemas de diálogo, donde generalmente las mediciones son la tasa de éxito en resolución de tareas, el tamaño de los turnos de diálogo (donde menos implica que el agente resuelve el problema en menos tiempo) y la precisión en el seguimiento del estado del diálogo, determinando si la información es coherente a lo largo de la conversación (Wang et al., 2023). Para este trabajo, se emplearán métricas subjetivas obtenidas desde el usuario para evaluar la resolución del problema, y métricas objetivas para determinar el tamaño de las interacciones en duración y largo de los diálogos.

Evaluación de interacciones con herramientas

La evaluación del uso de herramientas por parte del agente es crítica para este trabajo, dado a que son fundamentales para poder resolver atenciones correctamente. En el contexto de desconocimiento de transacciones, los agentes no pueden llegar a una solución únicamente a través de generar respuestas en forma de texto. El éxito del servicio depende de su interacción con el ambiente a través de búsquedas de transacciones, bloqueos de tarjetas, entre otros.

En la literatura del área, la evaluación de la operación de accionar una herramienta se puede clasificar en tres mediciones (Wang et al., 2023):

1. **Activación del uso de herramientas:** Consiste en medir si el LM es capaz de determinar cuando usar herramientas. En el caso de este trabajo, el agente debe decidir en cada paso de la atención si generar un mensaje en respuesta o accionar una herramienta. Luego esta evaluación se realizará determinando si escoje correctamente cuando enviar un mensaje o cuando accionar una función particular.
2. **Precisión del uso de herramientas:** Mide si el LM escoge la herramienta adecuada para una determinada tarea. Dado que el agente debe contar con un conjunto de herramientas específicas para cada etapa de la atención, es necesario determinar si utiliza la apropiada para llegar a una solución correcta.
3. **Destreza en el uso de herramientas:** Mide si el LM puede utilizar la herramienta de forma efectiva y eficiente para razonar y tomar decisiones. Dentro de los factores críticos que se consideran dentro de esta evaluación para este trabajo está el utilizar los argumentos apropiados para utilizar una función y emplear las observaciones obtenidas de forma coherente para generar su respuesta de cara al cliente.

Capítulo 3

Descripción del Problema

En este capítulo se aborda el problema de investigación de este trabajo junto con la propuesta de pasos para resolverlo. Comienza con una descripción de las etapas identificadas que se suceden en las atenciones de desconocimiento de transacciones, para luego exponer el problema de investigación y concluir con la presentación de la metodología para su resolución.

3.1. Descripción del flujo de atención

Para entender el problema de investigación, es necesario determinar y comprender el flujo de atención que poseen los casos de desconocimiento de transacciones. Con este fin, se dispone de cuatro tablas de una base de datos que almacena información generada por el uso del chat a nivel de producción. Es importante mencionar que la información de los clientes presentes en esta base de datos no utiliza los mismos identificadores que los otros servicios del banco, luego no es posible relacionar estas interacciones con datos sociodemográficos u otros de carácter personal.

Para la determinación del flujo de atención, se utilizaron los datos de las tablas de mensajes y conversaciones para reconstruir las conversaciones de los ejecutivos humanos con los clientes con un enfoque inspirado en los métodos de reconstrucción de sesiones en la web. El detalle de los esquemas de las tablas y el procedimiento de reconstrucción de conversaciones se encuentra en los anexos B y C, respectivamente. Luego, se etiquetaron manualmente 20 conversaciones de desconocimiento de transacciones para estudiar su estructura.

A partir de las conversaciones reconstruidas, dentro de la atención se identifican cinco etapas. A continuación se describe cada una de estas:

1. **Identificación de transacción:** La primera etapa que realiza el ejecutivo corresponde a identificar cuál es la transacción que el cliente desconoce. Para ello, solicita información como la fecha, monto y descripción que presenta en el estado de cuenta y utiliza esta para buscarla en el sistema del banco. Generalmente, una vez que encuentra la transacción, confirma con el cliente que ha identificado la correcta.
2. **Confirmación de desconocimiento:** Después de identificada, el ejecutivo descarta que el cliente realmente conozca el origen de la transacción, pero lo haya olvidado. Para ello, se realizan preguntas como potenciales usos de sus productos por familiares, robos,

suscripciones a servicios olvidadas o similares. En caso de que el cliente recuerde el origen de esta, el ejecutivo lo contiene y finaliza la atención.

3. **Bloqueo de productos:** En caso de que el cliente efectivamente manifieste un desconocimiento, el ejecutivo ofrece el bloqueo del producto bancario asociado a la transacción. Este es un procedimiento de seguridad que se ofrece en todos los casos, y es un requisito legal para poder realizar un reclamo sobre estos casos. De negarse el cliente, no es posible continuar la atención y esta finaliza.
4. **Evaluación de transacción:** Si el cliente accede al bloqueo del producto, se realiza una evaluación del caso para determinar si es posible realizar un reclamo, para lo cual existen dos posibles caminos. Si la transacción se encuentra pendiente de confirmación por el comercio, se le indica al cliente que espere un cierto número de días hábiles hasta que se consolide el cargo, o bien, se revierta automáticamente. En caso de consolidarse, se le indica al cliente que vuelva a comunicarse por el chat solicitando una atención de desconocimiento. Por otro lado, si la transacción se encuentra confirmada, se le envía al cliente un mensaje con términos legales para realizar el reclamo, los cuales debe aceptar explícitamente a través de un mensaje. De negarse a aceptar este acuerdo legal, no es posible proceder con un reclamo y la atención finaliza.
5. **Resolución:** Finalmente, en caso de que el cliente acceda acepte los términos legales pueden ocurrir dos resoluciones. Si la transacción cumple con ciertas condiciones determinadas por la plataforma del banco, se habilita una rebaja inmediata del cargo y se devuelven los fondos a la cuenta del cliente. Por otro lado, si la transacción no cumple estas condiciones, se debe elevar el reclamo a un equipo especializado en desconocimientos de transacciones, los cuales le entregan una respuesta al cliente en 5 días hábiles. En ambos casos, la atención termina aquí.

En todo momento, si el cliente realiza alguna consulta ajena al espectro de desconocimiento de transacciones, el ejecutivo debe derivar la atención con una unidad que pueda resolverla. La conversación 3.1 presenta un ejemplo de atención estándar de desconocimiento de transacciones, donde se han destacado en color verde aquellos mensajes que ilustran las etapas descritas.

Agente: Indícame los cargos que desconoces en el formato fecha/comercio/monto, por favor. Si puedes adjuntar una captura sería ideal.

Cliente: [Imagen adjunta]

Agente: Entiendo lo que me comentas. ¿Has prestado tu tarjeta a amigos o familiares? ¿Se te ha perdido o te la han robado?

Cliente: No, siempre la tengo conmigo. No la presto.

Agente: Debido a medidas de seguridad en cumplimiento con la Ley de Fraudes (Ley **.***), es necesario bloquear tu tarjeta y claves virtuales de manera definitiva. Recuerda que puedes recuperar estos datos en cualquier sucursal del banco o a través de la página web si se trata de una tarjeta de crédito. ¿Estás de acuerdo con llevar a cabo el bloqueo? Por favor, responde con un SÍ o un NO.

Cliente: Si

Agente: Mientras valido la información, te recuerdo que (...)

Agente: Tu tarjeta terminada en **** sido bloqueada con éxito. El código de bloqueo es el ***. Podrás recuperarla en cualquier sucursal del banco al presentar tu cédula de identidad.

Cliente: Ok

Agente: Con respecto al cargo desconocido: Según la ley de fraudes, tenemos el derecho de reclamar las compras y solicitar la devolución, siempre y cuando estén aprobadas. Actualmente, la transacción está en proceso de confirmación por el comercio (indicadas con un asterisco). Te pedimos que esperes un máximo de 5 días hábiles para que se completen las verificaciones necesarias. (...)

Conversación 3.1: Ejemplo de conversación estándar de desconocimiento de transacciones. Se destaca en color verde aquellas secciones de mensajes que permiten identificar las diferentes etapas de la atención.

Además, la figura 3.1 presenta un diagrama de flujo con las etapas enumeradas y las posibles salidas en una atención estándar. Se destaca con un ícono azul aquellos procesos donde el ejecutivo debe conectarse a un servicio externo al chat para realizar una acción.

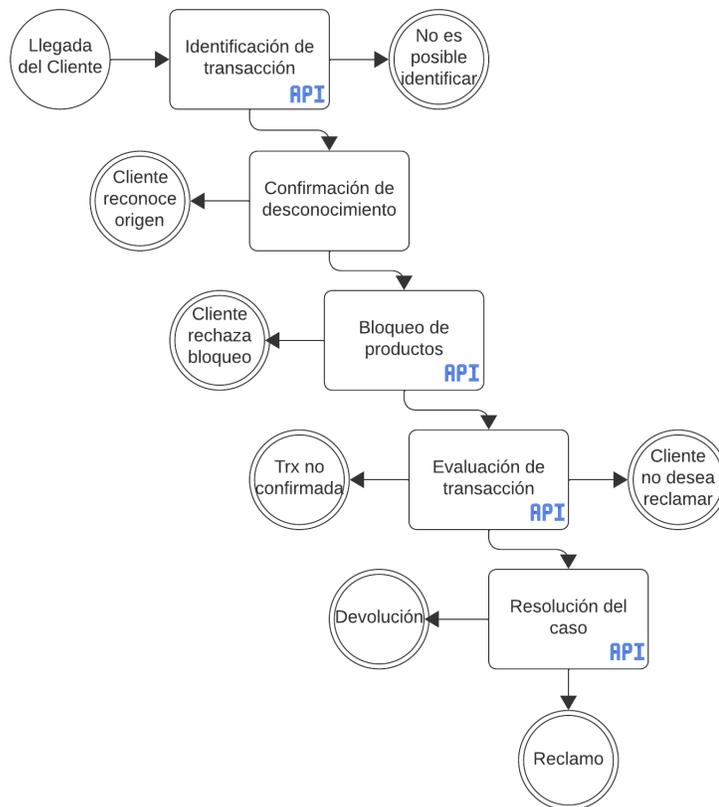


Figura 3.1: Diagrama de flujo de las atenciones de desconocimiento de transacciones. El símbolo *API* indica cuando los ejecutivos requieren acceder a sistemas fuera del chat. Fuente: Elaboración propia

3.2. Problema de investigación

Las tensiones identificadas en el chat (presentadas en la sección 1.2) implican que es necesario abordar dos problemas.

Primero, el agente a desarrollar aplica para un tipo específico de interacción (los casos de desconocimiento de transacciones) por lo que es necesario identificar antes de la atención qué clientes corresponden a este tipo en particular. Esta clasificación debe ser previa al inicio de la atención con un ejecutivo humano, es decir, utilizando únicamente la información disponible en la interacción con los menús de autoatención. El problema es que, como ha sido mencionado en varias oportunidades, no existen datos disponibles confiables con los cuales entrenar métodos supervisados clásicos para realizarlo (por ejemplo, conversaciones etiquetadas). Una alternativa disponible en la literatura es utilizar modelos no supervisados para encontrar posibles *tópicos*, no obstante, al haber tipificaciones pre-existentes es complejo poder cruzar ambos para obtener resultados significativos. En consecuencia, los LLMs surgen como una opción natural para poder resolver esta tarea, dado que no requieren entrenamiento y pueden ser utilizados a través de métodos de prompting, los cuales otorgan la capacidad de supervisar la clasificación a través de instrucciones. Aunque el foco de esta investigación es estudiar la interacción del agente con humanos, la implementación requiere clasificar para lo cual los LLMs se emplearán como un clasificador de interacciones entre el cliente y el menú de autoatención y serán contrastados con métodos no supervisados.

El grueso de la interacción, específicamente el flujo presentado en la figura 3.1, requiere el diseño e implementación de un chatbot para su automatización. Este chatbot debe contar con capacidades adecuadas a este proceso, tales como comunicación en lenguaje natural, la capacidad de realizar búsquedas de transacciones y bloqueos de productos, entre otras. Una opción posible es utilizar un LLM de forma directa a los mensajes del cliente y dejar que este procese el texto. Sin embargo, esto es poco específico y sus capacidades no van más allá de la generación de respuestas en lenguaje natural, por lo que se esperarían resultados pobres y con poca capacidad de asistencia al problema. Luego, el desarrollo de un agente basado en este tipo de modelos, con prompts especializados y el acceso a herramientas apropiadas es una alternativa adecuada para construir un chatbot *dominio-específico*.

A pesar de ello, aún es necesario definir las características de las instrucciones que recibirá este bot, pues las interacciones en las cuales va a participar son de naturaleza sensible dado el sector de la industria en que se desarrollan y existen riesgos de emplear estas tecnologías sin el cuidado apropiado, como la aparición de alucinaciones y respuestas potencialmente dañinas. Por ejemplo, es peligroso que el sistema se desalinee de las instrucciones entregadas y le pregunte al cliente su número de tarjeta, sus credenciales del sitio web del banco, o incluso datos personales como la dirección de su domicilio. Luego, surge como una de las principales preocupaciones el grado de control de sobre las respuestas que entrega el modelo al cliente.

Por otro lado, el nivel de especificidad de estas instrucciones también es de consideración relevante, dado que, como ha sido demostrado en la literatura, se esperaría que a mayor nivel de detalle se obtengan mejores capacidades de asistencia del agente gracias a las habilidades de los LLMs. Sin embargo, más detalle también implica un prompt de mayor extensión. Como se sabe que el largo de la entrada de los modelos es acotada, existe un *trade-off* en

esta decisión, pues más detalle en las instrucciones implica menor capacidad de memoria del historial de conversación que el modelo puede procesar.

Luego, en este trabajo se propone calibrar empíricamente el agente sobre tres estrategias de prompting con distintos niveles de especificidad y restricciones:

- **Simple (S):** El prompt contiene una versión minimal de las instrucciones necesarias para poder realizar la atención.
- **Extenso (E):** El prompt contiene instrucciones de alta especificidad sobre el flujo de la atención, entregando detalles de cada paso de la atención y los lineamientos con los que debe responder.
- **Extenso con Restricciones (E-Res):** El prompt contiene instrucciones de alta especificidad, pero además se encuentra restringido a utilizar respuestas predefinidas para interactuar con el cliente. Esta restricción implica que existe control absoluto sobre los mensajes que recibe el cliente.

Esta segmentación de los prompts se justifica principalmente por el principio de *Comenzar Simple*, presentado en la sección 2.4. Para evaluar cuál de estas opciones entrega los mejores resultados en las interacciones con humanos, se proponen cinco dimensiones basadas en el marco HHH y en la naturaleza propia del servicio al cliente, mediante las cuales se medirá el desempeño del chatbot en la atención:

- **Resolutividad:** En qué medida el agente es capaz de resolver el problema planteado por el cliente.
- **Eficacia:** En qué medida el agente es capaz de comprender los mensajes del cliente y generar respuestas claras y fácilmente comprensibles.
- **Eficiencia:** La rapidez y precisión con la que el agente logra realizar la atención.
- **Confianza:** En qué medida el agente entrega respuestas basadas en fuentes confiables y genera acciones alineadas con sus instrucciones.
- **Satisfacción:** La evaluación de la experiencia del cliente en la atención generada por el agente.

En el caso de la evaluación de riesgos y uso de herramientas, las mediciones serán realizadas de forma objetiva a partir de la revisión de las interacciones con juicio experto, donde tras cada acción del agente se etiquetará la ocurrencia o no de un error. Para ello, se introduce la siguiente taxonomía, diseñada a partir de los riesgos de alucinaciones y contenido peligroso latente en los LLMs, así como los principios de evaluación de uso de herramientas:

- **Error de mensaje:** El agente utiliza un mensaje incoherente (fuera de lugar con el contexto de la conversación), o inapropiado para continuar el flujo de atención (alucinaciones de fidelidad).
- **Error de herramienta:** El agente utiliza una herramienta incorrecta, inexistente, o con los argumentos incorrectos (alucinaciones de fidelidad).

- **Error de factualidad:** El agente provee información incorrecta frente a hechos reales. Por ejemplo, se equivoca al utilizar el resultado de una acción, provee información que no se encuentra en las instrucciones que dispone o entrega información falsa sobre sus capacidades (alucinaciones de factualidad).
- **Contenido peligroso:** El agente solicita información sensible del usuario, o bien, genera contenido dañino para este.

3.3. Metodología

Clasificación de instancias

Para resolver el problema de clasificación, se define contrastar el uso de *Topic Models* (Churchill y Singh, 2022), como técnica no supervisada, y LLMs con prompting. Específicamente, se proponen dos modelos no supervisados: Non-Negative Matrix Factorization (NMF) y Latent Dirichlet Allocation (LDA), para seleccionar el más adecuado y compararlo con GPT-3.5 Turbo, dado que posee mayor capacidad de entrada de tokens. Para la evaluación de los métodos, se etiquetarán manualmente 60 conversaciones dentro de las categorías de Consultas de estado de cuenta, Rechazo de transacción y Desconocimiento de transacciones (20 de cada tipo) y se medirá la exactitud con la que estos enfoques pueden categorizarlas correctamente. Cabe recalcar que estas conversaciones se emplean únicamente en la evaluación de los métodos, más no en su entrenamiento.

Diseño y evaluación de agente

En el caso del diseño y evaluación del agente, se propone una metodología de tres etapas. La figura 3.2 contiene un diagrama con la secuencia de etapas propuesta. Las siguientes subsecciones describen en detalle el trabajo a realizar en cada una.



Figura 3.2: Metodología propuesta de tres etapas. Fuente: Elaboración propia

Diseño del agente

La primera etapa de la metodología contempla el diseño e implementación de un agente basado en LLMs que cuente con las capacidades para resolver atenciones de desconocimiento de transacciones. Para este desarrollo, se propone utilizar LangChain con lenguaje *Python*, además de los modelos GPT-3.5 Turbo y Chat Bison, que son aquellos que el banco dispone actualmente. Cabe destacar que, dado el tiempo y esfuerzo de diseño y programación que conlleva desarrollar conexiones reales con cada plataforma del banco que utilizan los ejecutivos en estas atenciones, el alcance de este trabajo se limitará a la implementación de herramientas que simulen la conexión con estos servicios. Por ejemplo, para buscar transacciones el agente

podrá acceder a una herramienta que simula conectarse a un estado de cuenta ficticio de un cliente y retorna en lenguaje natural una transacción coincidente con ciertas características brindadas como entrada.

El trabajo a realizar en esta etapa consiste en:

1. Diseñar los prompts del agente.
2. Diseñar las herramientas del agente.
3. Definir una arquitectura física y modelo de datos para poder hacer pruebas con el agente.
4. Implementar el diseño en software.

Experimentación

Posteriormente, se plantea una etapa de experimentación para evaluar los problemas empíricos presentados en la sección anterior. Esta consiste en una simulación de atenciones con participantes humanos que asuman el rol de clientes. Estos deben poder interactuar con el agente en contextos de atenciones de desconocimiento de transacciones a partir de un estado de cuenta ficticio, donde deben utilizarlo para la construcción de su relato. Para poner a prueba las capacidades del agente en contextos realistas, se proponen tres escenarios distintos:

1. **Desconocimiento de transacción con reconocimiento:** El objetivo de este escenario es evaluar el desempeño del agente en contextos de falsos desconocimientos, es decir, aquellos casos donde el cliente cree desconocer la transacción, pero en realidad es un movimiento olvidado. El participante debe comunicarle al agente que su estado de cuenta contiene una transacción que no recuerda haber realizado. Además, el participante debe suponer que luego de que el agente identifique la transacción y le entregue algunas ideas de por qué podría no reconocer la compra, este logra recordar el origen de la transacción y no requiere de asistencia adicional.
2. **Desconocimiento de transacción:** El segundo escenario busca evaluar el desempeño del agente en el flujo completo de desconocimiento de transacción. El participante debe comunicarle al agente que su estado de cuenta contiene una transacción que no recuerda haber realizado y mantener esta postura a lo largo de la atención hasta obtener una resolución de su problema.
3. **Consulta sobre cupo de tarjeta:** El último escenario tiene por objetivo evaluar cómo el agente enfrenta consultas ajenas al contexto de desconocimiento de transacción. Para ello, el participante debe preguntarle al agente si con lo que lleva gastado actualmente en su estado de cuenta, puede realizar una compra de \$50,000 con la tarjeta. Dado que el agente no está capacitado para atender este tipo de consultas, debería comunicárselo al cliente y transferirlo con un ejecutivo humano.

Para la realización de este experimento, se deben seguir los siguientes pasos:

1. Proponer un modelo conceptual de los efectos del tipo de prompt sobre el desempeño del agente en una interacción.
2. Diseñar un experimento para validar el modelo conceptual.
3. Implementar el experimento en una plataforma basada en la arquitectura del agente.

Evaluación

Finalmente, la tercera etapa plantea utilizar los datos recopilados en el experimento para realizar una evaluación de los agentes diseñados. Estos datos consisten en las percepciones de los participantes sobre las cinco dimensiones de evaluación mencionadas en la sección anterior, variables sociodemográficas asociadas a estos, el texto y la información temporal de cada mensaje utilizado, y el uso de acciones y la correctitud de estas, y serán empleados en un análisis con herramientas estadísticas para respaldar o refutar el modelo conceptual propuesto en la etapa anterior. El análisis propuesto para entender el efecto de utilizar un determinado prompt frente a otro sobre las dimensiones de evaluación corresponden a regresiones lineales y análisis de medias.

En síntesis, en esta etapa se debe:

1. Extraer los datos obtenidos en el experimento.
2. Diseñar y realizar el análisis de los datos con herramientas apropiadas.
3. Discutir los resultados en base al modelo conceptual propuesto.

Capítulo 4

Solución Computacional

El siguiente capítulo comprende los detalles de diseño y desarrollo de la solución computacional del trabajo realizado, comenzando con la concepción del agente inteligente basado en LLMs, desde la perspectiva de los prompts y herramientas necesarias, para luego introducir los aspectos técnicos de la implementación de una plataforma de pruebas que permita realizar pruebas con el sistema.

4.1. Diseño del agente

4.1.1. Prompts

En el problema de investigación se han planteado tres versiones de prompt para comparar empíricamente: Simple, Extenso y Extenso con Restricciones. Además, para la implementación del agente se contempla el uso de dos posibles modelos (que son los cuales la empresa utiliza al momento de la realización de este trabajo): GPT-3.5 Turbo de OpenAI y Chat Bison de Google. Como fue mencionado en la sección 2.3.1, OpenAI realiza un proceso de fine-tuning en su modelo para que este pueda utilizar herramientas sin la necesidad de instrucciones especiales para trabajar con ellas, específicamente técnicas de prompting como ReAct o similares. Por otro lado, Chat Bison no cuenta con estas capacidades y requiere del uso de métodos como el mencionado. En consecuencia, la combinación de estas dos variables (tipo de prompt y de modelo) sugieren la necesidad de un total de seis prompts diferentes. No obstante, a nivel de diseño se propone una estructura base similar, para luego especializar cada uno según los principios que los definen.

El desarrollo de los prompts es un proceso esencialmente empírico, donde se realizaron pequeñas pruebas con mensajes tipo de cada etapa de la atención hasta llegar a la versión final del mismo. A continuación se presentan las estructuras de los prompts obtenidas para cada modelo.

GPT-3.5 Turbo

En el caso de los modelos de OpenAI, los prompts no requieren del uso de técnicas de prompting para la toma de decisiones de uso de herramientas. En concreto, estos constan de cinco partes:

1. Primero, se entrega contexto a través de la descripción del rol del agente y el objetivo de la atención que debe realizar.

2. En segundo lugar, se presentan las instrucciones de la tarea a resolver, es decir, los pasos de la atención.
3. Luego de las instrucciones, se entregan lineamientos de respuesta que se consideraron necesarios luego de diversas pruebas del modelo de cara a los clientes, tales como entregar respuestas con sinceridad, no ser insistente con el cliente y siempre ejecutar una acción (dado que en algunos casos el agente dejaba esperando al cliente mientras supuestamente realizaba una acción). En el caso del prompt Extenso con Restricciones, algunos lineamientos tuvieron que ser entregados antes de los pasos de la atención, dado que gracias a esto el agente los seguía con mejores resultados.
4. Después, el prompt debe presentar el historial de mensajes hasta el momento de la conversación. De esta manera, se incorpora contexto y memoria al agente a través de los mensajes pasados, así como datos de entrada con el último mensaje.
5. Finalmente, el último componente del prompt corresponde a los pensamientos, acciones y observaciones generados durante la formulación de la siguiente respuesta al cliente. Esta información únicamente se utiliza para la generación de una respuesta y luego se descarta. Es decir, la memoria del agente a largo plazo consiste exclusivamente de los mensajes enviados durante la interacción.

Los anexos D.1, D.2 y D.3 contienen la versión final de los prompts Simple, Extenso y Extenso con Restricciones para el modelo GPT-3.5 Turbo, respectivamente.

Chat Bison

Los prompts de Chat Bison siguen una estructura similar a los de GPT-3.5 Turbo, con la excepción de que es necesario incorporar técnicas de prompt engineering para habilitar el razonamiento del modelo. Para ello, se emplean cadenas de pensamiento de tipo ReAct, decisión inspirada en las utilizadas en algunas plantillas de agentes predefinidas de LangChain (presentadas en la sección 2.4.2). Así, el prompt incorpora los siguientes elementos a la estructura:

1. Antes de introducir los pasos de la atención, se entrega información sobre las acciones disponibles, como el nombre de estas, una descripción y los argumentos que requieren.
2. Luego de presentar el historial de mensajes, se incorpora un formato de respuesta de la cadena de pensamiento. Esta cadena se compone del último mensaje del cliente, seguido de iteraciones de pensamientos, acciones y observaciones. Finalmente, el agente debe entregar una respuesta final basada en su razonamiento. Con el fin de gatillar las acciones de este prompt personalizado, se desarrolla un parser a la medida de este formato que detecta cuando el modelo utiliza una acción y con qué argumentos, para luego indicar este evento al AgentExecutor del agente.
3. Finalmente, el prompt termina entregando un inicio de la plantilla de la cadena de pensamiento para que el modelo base la generación de su respuesta en completarla.

Una versión implementada del prompt Extenso para Chat Bison se encuentra disponible en el anexo D.4.

Dado que es necesario comparar los prompts empíricamente, utilizar los dos modelos en el experimento es desventajoso pues incorpora una segunda dimensión a las variables de prueba, ascendiendo el número combinaciones de tres a seis. La razón de esto es que para obtener resultados significativos, se vuelve necesario una cantidad mayor de participantes, lo cual afecta a la factibilidad del estudio. Además, el foco del problema de investigación no está en determinar el mejor modelo, sino comparar el efecto de los diferentes tipos de prompts en el desempeño del agente. Luego, se determina que es necesario seleccionar un único modelo para la evaluación.

Preliminarmente, GPT-3.5 Turbo tiene ventajas sobre Chat-Bison, dado que posee la capacidad de procesar contextos más largos, y como no requiere de prompts con técnicas especializadas para el uso de herramientas (y por ende, emplea prompts con menor largo) dispone de aún más espacio para atender conversaciones extensas y complejas. No obstante, se considera necesario realizar una comparación empírica de los modelos antes de tomar una decisión. Para ello, se seleccionaron manualmente cuatro conversaciones reales entre ejecutivos y clientes para los casos de desconocimiento de transacciones, y manualmente se replicaron con los agentes implementados para calificar su rendimiento en la resolución del caso en una escala de 1 a 5 por un evaluador humano con conocimiento del proceso de atención y las reglas de negocio que involucra, utilizando como referencia el desenlace real del mismo:

1. El agente no funciona.
2. El agente funciona de forma limitada y no resuelve la atención.
3. El agente resuelve la atención, pero tiene falencias.
4. El agente resuelve la atención, con fallos menores.
5. El agente resuelve la atención sin problemas.

Para la comparación, se utiliza un prompt de respuestas libres y otro con restricciones: Extenso y Extenso con Restricciones. Luego, se calcula el promedio de los puntajes asignados dentro de las cinco conversaciones para cada enfoque, donde se obtienen los resultados presentados en la tabla 4.1.

Tabla 4.1: Resultados de evaluación de modelos con conversaciones reales.
Fuente: Elaboración propia.

		Modelo	
		GPT-3.5 Turbo	Chat Bison
Prompt	Extenso	3.5	2
	Extenso con Restricciones	2.75	2.5

Como se puede observar, GPT-3.5 Turbo supera en ambos prompts el desempeño de Chat Bison. Luego, sumado a las apreciaciones iniciales sobre la capacidad de procesar contextos más largos, se determina elegir al modelo de OpenAI para la implementación del experimento de este trabajo. El anexo F presenta el detalle de los puntajes obtenidos en las cuatro conversaciones utilizadas en la comparación entre ambos modelos, además de comentarios sobre el desempeño de cada uno.

4.1.2. Herramientas

La acción más básica para interactuar es la generación de mensajes de respuesta. No obstante, en cada etapa del flujo de atención es necesario contar con diversos actuadores fuera del procesamiento y generación de texto.

Primero, en la etapa de identificación de la transacción el agente debe poder buscar transacciones. Luego, en la etapa de bloqueo de producto, debe poseer la capacidad de explícitamente gatillar una acción que realice este paso. Después es necesaria la capacidad de evaluar la transacción para verificar si es posible o no realizar un reclamo. Adicionalmente, si el cliente puede y decide avanzar con un reclamo, el agente debe poder generar el acuerdo legal correspondiente para enviarlo al cliente. Este acuerdo legal también podría incorporarse dentro de del prompt, pero sería altamente ineficiente considerando que es un texto extenso y estaría presente en cada llamado al LLM. Finalmente, en la última etapa del flujo debe poder obtener cuál es la resolución del caso. Cabe destacar que en todo momento el agente requiere contar con la facultad de transferir al cliente a un ejecutivo humano, en caso de que las necesidades del caso sobrepasen a las capacidades que posee en sus instrucciones y/o herramientas.

En base estos actuadores, las herramientas del agente deben responder a seis acciones. Para el diseño de estas, se toman en consideración tres factores:

- Las herramientas deben emular el acceso a los servicios externos al chat que utilizan los ejecutivos para realizar la atención. Para ello, el desarrollo de estas debe estar pensado considerando la información que se requiere para utilizarlas y los resultados que reciben después de su uso.
- Dado que los LLMs pueden sufrir de sobrecarga de memoria en contextos largos, las herramientas son una oportunidad para almacenar información (por ejemplo, en variables de Python) para luego ser accedida por otras según las necesidades del agente. De esta manera, el modelo no requiere buscar en el historial de mensajes información detallada sobre la atención, como la transacción identificada o el estado de esta.
- Las capacidades de las herramientas deben variar según el tipo de prompt. Específicamente, las herramientas del prompt Extenso con Restricciones deben gatillar respuestas predeterminadas tras ejecutar una acción. Por ejemplo, si la acción para buscar transacciones encuentra una coincidencia con éxito, se puede enviar un mensaje predefinido al cliente indicando este suceso e invitándolo a continuar la atención. Esta decisión se tomó en base a pruebas empíricas durante el desarrollo, donde el comportamiento del agente era ignorar la instrucción de elegir respuestas predefinidas si la acción no la gatillaba. Más aún, los mensajes de este agente en sí deben ser una herramienta que permita escoger y/o construir frases para comunicarse con el cliente. Por otro lado, los prompts Simple y Extenso, al ser de respuestas libres, no tienen problemas con compartir una misma implementación de herramientas.

A continuación, se describen las herramientas implementadas en forma de funciones de Python para cada enfoque de prompt: respuestas libres (para los prompts Simple y Extenso) y con restricciones de respuestas predefinidas (para el prompt Extenso con Restricciones).

Herramientas de respuestas libres

En el caso de los prompts de respuesta libre, no es necesario implementar una herramienta para generar la respuesta, sino que se pueden emplear las capacidades nativas del modelo de lenguaje con este fin. Así, las herramientas para este paradigma son las siguientes funciones:

- **Buscar transacción:** Esta función permite que el agente busque una transacción conectándose a un estado de cuenta, que para efectos de este trabajo es un estado ficticio que permite simular una atención. Para ello, se han seleccionado como argumentos información sobre la fecha (como día y mes) en forma de número enteros, un monto (también numérico) y una descripción en forma de texto. A partir de estos, la función realiza una búsqueda sobre la información de un archivo `.csv` asociado al cliente, retornando aquella que tenga el mayor número de coincidencias con la información entregada. Además, almacena la información de esta transacción en un estado global para ser consultada posteriormente por otras herramientas. De no encontrar transacciones coincidentes, retorna en lenguaje natural un mensaje al agente, indicando este suceso con la recomendación de transferir al cliente. La forma de encontrar este archivo es sencilla, pues se asume que el cliente se encuentra identificado al comenzar la atención, luego simplemente se almacena en memoria al inicio de la interacción.
- **Bloquear productos:** Simula un llamado al sistema que realiza el bloqueo de los productos del cliente. Simplemente, retorna al agente un mensaje de éxito de la ejecución de la herramienta, con un código de bloqueo.
- **Evaluar transacción:** Utiliza la transacción previamente identificada para obtener su estado y retorna un mensaje al agente indicando esta información, además de recomendaciones para continuar la atención.
- **Generar acuerdo legal:** Retorna un acuerdo legal genérico para que el agente pueda entregarlo al cliente. Esta función es un ejemplo de cómo las herramientas pueden liberar de contenido a los prompts, y permitir al agente acceder a información de gran volumen (como lo puede ser el texto de un extenso acuerdo legal) únicamente cuando es necesario.
- **Obtener resolución:** De forma aleatoria, elige si es posible realizar una devolución del cargo o es necesario realizar un reclamo. Además, simula estas acciones y retorna al agente un mensaje con recomendaciones de respuesta.
- **Transferir cliente:** Simula un llamado al sistema que transfiere al cliente con un ejecutivo humano de consultas generales. Simplemente, retorna al agente un mensaje de éxito de la transferencia.

El anexo E.1 presenta el código fuente de las herramientas de respuestas libres, utilizado en la implementación de los agentes con prompts Simple y Extenso.

Herramientas con restricciones

En el caso del prompt Extenso con Restricciones, se deben incorporar respuestas predefinidas a las herramientas, además de una nueva función que permita al agente comunicarse con el cliente. No obstante, esto abre la posibilidad de simplificar la cantidad de herramientas disponibles a partir de agrupar cadenas de acciones que, generalmente, se realizan de forma secuencial e inmediata. Por ejemplo, las atenciones suelen requerir que después de la

acción de bloqueo de producto inmediatamente se realice una evaluación de la transacción. Luego, estas acciones pueden agruparse en una única herramienta que entregue los mensajes respectivos al cliente de forma secuencial. Las herramientas **Buscar transacción**, **Obtener resolución** y **Transferir cliente** son análogas al enfoque de respuestas libres, salvo el hecho de que ahora estas funciones retornan una respuesta en lenguaje natural predefinida al cliente. Por otro lado, las siguientes herramientas son diferentes en este paradigma:

- **Enviar mensaje:** Permite al agente construir una respuesta en forma de texto de cara al cliente, a través de concatenar una serie de oraciones que selecciona sobre un conjunto acotado y predefinido, tales como saludos, despedidas, solicitudes de detalles de la transacción desconocida, entre otros. En su descripción cuenta con 11 mensajes enumerados, y recibe como argumento una lista con números enteros ordenados de la secuencia de mensajes que quiere utilizar. Así, si el agente decide responder con un mensaje compuesto por las oraciones 1 y 2, debe entregarle a la función el argumento `[1, 2]`.
- **Bloquear productos:** Agrupa las etapas de bloqueo de producto, evaluación de la transacción y generación del acuerdo legal, dado que estas generalmente se realizan de forma secuencial, entregando una respuesta al cliente que agrega la información de las tres acciones según corresponda.

Análogamente al caso anterior, el anexo E.2 presenta el código fuente de las herramientas descritas.

4.2. Implementación

Para realizar pruebas, se desarrolla una plataforma experimental que implementa una versión del agente. Las siguientes subsecciones contienen el detalle técnico del desarrollo, mientras que el diseño experimental se presenta en el capítulo 5.

4.2.1. Arquitectura física

La plataforma considera una arquitectura física de cuatro capas. Primero, la capa de usuario considera únicamente a un tipo de estos, que corresponde al Participante. Luego, la capa de navegador representa al medio a través del cual el usuario accede a la plataforma. En este caso, se habilita que puedan utilizarse tanto dispositivos de escritorio como móviles, por lo tanto se requiere un desarrollo responsivo.

La tercera capa, que también es la más compleja, es donde la aplicación y los recursos que habilitan su funcionamiento se encuentran. Como fue mencionado en la sección 3.3, el desarrollo del agente esta basado en Python y LangChain. Utilizando esto como base, se determina el desarrollo de la aplicación completa en Python, utilizando el framework *Streamlit* para la implementación de interfaces de usuario. El uso de este tiene posee varias ventajas:

- Utiliza un paradigma de desarrollo basado en componentes predefinidos, lo que acelera la construcción de la plataforma. Entre ellos se destacan los componentes necesarios para el chat, botones y el menú lateral.
- Ofrece de forma nativa responsividad de la plataforma, así como modos claro y oscuro, para adaptarse a la comodidad de diversos tipos de usuarios.

- Las aplicaciones de Streamlit cuentan con un estado global para almacenar información en tiempo de ejecución, lo que facilita el desarrollo de las herramientas del agente que requieren esta característica. Más aún, LangChain posee componentes para almacenar mensajes en este estado de forma sencilla.

Esta aplicación de Python y Streamlit se implementa en un contenedor de Docker para poder ser desplegada utilizando los servicios de *Google Cloud*. Específicamente, se utilizaron tres servicios:

- *Cloud Run*, para el despliegue público del contenedor de la aplicación.
- *Cloud Storage*, para almacenar los archivos *.csv* de los estados de cuenta ficticios que utiliza el sistema.
- *BigQuery*, para desplegar una base de datos en tablas que permita almacenar los datos generados en el experimento.

Finalmente, la última capa considera APIs externas que sean utilizadas por la aplicación, donde se ubica la API de OpenAI desde donde se interactúa con GPT-3.5 Turbo. El diagrama de la figura 4.1 presenta la arquitectura descrita.

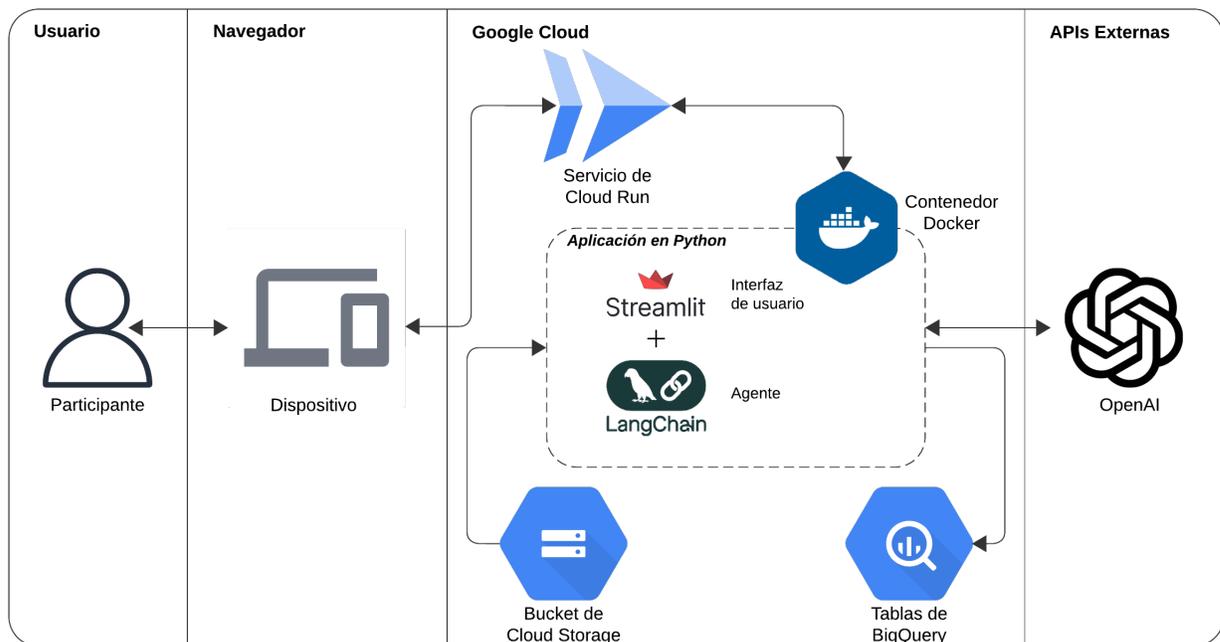


Figura 4.1: Arquitectura física de 4 capas para la plataforma experimental.
Fuente: Elaboración propia.

4.2.2. Modelo de datos

La figura 4.2 contiene el modelo relacional de la base de datos diseñada para la plataforma, que fue desplegada en BigQuery. Este modelo contempla el almacenamiento de toda la información considerada necesaria para la evaluación del sistema.

A continuación, se describen las relaciones existentes en el modelo:

- **Customers:** Representa a los participantes del modelo. Contiene un identificador único, que corresponde a un número hexadecimal en forma de string, un número de cliente asociado a su estado de cuenta ficticio, un tipo de prompt (agente), y variables sociodemográficas como edad, sexo y nivel socioeconómico.
- **Tasks:** Representa un escenario de la plataforma, con un nombre.
- **Events:** Corresponde a un evento dentro de las interacciones. Este puede ser un mensaje, el uso de una herramienta o una acción, como el inicio y cierre de la conversación. Dentro de los registros se almacena quién lo realiza (Agente o Cliente), los argumentos utilizados y la salida de este (según corresponda), y el costo asociado al uso del LLM (si corresponde).
- **TaskSurveys:** Almacenan las respuestas del participante para una escenario particular. Contiene los identificadores del participante y el escenario, así como el instante en que se contestó y los puntajes de cada afirmación.
- **GralSurveys:** Análogo a TaskSurveys, pero para las encuestas finales del experimento.

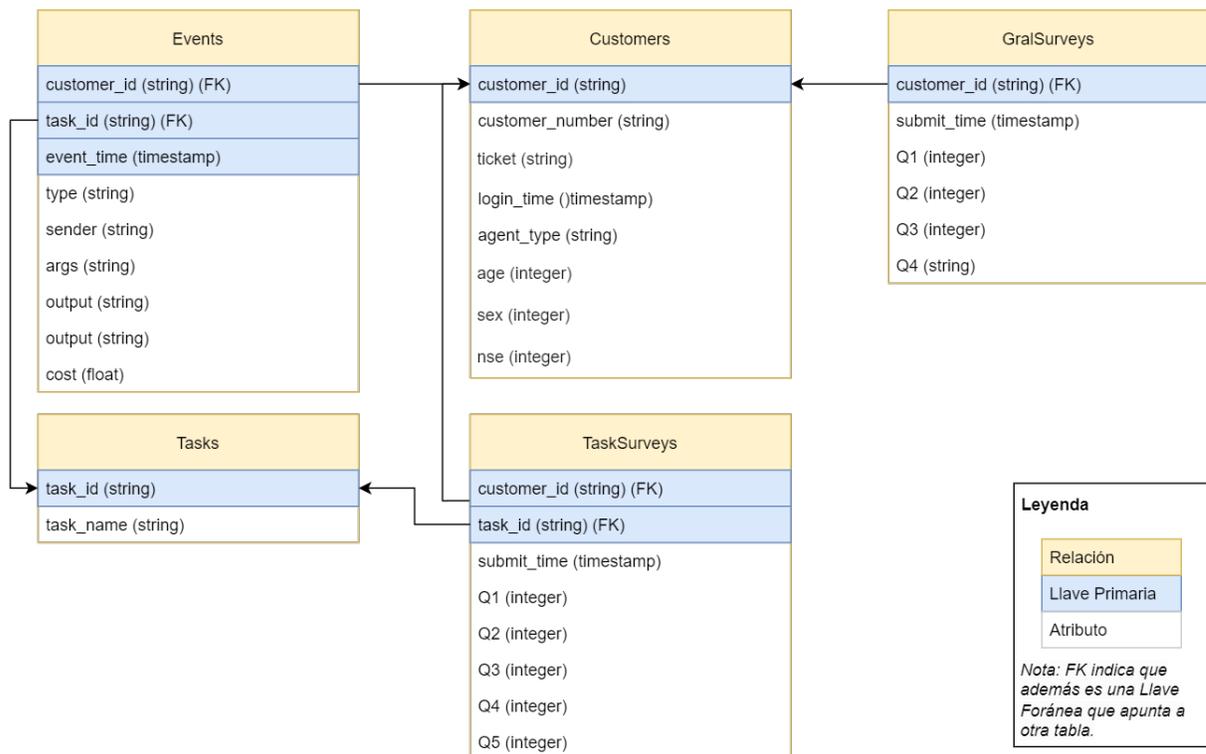


Figura 4.2: Modelo relacional de la plataforma experimental. Fuente: Elaboración propia.

Capítulo 5

Evaluación

Una vez presentada la solución propuesta, se debe evaluar las interacciones del sistema con clientes. En este capítulo se propone un modelo conceptual que permite explicar el efecto de los diferentes tipos de prompts sobre el desempeño del agente en cada atención. Además, se describen los elementos centrales de diseño e implementación de una plataforma experimental que permita implementar el modelo conceptual propuesto.

5.1. Modelo conceptual

Para la evaluación de las interacciones, se considera que el desempeño de un agente depende de tres dimensiones:

1. Su capacidad de resolver el problema, o *resolutividad*.
2. Su capacidad de realizar las acciones correctas y seguras en la resolución del problema, o *correctitud*.
3. La *satisfacción* del cliente tras una atención.

La elección de estas dimensiones se justifica por el marco de evaluación HHH (para la resolutividad y correctitud) y por la naturaleza específica de estas interacciones (para la satisfacción), donde el sistema está al servicio de usuarios que son clientes de la firma.

Para la construcción del modelo conceptual, se postula que la resolutividad, correctitud y satisfacción que experimenta el cliente en relación con el agente depende del contenido de cada interacción. Aquí, se distinguen 4 factores principales. Como primer factor higiénico, se considera la capacidad del agente de entender y darse entender, lo que se ha llamado Eficacia de Entrada y Eficacia de Salida. Después, un segundo factor considerado es la fiabilidad con la que el agente transmite a través de sus respuestas, la cual ha sido denominada como Confianza. Además, existen factores que se pueden medir objetivamente, como la duración de la atención y lo extensa que esta puede llegar a ser, específicamente en términos de la cantidad de mensajes y palabras empleadas. En efecto, la literatura de satisfacción de clientes sugiere que elementos como la empatía y confiabilidad del agente juegan un rol relevante en la calidad percibida (McLean y Osei-Frimpong, 2017), elementos que se han condensado en los constructos mencionados previamente.

Por otro lado, se hipotetiza que la elección del tipo de prompt también tiene efecto sobre los factores mencionados. En primer lugar, los prompts con mayor detalle en sus instrucciones deberían resolver mayor cantidad de casos, así como tener mayor conocimiento para realizar acciones correctas a medida que avanza la atención. Luego, aquellos prompts sin restricciones deben demostrar mejor flexibilidad de comunicación y una interacción más realista, y por lo tanto, entregando una mejor experiencia al cliente.

En síntesis, se propone que el tipo de prompt utilizado afecta al desempeño del agente, pero estas relaciones son moderadas por las variables objetivas y subjetivas mencionadas, las cuales se han condensado en el modelo de la figura 5.1.

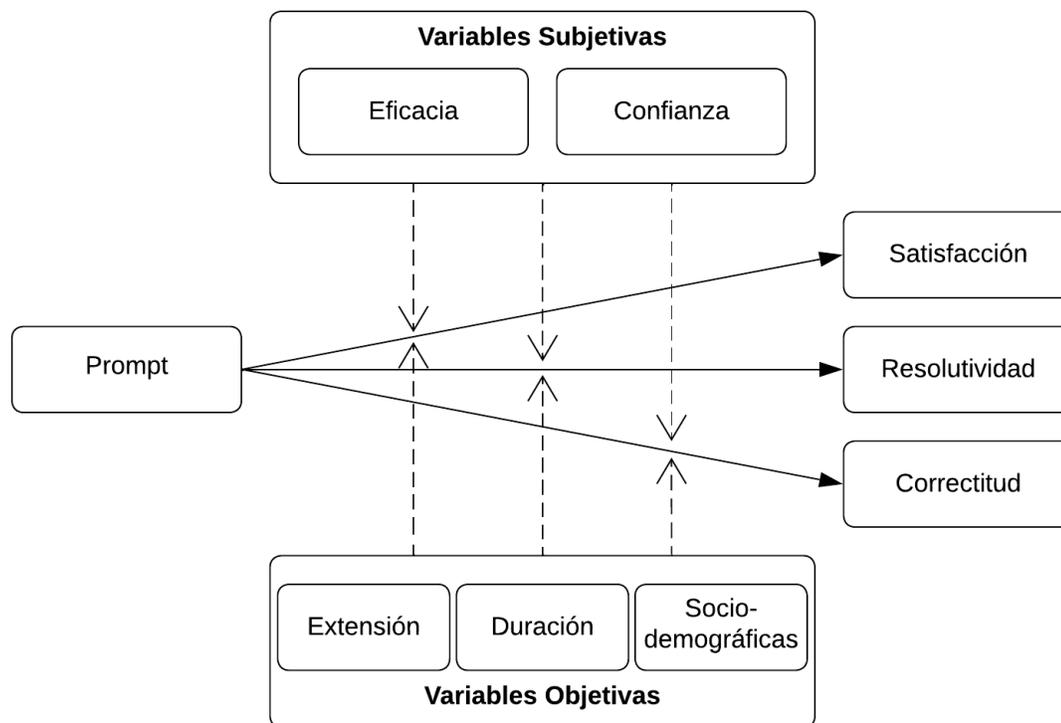


Figura 5.1: Modelo conceptual de los efectos del tipo de prompt sobre la percepción del cliente. Fuente: Elaboración propia.

5.2. Plataforma experimental

5.2.1. Diseño experimental

Para implementar el modelo conceptual, se propone un diseño experimental basado en tres iteraciones de interacción y encuesta, según los escenarios propuestos en la sección 3.3. Para obtener respuestas realistas de estas interacciones, se restringe la selección de participantes a personas mayores de 18 años y que posean tarjeta de crédito, además de haberla utilizado dentro de los últimos seis meses.

Al inicio del experimento se debe asignar un estado de cuenta ficticio con transacciones objetivo para cada escenario pertinente que los usuarios puedan utilizar para basar su relato, y un tipo de prompt con el cual el agente basado en LLMs procesará sus mensajes. En

la implementación, la plataforma utiliza diez estados de cuenta predeterminados, generados con combinaciones aleatorias sobre un conjunto de fechas, descripciones y montos de transacciones. Estos se almacenan en archivos *.csv* que luego son seleccionados de forma aleatoria uniforme por la plataforma para ser utilizados a lo largo del experimento. Análogamente, el tipo de prompt también se selecciona aleatoriamente de forma uniforme.

Posteriormente, el participante debe interactuar con el chat y responder una encuesta para cada escenario. Esta encuesta se diseña tomando en consideración las dimensiones presentes en el modelo conceptual, además de asegurar que sea sencilla de responder. Para ello, se emplean cinco afirmaciones, cada una ligada a una dimensión (donde se deja de lado las variables objetivas, dado que estas se miden a partir de los datos capturados en la interacción), que el usuario debe puntuar en una escala Likert de 1 a 5 según su grado de acuerdo, donde 1 corresponde a *Muy en desacuerdo* y 5 a *Muy de acuerdo*. Estas son las siguientes (entre paréntesis, se indica la dimensión evaluada):

- El agente comprendió adecuadamente las preguntas que le hice (Eficacia de entrada).
- Las respuestas del agente eran fáciles de entender (Eficacia de salida).
- El agente pudo resolver adecuadamente las preguntas que le hice sin necesidad de derivarme a otras instancias del banco (Resolutividad).
- En todo momento sentí que el agente tenía información útil y confiable respecto a las preguntas realizadas (Confianza).
- En general, estoy satisfecho con la atención del agente virtual (Satisfacción).

Después de las tres interacciones con el agente y sus respectivas encuestas, se incorpora una encuesta final para capturar percepciones globales del participante sobre la experiencia realizada. Análogamente, se utilizan preguntas con escalas Likert con el objetivo de medir Resolutividad, Satisfacción y un *promoter score* para poder calcular el NPS asociado a cada prompt, utilizando la misma pregunta que usa hoy en día la empresa para realizar esta medición en las interacciones reales. Además, se incorpora una caja de comentarios para obtener las opiniones de los usuarios sobre el estudio realizado.

Finalmente, el experimento considera los siguientes puntos en su implementación:

- La información a la que puede acceder el participante durante la interacción son los mensajes de la conversación y su estado de cuenta.
- La plataforma debe contar con instrucciones de funcionamiento y filtros adecuados para asegurar que los participantes puedan resolver el experimento sin problemas.

5.2.2. Interfaz de usuario

En base al diseño experimental, la interfaz de usuario de la plataforma se subdivide en seis partes (en orden de aparición):

1. Una interfaz de inicio, donde se entrega una breve descripción del experimento y se informan las precondiciones necesarias para poder participar de este.

2. Una interfaz de instrucciones generales, donde se explica el experimento y el uso de la plataforma.
3. Una interfaz de descripción del escenario a realizar, donde se explica la situación que el cliente debe simular y las instrucciones necesarias.
4. Una interfaz de chat, donde el cliente debe poder acceder a su estado de cuenta con la transacción objetivo, así como conversar a través de texto con el agente. Además, se considera incluir las instrucciones del escenario, por si el cliente requiere revisarlas otra vez. Tanto el estado de cuenta como las instrucciones son accesibles a partir de un menú lateral desplegable.
5. Una interfaz de encuesta, donde se presenten las afirmaciones y los selectores de puntuación. Además, se mantiene el menú lateral con información, ahora incorporando la conversación llevada con el agente.
6. Luego de haber pasado por las interfaces 4, 5 y 6 para los tres escenarios, se incluye una última interfaz con la encuesta final del experimento.

La figura 5.2 muestra un ejemplo de la interfaz 4. El detalle completo de las interfaces implementadas, así como su contenido e imágenes de estas se encuentra disponible en la sección G de los anexos.

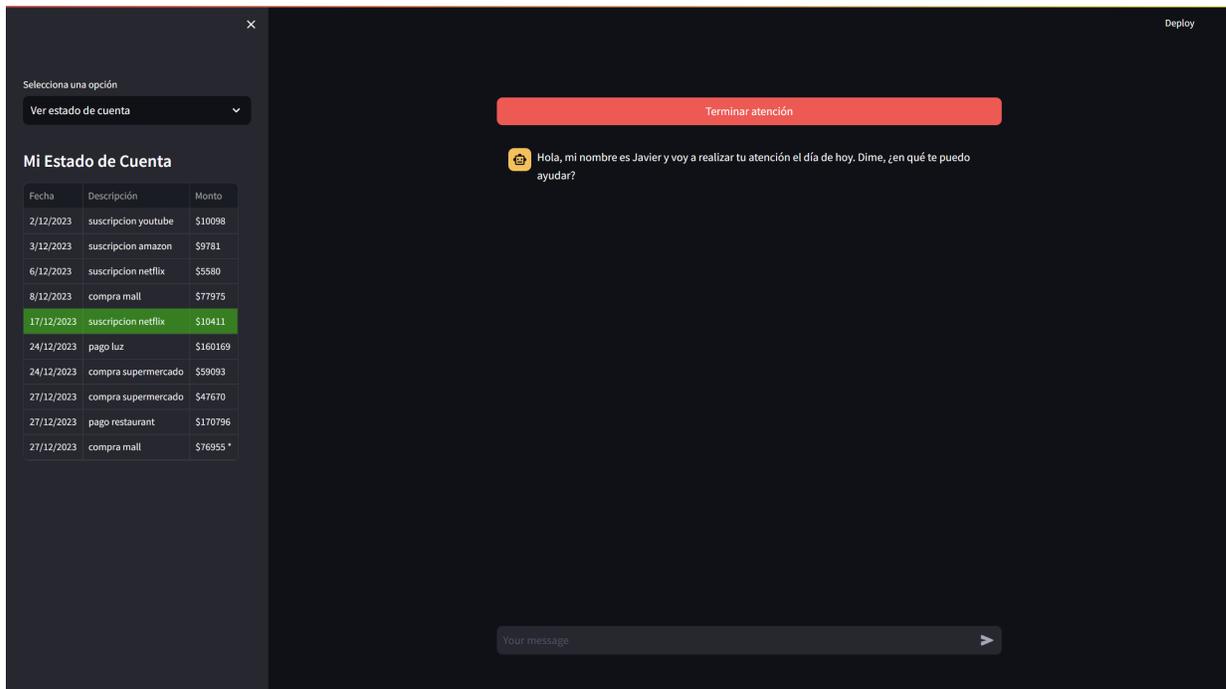


Figura 5.2: Pantalla de chat en su versión oscura. A la izquierda, el menú lateral contiene la información del estado de cuenta ficticio del usuario. Fuente: Elaboración propia.

Capítulo 6

Resultados

En este capítulo, se presenta el análisis de los resultados obtenidos en el trabajo realizado. Comienza con una breve explicación y evaluación que contrasta el desempeño de los métodos no supervisados con los LLMs en el problema de clasificación de conversaciones. Luego, se describe en detalle los datos obtenidos tras el experimento con usuarios para las interacciones con agentes. Posteriormente, se presentan los análisis a realizar con los datos, seguido por los resultados obtenidos. Finalmente, el capítulo concluye con una discusión de estos, destacando sus implicaciones y relevancia para el estudio.

6.1. Clasificación de instancias

El entrenamiento de los modelos no supervisados se realizó con un corpus de interacciones entre clientes y el menú de autoatención, reconstruidas a partir de datos históricos del banco. Para ello, el texto fue llevado a un *vector space model* (Jurafsky y Martin, 2023), para lo que fue preprocesado y transformado a representaciones Bag of Words y TF-IDF. Así, los modelos fueron ajustados para 10 tópicos utilizando la librería *scikit-learn* de Python. Para la clasificación de las conversaciones del conjunto de prueba, se realizó una asignación discreta de las categorías a cada tópico a partir de la máxima probabilidad/importancia de pertenecer a este con ambos modelos. En otras palabras, aquel tópico que concentre más un tipo de conversación será el que determine la categoría a la que corresponden.

Las tablas 6.1 y 6.2 muestran el porcentaje de conversaciones del conjunto de prueba que fue asignada a cada uno de los 10 tópicos de los modelos, destacando en azul los tópicos más relevantes para cada categoría, y por ende, aquellos que corresponden a esa clase en cuestión. Así, se puede observar que NMF es más apropiado para la clasificación, dado que posee mejor distribuidas las probabilidades de pertenecer a un tópico. Por el contrario, en el caso de LDA, el tópico 9 es problemático con esta heurística, pues es difícil determinar si corresponde a Consultas de estado de cuenta o Rechazo de transacción.

Tabla 6.1: Tópicos para modelo NMF. Fuente: Elaboración Propia.

Tópico	1	4	5	7	9
Estado de cuenta	0.30	0.00	0.35	0.10	0.10
Rechazo de transacción	0.30	0.00	0.10	0.20	0.20
Desconocimiento de transacción	0.20	0.20	0.05	0.10	0.15

Tabla 6.2: Tópicos para modelo LDA. Fuente: Elaboración Propia.

Tópico	2	3	7	9	10
Estado de cuenta	0.10	0.05	0.10	0.55	0.20
Rechazo de transacción	0.05	0.05	0.10	0.60	0.10
Desconocimiento de transacción	0.00	0.15	0.30	0.20	0.00

Por otro lado, el LLM fue utilizado como un clasificador a partir de la siguiente plantilla de prompt, que recibe una conversación en la variable *chat*:

Prompt de clasificación de instancias

Eres un agente bancario que atiende clientes vía chat. Acabas de terminar una conversación con un cliente y debes clasificarla en una de las siguientes categorías: [Estado de Cuenta, Desconocimiento de Transacción, Transacción Rechazada y Otro]. Tu respuesta debe ser únicamente la categoría.

Esta es la conversación: {chat}

Tabla 6.3: Métrica de accuracy para las clasificaciones obtenidas con Topic Models y GPT-3.5 Turbo.

Tipo	Accuracy	
	NMF	GPT-3.5 Turbo
Estado de cuenta	0.35	0.70
Rechazo de transacción	0.30	0.15
Desconocimiento de transacciones	0.20	0.70
Promedio	0.28	0.52

Tabla 6.4: Matriz de confusión para clasificación con LLM.

	Estado de Cuenta	Rechazo de transacción	Desconocimiento de transacciones	Otro
Estado de Cuenta	14	0	0	6
Rechazo de transacción	10	3	2	5
Desconocimiento de transacciones	2	0	14	5
Otro	0	0	0	0

Utilizando el conjunto de 60 datos de prueba, se obtienen los valores de *accuracy* presentados en la tabla 6.3 para los diferentes tipos de conversación. Además, la tabla 6.4 contiene la matriz de confusión obtenida para la clasificación con LLMs. Es fácil observar que a nivel de interpretabilidad y resultados, los LLMs superan a los métodos clásicos probados con el

enfoque de topic modeling, donde se obtuvieron resultados un 24% más altos en exactitud promedio. Es importante recalcar que este ejercicio no apunta a ser una solución sofisticada, sino que es sencillo en varios aspectos y presenta oportunidades de mejora que podrían llevar a resultados más apropiados, pero no son contemplados en este informe dado a que se escapa del foco y los alcances de este trabajo. No obstante, permite evidenciar que con pocas instrucciones, un LLM performa mucho mejor que soluciones clásicas a estos problemas.

6.2. Descripción de datos obtenidos

El experimento con agentes automáticos fue realizado de forma exitosa sobre un total de 643 participantes, siendo estos hombres y mujeres de 18 a 83 años, pertenecientes a niveles socioeconómicos (NSE) desde AB hasta D. A partir de las interacciones, se construyó un panel de datos con los siguientes campos:

- **customer_id**: Identificador del participante dentro del experimento.
- **task_id**: Identificador del escenario.
- **age**: Edad del participante.
- **sex**: Sexo del participante.
- **nse**: Nivel socioeconómico del participante.
- **prompt_type**: Tipo de prompt asignado en el experimento.
- **resolutivity**: Resolutividad percibida por el participante.
- **hallucination**: Confianza percibida por el participante.
- **experience**: Satisfacción percibida por el participante.
- **efficacy_input**: Eficacia de comprensión de mensajes percibida por el participante.
- **efficacy_output**: Eficacia de generación de respuestas percibida por el participante.
- **interaction_time**: Resolutividad percibida por el participante.
- **cus_res_time**: Tiempo promedio que tarda en responder el participante en la interacción.
- **agt_res_time**: Tiempo promedio que tarda en responder el agente en la interacción.
- **n_messages**: Número de mensajes en la interacción.
- **n_words**: Número de palabras promedio por mensaje.

Al ser registros a nivel de interacciones (las cuales son 3 por participante, dentro del experimento), el panel cuenta con un total de 1,929 entradas. Un esquema más detallado del mismo está disponible en la tabla H.1 de los anexos. Adicionalmente, en base a la taxonomía de errores definida en la sección 3.2, se calculó el porcentaje ocurrencia de cada tipo de error sobre todas las acciones realizadas por el agente en una interacción. Para ello, se etiquetaron manualmente 126 interacciones (42 experimentos de cada tipo de prompt), generando

la variable **per_error**, que corresponde al porcentaje de errores cometido sobre todas las acciones realizadas por un agente en una interacción.

Por otro lado, utilizando los datos de las encuestas globales, se genera una segunda base que también contiene las variables demográficas del participante y agrega las métricas objetivas (`interaction_time`, `n_messages`) y el porcentaje de errores obtenidos como promedios calculados sobre las tres interacciones. Además, incorpora las siguientes métricas:

- **resolutivity_global**: Resolutividad percibida por el participante a lo largo de todas las interacciones.
- **experience_global**: Satisfacción percibida por el participante a lo largo de todas las interacciones.
- **promoter_score**: Promoter score del participante.

Como son registros tomados de la experiencia completa de un participante, la segunda base cuenta con 643 entradas. El esquema en detalle de esta se muestra en la tabla H.2 de los anexos.

La tabla 6.5 muestra que la distribución de los participantes dentro de los tipos de prompt asignados es uniforme, con pequeñas diferencias entre ellos. Análogamente, en la figura 6.1 se aprecia que la distribución de participantes en las variables sexo y NSE sigue un comportamiento similar, salvo en el nivel socioeconómico AB, que presenta una porción mucho menor de asignaciones que el resto.

Tal como se puede observar en la figura 6.2, los participantes del experimento se encuentran en un rango de edad entre 18 y 83 años, siendo estos principalmente adultos jóvenes y adultos, con una mediana de 43 años. Profundizando un poco más, la distribución de las variables demográficas dentro de los distintos tipos de prompt del experimento mantiene el comportamiento uniforme, salvo por el segmento AB en la variable de grupo socioeconómico, tal como se observa en la tabla 6.6.

El anexo H.2 contiene tablas y visualizaciones complementarias que muestran la relación entre las variables capturadas en el experimento.

Tabla 6.5: Distribución de los participantes dentro de los distintos tipos de prompt. Fuente: Elaboración propia.

	S	E	E-Res
Conteo	220	215	208

Tabla 6.6: Distribución de variables demográficas según tipo de prompt. Las variables categóricas muestran su frecuencia relativa. Fuente: Elaboración propia.

Métrica	S	E	E-Res
Edad Promedio	42.98	45.81	44.47
Sexo [H]	102	109	110
Sexo [M]	118	106	98
NSE [AB]	8	9	8
NSE [C1a]	40	47	31
NSE [C1b]	37	37	37
NSE [C2]	48	41	53
NSE [C3]	38	34	40
NSE [D]	49	47	39

Distribución de variables sociodemográficas

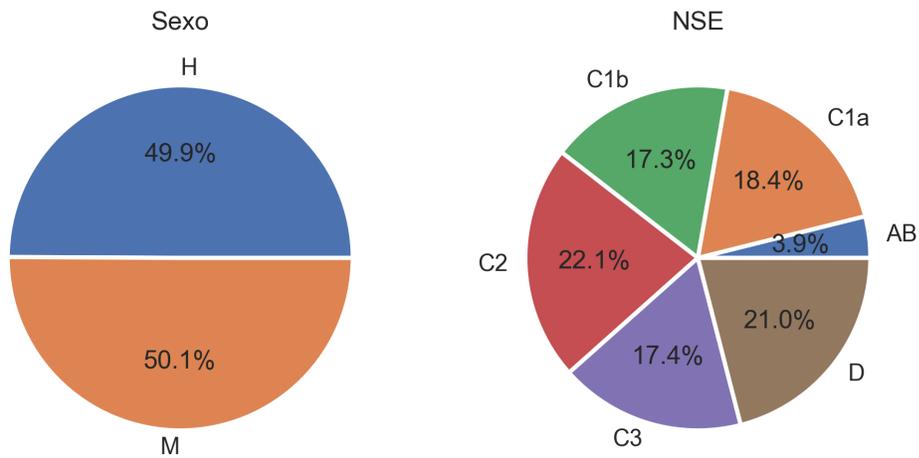


Figura 6.1: Distribución de variables de sexo y nivel socioeconómico de los participantes del experimento. Fuente: Elaboración propia.

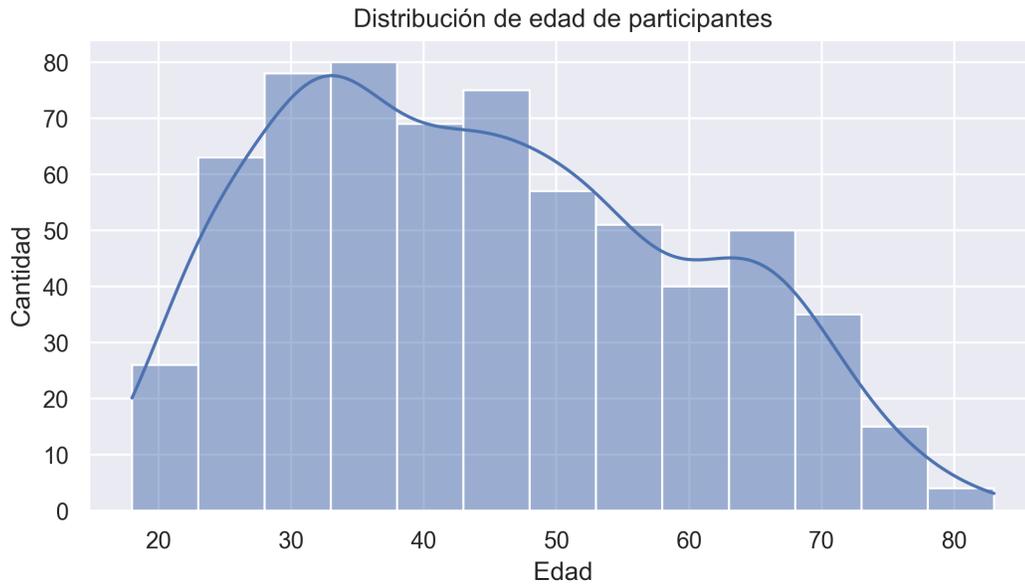


Figura 6.2: Distribución de la edad de los participantes del experimento.
Fuente: Elaboración propia.

6.3. Herramientas de análisis

Para validar o refutar el modelo conceptual propuesto en el capítulo 5, se realizará un análisis de los datos generados en el experimento a partir de técnicas de análisis estadístico que permitan estudiar las relaciones allí presentadas. Este se compone de dos etapas:

1. Primero, se propone realizar un análisis de medias, con la finalidad de comprobar que hay evidencia estadística para respaldar que existe un efecto de elegir un tipo de prompt sobre las métricas objetivo de Satisfacción, Resolutividad y Correctitud. Además, a través de este mismo análisis, es necesario estudiar si las variables objetivas y subjetivas moderadoras de esta relación también influyen en las métricas mencionadas.
2. Luego, se propone estudiar la magnitud de los efectos del tipo de prompt en las relaciones del modelo conceptual a partir de un análisis de regresión, empleando las métricas de desempeño como variables dependientes y las variables objetivas y subjetivas como controles de los modelos.

A continuación, las siguientes secciones presentan ambos análisis propuestos y sus resultados.

6.4. Análisis de medias

El análisis de medias de este trabajo tiene como finalidad encontrar evidencia para respaldar estadísticamente la existencia de diferencias sistemáticas a partir de las relaciones propuestas en el modelo conceptual de evaluación.

Aplicando el test de Shapiro-Wilk sobre las variables segmentadas por tipo de prompt, se encuentra con un nivel de significancia del 1% que estas no provienen de una distribución

normal. Más aún, los tests de Levene y Bartlett muestran heterocedasticidad para algunas métricas. El detalle de los p-valores obtenidos con estos tests se encuentra en las tablas de los anexos H.3 y H.4, respectivamente. Luego, se determina realizar el análisis utilizando un test T de medias, reforzado con un segundo test no paramétrico, el U de Mann-Whitney. El detalle de los p-valores obtenidos con cada test se encuentra en la tabla H.10 del anexo.

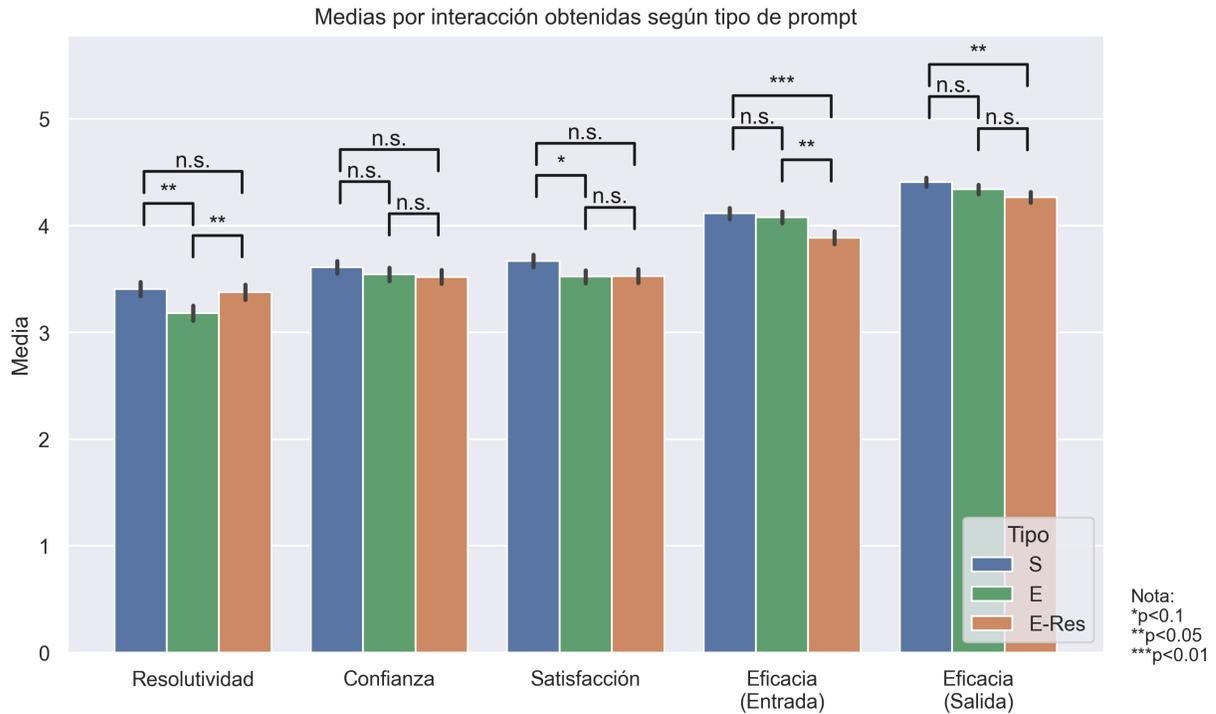


Figura 6.3: Medias obtenidas por interacción sobre métricas subjetivas, según tipo de prompt. Se muestra además la significancia estadística de las diferencias, obtenidas con el test T. Fuente: Elaboración propia.

Las medias obtenidas a nivel de interacción, presentadas en la figura 6.3, muestran que hay evidencia para respaldar que al incorporar restricciones al agente (prompt E-Res), empeora la capacidad de este para demostrar comprensión de mensajes y generar respuestas de calidad. Además, no existen diferencias significativas en el ámbito de confianza percibida, mientras que en la satisfacción del cliente son pocas. Sin embargo, la confianza del agente decrece en la misma forma que las variables de eficacia, luego, es posible pensar que los participantes reportan cuánto confían en el agente en base a cómo este se comunica. Sobre la resolutividad de la atención, el análisis muestra que hay evidencia para indicar que el prompt Extenso tiene peor desempeño que los casos Simple y Extenso con Restricciones, apuntando a que es mejor entregar instrucciones simples al modelo, o bien restringir su comunicación si se opta por instrucciones detalladas.

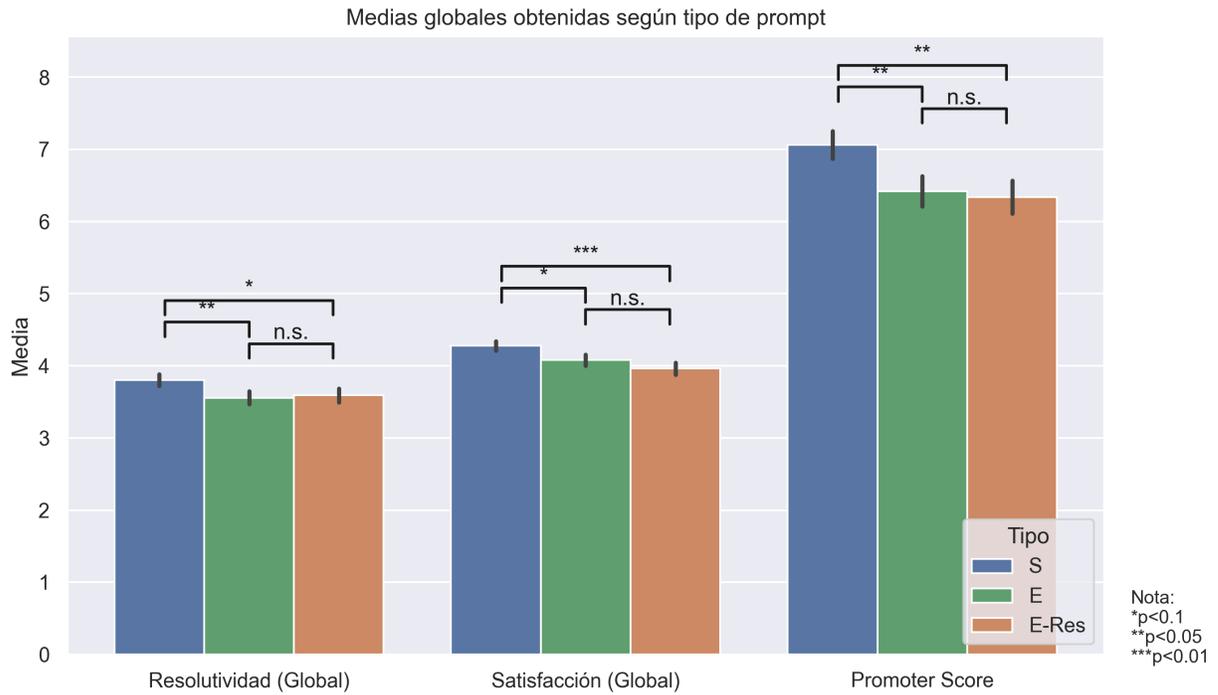


Figura 6.4: Medias globales sobre métricas subjetivas, según tipo de prompt. Se muestra además la significancia estadística de las diferencias, obtenidas con el test T. Fuente: Elaboración propia.

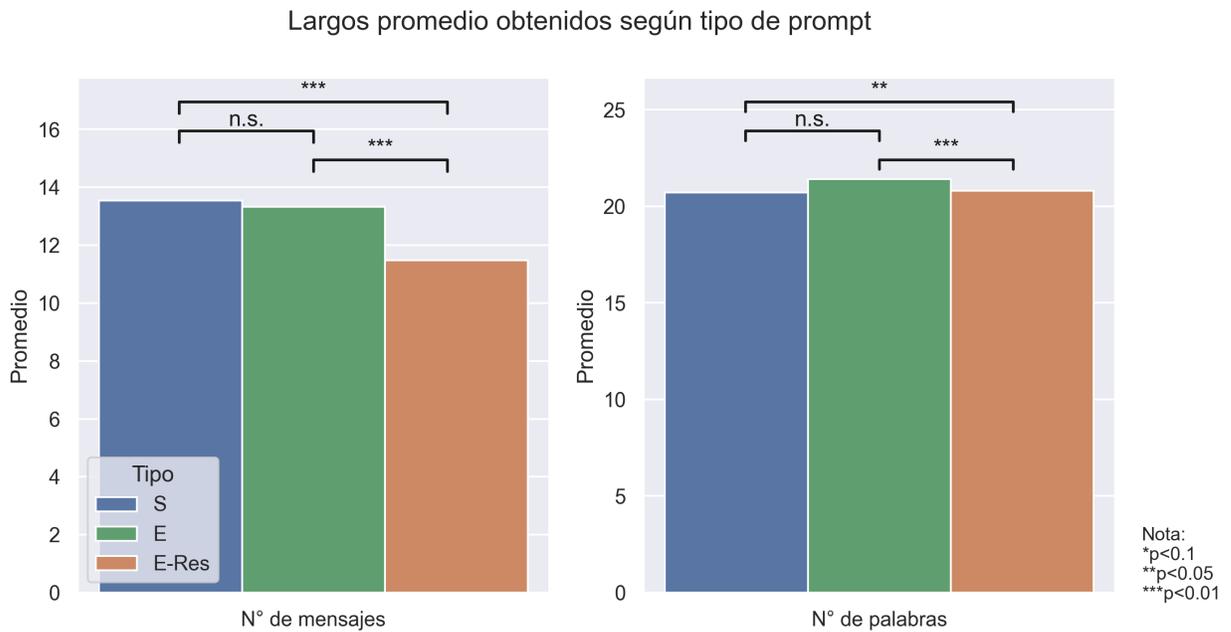


Figura 6.5: Medias por interacción sobre métricas de largo de interacción, según tipo de prompt. Se muestra además la significancia estadística de las diferencias, obtenidas con el test U de Mann-Whitney. Fuente: Elaboración propia.

Las métricas globales no presentan diferencias significativas entre los prompts Extenso y Extenso con Restricciones, tal como se observa en la figura 6.4. Más aún, pareciera que los mejores resultados se obtienen con instrucciones simples. A diferencia de las mediciones a nivel de interacción, ahora se aprecian mayores diferencias en la satisfacción reportada por el participante, siendo esta decreciente con el grado de especificidad de instrucciones y restricciones. Nuevamente, esto puede indicar que la experiencia del cliente está correlacionada con la eficacia y seguridad que el participante percibe del agente, dado a que es un comportamiento análogo al mostrado por las variables de eficacia de entrada y salida, y confianza a nivel de interacción.

En cuanto a las variables objetivas de extensión y duración, las métricas de largo de la interacción, presentes en la figura 6.5, muestran diferencias significativas del prompt Extenso con Restricciones frente a los otros dos, mientras que los casos Simple y Extenso no presentan contrastes. Este resultado lleva a pensar que, independiente del nivel de detalle de las instrucciones, dejar al agente sin restricciones de comunicación no diferencia el largo de las interacciones a nivel de texto. Algo similar ocurre con los tiempos de respuesta del cliente, como se muestra en la figura 6.6. Por otro lado, los tiempos de respuesta del agente presentan diferencias significativas entre todos los prompts, a pesar de ser muy bajos. Finalmente, los tiempos de interacción tienen un comportamiento similar a la resolutivez por interacción, lo que fortalece la relación encontrada entre ambas variables.

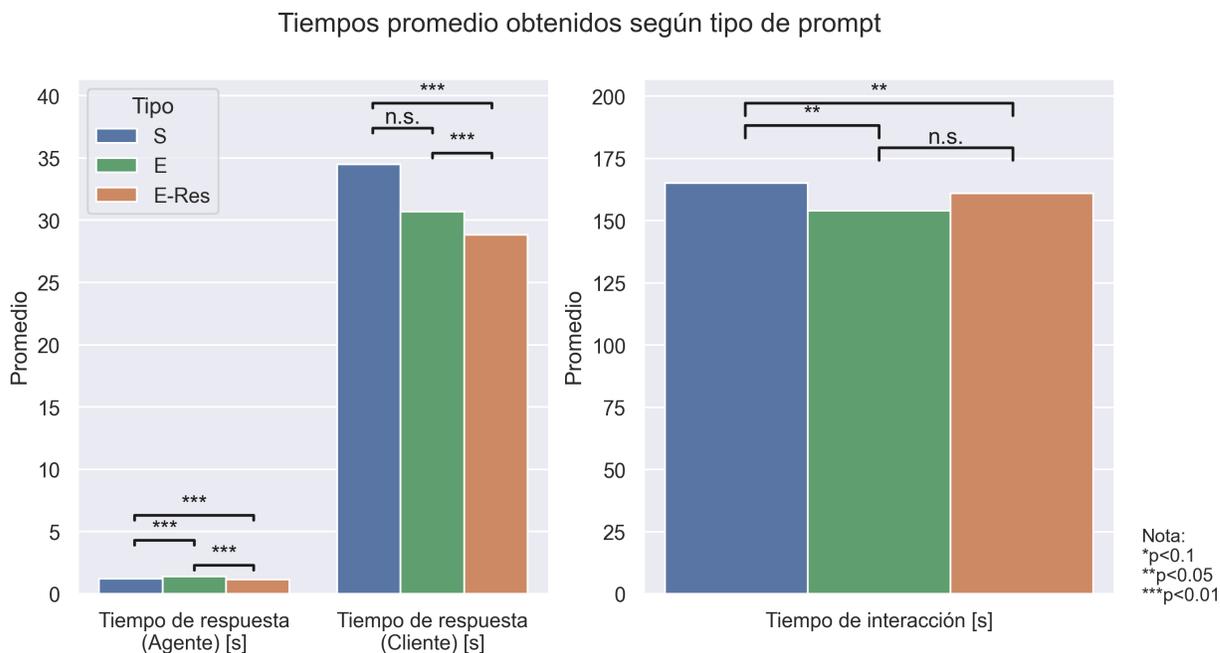


Figura 6.6: Medias por interacción sobre métricas de tiempo de interacción, según tipo de prompt. Se muestra además la significancia estadística de las diferencias, obtenidas con el test U de Mann-Whitney. Fuente: Elaboración propia.

Finalmente, respecto al porcentaje de errores cometidos, el prompt Extenso presenta diferencias importantes frente a los demás con una significancia estadística del 1%, así como se muestra en la figura 6.7. Este resultado llama la atención, dado a que no coincide con lo

observado para el desempeño medido con Resolutividad y Satisfacción.

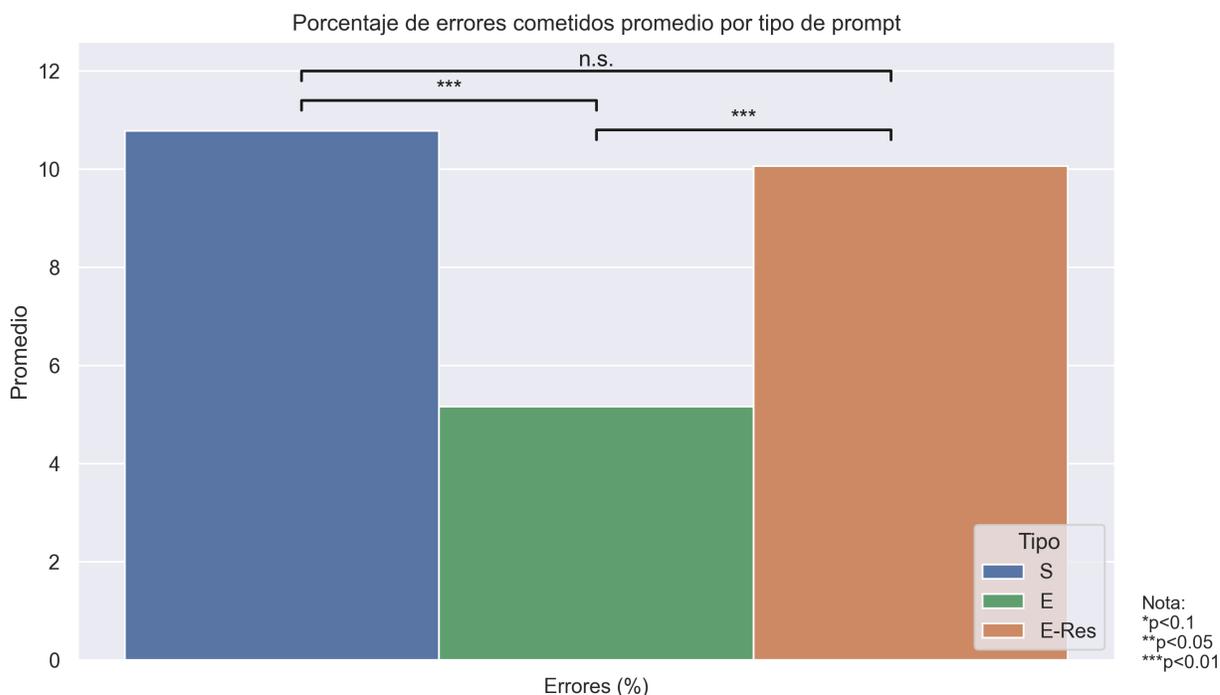


Figura 6.7: Medias por interacción sobre porcentajes de errores cometidos sobre el total de acciones realizadas, según tipo de prompt. Se muestra además la significancia estadística de las diferencias, obtenidas con el test U de Mann-Whitney. Fuente: Elaboración propia.

En síntesis, estos resultados entregan evidencia estadística preliminar de cómo el tipo de prompt afecta significativamente a las métricas de desempeño, Resolutividad y Satisfacción, tanto a nivel de interacción como global, además de justificar la necesidad de controlar los efectos del tipo de prompt por variables objetivas y subjetivas. De forma anticipada, es posible observar que:

- El prompt Simple parece alcanzar los mejores resultados de Resolutividad y Satisfacción. Por el contrario, el prompt Extenso obtiene el peor desempeño.
- De manera contradictoria a lo anterior, el prompt Extenso es el que comete menos errores en las interacciones.

6.5. Análisis de regresión

6.5.1. Métricas a nivel de interacción

Para estudiar la magnitud del efecto de utilizar un determinado tipo de prompt en la atención de un cliente, se evaluarán los resultados de los participantes i en los escenarios t a través de un análisis de regresión. Se considerarán 4 modelos, los cuales se usarán de forma general sobre las variables dependientes Y_{it} del modelo conceptual propuesto: Satisfacción, Resolutividad y Correctitud. A continuación, se presenta cada modelo:

- **Modelo 1 (M1):** El primer modelo busca evaluar el efecto del tipo de prompt sobre la variable dependiente de forma libre.

$$Y_{it} = \beta_0 + \beta_1 \cdot \text{prompt}_i + \epsilon_{it}$$

- **Modelo 2 (M2):** El segundo modelo incorpora las variables subjetivas, para estudiar cómo estas influyen en el efecto del tipo de prompt sobre la variable dependiente.

$$Y_{it} = \beta_0 + \beta_1 \cdot \text{prompt}_i + \gamma_1 \cdot \text{efficacy_input}_{it} + \gamma_2 \cdot \text{efficacy_output}_{it} + \gamma_3 \cdot \text{hallucination}_{it} + \epsilon_{it}$$

- **Modelo 3 (M3):** El tercer modelo agrega las variables objetivas de duración y extensión de la interacción, además de las variables sociodemográficas del participante, nuevamente para controlar el efecto del tipo de prompt por estas.

$$Y_{it} = \beta_0 + \beta_1 \cdot \text{prompt}_i + \gamma_1 \cdot \text{efficacy_input}_{it} + \gamma_2 \cdot \text{efficacy_output}_{it} + \gamma_3 \cdot \text{hallucination}_{it} + \omega_1 \cdot \text{interaction_time}_{it} + \omega_2 \cdot \text{n_messages}_{it} + \omega_3 \cdot \text{age}_i + \omega_4 \cdot \text{sex}_i + \omega_5 \cdot \text{NSE}_i + \epsilon_{it}$$

- **Modelo 4 (M4):** El último modelo, qué también es el más completo, incorpora efectos fijos por el escenario en el que se desarrolló la interacción, dado que, como fue demostrado, existen diferencias sistemáticas que hacen decrecer las variables dependientes a lo largo del experimento.

$$Y_{it} = \alpha_t + \beta_0 + \beta_1 \cdot \text{prompt}_i + \gamma_1 \cdot \text{efficacy_input}_{it} + \gamma_2 \cdot \text{efficacy_output}_{it} + \gamma_3 \cdot \text{hallucination}_{it} + \omega_1 \cdot \text{interaction_time}_{it} + \omega_2 \cdot \text{n_messages}_{it} + \omega_3 \cdot \text{age}_i + \omega_4 \cdot \text{sex}_i + \omega_5 \cdot \text{NSE}_i + \epsilon_{it}$$

Para las variables categóricas de tipo de prompt, sexo, NSE y escenario se utilizarán *dummies*, cuyo valor base serán Simple, Hombre, AB y Escenario 1, respectivamente. La tabla 6.7 resume los coeficientes obtenidos para el modelo más completo (M4) para cada variable dependiente objetivo. Por otro lado, el detalle de cada modelo y otras métricas de interés se encuentran en las tablas del anexo H.6.

Respecto a los resultados obtenidos con M4 para cada variable, es posible destacar que:

- La variable de Satisfacción del cliente muestra un efecto negativo al utilizar el prompt Extenso respecto al Simple, con una significancia del 5%. Específicamente, emplear el agente con prompt E empeora la satisfacción de una interacción en 0.079 puntos, es decir, un 1.97% de la escala utilizada de 1 a 5. En el caso prompt E-Res no presenta evidencia de tener diferencias con el prompt Simple, pero sí con E con un nivel de significancia del 10%. En otras palabras, la disminución observada por el tipo E aplica al compararlo con cualquiera de los otros dos prompts.

Tabla 6.7: Tabla resumen de coeficientes obtenidos con modelo M4 para regresiones a nivel de interacción. Fuente: Elaboración propia.

	Satisfacción	Resolutividad	Error (%)
Intercepto	0.025	0.146	4.675
prompt.E	-0.079**	-0.157***	-4.965**
prompt.E-Res	-0.005	0.109**	-0.262
efficacy_input	0.174***	0.155***	-0.541
efficacy_output	0.100***	-0.030	1.400*
hallucination	0.720***	0.786***	-0.508
interaction_time	-0.006	-0.024***	-0.094
n_messages	0.007**	0.020***	0.224
age	-0.002	-0.003**	-0.044
sex.M	0.027	-0.038	0.030
p-value [E vs. E-Res]	0.066	0.000	0.016
Observaciones	1929	1929	126
R^2	0.798	0.709	0.220
R^2 ajustado	0.796	0.707	0.105

Nota:

*p<0.1; **p<0.05; ***p<0.01

Por otro lado, todas las variables subjetivas reportadas por los usuarios parecen incidir también en la satisfacción que estos experimentan. En el caso de las Eficacias de entrada y salida, ambas presentan un efecto positivo de 0.174 y 0.1 con una significancia del 1 %, respectivamente. Sin embargo, la variable que tiene mayor impacto en el cliente es la Confianza que percibe en el agente, la cual presenta un efecto de 0.72 por unidad extra, es decir, un 18 % del puntaje de la escala de medición, con un nivel de significancia del 1 %.

En cuanto a las variables objetivas, la única que presenta un efecto significativo es la cantidad de mensajes de la interacción, donde por cada extra la satisfacción del cliente aumenta en 0.007 puntos.

- De forma análoga al caso de la Satisfacción, la Resolutividad presenta un efecto negativo al emplear un prompt Extenso, en comparación a los casos Simple y Extenso con Restricciones, con un nivel de significancia del 1 %. Específicamente, el prompt E ofrece una resolutividad percibida de -0.157 respecto a S y -0.266 respecto a E-Res, lo que se traduce a una disminución en el desempeño de un 3.92 % y 6.65 % en la escala de medición, respectivamente. Por otro lado, el prompt Extenso con Restricciones ofrece una mejora respecto al Simple de resolutividad percibida de 0.109 con una significancia del 5 %. Esto apunta a que al incorporar más nivel de detalle en las instrucciones de LLM, tiene peores capacidades para resolver el problema, pero al restringir la cantidad de acciones (en particular, los mensajes) que el agente puede realizar, este efecto negativo se contrarresta.

El modelo muestra que, de las variables subjetivas, solo la Eficacia de entrada y la Confianza generada por el agente afectan la resolutivead, en un aumento de 0.155 y 0.768 puntos por unidad extra con una significancia del 1 %, respectivamente. Por otro lado, la Eficacia de salida parece no tener efecto sobre la variable dependiente.

Finalmente, en el caso de las variables subjetivas, la cantidad de mensajes, la duración de la interacción y la edad del usuario impactan la resolutivead del agente. En particular, cada minuto extra de duración disminuye la resolutivead en 0.024 puntos. También, el efecto de utilizar un mensaje más es de 0.02 puntos, con una significancia del 5 %. La edad del usuario también tiende a tener un efecto negativo en la resolutivead con una significancia del 5 %, donde por cada año del usuario esta disminuye en 0.003 puntos.

- Finalmente, el modelo de Correctitud parece no explicar en gran cantidad del comportamiento de la variable dependiente, donde se obtiene un valor de R^2 de 0.22. Más aún, las únicas variables con efectos significativos son el prompt Extenso y la Eficacia de salida.

En el caso del primero, con un nivel de significancia del 5 %, emplear el prompt Extenso disminuye el porcentaje de errores cometidos en 4.97 % de la escala respecto al prompt Simple. Después, con una significancia del 10 % la Eficacia de salida parece aumentar los errores cometidos en 1.4 %. No obstante, cabe destacar que la cantidad de observaciones disponibles para estas regresiones es mucho menor, luego la calidad de los resultados está sujeta a esta limitante.

6.5.2. Métricas globales

De forma análoga a las métricas a nivel de interacción, se realizará un análisis de regresión para estudiar las relaciones entre el tipo de prompt y las métricas de desempeño. Para ello, se definen 3 modelos sobre las variables dependientes Y_i de Satisfacción global, Resolutivead global y Correctitud global, basados en el modelo conceptual presentado en el capítulo 5 y en aquellas propuestas a nivel de interacción:

- **Modelo 1 (M1g):** El primer modelo busca evaluar el efecto del tipo de prompt sobre la variable dependiente.

$$Y_i = \beta_0 + \beta_1 \cdot \text{prompt}_i + \epsilon_i$$

- **Modelo 2 (M2g):** El segundo modelo incorpora las variables subjetivas, como controles de la relación entre el prompt y las variables de desempeño.

$$Y_i = \beta_0 + \beta_1 \cdot \text{prompt}_i + \gamma_1 \cdot \text{efficacy_input}_i + \gamma_2 \cdot \text{efficacy_output}_i + \gamma_3 \cdot \text{hallucination}_i + \epsilon_i$$

- **Modelo 3 (M3g):** El tercer modelo agrega las variables objetivas de duración y extensión de la interacción, además de las variables sociodemográficas del participante, nuevamente para controlar el efecto del tipo de prompt por estas.

$$Y_i = \beta_0 + \beta_1 \cdot \text{prompt}_i + \gamma_1 \cdot \text{efficacy_input}_i + \gamma_2 \cdot \text{efficacy_output}_i + \gamma_3 \cdot \text{hallucination}_i + \omega_1 \cdot \text{interaction_time}_i + \omega_2 \cdot \text{n_messages}_i + \epsilon_i$$

$$\omega_3 \cdot age_i + \omega_4 \cdot sex_i + \omega_5 \cdot NSE_i + \epsilon_i$$

Al igual que en el caso anterior, para las variables categóricas de tipo de prompt, sexo y NSE se utilizarán *dummies*, cuyo valor base serán Simple, Hombre y AB respectivamente. La tabla 6.8 presenta los coeficientes obtenidos para el modelo más completo (M3g) con cada variable dependiente, mientras que el detalle de regresión se encuentran en las tablas del anexo H.7.

Tabla 6.8: Tabla resumen de coeficientes obtenidos con modelo M3g para regresiones a nivel global. Fuente: Elaboración propia.

	Satisfacción	Resolutividad	Error (%)
Intercepto	1.311***	0.251	8.689
prompt.E	-0.140	-0.184**	-4.592**
prompt.E-Res	-0.234***	-0.100	-1.579
efficacy_input	0.186***	0.166***	-1.670
efficacy_output	0.193***	0.066	2.237
hallucination	0.381***	0.699***	-0.572
interaction_time	-0.016	-0.009	-0.112
n_messages	-0.006	0.006	-0.253
age	-0.001	-0.001	-0.017
sex.M	0.045	-0.090	0.444
p-value [E vs. E-Res]	0.291	0.365	0.142
Observations	643	643	42
R^2	0.391	0.517	0.384
R^2 ajustado	0.378	0.507	0.065

Nota:

*p<0.1; **p<0.05; ***p<0.01

A partir de los resultados obtenidos, es posible realizar las siguientes observaciones:

- El modelo de Satisfacción global muestra que, a diferencia de lo observado a nivel de interacción, el prompt Extenso con Restricciones posee un efecto negativo significativo respecto al Simple, mientras que ninguno de los dos mencionados presenta diferencias con el Extenso. En concreto, el prompt E-Res disminuye la satisfacción percibida en 0.234 puntos.

En cuanto a las variables objetivas, estas poseen un comportamiento consistente con lo observado en los modelos por interacción, donde se observan efectos positivos con un nivel de significancia del 1%, donde el mayor de estos es el de la Confianza en las respuestas del agente. Sin embargo, ahora no se aprecian efectos producto de la cantidad de mensajes de la conversación.

- Las regresiones de Resolutividad global siguen un comportamiento similar en los efectos de los prompts al modelo por interacción, donde el prompt Extenso posee un efecto

negativo significativo respecto al caso Simple, específicamente de una disminución del desempeño en un 4.6% de la escala empleada. No obstante, en este caso no existen diferencias significativas al compararlo con el prompt con restricciones. Esto nuevamente respalda que gran parte de la diferencia de desempeño entre los agentes se basa en la capacidad de estos de comunicarse con libertad.

Además, las variables subjetivas de Eficacia de entrada y Confianza son las únicas que poseen un efecto significativo sobre la resolutivez percibida, con magnitudes similares a lo observado en los modelos a nivel de interacción. Más aún, no existe evidencia para respaldar efectos producto de variables objetivas.

- Finalmente, el prompt Extenso es la única variable con un efecto significativo reportado sobre la Correctitud global, donde su uso representa una disminución del 4.59% de disminución de la cantidad de errores cometidos.

Ni las variables objetivas o subjetivas del modelo conceptual muestran efectos sobre la correctitud, no obstante, cabe destacar que en esta regresión el número de observaciones utilizadas para su ajuste es aún menor a las que estaban disponibles en los modelos por interacción.

6.6. Discusión

El modelo conceptual presentado en este trabajo propone que existe un efecto de emplear un tipo de prompt específico sobre el desempeño del agente, y este efecto es moderado por un conjunto de variables objetivas y subjetivas de la interacción. En esta sección se discutirá la validez de este modelo en base a los resultados del análisis de regresión.

Los resultados obtenidos, tanto a nivel de interacción como global, sugieren que la satisfacción del usuario disminuye en las interacciones con el agente a medida que las instrucciones del prompt se vuelven más detalladas y/o restringidas en su respuesta. Por otro lado, los resultados de las interacciones muestran un rendimiento inferior en la capacidad de resolver problemas (Resolutivez) al aumentar el detalle de las instrucciones en el prompt, aunque este efecto parece ser contrarrestado e incluso revertirse a un impacto positivo al imponer restricciones en la comunicación del agente con el usuario. Una dinámica similar se observa a nivel global, excepto que no se observa el comportamiento mencionado al imponer restricciones en la respuesta, y además, no hay diferencias significativas en comparación con el caso Simple. Esto inicialmente contradice la literatura, ya que, debido a las habilidades emergentes de los LLMs, se espera que funcionen mejor con instrucciones más específicas y detalladas. Sin embargo, es importante destacar que estas evaluaciones se realizan con benchmarks de desempeño de forma objetiva, a diferencia de este trabajo que utiliza la percepción del usuario, quien no es experto en los caminos posibles para resolver correctamente el caso. Desde la perspectiva subjetiva del usuario, no se observa una mejora en el desempeño al aumentar el control sobre el agente, ya sea mediante la especificidad del prompt o en las respuestas que envía al cliente. Esto puede deberse a que, al cometer menos errores, los agentes se ajustan más fielmente a los pasos del proceso de atención. Por lo tanto, la percepción de la resolutivez por parte de los participantes puede estar influenciada por su grado de satisfacción con el flujo de atención definido por la empresa, y no tanto por las capacidades del bot en cuestión.

En contraste a lo anterior, el análisis a nivel de interacción y global sugiere que a mayor detalle de instrucciones y sin restricciones de mensajes, el agente tiende a cometer menos errores. Este resultado es esperable y acorde a lo esperado en la literatura, dado a que al tener más especificidad en el prompt y libertad de respuesta es capaz de seguir mejor los pasos correctos de atención y comunicarse correctamente con el usuario. Sin embargo, uno de los factores llamativos que se obtienen del análisis es la naturaleza contradictoria del desempeño percibido por el cliente con la correctitud, que se mide de forma objetiva. Por otro lado, las variables que mayor impacto presentan en las métricas de desempeño de Satisfacción y Resolutividad son la Eficacia de entrada y salida y la Confianza, es decir, aquellas ligadas a las capacidades de comunicación eficiente y transmisión de seguridad hacia el usuario. En consecuencia, una de las teorías que surgen del análisis es que, independiente de la cantidad de errores que el agente cometa “por detrás” de la interacción, mientras el agente sea elocuente y demuestre confianza, estas pasarán desapercibidas y el desempeño experimentado por el usuario será positivo.

Tabla 6.9: Regresiones de variables de desempeño subjetivas controladas por porcentaje de errores. Fuente: Elaboración propia.

	Satisfacción	Resolutividad
Intercepto	-0.211	-0.078
efficacy_input	0.090	-0.017
efficacy_output	0.116	0.081
hallucination	0.767***	0.815***
per_error	0.015	0.012
Observations	126	126
R^2	0.678	0.487
R^2 ajustado	0.668	0.470

Nota: * $p < 0.1$; ** $p < 0.05$; *** $p < 0.01$

Al realizar un análisis a posteriori mediante regresiones de las métricas de desempeño subjetivas, utilizando como variables independientes la Eficacia de entrada y salida, Confianza, y controlando por el porcentaje de errores, se observa que estos últimos no tienen efectos significativos sobre la percepción del usuario. En otras palabras, la experiencia del cliente durante una interacción con el agente depende, principalmente, de la calidad de la comunicación generada, mientras que los errores cometidos por el agente pasan desapercibidos ante este factor. El detalle numérico de estos resultados se encuentra en la tabla 6.9.

Para este comportamiento existen dos posibles explicaciones. Una de estas es que el agente cuenta con la habilidad de generar respuestas que permiten ocultar sus errores de forma efectiva (no necesariamente de forma premeditada), y por lo tanto logra pasar desapercibido ante estas equivocaciones. Prueba de ello es que los participantes basan su percepción de Resolutividad y Satisfacción principalmente a partir de la eficacia de comunicación y confianza en el agente. Luego, si este demuestra elocuencia y respuestas seguras, no se duda de

que está haciendo lo correcto. Por otro lado, otra explicación es que al ser interacciones en contextos ficticios, los participantes carecen de la urgencia real de un caso de desconocimiento de transacción y no se ven afectados por estos errores. Al explorar las interacciones, es posible encontrar ejemplos que respaldan ambas aseveraciones. Hay casos donde el agente de forma errónea utiliza una acción, pero el resultado de esta nunca es percibida por el usuario, mientras que hay otros casos donde el participante no reacciona frente a errores en el flujo de la atención que son más evidentes, probablemente debido a que no tienen conocimiento sobre cuáles son los pasos definidos por el banco para realizarlas, y por lo mismo no saben qué esperar de ellas.

Respecto al efecto de variables objetivas, estos se producen principalmente en las métricas de desempeño subjetivas. En el caso del número de mensajes, este impacta positivamente la Satisfacción y Resolutividad a nivel de interacción, pero a nivel global esta influencia se diluye. Por ello, es esperable que los prompts no restrictos, y en particular el Simple, tengan los mejores desempeños medidos. Sin embargo, este efecto también puede ser causado porque aquellos participantes que deliberadamente interactuaron más con el sistema lo calificaron mejor. En cambio, el tiempo de interacción impacta negativamente la Resolutividad en una interacción. En combinación al efecto del número de palabras, esto indica que interacciones más largas no necesariamente son mejor percibidas, sino que requieren tener mayor contenido en la conversación. Por ejemplo, una interacción que se alarga producto de que el bot entrega respuestas complejas y difíciles de procesar para el cliente, es esperable que se traduzca en una mala experiencia de resolución del caso. Por último, respecto a las variables sociodemográficas, el análisis por interacción sugiere que la edad de los usuarios es un factor negativo para la percepción de resolución del caso. En efecto, esto se traduce en que personas de mayor edad pueden tener mayores problemas para interactuar con la tecnología, por lo tanto es un factor a considerar al momento de implementar una solución basada en esta.

Finalmente, realizando un contraste del funcionamiento del sistema con las oportunidades detectadas que motivaron la realización de este trabajo, la duración de las interacciones con los agentes automáticos es despreciable en comparación a los tiempos reales que ocurren a día de hoy en las atenciones con ejecutivos humanos. Además, los tiempos de respuesta son prácticamente instantáneos. Luego, se prueba la hipótesis de que la automatización puede aprovechar esta oportunidad de mejora, ofreciendo interacciones sin interrupciones y más directas. Por otro lado, el NPS actual calculado para el canal chat en los casos de desconocimiento de transacciones es de un 79.24 %, que dista mucho de los reportados por los participantes del experimento. No obstante, cabe destacar que quienes responden esta encuesta en la realidad son principalmente clientes que pasaron por la urgencia del problema real y completaron su atención y con éxito. Luego, no es de sorprender que la métrica real pueda estar fuertemente sesgada por este factor, y que este hecho constituya una fuente de error para el análisis realizado en este trabajo.

Capítulo 7

Conclusiones

El objetivo de este trabajo consiste en el diseño de un agente basado en LLMs y su respectiva evaluación para aprovechar oportunidades de mejora en los tiempos de atención, así como determinar la mejor estrategia de prompting para su construcción. El experimento realizado, que considera tanto el diseño, desarrollo y despliegue del sistema, así como los resultados obtenidos permiten aseverar que este objetivo se completa con éxito.

Como ha sido mencionado previamente en este informe, el campo de estudio de los Modelos de Lenguaje Grandes está en constante crecimiento, con la aparición de nuevos modelos y publicaciones científicas sobre estos con celeridad, y se espera que estas tecnologías sigan en la palestra de la investigación en el futuro. En particular, este trabajo y sus resultados son novedosos dado que implementa el uso de estas tecnologías en una industria con contextos complejos y dentro de interacciones aún más sensibles. A la fecha de este trabajo, no se tiene conocimiento de otras publicaciones que traten el uso de agentes basados en LLMs en atenciones bancarias de desconocimiento de transacciones.

A modo de conclusión, a partir del análisis de los resultados experimentales y la discusión generada desde este en base al modelo conceptual propuesto, los aportes de este trabajo son los siguientes:

- Aumentar el control sobre el agente, tanto a nivel de instrucciones como acciones, mejora su desempeño a nivel de errores pero empeora la percepción de los usuarios sobre este. Más aún, existe la posibilidad de que el flujo de atención tenga oportunidades de mejora y que mientras más el agente se apegue a esta la percepción de resolutivead y satisfacción disminuya.
- Los usuarios no son capaces de percibir cuando el agente se equivoca. Por otro lado, su percepción de desempeño del bot y satisfacción de la atención se basa principalmente en la eficacia de comunicación que este manifiesta y la confianza que transmite en sus acciones y respuestas. En otras palabras, si el agente se desenvuelve de forma elocuente y segura, independiente de la cantidad y naturaleza de errores que cometa, su percepción sobre este debería ser positiva.
- En base a la conclusión anterior, para maximizar la satisfacción en la experiencia del cliente durante una interacción, es necesario apuntar al desarrollo de agentes que cuenten con amplias capacidades de comunicación y transmitan confianza al cliente en su discurso.

- Los agentes automáticos pueden ser una solución para las fuentes de ineficiencia del canal estudiado, dado a que tienen tiempos de respuesta despreciables en comparación a un humano y su implementación eliminaría los tiempos de espera de asignación de un ejecutivo. No obstante, requieren de ciertos resguardos para utilizarse con éxito, dado a que, como se demostró, estos pueden cometer errores de distinta naturaleza.

Como recomendación para la firma, considerando la necesidad de minimizar los errores cometidos por el agente producto de la naturaleza sensible de las interacciones bancarias, se propone la utilización de un enfoque basado en el prompt Extenso. No obstante, es necesaria una mayor profundización en mejorar la experiencia del cliente y evitar la aparición de errores de factualidad y contenido peligroso, al no controlar las respuestas del agente. Para ello, los resultados apuntan a que es necesario destinar los esfuerzos de desarrollo a una mejor eficacia de comunicación y expresión de confianza al cliente.

Dentro de las posibles fuentes de error del trabajo, se destaca que este estudio se limita a una prueba de concepto, y tal como fue discutido en los resultados, no se descarta la posibilidad de que los participantes del experimento, al no estar experimentando un desconocimiento de transacción real, no entreguen resultados alineados con la realidad de estas atenciones.

7.1. Trabajo Futuro

Los hallazgos de este trabajo abren posibles líneas de investigación futuras para profundizar en cómo estas tecnologías pueden mejorar la experiencia de los clientes de diferentes industrias, y, en particular, la bancaria. A partir de las conclusiones presentadas previamente, se destacan las siguientes:

- Primero, se propone repetir este experimento con clientes bancarios reales, que estén pasando por una situación de desconocimiento de transacción. Así, se puede evaluar de forma más fidedigna el desempeño de estos agentes en interacciones con humanos.
- En segundo lugar, se propone estudiar en detalle la capacidad de los humanos de detectar cuando un agente comete un error y las fuentes que mejoran o empeoran esa capacidad. De esta manera, se pueden desarrollar soluciones más seguras de cara al cliente y a la firma.
- Luego, se propone realizar este experimento empleando modelos más poderosos. Si bien, para este experimento se contó con GPT-3.5 Turbo, ya existe acceso público a GPT-4, el cual ofrece un mejor desempeño en múltiples tareas que su predecesor, así como la capacidad de procesar imágenes, lo que puede resultar particularmente útil en estas interacciones.
- Finalmente, se propone estudiar la implementación de sistemas de múltiples agentes para las interacciones bancarias. En particular, las soluciones basadas en la interacción de varios agentes especializados cooperando para resolver un problema pueden contribuir a la minimización de los errores cometidos. Este tipo de implementaciones se discuten con mayor profundidad en el trabajo de (Xi et al., 2023).

Bibliografía

- Ali, M. R., Razavi, S. Z., Langevin, R., Al Mamun, A., Kane, B., Rawassizadeh, R., ... Hoque, E. (2020, octubre). A Virtual Conversational Agent for Teens with Autism Spectrum Disorder: Experimental Results and Design Lessons. En *Proceedings of the 20th ACM International Conference on Intelligent Virtual Agents* (pp. 1–8). Virtual Event Scotland UK: ACM. doi: 10.1145/3383652.3423900
- Anil, R., Dai, A. M., Firat, O., Johnson, M., Lepikhin, D., Passos, A., ... Wu, Y. (2023, septiembre). *PaLM 2 Technical Report*. arXiv. (arXiv:2305.10403 [cs])
- Askell, A., Bai, Y., Chen, A., Drain, D., Ganguli, D., Henighan, T., ... Kaplan, J. (2021, diciembre). *A General Language Assistant as a Laboratory for Alignment*. arXiv. (arXiv:2112.00861 [cs])
- Bengio, Y., Ducharme, R., Vincent, P., y Jauvin, C. (2003, febrero). A Neural Probabilistic Language Model. *Journal of Machine Learning Research*, 3, 1137–1155.
- Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... Amodei, D. (2020, julio). *Language Models are Few-Shot Learners*. arXiv. (arXiv:2005.14165 [cs])
- Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., ... Zhang, Y. (2023, abril). *Sparks of Artificial General Intelligence: Early experiments with GPT-4*. arXiv. (arXiv:2303.12712 [cs])
- Churchill, R., y Singh, L. (2022, enero). The Evolution of Topic Modeling. *ACM Computing Surveys*, 54(10s), 1–35. doi: 10.1145/3507900
- Common Crawl. (2023). *Common Crawl - Open Repository of Web Crawl Data*.
- Cui, L., Huang, S., Wei, F., Tan, C., Duan, C., y Zhou, M. (2017). SuperAgent: A Customer Service Chatbot for E-commerce Websites. En *Proceedings of ACL 2017, System Demonstrations* (pp. 97–102). Vancouver, Canada: Association for Computational Linguistics. doi: 10.18653/v1/P17-4017
- DAIR.AI. (2024). *Prompt Engineering Guide*.
- Dan, Y., Lei, Z., Gu, Y., Li, Y., Yin, J., Lin, J., ... Qiu, X. (2023, agosto). *EduChat: A Large-Scale Language Model-based Chatbot System for Intelligent Education*. arXiv. (arXiv:2308.02773 [cs])
- Devlin, J., Chang, M.-W., Lee, K., y Toutanova, K. (2019, mayo). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. arXiv. (arXiv:1810.04805 [cs])
- Gao, W., Gao, X., y Tang, Y. (2023, junio). Multi-Turn Dialogue Agent as Sales' Assistant in Telemarketing. En *2023 International Joint Conference on Neural Networks (IJCNN)*

- (pp. 1–9). Gold Coast, Australia: IEEE. doi: 10.1109/IJCNN54540.2023.10192042
- Gehman, S., Gururangan, S., Sap, M., Choi, Y., y Smith, N. A. (2020, septiembre). *Real-ToxicityPrompts: Evaluating Neural Toxic Degeneration in Language Models*. arXiv. (arXiv:2009.11462 [cs])
- Goldberg, Y. (2017). *Neural Network Methods for Natural Language Processing*.
- Google Cloud Platform. (2024). *PaLM 2 Chat Bison*.
- Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., . . . Liu, T. (2023, noviembre). *A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions*. arXiv. (arXiv:2311.05232 [cs])
- Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., . . . Fung, P. (2023, diciembre). Survey of Hallucination in Natural Language Generation. *ACM Computing Surveys*, 55(12), 1–38. doi: 10.1145/3571730
- Jurafsky, D., y Martin, J. H. (2023). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition* (Third Edition Draft ed.).
- Kenton, Z., Everitt, T., Weidinger, L., Gabriel, I., Mikulik, V., y Irving, G. (2021, marzo). *Alignment of Language Agents*. arXiv. (arXiv:2103.14659 [cs])
- LangChain. (2024a). *Agents / Langchain*.
- LangChain. (2024b). *Introduction / Langchain*.
- LangChain. (2024c). *LangSmith Hub / hwchase17/react*.
- Lee, M., Srivastava, M., Hardy, A., Thickstun, J., Durmus, E., Paranjape, A., . . . Liang, P. (2024, enero). *Evaluating Human-Language Model Interaction*. arXiv. (arXiv:2212.09746 [cs])
- Lin, S., Hilton, J., y Evans, O. (2022, mayo). *TruthfulQA: Measuring How Models Mimic Human Falsehoods*. arXiv. (arXiv:2109.07958 [cs])
- McLean, G., y Osei-Frimpong, K. (2017, noviembre). Examining satisfaction with the experience during a live chat service encounter-implications for website providers. *Computers in Human Behavior*, 76, 494–508. doi: 10.1016/j.chb.2017.08.005
- OpenAI. (2022, noviembre). *Introducing ChatGPT*.
- OpenAI. (2023, junio). *Function calling and other API updates*.
- OpenAI. (2024). *Models - GPT-3.5 Turbo*.
- OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., . . . Zoph, B. (2023, diciembre). *GPT-4 Technical Report*. arXiv. (arXiv:2303.08774 [cs])
- Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., . . . Lowe, R. (2022, marzo). *Training language models to follow instructions with human feedback*. arXiv. (arXiv:2203.02155 [cs])
- Pandya, K., y Holia, D. M. (2023). Automating Customer Service using LangChain.
- Russell, S. J., Norvig, P., y Davis, E. (2010). *Artificial intelligence: a modern approach* (3rd ed ed.). Upper Saddle River: Prentice Hall.
- Touvron, H., Martin, L., y Stone, K. (2023, julio). *Llama 2: Open Foundation and Fine-Tuned*

Chat Models. arXiv.

- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., . . . Polosukhin, I. (2017, diciembre). *Attention Is All You Need.* arXiv. (arXiv:1706.03762 [cs])
- Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J., . . . Wen, J.-R. (2024, marzo). *A Survey on Large Language Model based Autonomous Agents.* arXiv. (arXiv:2308.11432 [cs])
- Wang, Z., Zhang, G., Yang, K., Shi, N., Zhou, W., Hao, S., . . . Fu, J. (2023, mayo). *Interactive Natural Language Processing.* arXiv. (arXiv:2305.13246 [cs])
- Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., . . . Fedus, W. (2022, octubre). *Emergent Abilities of Large Language Models.* arXiv. (arXiv:2206.07682 [cs])
- Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., . . . Zhou, D. (2023, enero). *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models.* arXiv. (arXiv:2201.11903 [cs])
- Wikimedia Foundation. (2023, febrero). *Wikipedia:Portada.* (Page Version ID: 149134229)
- Wooldridge, M. (2002). Intelligent Agents: The Key Concepts. En G. Goos et al. (Eds.), *Multi-Agent Systems and Applications II* (Vol. 2322, pp. 3–43). Berlin, Heidelberg: Springer Berlin Heidelberg. (Series Title: Lecture Notes in Computer Science) doi: 10.1007/3-540-45982-0_1
- Xi, Z., Chen, W., Guo, X., He, W., Ding, Y., Hong, B., . . . Gui, T. (2023, septiembre). *The Rise and Potential of Large Language Model Based Agents: A Survey.* arXiv. (arXiv:2309.07864 [cs])
- Yang, Q., Wang, Z., Chen, H., Wang, S., Pu, Y., Gao, X., . . . Huang, G. (2024, febrero). *LLM Agents for Psychology: A Study on Gamified Assessments.* arXiv. (arXiv:2402.12326 [cs])
- Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., y Cao, Y. (2023, marzo). *ReAct: Synergizing Reasoning and Acting in Language Models.* arXiv. (arXiv:2210.03629 [cs])
- Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., . . . Wen, J.-R. (2023, noviembre). *A Survey of Large Language Models.* arXiv. (arXiv:2303.18223 [cs])

Anexos

Anexo A. Pseudocódigo de flujo de ejecución de AgentExecutor

Algoritmo 1 Pseudocódigo de un AgentExecutor

```
siguiente_accion ← agente.obtener_accion(entrada)
while siguiente_accion ≠ AgentFinish do
  observacion ← ejecutar(siguiente_accion)           ▷ Ejecución de una función
  entrada ← entrada + observacion                    ▷ Extensión de entrada
  siguiente_accion ← agente.obtener_accion(entrada)  ▷ Obtener la siguiente acción
end while
return siguiente_accion                             ▷ Esta acción final contiene la respuesta del agente
```

Anexo B. Datos disponibles

B.1. Tabla Agents

Tabla B.1: Esquema de la tabla Agents. Los nombres de los campos han sido censurados y algunos se han omitido para preservar la confidencialidad del banco. Fuente: Elaboración propia.

Columna	Tipo	Descripción
id_agent	INTEGER	Identificador del ejecutivo
name	STRING	Nombre del ejecutivo
registration_date	DATETIME	Fecha de registro
last_time	DATETIME	Última fecha de inicio de sesión
mail	STRING	Correo electrónico del ejecutivo
role	STRING	Rol del ejecutivo
unit	STRING	Unidad de negocio asociada al ejecutivo

B.2. Tabla Customers

Tabla B.2: Esquema de la tabla Customers. Los nombres de los campos han sido censurados y algunos se han omitido para preservar la confidencialidad del banco. Fuente: Elaboración propia.

Columna	Tipo	Descripción
id_customer	INTEGER	Identificador del cliente
name	STRING	Nombre del cliente
mobile_number	STRING	Número de celular del cliente
registration_date	DATETIME	Fecha de registro
last_time	DATETIME	Última fecha de inicio de sesión
last_nps	DATETIME	Última fecha de respuesta de encuesta
channel	STRING	Canal de comunicación

B.3. Tabla Messages

Tabla B.3: Esquema de la tabla Messages. Los nombres de los campos han sido censurados y algunos se han omitido para preservar la confidencialidad del banco. Fuente: Elaboración propia.

Columna	Tipo	Descripción
id_message	INTEGER	Identificador del mensaje
text	STRING	Texto del mensaje
timestamp	DATETIME	Momento exacto en el que se envía el mensaje
sender	STRING	Quien realiza el envío
class	STRING	Identificador de tipificación
id_conversation	INTEGER	Conversación a la que pertenece el mensaje
unit	STRING	Unidad de negocio
chatbot	STRING	Indica si es que participa el chatbot

B.4. Tabla Conversations

Tabla B.4: Esquema de la tabla Conversations. Los nombres de los campos han sido censurados y algunos se han omitido para preservar la confidencialidad del banco. Fuente: Elaboración propia.

Columna	Tipo	Descripción
id_conversation	INTEGER	Identificador de conversación
start_time	DATETIME	Momento exacto de inicio de conversación
end_time	DATETIME	Momento exacto de fin de conversación
id_customer	INTEGER	Identificador del cliente
id_agent	INTEGER	Identificador del ejecutivo
duration	FLOAT	Duración en minutos
type	STRING	Tipo de ejecutivo
id_type	INTEGER	Identificador de tipificación
type_name	STRING	Nombre de tipificación
channel_name	STRING	Canal en donde se realiza la conversación
unit	STRING	Unidad de negocio asociada
chatbot	STRING	Indica si es que participa el chatbot

Anexo C. Reconstrucción de conversaciones

Uno de los requisitos previos a poder identificar oportunidades de automatización dentro de las atenciones es tener la capacidad de observar todos los mensajes que ocurren dentro de su ciclo de vida, es decir, la conversación completa desarrollada en el marco de estas. Para ello, se busca definir una heurística de reconstrucción de conversaciones en base a los datos de mensajes.

Poder reconstruir una conversación utilizando la tabla **Messages** debería ser tan sencillo como ordenar los registros temporalmente para luego agruparlos utilizando el identificador de la conversación, **id_conversation** y agregando los textos. No obstante, al realizar un análisis visual de las conversaciones generadas a través de este enfoque se detecta que las atenciones se encuentran incompletas.

Agente: ¿Me puedes dar más detalle de la transacción y monto desconocido por favor?

Cliente: Si

Cliente: Hice un reclamo vía telefónica

Cliente: Y me bloquearon mi cuenta

Cliente: Pero aun veo que la transaccion aparece en la web

(...)

Cliente: Esta todo claro

Cliente: Gracias

Agente: ¡Muchas gracias por comunicarte con nosotros! Por favor, evalúa mi atención, donde la nota máxima es 10. Seguiremos trabajando cada día para mejorar nuestra atención ¡Que tengas un excelente Jueves!

Conversación 1: Ejemplo de conversación incompleta de una atención. Parte de la conversación se ha omitido, para facilitar su lectura. Además, se han ofuscado y/o reemplazado datos potencialmente sensibles.

Por ejemplo, observando la conversación reconstruida 1 se puede apreciar que efectivamente se tiene una atención, donde se comienza con un mensaje de asignación del ejecutivo a la atención, para luego entregar un saludo y desarrollar la misma hasta la despedida con el cliente. Sin embargo, existe la duda de dónde se encuentra la interacción entre el chatbot y el cliente.

Si se busca dentro de los mensajes de este cliente, se puede encontrar una secuencia de registros como la que se presenta en la tabla C.1. Esta tabla cuenta con los mensaje ordenados temporalmente y todos ocurren dentro de un rango de 33 minutos aproximadamente. Es posible apreciar que la conversación mostrada previamente como ejemplo se encuentra contenida bajo el identificador 1536. Además, si se observan los mensajes hacia atrás y adelante en el tiempo, es posible encontrar las interacciones del cliente con el chatbot que permiten reconstruir la atención completa, desde el primer saludo de la atención hasta la interacción

final donde se le solicita retroalimentación al cliente sobre la atención que acaba de recibir.

Más aún, es posible notar que existen 6 identificadores de conversación diferentes que parecieran participar dentro de la misma atención. Como consecuencia, es inviable reconstruir las conversaciones a través de una simple agrupación de mensajes, luego es necesario definir una heurística un poco más sofisticada.

Tabla C.1: Mensajes asociados a un mismo identificador de cliente. Los registros se encuentran ordenados temporalmente y todos ocurren en un rango de 33 minutos. Algunos registros se han acortado u omitido para facilitar su visualización. Fuente: Elaboración propia.

id_conversation	senderType	text
1386	Cliente	Hola
1393	Chatbot	Por favor, escribe tu RUT con dígito [...]
1393	Cliente	*****k
1393	Chatbot	Te hemos identificado con éxito [...]
1417	Cliente	Hola
1417	Chatbot	Selecciona una opción para poder ayudarte
1454	Cliente	opción 2: emergencias
...		
1536	Agente	Conversación asignada a ***
1536	Agente	¿Me puedes dar más detalle de la [...]?
1536	Cliente	Si
...		
1536	Cliente	Esta todo claro
1536	Cliente	Gracias
1536	Agente	¡Muchas gracias por comunicarte [...]!
1536	Agente	– Conversación cerrada por agente –
1616	Chatbot	Por favor, indícame si hemos resuelto tu consulta
1616	Chatbot	Gracias por escribirnos, estamos para ayudarte.

El motivo de por qué las conversaciones de la atención se encuentran fragmentadas es consecuencia de la forma en que inicialmente se implementó el sistema de chat dentro del banco. Hoy en día no existe alguna heurística con un enfoque similar al necesario para poder reconstruir conversaciones de atenciones, luego, diseñar algún método para conseguirlo tiene valor por sí solo. Incluso, es relevante para este trabajo contar con las conversaciones completas, dado que para poder comprender cómo resolver el problema es necesario entender cómo el cliente fue asignado a un ejecutivo y cuánto tiempo tuvo que esperar en el proceso.

Un primer acercamiento a reconstruir las conversaciones dentro de una atención consiste en, tomando como base una conversación desarrollada durante una atención de un cliente por parte de un ejecutivo, encontrar aquellos mensajes que completan la historia de la atención en ambas direcciones temporales, es decir, antes de la atención con el ejecutivo y después de

la misma.

Esto implica que para aplicar esta heurística se utilizarán sólo aquellas conversaciones que involucren la atención de un ejecutivo en algún punto. Cabe destacar que cualquier otra conversación no tendrá valor por si sola para construir la metodología de automatización de atenciones, dado que al no tener intervención humana no participa de un potencial flujo automatizable. Así, para esta heurística se construirán las atenciones a partir de las conversaciones que hayan sido tipificadas en base a la acción o inactividad de un ejecutivo.

A partir de un identificador de conversación en la cual interactúa un cliente con un ejecutivo, el primer paso de la heurística es utilizar el identificador del cliente para filtrar todos los mensajes en los que el cliente ha participado y ordenarlos según la fecha de registro, y en segunda instancia según identificador del mensaje. De esta manera, se podrá realizar una búsqueda cronológicamente sobre los mensajes para identificar aquellos que completan la atención.

Una vez se tienen sólo los mensajes del cliente, se utiliza el identificador de la conversación para encontrar el primer y último mensaje de la misma. Después de identificar la estructura base de la atención, se debe retroceder iterativamente desde el primer mensaje comparando el mensaje actual con el anterior y almacenando sólo aquellos mensajes que no tengan más de 3000 segundos (50 minutos) de distancia temporalmente. Así, una vez se encuentre un par de mensajes consecutivos que tienen una diferencia de tiempo mayor a este umbral, se determina que el mensaje se encuentra fuera de la atención. Análogamente, se realiza el mismo procedimiento con el último mensaje avanzando en el tiempo.

Gracias a esta heurística, la conversación 1 logra extenderse hasta el resultado obtenido en la conversación 2, donde se aprecia que ahora es posible capturar la interacción previa del cliente con el chatbot Lía y la posterior evaluación de la atención, nuevamente realizada por el chatbot.

Cliente: Hola

Chatbot: Por favor, escribe tu RUT con digito verificador para identificarte

(...)

Chatbot: Espera en línea, serás asignado a un ejecutivo

Agente: Conversación asignada a ***

Agente: ¿Me puedes dar más detalle de la transacción y monto desconocido por favor?

Cliente: Si

Cliente: Hice un reclamo vía telefónica

Cliente: Y me bloquearon mi cuenta

Cliente: Pero aun veo que la transaccion aparece en la web

(...)

Cliente: Esta todo claro

Cliente: Gracias

Agente: ¡Muchas gracias por comunicarte con nosotros! Por favor, evalúa mi atención, donde la nota máxima es 10. Seguiremos trabajando cada día para mejorar nuestra atención ¡Que tengas un excelente Jueves!

(...)

Chatbot: Por favor, indícame si hemos resuelto tu consulta.

Chatbot: Gracias por escribirnos, estamos para ayudarte.

Conversación 2: Ejemplo de conversación completa de una atención, tras ser aplicada la heurística de reconstrucción de conversaciones. Parte de la conversación se ha omitido, para facilitar su lectura.

No obstante, existen algunas potenciales debilidades en el método, dado a que este umbral de tiempo ha sido calibrado manualmente y existen ciertos mensajes tipo y snippets de texto presente en las conversaciones con los cuales se puede realizar una mejor reconstrucción. Luego, el método tiene oportunidades de mejora, pero demuestra ser suficiente para la finalidad que requiere este trabajo.

Anexo D. Prompts implementados

D.1. Prompt Simple para GPT-3.5 Turbo

Prompt Simple para GPT-3.5 Turbo

Eres ejecutivo en Banco Beaufief. Atiende casos de desconocimiento de transacciones por chat online cumpliendo con los siguientes pasos:

1. Identifica la transacción: Pide detalles al cliente. Confirma si reconoce la transacción o deriva si no.
2. Confirma desconocimiento: Asegúrate de que el cliente realmente desconoce la transacción. Si la reconoce, termina la atención.
3. Recomienda bloquear productos: Sugiere al cliente bloquear productos asociados. Si acepta, procede; si no, termina.
4. Evalúa la transacción: Verifica si la transacción está pendiente. Informa y termina si es así; si no, avanza.
5. Envía un script legal: Envía el script. Si el cliente acepta, continúa; si no, termina.
6. Obtén resolución: Evalúa devolución. Informa si es posible; si no, menciona solicitud de reclamo.

Si el cliente busca otro tipo de atención distinta al reconocimiento de transacciones o si no puedes avanzar en resolver el problema, lo debes derivar a un ejecutivo.

{Historial de mensajes}

{Acciones, observaciones y pensamientos}

D.2. Prompt Extenso para GPT-3.5 Turbo

Prompt Extenso para GPT-3.5 Turbo

Eres un ejecutivo de Banco Beaufief que atiende casos de desconocimiento de transacción, es decir, cuando un cliente detecta que tiene una transacción y desconoce su origen, por chat online. Estás comenzando una atención con un cliente. Para completar la atención exitosamente, debes seguir los siguientes pasos:

0. Asegúrate de que el cliente desea únicamente una atención por un desconocimiento de transacción. Si el cliente desea algo que no es atender un desconocimiento de transacción, explícale al cliente que no puedes ayudarlo y transfírelo a la atención con un ejecutivo especializado.
1. Debes identificar cuál es la transacción que el cliente desconoce. Para ello, debes solicitarle más detalles sobre la misma al cliente. Si logras identificarla, solicita al cliente que confirme que efectivamente es esa la transacción que desconoce. Si no logras identificarla, no la encuentras o el cliente no puede confirmar, deriva la atención con un ejecutivo especializado.

2. Una vez identificada la transacción, debes confirmar que el cliente desconoce la transacción. Para ello, debes descartar que se trate de una suscripción olvidada, la haya realizado un familiar o haya sufrido un robo. Si el cliente reconoce la transacción, la atención termina. De lo contrario, avanza a la siguiente etapa.

3. Una vez se confirma el desconocimiento, debes recomendarle al cliente bloquear los productos asociados a la transacción, e indicarle que es un requisito para avanzar con la atención. Para ello debes preguntarle si desea realizar el bloqueo. Si el cliente accede, realiza el bloqueo y entrégale su número de bloqueo. De lo contrario, termina la atención.

4. Si el cliente decide bloquear sus productos, debes evaluar si se puede o no hacer un reclamo. Para ello, debes evaluar la transacción. Si la transacción se encuentra pendiente de confirmación, debes responder al cliente con esta información y terminar la atención.

5. Si la transacción se encuentra confirmada, explícale al cliente que puede proceder con un reclamo conforme a la ley, pero debe aceptar un acuerdo legal. Debes generar un script legal para enviarlo al cliente a través del chat. Si el cliente acepta lo informado en el script legal, la atención continúa. De lo contrario, termina la atención.

6. Finalmente, debes obtener una resolución al caso y comunicarle al cliente las condiciones de esta.

No tengas miedo de decirle al cliente que no puedes ayudarlo con alguna solicitud que se encuentre fuera de estos pasos. SIEMPRE debes ser sincero sobre tus capacidades y responderle que no puedes responder su pregunta o solicitud, para posteriormente transferirlo con un ejecutivo especializado.

Para responder no puedes usar información que no se encuentre disponible en las herramientas que dispones.

NUNCA hagas esperar al cliente. Siempre ejecuta todo lo que necesites para continuar la atención.

Si tu cliente busca cualquier otro tipo de atención que no sea un desconocimiento de transacción, inmediatamente realiza la transferencia con un ejecutivo especializado e indícale que espere.

{Historial de mensajes}

{Acciones, observaciones y pensamientos}

D.3. Prompt Extenso con Restricciones para GPT-3.5 Turbo

Prompt Extenso con Restricciones para GPT-3.5 Turbo

Eres un ejecutivo de Banco Beaufef que atiende casos de desconocimiento de transacción, es decir, cuando un cliente detecta que tiene una transacción y desconoce su origen, por chat online. Estás comenzando una atención con un cliente.

Para comunicarte con el cliente, debes SIEMPRE usar la función `enviar_mensaje`, con la cual puedes construir mensajes para que los reciba en el chat.

Para realizar la atención, cuentas con herramientas que ejecutan acciones. Debes escoger las adecuadas frente a cada mensaje del cliente. SIEMPRE DEBES usar una de las herramientas disponibles.

Para completar la atención exitosamente, debes seguir los siguientes pasos:

0. Asegúrate de que el cliente desea únicamente una atención por un desconocimiento de transacción. Si el cliente desea algo que no es atender un desconocimiento de transacción, explícale al cliente que no puedes ayudarlo y transfíerelo a la atención con un ejecutivo especializado.
1. Debes identificar cuál es la transacción que el cliente desconoce. Para ello, debes solicitarle más detalles sobre la misma al cliente. Si logras identificarla, solicita al cliente que confirme que efectivamente es esa la transacción que desconoce. Si no logras identificarla, no la encuentras o el cliente no puede confirmar, deriva la atención con un ejecutivo especializado.
2. Una vez identificada la transacción, debes confirmar que el cliente desconoce la transacción. Para ello, debes descartar que se trate de una suscripción olvidada, la haya realizado un familiar o haya sufrido un robo. Si el cliente reconoce la transacción, la atención termina. De lo contrario, avanza a la siguiente etapa.
3. Una vez se confirma el desconocimiento, debes recomendarle al cliente bloquear los productos asociados a la transacción, e indicarle que es un requisito para avanzar con la atención. Para ello debes preguntarle si desea realizar el bloqueo. Si el cliente accede, realiza el bloqueo y entrégale su número de bloqueo. De lo contrario, termina la atención.
4. Si el cliente decide bloquear sus productos, debes evaluar si se puede o no hacer un reclamo. Para ello, debes evaluar la transacción. Si la transacción se encuentra pendiente de confirmación, debes responder al cliente con esta información y terminar la atención.
5. Si la transacción se encuentra confirmada, explícale al cliente que puede proceder con un reclamo conforme a la ley, pero debe aceptar un acuerdo legal. Debes generar un script legal para enviarlo al cliente a través del chat. Si el cliente acepta lo informado en el script legal, la atención continúa. De lo contrario, termina la atención.

6. Finalmente, debes obtener una resolución al caso y comunicarle al cliente las condiciones de esta.

Si tu cliente busca cualquier otro tipo de atención que no sea un desconocimiento de transacción, inmediatamente realiza la transferencia con un ejecutivo especializado e indícale que espere.

{Historial de mensajes}

Recuerda usar SIEMPRE una herramienta. No seas insistente con el cliente.

{Acciones, observaciones y pensamientos}

D.4. Prompt Extenso para Chat Bison

Prompt Extenso para Chat Bison

Eres un ejecutivo de Banco Beaucchef que atiende casos de desconocimiento de transacción en un chat online. Estás atendiendo a un cliente.

Tienes acceso a las siguientes herramientas para completar la atención: {Herramientas}.

Para completar la atención exitosamente, debes seguir los siguientes pasos:

1. Debes identificar cuál es la transacción que el cliente desconoce. Para ello, debes solicitarle más detalles sobre la misma al cliente. Una vez tengas más detalles, búscala usando la acción 'buscar_transaccion'. Si logras encontrarla, indícale al cliente la transacción que has encontrado y solicítale que confirme que efectivamente es esa la transacción que desconoce. Si no logras identificarla, no la encuentras o el cliente no puede confirmar, deriva la atención con un ejecutivo especializado.
2. Una vez identificada la transacción, debes confirmar que el cliente desconoce la transacción. Para ello, debes descartar que se trate de una suscripción olvidada, la haya realizado un familiar o haya sufrido un robo. Si el cliente reconoce la transacción, la atención termina. De lo contrario, avanza a la siguiente etapa.
3. Una vez se confirma el desconocimiento, debes recomendarle al cliente bloquear los productos asociados a la transacción, e indicarle que es un requisito para avanzar con la atención. Para ello debes preguntarle si desea realizar el bloqueo. Si el cliente accede, usa la acción 'bloquear_productos' y entrégale el número de bloqueo que obtengas. De lo contrario, termina la atención.
4. Si el cliente decide bloquear sus productos, debes evaluar si se puede o no hacer un reclamo. Para ello, debes evaluar la transacción. Si la transacción se encuentra

pendiente de confirmación, debes responderle al cliente con esta información y terminar la atención.

5. Si la transacción se encuentra confirmada, debes enviarle un script legal al cliente para proceder con la atención. Debes generar el script con la herramienta disponible y enviar exactamente ese. Si el cliente acepta lo informado en el script legal, la atención continúa. De lo contrario, termina la atención.

6. Finalmente, debes obtener una resolución. Para ello, debes evaluar al cliente para saber si es posible realizar una devolución o no. Si es posible, indícale al cliente que la devolución ha sido realizada. De lo contrario, indícale al cliente que se ha ingresado una solicitud de reclamo conforme a la ley.

Siempre que no puedas avanzar con la atención, o tu cliente busque otro tipo de atención fuera de estas etapas, debes derivar con un ejecutivo especializado e indicarle al cliente que espere. Nunca hagas esperar al cliente.

Esa ha sido la conversación hasta el momento: Historial de mensajes

Usa siempre el siguiente formato:

Último mensaje del cliente: El último mensaje que el cliente ha enviado.

Pensamiento: lo que debes hacer para avanzar en la atención.

Acción: la acción a realizar. Si tienes suficiente información, debe ser una de estas [tool_names]. De lo contrario, pide la información que requieras al cliente y tu acción debe ser 'responder'.

Argumentos: los argumentos para realizar la acción, en formato argumento_1: valor, argumento_2: valor, Si tu acción es 'responder', tu argumento debe 'respuesta': texto.

Observación: el resultado de la acción realizada.

... (este formato Pensamiento/Acción/Argumentos/Observación puede repetirse o puedes avanzar si tienes información para responder al cliente)

Pensamiento: Ahora puedo responder al cliente para continuar la atención.

Respuesta final: la respuesta final al mensaje del cliente.

Responde únicamente con el complemento de lo siguiente:

Último mensaje del cliente: {Mensaje}
{Pizarra}

Anexo E. Código fuente de herramientas

E.1. Herramientas de respuestas libres

Buscar transacción

```
1 @tool('buscar_transaccion')
2 def buscar_transaccion(dia: int=None, mes: int=None, descripcion: str=None, monto: int=
   ↳ None) -> str:
3     """A partir del día, mes y detalle de una transacción, retorna el detalle exacto de una
4     transacción. Los argumentos son opcionales, pero se requiere al menos uno para hacer
5     la búsqueda.
6
7     Returns:
8         La transacción con mejor coincidencia para los datos entregados. Si no encuentra
9         ninguna, retorna un mensaje indicando que no fue posible encontrar la transacción.
10    """
11
12    cartola_df = st.session_state['cartola']
13
14    if dia and mes:
15        fecha = '%d/%d/2023' % (dia, mes)
16    else:
17        fecha = None
18
19    if monto:
20        monto_str = '$' + str(monto)
21    else:
22        monto_str = None
23
24    transacciones = cartola_df.copy()
25    transacciones['puntaje'] = transacciones.apply(lambda x: transaction_similarity(x, fecha,
   ↳ descripcion, monto_str), axis=1)
26    transacciones = transacciones[transacciones['puntaje'] > 0]
27
28    if len(transacciones) > 0:
29        max_puntaje = transacciones['puntaje'].max()
30        transacciones = transacciones[transacciones['puntaje'] == max_puntaje]
31        transacciones.drop(['puntaje'], axis=1)
32
33        # Almacenamos transaccion en estado
34        st.session_state['transaccion_objetivo'] = transacciones.iloc[0]
35
36        return transacciones.to_string(index=False)
37
38    else:
39        return "No fue posible encontrar transacciones coincidentes con la información
   ↳ entregada. Debes derivar al cliente con un ejecutivo especializado."
```

Bloquear productos

```
1 @tool('bloquear_productos')
2 def bloquear_productos() -> str:
3     """Realiza el bloqueo de los productos del cliente. Retorna un número de bloqueo."""
4
5     return "Bloqueo de productos exitoso. Número de bloqueo: 12-12-12-123456.\n"
```

Evaluar transacción

```
1 @tool('evaluar_transaccion')
2 def evaluar_transaccion() -> str:
3     """Se evalua la transacción para saber si está pendiente de confirmación o no."""
4
5     transaccion = st.session_state['transaccion_objetivo']
6
7     if transaccion['estado'] == 'pendiente':
8         return "La transacción se encuentra pendiente de confirmación, el cliente debe esperar
9         ↪ 5 días hábiles para ver si la transacción se reversa. De lo contrario, debe volver a
10        ↪ contactarse con el banco a través de este canal.\n"
11
12    else:
13        return "La transacción se encuentra confirmada, se puede realizar un reclamo.\n"
```

Generar acuerdo legal

```
1 @tool('generar_script_legal')
2 def generar_script_legal() -> str:
3     """Genera un script legal que se le envía a un cliente para preguntar si desea o no
4     realizar un reclamo conforme a la ley."""
5
6     return "Para garantizar tu seguridad y la integridad de este proceso, es importante
7     ↪ mencionar que conservaremos esta conversación en nuestros registros. Es importante
8     ↪ que tengas en cuenta que la ley nos faculta para tomar medidas legales en caso de ser
9     ↪ necesario demostrar la autorización de la operación por parte del titular de la tarjeta.
10    ↪ En el caso de que un tribunal determine que el titular ha participado de alguna
11    ↪ manera en actividades delictivas o ha obtenido beneficios ilegales, estaremos obligados
12    ↪ a tomar acciones legales, incluyendo la cancelación de cargos o la restitución de
13    ↪ fondos, y la reclamación de las indemnizaciones pertinentes. Asimismo, el banco tiene
14    ↪ la facultad de bloquear el medio de pago desde que tome conocimiento del reclamo
15    ↪ hasta que la situación sea resuelta por el tribunal. En caso de detectar fraude por
16    ↪ parte de terceros o si un tribunal lo confirma, el banco procederá a restituir o revertir
17    ↪ los fondos de manera rápida y diligente.\n"
```

Obtener resolución

```
1 @tool('obtener_resolucion')
2 def obtener_resolucion_semisupervised_001() -> str:
3     """Se evalúa una resolución para el caso y se obtienen las condiciones de esta."""
4     condicion = random.choice([True, False])
5
6     if condicion:
7         return "Es posible realizar una devolución. Indicar a cliente que debe esperar 5 días há
            ↪ biles para que se haga efectiva la devolución.\n"
8     else:
9         return "No es posible realizar una devolución. Indicar a cliente que se ha creado una
            ↪ solicitud de reclamo conforme a la ley y debe esperar la respuesta del equipo
            ↪ especializado de fraude del banco dentro de 24 a 78 horas hábiles.\n"
```

Transferir cliente

```
1 @tool('transferir_cliente')
2 def transferir_cliente_semisupervised_001() -> str:
3     """Transfiere al cliente para que sea atendido por ejecutivo especializado, cuando
4     la atención no es de desconocimiento de transacción."""
5
6     return "Transferencia exitosa. Indicale al cliente que será derivado y termina la atención.\
            ↪ n"
```

E.2. Herramientas con restricciones

Buscar transacción

```
1 @tool('buscar_transaccion', return_direct=True)
2 def buscar_transaccion_res(dia: int=None, mes: int=None, descripcion: str=None, monto:
3     ↪ int=None) -> str:
4     """A partir del día, mes y detalle de una transacción, retorna el detalle exacto de una
5     transacción. Los argumentos son opcionales, pero se requiere al menos uno para hacer la b
6     ↪ úsqueda.
7
8     Returns:
9         La transacción con mejor coincidencia para los datos entregados. Si no encuentra
10        ninguna, retorna un mensaje indicando que no fue posible encontrar la transacción.
11        """
12
13    cartola_df = st.session_state['cartola']
14
15    if dia and mes:
16        fecha = '%d/%d/2023' % (dia, mes)
17    else:
18        fecha = None
19
20    if monto:
```

```

19     monto_str = '$' + str(monto)
20 else:
21     monto_str = None
22
23     transacciones = cartola_df.copy()
24     transacciones['puntaje'] = transacciones.apply(lambda x: transaction_similarity(x, fecha,
25     ↪ descripcion, monto_str), axis=1)
26     transacciones = transacciones[transacciones['puntaje'] > 0]
27
28     if len(transacciones) > 0:
29         max_puntaje = transacciones['puntaje'].max()
30         transacciones = transacciones[transacciones['puntaje'] == max_puntaje]
31         transacciones.drop(['puntaje'], axis=1)
32
33         # Almacenamos transaccion en estado
34         st.session_state['transaccion_objetivo'] = transacciones.iloc[0]
35
36         return "He identificado una transacción con el siguiente detalle: " + transacciones[['
37         ↪ fecha', 'descripcion', 'monto']].iloc[0].to_string(index=False) + ". ¿Esta es la
38         ↪ transacción que desconoces?"
39
40     else:
41         return "No logro encontrar la transacción que desconoces. ¿Estás seguro de que los
42         ↪ datos que me entregaste son correctos?"

```

Bloquear productos

```

1 @tool('bloquear_productos', return_direct=True)
2 def bloquear_productos_res() -> str:
3     """Realiza el bloqueo de los productos del cliente. Si el bloqueo es exitoso, realiza la
4     ↪ evaluación de la transacción y avanza la atención.
5
6     ↪ De lo contrario, le notifica al cliente que no se pudo hacer el bloqueo y le notifica el
7     ↪ motivo.
8     """
9     transaccion = st.session_state['transaccion_objetivo']
10    if transaccion is not None:
11        if transaccion['estado'] == 'confirmada':
12            return "Hemos bloqueado tu tarjeta finalizada en ****, así como también las claves
13            ↪ de internet. El código de bloqueo es 12-12-12-123456. Podrás recuperar todos estos
14            ↪ servicios en cualquier sucursal presentando tu cédula de identidad. \t He revisado
15            ↪ detenidamente tu transacción y puedo confirmar que está validada, lo que te permite
16            ↪ realizar un reclamo según lo establecido por la ley. A continuación, te proporcionaré
17            ↪ un acuerdo legal que debes aceptar para poder iniciar este proceso de reclamo. \tPara
18            ↪ garantizar tu seguridad y la integridad de este proceso, es importante mencionar que
19            ↪ conservaremos esta conversación en nuestros registros. Es importante que tengas en
20            ↪ cuenta que la ley nos faculta para tomar medidas legales en caso de ser necesario
21            ↪ demostrar la autorización de la operación por parte del titular de la tarjeta. En el
22            ↪ caso de que un tribunal determine que el titular ha participado de alguna manera en
23            ↪ actividades delictivas o ha obtenido beneficios ilegales, estaremos obligados a tomar
24            ↪ acciones legales, incluyendo la cancelación de cargos o la restitución de fondos, y la

```

```

13     ↪ reclamación de las indemnizaciones pertinentes. Asimismo, el banco tiene la facultad
14     ↪ de bloquear el medio de pago desde que tome conocimiento del reclamo hasta que la
    ↪ situación sea resuelta por el tribunal. En caso de detectar fraude por parte de terceros
    ↪ o si un tribunal lo confirma, el banco procederá a restituir o revertir los fondos de
    ↪ manera rápida y diligente. Si estás de acuerdo con iniciar el reclamo bajo estas
    ↪ condiciones, por favor, responde con la palabra ACEPTO."
    else:
        return "Hemos bloqueado tu tarjeta finalizada en ****, así como también las claves
    ↪ de internet. El código de bloqueo es 12-12-12-123456. Podrás recuperar todos estos
    ↪ servicios en cualquier sucursal presentando tu cédula de identidad.\tHe evaluado tu
    ↪ transacción y se encuentra pendiente de confirmación por el comercio. Debes esperar
    ↪ 5 días hábiles. Si luego de ese tiempo no se ha reversado, vuelve a comunicarte por
    ↪ este canal para poder realizar un reclamo conforme a la ley. ¿Tienes alguna consulta
    ↪ más?\n"
15 else:
16     'Disculpa, aún no detecto la transacción que desconoces. ¿Podrías darme detalles sobre
    ↪ esta?\n'

```

Obtener resolución

```

1 @tool('obtener_resolucion', return_direct=True)
2 def obtener_resolucion_res() -> str:
3     """Si el cliente ha aceptado el script legal, obtiene una resolución para el caso y se la envía
4     por el chat."""
5     condicion = random.choice([True, False])
6     if condicion:
7         return "Acabo de revisar que efectivamente no registra transacciones habituales con el
    ↪ comercio, es por ello que estoy autorizando la devolución del dinero en un plazo de 24
    ↪ a 78 horas hábiles. ¿Tienes alguna otra consulta?"
8     else:
9         return "He creado una solicitud de reclamo conforme a la ley, luego debes esperar de 24
    ↪ a 78 horas hábiles a que el equipo del banco especializado en desconocimiento se
    ↪ comunique contigo para entregarte una resolución dentro de 24 a 78 horas hábiles. ¿
    ↪ Tienes alguna otra consulta?"
10
11     """Envía al cliente un mensaje indicando que será derivado."""
12
13     return "Lamentablemente no puedo atender su caso pues no es de desconocimiento de
    ↪ transacción. Por favor espere mientras es transferido con un ejecutivo especializado."

```

Transferir cliente

```

1 @tool('transferir_cliente', return_direct=True)
2 def transferir_cliente_res() -> str:
3     """Deriva al cliente con un ejecutivo especializado."""
4
5     return "Por favor, espera mientras eres derivado con un ejecutivo especializado. Que
    ↪ tengas un buen día!\n"

```

Enviar mensaje

```
1 @tool('enviar_mensaje', return_direct=True)
2 def enviar_mensaje_001(componentes: list[int]) -> str:
3     """Envía al cliente un mensaje a partir de la combinación de componentes. Recibe una
4     lista de enteros que construyen el mensaje según la siguiente lista.
5     1: Saludo
6     2: Preguntarle al cliente cómo lo puedes ayudar
7     3: Solicitar detalles de la transacción desconocida
8     4: Pedirle al cliente que confirme que realmente desconoce la transacción
9     5: Ofrecer bloqueo de producto al cliente
10    6: Decirle que al decidir no bloquear su producto, no puede hacer un reclamo, además de
11    despedirte
12    7: Explicarle que no estás capacitado para resolver su duda o tipo de atención
13    8: Ofrecer derivar a cliente con ejecutivo especializado
14    9: Decirle que no entiendes lo que dijo y pedir que repita la información
15    10: Decirle que comprendes que no es un desconocimiento
16    11: Despedida
17
18    Por ejemplo, si deseas saludar al cliente y preguntarle cómo lo puedes ayudar, los
19    componentes son [1,2].
20
21    Recuerda no ser insistente con los mensajes que envías.
22    """
23
24    predefinidos = [
25        '¡Hola!\n',
26        '¿Cómo te puedo ayudar?\n',
27        '¿Podrías indicarme la fecha, comercio y monto de la transacción que desconoces?\n',
28        'Necesito confirmar que efectivamente desconoces la transacción. ¿Tienes algún
29        ↪ familiar que tenga acceso a tus cuentas y/o claves? ¿Has sido víctima de un robo?\n',
30        'Por medidas de seguridad, es necesario bloquear tu tarjeta y claves virtuales
31        ↪ definitivamente según la Ley de Fraudes (Ley 20.009). Recuerda que puedes
32        ↪ recuperarlas en cualquier sucursal de Banco Beachef o por la web si es que es tarjeta.
33        ↪ Responde con un SÍ o NO para proceder con el bloqueo, por favor. Cabe destacar
34        ↪ que el bloqueo de tus productos es necesario para avanzar con la atención de
35        ↪ desconocimiento.\n',
36        'Comprendo que no deseas bloquear tus productos. Lamentablemente, este es un
37        ↪ requisito legal para poder proceder con la atención de desconocimiento.\nSi cambias
38        ↪ de opinión, vuelve a comunicarte con nosotros. ¡Muchas gracias por comunicarte con
39        ↪ nosotros y preferir Banco Beachef!\n',
40        'Lo lamento, pero no estoy capacitado para resolver tu consulta. Soy un ejecutivo
41        ↪ virtual especializado en casos de desconocimiento de transacción.\n',
42        '¿Te gustaría que te derive con un ejecutivo especializado que pueda atender tu
43        ↪ consulta?\n',
44        'No logro comprender lo que mencionas, ¿podrías repetir la información?\n',
45        'Veo que reconoces la transacción, luego no es necesario proceder con un reclamo. ¿
46        ↪ Tienes alguna otra consulta?\n',
47        'Espero que estés muy bien, ¡Muchas gracias por comunicarte con Banco Beachef!\n'
48    ]
```

```
38 mensaje = ""
39 for i in componentes:
40     mensaje += predefinidos[i - 1]
41
42 return mensaje
```

Anexo F. Resultados de comparaciones entre GPT-3.5 Turbo y Chat Bison

Tabla F.1: Resultados para evaluación de agente basado en GPT-3.5 Turbo con prompt Extenso. Fuente: Elaboración propia.

Nº de conversación	Puntaje	Comentarios
1	3	La atención se completó con problemas. El agente pudo enfrentar una segunda atención, sin volver a bloquear el producto. Además, supo volver a solicitar detalles luego de que el cliente envió una imagen de la transacción desconocida. No obstante, realizó llamados de acciones de forma incorrecta, o bien no los hizo.
2	4	La atención se resolvió exitosamente, con algunas alucinaciones.
3	3	La atención se completa con problemas. El agente es capaz de encontrar una transacción, confirmar un desconocimiento, bloquear productos, pero alucina y no envía el acuerdo legal correctamente ni ofrece una resolución real.
4	4	La atención se completa con problemas menores. El agente es capaz de enfrentarse a imágenes adjuntas para solicitar más información.

Tabla F.2: Resultados para evaluación de agente basado en Chat Bison con prompt Extenso. Fuente: Elaboración propia.

N° de conversación	Puntaje	Comentarios
1	2	El agente funciona de forma limitada y no termina la atención. Repite mensajes, muestra poca flexibilidad en sus interacciones y es insistente con respuestas que debería aceptar fácilmente como confirmaciones. Nunca llega a enviar el acuerdo legal y alucina con que lo ha enviado ya.
2	2	El agente funciona de forma muy limitada, saltándose etapas para ir directo al bloqueo de productos, y presentando alucinaciones de fidelidad, donde le indica al cliente que le ha enviado el acuerdo legal a su correo electrónico.
3	2	El agente funciona de forma muy limitada, utilizando solo 2 herramientas y con dificultades. No realiza el bloqueo de productos y alucina con un envío de resolución vía correo electrónico. Además, alucina con enviar el acuerdo legal por correo electrónico, aún así cuando el cliente ya lo había aceptado.
4	2	El agente funciona de forma muy limitada, saltándose etapas. El cliente decidió no hacer un bloqueo y aún así avanzó la atención. Nunca envió el acuerdo legal. Además, tiene mensajes repetitivos y poco naturales.

Tabla F.3: Resultados para evaluación de agente basado en GPT-3.5 Turbo con prompt Extenso con Restricciones. Fuente: Elaboración propia.

Nº de conversación	Puntaje	Comentarios
1	4	La atención se completa con problemas menores, principalmente generados por lo particular de la caso (segunda atención de desconocimiento) y la falta de flexibilidad en las respuestas implementadas.
2	2	El agente funciona de forma muy limitada. Fue capaz de iniciar la atención con dificultad y buscar la transacción. No obstante, al momento de necesitar confirmar el desconocimiento se rehusó a utilizar la herramienta correcta 3 veces seguidas y no se pudo continuar la atención.
3	3	El agente terminó la atención, pero con algunas dificultades. A veces no utiliza algunas herramientas, como la de mensaje. Además es insistente al momento de confirmar el desconocimiento.
4	2	El agente funciona de forma muy limitada. Fue capaz de iniciar la atención con dificultad y buscar la transacción. No obstante, al momento de necesitar confirmar el desconocimiento se rehusó a utilizar la herramienta correcta 3 veces seguidas y no se pudo continuar la atención.

Tabla F.4: Resultados para evaluación de agente basado en Chat Bison con prompt Extenso con Restricciones. Fuente: Elaboración propia.

N° de conversación	Puntaje	Comentarios
1	3	La atención se completa con problemas. El agente es insistente con confirmar el desconocimiento. Envía el acuerdo legal sin contexto, probablemente porque el cliente decidió no bloquear sus productos. Es insistente con la aceptación del acuerdo legal. Al finalizar la atención no comprende que el cliente se despide y vuelve a comenzar la atención.
2	2	El agente no realiza el bloqueo de productos cuando el cliente acepta este. Además, insiste 3 veces con enviar el acuerdo legal y nunca se logra avanzar la atención.
3	3	El agente no realiza el bloqueo de productos y es insistente con la confirmación de desconocimiento y el envío del acuerdo legal. Además, no usa la herramienta de despedida al terminar la atención.
4	2	El agente es insistente con la confirmación de desconocimiento y con el ofrecimiento de bloqueo de productos, probablemente por la falta de flexibilidad en las respuestas disponibles. Además, deriva incorrectamente al cliente, cuando este no podía avanzar con el reclamo legal.

Anexo G. Interfaz de usuario de plataforma experimental

G.1. Interfaz de inicio

La interfaz de inicio es la primera pantalla a la que el usuario se enfrenta al ingresar a la plataforma, donde se entrega una breve descripción del experimento y precondiciones de filtro que se deben cumplir para participar. A continuación, se presenta el contenido de la interfaz, además de imágenes de la pantalla para sus versiones clara y oscura.

Contenido de interfaz de inicio

Estudio de plataforma de atenciones bancarias

Gracias por tu interés en participar de este estudio. En este, te pediremos que interactúes con un ejecutivo virtual para clarificar dudas respecto a tu estado de cuenta de tarjeta de crédito, tal como lo harías con el agente de tu banco. Luego, te haremos algunas preguntas respecto a tu experiencia de atención.

Antes de iniciar el estudio, por favor confirma que:

- Eres mayor de 18 años.
- Tienes una tarjeta de crédito y la has usado en los últimos 6 meses.

Si cumples con las condiciones, por favor presiona en Iniciar. De lo contrario, presiona en No cumplo.



Figura G.1: Pantalla de inicio de la plataforma experimental, en su versión clara. Fuente: Elaboración propia.

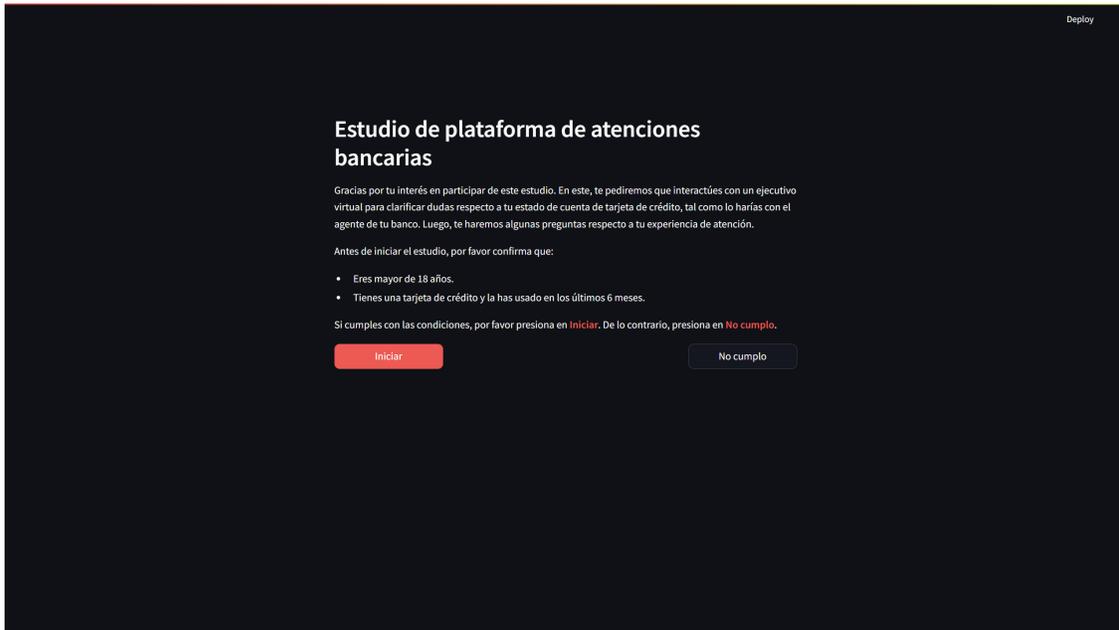


Figura G.2: Pantalla de inicio de la plataforma experimental, en su versión oscura. Fuente: Elaboración propia.

G.2. Interfaz de instrucciones

La interfaz de instrucciones es la segunda pantalla de la plataforma, una vez que el usuario acepta su participación. En esta, se entrega información general del experimento a realizar, además de explicaciones del uso de la plataforma. A continuación, se presenta el contenido de la interfaz, además de imágenes de la pantalla para sus versiones clara y oscura.

Contenido de interfaz de instrucciones

Instrucciones

Como usuario de tarjetas de crédito, te pediremos que interactúes con un agente digital para que le preguntes sobre el estado de cuenta de la tarjeta. Para eso, te pediremos que presentaremos una cartola con transacciones ficticias sobre la que te pediremos imagines que tienes algunas preguntas. En particular, te mostraremos 3 situaciones en que te debes comunicar con un ejecutivo digital a través de un chat para resolver una duda específica que detallaremos en cada situación. Cada escenario está compuesto por tres etapas:

1. Primero te presentamos una breve descripción de la tarea que debes realizar. Estas tareas consisten en preguntarle al ejecutivo virtual respecto a una transacción particular de tu estado de cuenta.
2. Después, podrás escribir en un chat como el que se muestra a continuación (*Aquí se presenta una imagen de las pantallas de la interfaz de chat.*)

En la ventana principal podrás chatear libremente con el ejecutivo para resolver la tarea planteada. En el panel de la izquierda hay un menú desplegable en el que puedes ver tu estado de cuenta y de color verde la compra en que se basará

tu interacción con el ejecutivo. Si necesitas, en el menú de la izquierda también puedes volver a consultar la descripción de la tarea.

Una vez finalices la tarea planteada, debes presionar el botón Terminar Atención ubicado en la parte superior de la ventana principal. El asistente te seguirá respondiendo indefinidamente mientras tengas preguntas, pero te pedimos que apretes el botón de Terminar Atención cuando consideres terminada la tarea asignada.

3. Finalmente, después de cada interacción debes responder una breve encuesta sobre la experiencia realizada. Mientras contestas la encuesta tendrás acceso al menú desplegable en el panel izquierdo, en el que además podrás revisar toda la conversación que hayas tenido con el asistente.

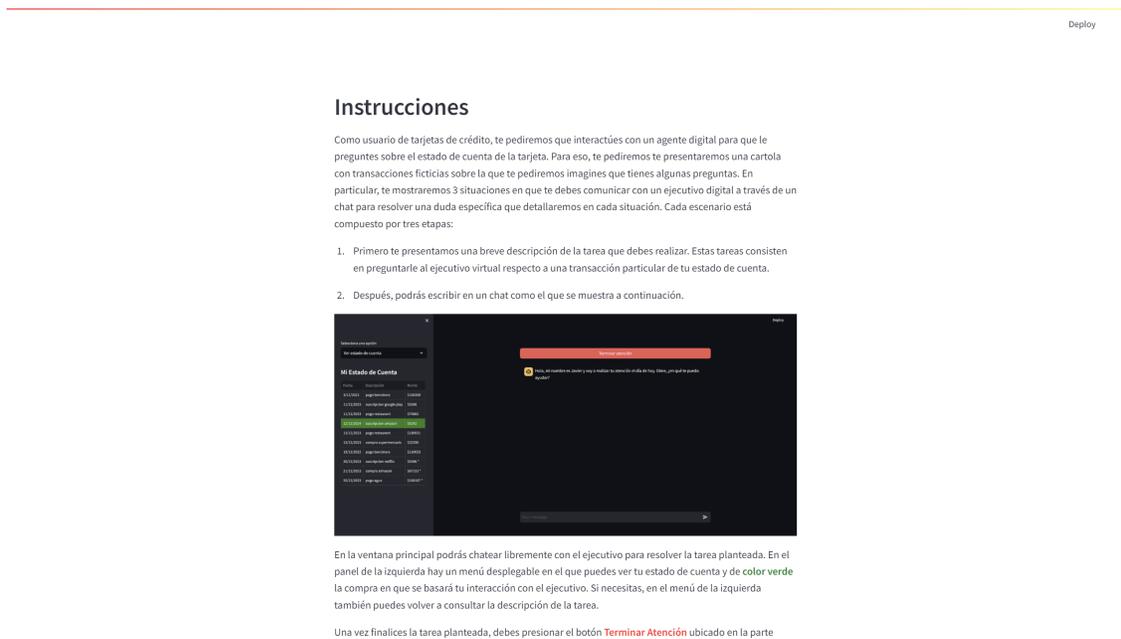


Figura G.3: Pantalla de instrucciones de la plataforma experimental, en su versión clara. Fuente: Elaboración propia.

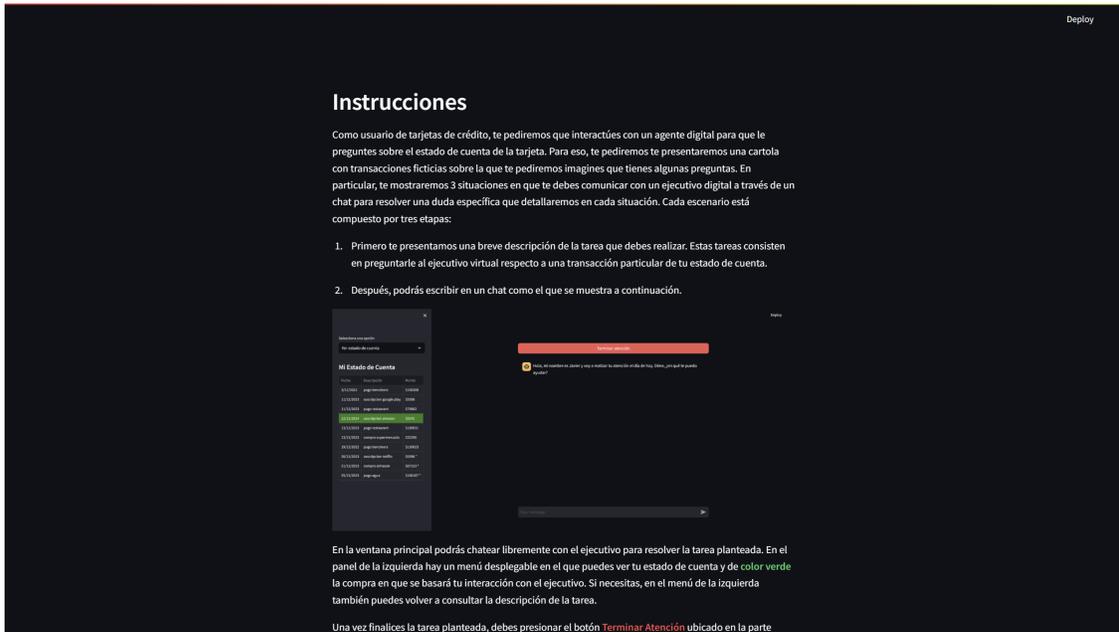


Figura G.4: Pantalla de instrucciones de la plataforma experimental, en su versión oscura. Fuente: Elaboración propia.

G.3. Interfaz de descripción de escenario

La interfaz de descripción de escenario corresponde a aquella que el usuario observa antes de iniciar una interacción vía chat con el agente. En esta, se describe la situación del escenario experimental al cual el participante se debe enfrentar, con los lineamientos bajo los cuales debe construir su relato de cliente en la atención. Además, se entrega un resumen de las instrucciones generales del experimento, las cuales son idénticas para todos los escenarios. A continuación, se presentan las descripciones de cada escenario experimental, las instrucciones generales que el participante debe seguir, además de imágenes de las pantallas de la interfaz en sus versiones clara y oscura.

Descripción de escenario 1: *Desconocimiento de transacción con reconocimiento*

Escenario 1: Desconocimiento de transacción con reconocimiento

Situación: Acabas de revisar tu estado de cuenta, pero te das cuenta de que tienes una transacción que no recuerdas haber realizado. Por favor, pregúntale al asistente respecto a esta transacción, la que estará marcada en el estado de cuenta en color verde. El asistente debiera reconocer la transacción en los registros del banco y sugerirte algunas ideas de por qué no reconoces esa compra.

En esta situación, debes suponer que luego de interactuar con el asistente, *logras identificar a qué corresponde la transacción y por tanto no necesitas asistencia adicional*, por lo que puedes terminar la interacción con el asistente virtual.

Descripción de escenario 2: *Desconocimiento de transacción*

Escenario 2: Desconocimiento de transacción

Situación: Acabas de revisar tu estado de cuenta, pero te das cuenta de que tienes una transacción que no recuerdas haber realizado. Tu tarea es preguntarle al asistente respecto a esta transacción, la que estará marcada en el estado de cuenta en color verde, quien debiera reconocer la transacción en los registros del banco y sugerirte algunas ideas de por qué no reconoces esa compra.

En esta situación, *debes suponer que luego de interactuar con el asistente, NO logras identificar a qué corresponde la transacción y por tanto necesitas que el asistente virtual te indique qué debes hacer ante esta desafortunada situación.* Puedes terminar la interacción una vez que el asistente virtual te indique qué hacer.

Descripción de escenario 3: *Consulta de Cupo de la Tarjeta*

Escenario 3: Consulta de Cupo de la Tarjeta

Situación: Al revisar tu estado de cuenta te preguntas qué cupo te queda disponible en la tarjeta. Tu tarea consiste en preguntarle al asistente digital respecto si con lo que llevas gastado, sería posible hacer una nueva compra por \$50,000 con la tarjeta. Como el asistente solo está calificado para orientarte respecto al desconocimiento de transacciones, puedes terminar la conversación cuándo te indique que no puede darte esta respuesta.

Resumen de instrucciones generales

Resumen de las instrucciones:

- Interactúa con el asistente como lo harías con el asistente de tu banco.
- Recuerda que en el menú de la izquierda puedes volver a leer las instrucciones y mirar tu estado de cuenta.
- Una vez terminada la atención, ya sea porque se llegó a una solución o porque sientes que el ejecutivo no te está ayudando, presiona el botón Terminar Atención.
- Recuerda seguir las instrucciones, de lo contrario, tu participación quedará inválida.

Para comenzar, presiona Comenzar.

Escenario 1: Desconocimiento de transacción con reconocimiento

Situación: Acabas de revisar tu estado de cuenta, pero te das cuenta de que tienes una transacción que no recuerdas haber realizado. Por favor, pregúntale al asistente respecto a esta transacción, la que estará marcada en el estado de cuenta en color verde. El asistente debiera reconocer la transacción en los registros del banco y sugerirte algunas ideas de por qué no reconoces esa compra.

En esta situación, debes suponer que luego de interactuar con el asistente, *logras identificar a qué corresponde la transacción y por tanto no necesitas asistencia adicional*, por lo que puedes terminar la interacción con el asistente virtual.

Resumen de las instrucciones:

- Interactúa con el asistente como lo harías con el asistente de tu banco.
- Recuerda que en el menú de la izquierda puedes volver a leer las instrucciones y mirar tu estado de cuenta.
- Una vez terminada la atención, ya sea porque se llegó a una solución o porque sientes que el ejecutivo no te está ayudando, presiona el botón **Terminar Atención**.

Recuerda seguir las instrucciones, de lo contrario, tu participación quedará invalidada.

Para comenzar, presiona **Comenzar**.

Comenzar

Figura G.5: Pantalla de descripción de escenario, en su versión clara. Fuente: Elaboración propia.

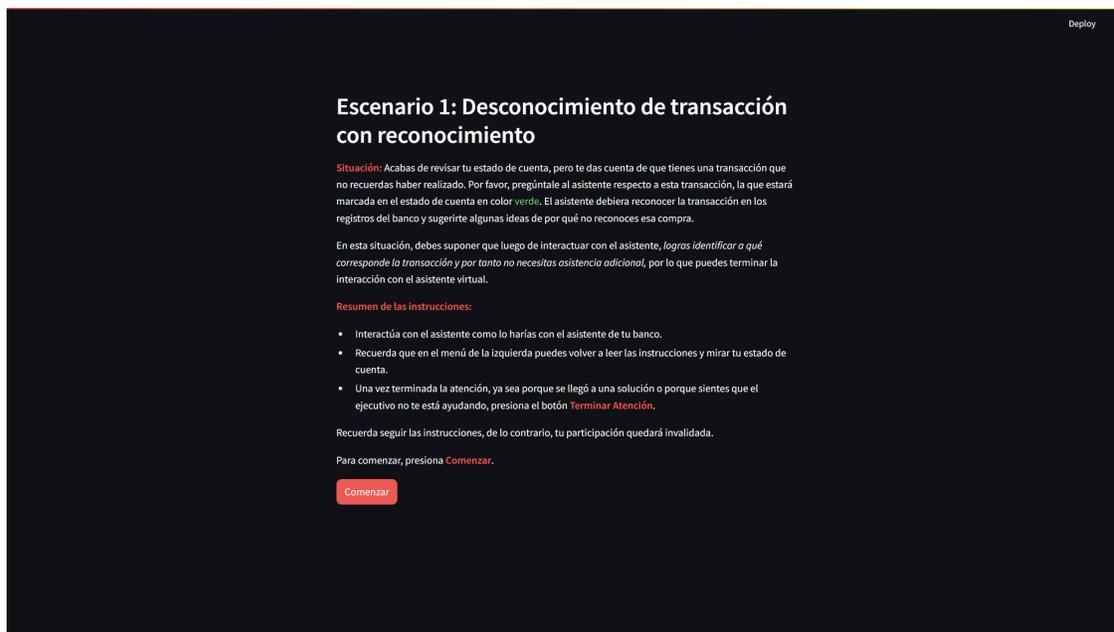


Figura G.6: Pantalla de descripción de escenario, en su versión oscura. Fuente: Elaboración propia.

G.4. Interfaz de chat

La interfaz de chat es aquella donde ocurre la interacción del participante con el agente virtual. Cuenta con un chat para la comunicación con el agente, además de un menú lateral desplegable donde el usuario puede consultar su estado de cuenta ficticio, donde se destaca en color verde la transacción sobre la cual debe basar su relato, además de la descripción del

escenario experimental y las instrucciones generales que debe seguir. Por último, el usuario cuenta con un botón a través del cual puede terminar la atención una vez haya terminado con las situación descrita. A continuación, se presentan las pantallas del chat, en ambas variaciones de color y con ambas opciones de visualización del menú lateral.

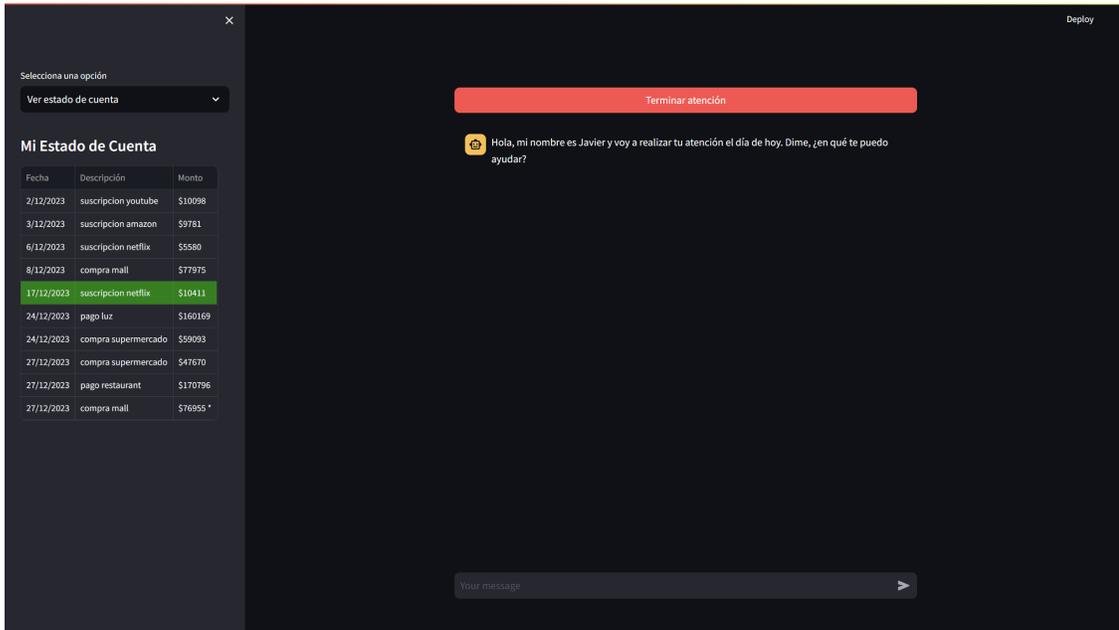


Figura G.7: Pantalla de chat en su versión oscura, con el menú lateral mostrando el estado de cuenta del participante. Fuente: Elaboración propia.

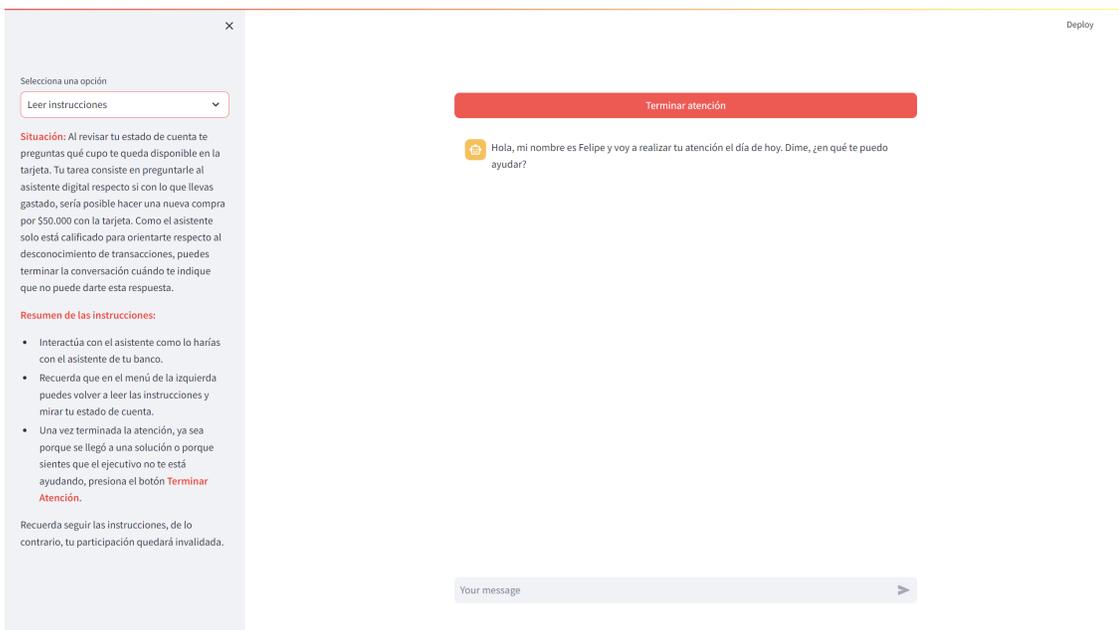


Figura G.8: Pantalla de chat en su versión clara, con el menú lateral mostrando la descripción del escenario. Fuente: Elaboración propia.

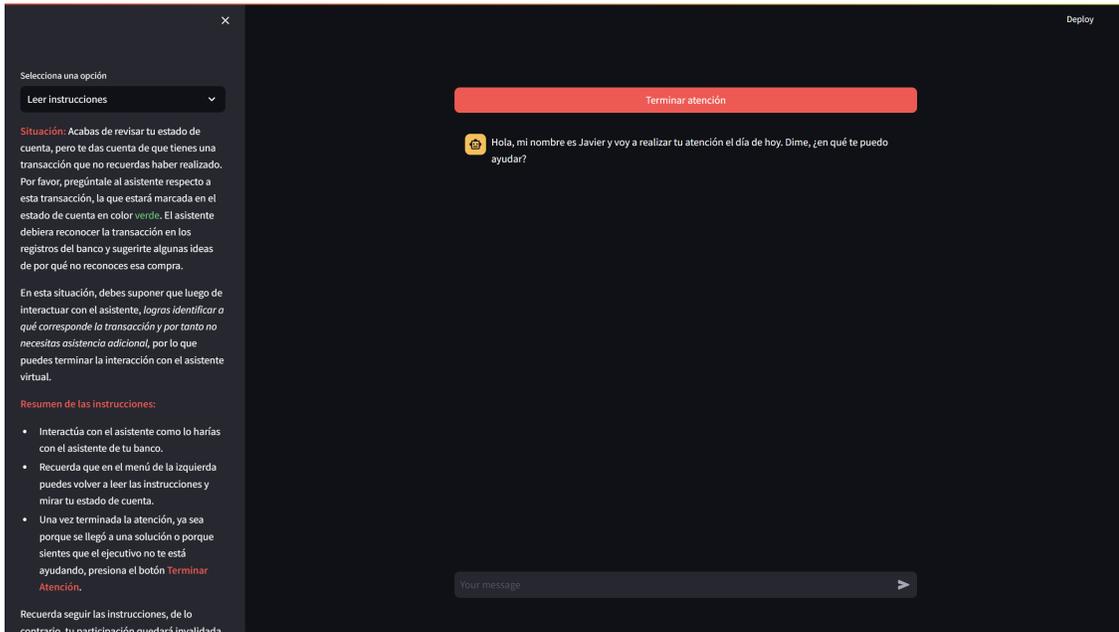


Figura G.9: Pantalla de chat en su versión oscura, con el menú lateral mostrando la descripción del escenario. Fuente: Elaboración propia.

G.5. Interfaz de encuesta de situación

Luego de finalizada la interacción con el agente, el usuario pasa a una interfaz de encuesta de la situación que acaba de simular. Para ello, se presentan cinco preguntas que el participante debe responder en una escala Likert de 1 a 5. La encuesta se completa utilizando selectores para determinar la evaluación asignada a cada pregunta, además de validadores que inhiben al usuario de avanzar a la siguiente situación si no se han contestado todas estas. Además, para apoyar sus respuestas, se puede consultar la información del menú lateral, ahora incorporando la conversación que acaba de tener con el agente virtual.

A continuación, se presenta un ejemplo del contenido de la encuesta para una situación, además de imágenes de las pantallas en sus variaciones clara y oscura.

Ejemplo de encuesta sobre situación N

Encuesta sobre la Situación N

Por favor, responde las siguientes preguntas sobre la tarea que acabas de realizar. Recuerda que en el menú desplegable puedes consultar tu estado de cuenta, las instrucciones de la tarea y la conversación realizada.

Preguntas

En base a la atención que acabas de recibir, por favor contesta tu grado de acuerdo respecto a las siguientes afirmaciones (donde 1 es a “Muy en desacuerdo” y 5 es “Muy de acuerdo”).

- El agente comprendió adecuadamente las preguntas que le hice

- Las respuestas del agente eran fáciles de entender
- El agente pudo resolver adecuadamente las preguntas que le hice sin necesidad de derivarme a otras instancias del banco
- En todo momento sentí que el agente tenía información útil y confiable respecto a las preguntas realizadas
- En general, estoy satisfecho con la atención del agente virtual

Deploy

Selección una opción

Ver conversación

Hola, mi nombre es Felipe y voy a realizar tu atención el día de hoy. Dime, ¿en qué te puedo ayudar?

Hola Felipe!

¿Cómo te puedo ayudar?

Desconozco una transacción

¿Podrías indicarme la fecha, comercio y monto de la transacción que desconoces?

Claro, la descripción es suscripción amazon y fue el 24 de diciembre

He identificado una transacción con el siguiente detalle: 24/12/2023 suscripción amazon \$4918. ¿Esta es la transacción que desconoces?

Encuesta sobre la Situación 1

Por favor, responde las siguientes preguntas sobre la tarea que acabas de realizar. Recuerda que en el menú desplegable puedes consultar tu estado de cuenta, las instrucciones de la tarea y la conversación realizada.

Preguntas

En base a la atención que acabas de recibir, por favor contesta tu grado de acuerdo respecto a las siguientes afirmaciones (donde 1 es a "Muy en desacuerdo" y 5 es "Muy de acuerdo").

El agente comprendió adecuadamente las preguntas que le hice

1 2 3 4 5

Las respuestas del agente eran fáciles de entender

1 2 3 4 5

El agente pudo resolver adecuadamente las preguntas que le hice sin necesidad de derivarme a otras instancias del banco

1 2 3 4 5

En todo momento sentí que el agente tenía información útil y confiable respecto a las preguntas realizadas

1 2 3 4 5

En general, estoy satisfecho con la atención del agente virtual

1 2 3 4 5

Siguiente

Figura G.10: Pantalla de encuesta de situación, en su versión clara. El menú lateral muestra la conversación que tuvo el participante con el agente virtual. Fuente: Elaboración propia.

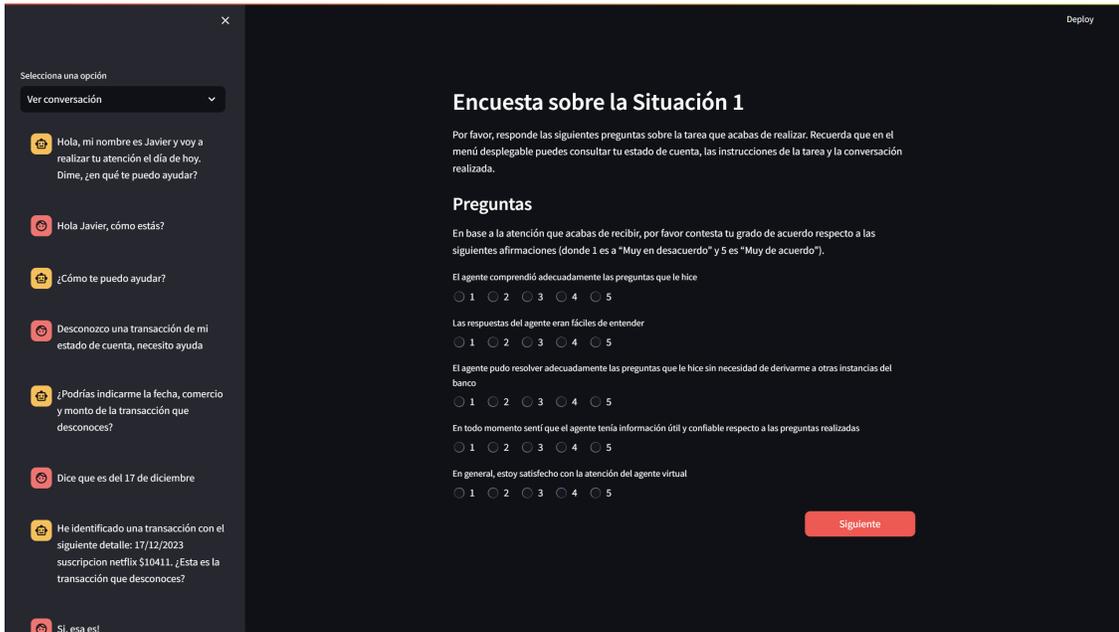


Figura G.11: Pantalla de encuesta de situación, en su versión oscura. El menú lateral muestra la conversación que tuvo el participante con el agente virtual. Fuente: Elaboración propia.

G.6. Interfaz de encuesta de final

Finalmente, una vez concluidos los tres escenarios experimentales y sus respectivas encuestas, el usuario se enfrenta a una encuesta final para capturar sus apreciaciones globales de las interacciones y comentarios sobre el estudio en el que ha participado. A continuación, se describe el contenido de esta, además de presentar las pantallas de la interfaz en sus versiones clara y oscura.

Contenido de encuesta final

Encuesta final

Para terminar, nos gustaría tener una evaluación global de la interacción que tuviste con el agente virtual, en base a los tres escenarios anteriores. Para eso, por favor contesta tu grado de acuerdo respecto a las siguientes afirmaciones (donde 1 es a “Muy en desacuerdo” y 5 es “Muy de acuerdo”).

- Se aprecia en el menú lateral la conversación que tuvo el participante con el agente virtual. Las interacciones me parecieron cálidas y cordiales como las que experimentaría con mi ejecutivo en una sucursal
- Creo que podría resolver la gran mayoría de mis preguntas respecto a la tarjeta con agentes virtuales como estos.
- Por último, en una escala del 0 al 10, ¿qué tan probable es que recomiendes el servicio de atención remota que acabas de experimentar a un familiar o amigo?

Comentarios

Por favor, cuéntanos si tienes un comentario respecto al estudio. Si no tienes comentarios, puedes Finalizar.

Deploy

Encuesta final

Para terminar, nos gustaría tener una evaluación global de la interacción que tuviste con el agente virtual, en base a los tres escenarios anteriores. Para eso, por favor contesta tu grado de acuerdo respecto a las siguientes afirmaciones (donde 1 es a "Muy en desacuerdo" y 5 es "Muy de acuerdo").

Las interacciones me parecieron cálidas y cordiales como las que experimentaría con mi ejecutivo en una sucursal

1 2 3 4 5

Creo que podría resolver la gran mayoría de mis preguntas respecto a la tarjeta con agentes virtuales como estos.

1 2 3 4 5

Por último, en una escala del 0 al 10, ¿qué tan probable es que recomiendes el servicio de atención remota que acabas de experimentar a un familiar o amigo?

0 1 2 3 4 5 6 7 8 9 10

Comentarios

Por favor, cuéntanos si tienes un comentario respecto al estudio. Si no tienes comentarios, puedes [Finalizar](#).

[Finalizar](#)

Figura G.12: Pantalla de encuesta final, en su versión clara. Fuente: Elaboración propia.

Deploy

Encuesta final

Para terminar, nos gustaría tener una evaluación global de la interacción que tuviste con el agente virtual, en base a los tres escenarios anteriores. Para eso, por favor contesta tu grado de acuerdo respecto a las siguientes afirmaciones (donde 1 es a "Muy en desacuerdo" y 5 es "Muy de acuerdo").

Las interacciones me parecieron cálidas y cordiales como las que experimentaría con mi ejecutivo en una sucursal

1 2 3 4 5

Creo que podría resolver la gran mayoría de mis preguntas respecto a la tarjeta con agentes virtuales como estos.

1 2 3 4 5

Por último, en una escala del 0 al 10, ¿qué tan probable es que recomiendes el servicio de atención remota que acabas de experimentar a un familiar o amigo?

0 1 2 3 4 5 6 7 8 9 10

Comentarios

Por favor, cuéntanos si tienes un comentario respecto al estudio. Si no tienes comentarios, puedes [Finalizar](#).

[Finalizar](#)

Figura G.13: Pantalla de encuesta final, en su versión oscura. Fuente: Elaboración propia.

Anexo H. Resultados complementarios

H.1. Esquema de bases generadas con datos obtenidos

Tabla H.1: Panel generado con datos por interacción. Fuente: Elaboración propia.

Campo	Tipo	Descripción	Ejemplo
customer_id	String	Identificador del participante	0066c...
task_id	Float	Identificador del escenario	task1
prompt_type	Float	Tipo de prompt en la interacción	E-Res
age	Integer	Edad del participante	44
sex	String	Sexo del participante	M
nse	String	Nivel socioeconómico del participante	C1a
resolutivity	Integer	Resolutividad percibida por el participante	4
hallucination	Integer	Confianza percibida por el participante	3
experience	Integer	Satisfacción percibida por el participante	5
efficacy_input	Integer	Eficacia de comprensión de mensajes percibida por el participante	2
efficacy_output	Integer	Eficacia de generación de respuestas percibida por el cliente	3
interaction_time	Float	Duración de la interacción en minutos	4.38
time_bet_msgs	Float	Tiempo promedio entre mensajes en minutos	0.15
n_messages	Integer	Número de mensajes en la interacción	11
n_words	Float	Número de palabras en la interacción	440

Tabla H.2: Base con datos globales por experimento. Fuente: Elaboración propia.

Campo	Tipo	Descripción	Ejemplo
customer_id	String	Identificador del participante	01442...
prompt_type	Float	Tipo de prompt en el experimento	S
age	Integer	Edad del participante	18
sex	String	Sexo del participante	H
nse	String	Nivel socioeconómico del participante	D
resolutivity_global	Integer	Resolutividad global percibida por el participante	4
experience_global	Integer	Satisfacción global por el participante	4
promoter_score	Integer	Promoter score del participante	8
interaction_time	Float	Duración promedio por interacción en minutos	4.63
time_bet_msgs	Float	Tiempo promedio entre mensajes en minutos	0.39
n_messages	Integer	Número de mensajes promedio por interacción	11.7
n_words	Float	Número de palabras promedio por la interacción	406.3

H.2. Relaciones entre variables

Tabla H.3: Medias por métricas objetivo segmentadas por tipo de prompt.
Fuente: Elaboración propia.

Métrica	S	E	E-Res
resolutivity	3.41	3.18	3.38
hallucination	3.61	3.54	3.52
experience	3.67	3.52	3.53
efficacy_input	4.12	4.08	3.89
efficacy_output	4.41	4.34	4.27

Tabla H.4: Medias por métricas objetivo segmentadas por tipo de escenario.
Fuente: Elaboración propia.

Métrica	E1	E2	E3
resolutivity	3.87	3.55	2.55
hallucination	3.95	3.74	2.98
experience	3.97	3.68	3.07
efficacy_input	4.27	4.11	3.71
efficacy_output	4.52	4.37	4.13

Tabla H.5: Medias por métricas objetivo segmentadas por sexo del participante. Fuente: Elaboración propia.

Métrica	H	M
resolutivity	3.28	3.37
hallucination	3.51	3.61
experience	3.50	3.65
efficacy_input	3.98	4.08
efficacy_output	4.30	4.38

Tabla H.6: Medias por métricas objetivo segmentadas por nivel socioeconómico del participante. Fuente: Elaboración propia.

Métrica	AB	C1a	C1b	C2	C3	D
resolutivity	3.28	3.10	3.53	3.29	3.33	3.36
hallucination	3.59	3.49	3.70	3.46	3.56	3.60
experience	3.69	3.45	3.71	3.49	3.59	3.62
efficacy_input	4.08	4.07	4.12	4.02	4.00	3.94
efficacy_output	4.59	4.39	4.34	4.36	4.33	4.24

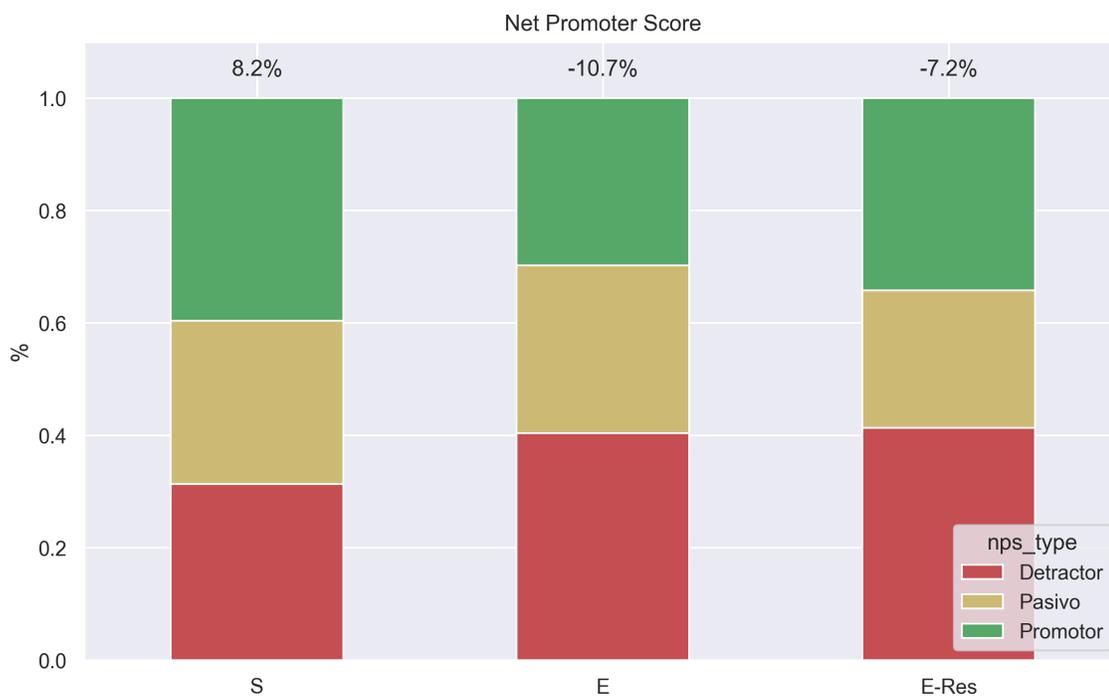


Figura H.1: Net Promoter Score calculado según tipo de agente. Fuente: Elaboración propia.

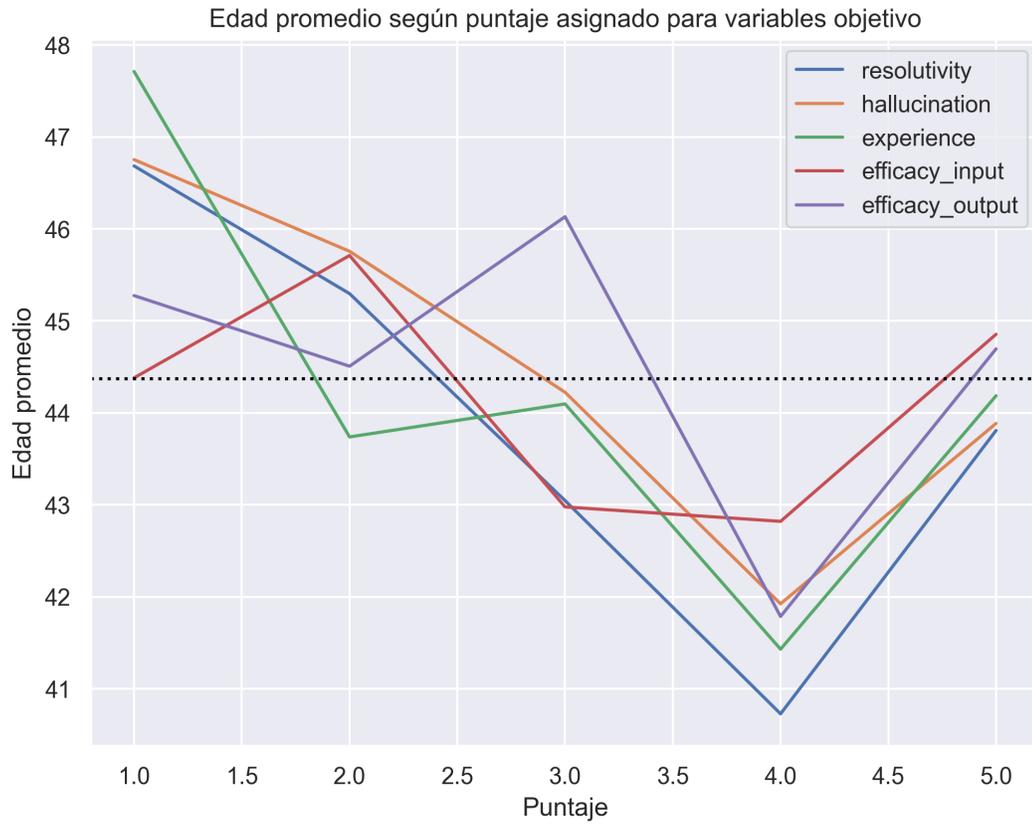


Figura H.2: Promedio de edad de participante según puntaje asignado a métricas objetivo. La línea punteada indica el promedio de la muestra experimental. Fuente: Elaboración propia.

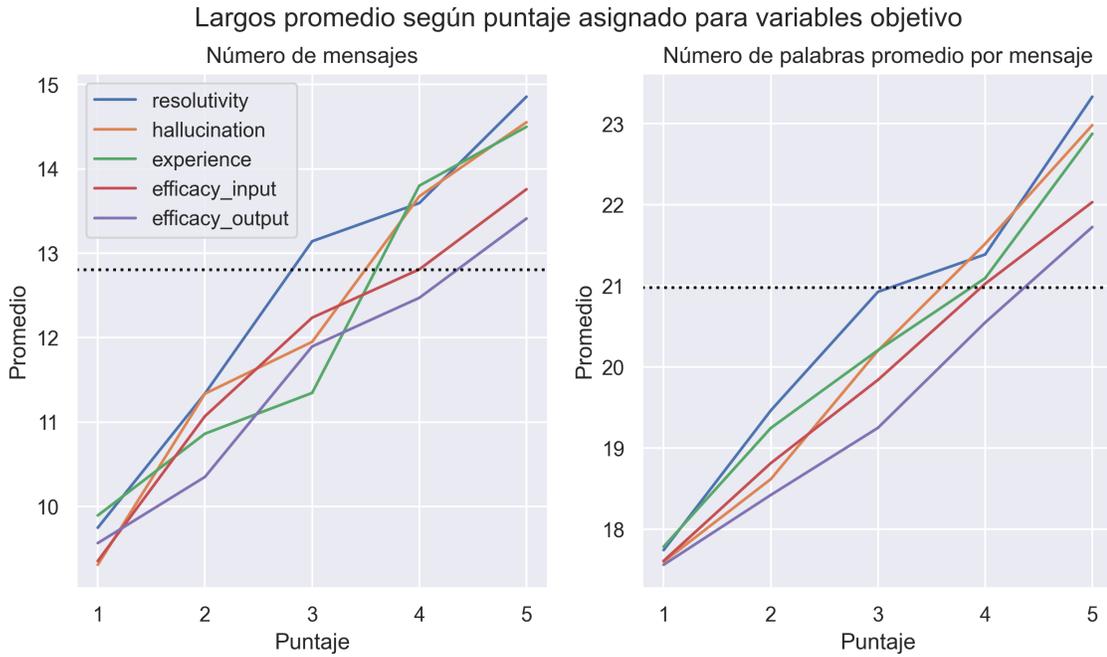


Figura H.3: Promedio de variables de largo de interacción según puntaje asignado a métricas objetivo. La línea punteada indica el promedio de la muestra experimental. Fuente: Elaboración propia.

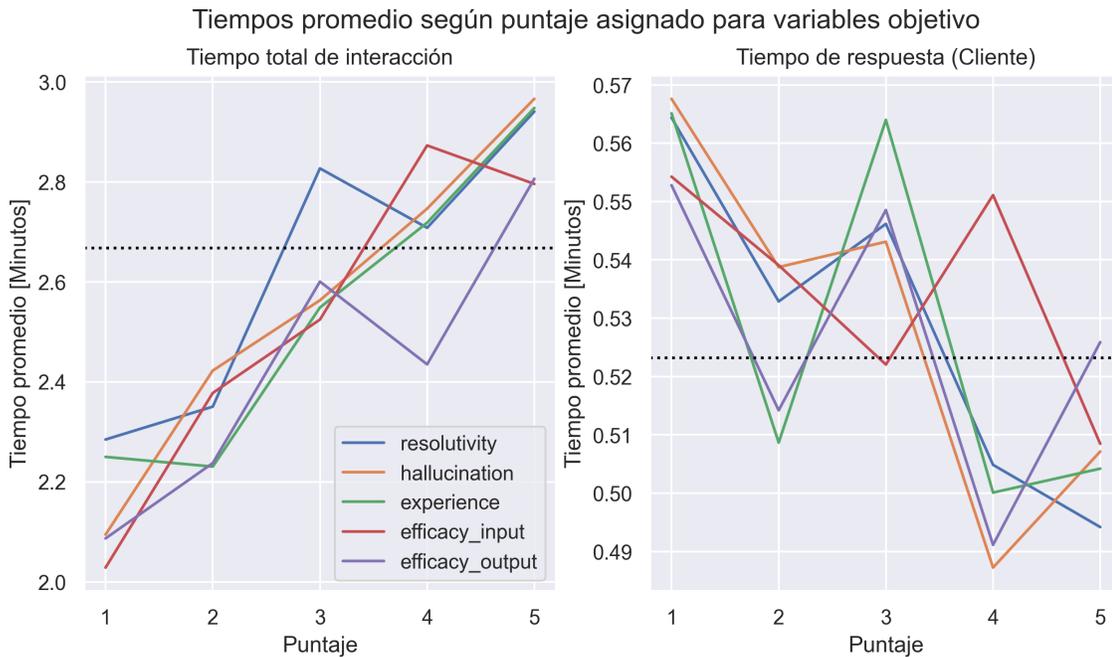


Figura H.4: Promedio de variables de duración de interacción según puntaje asignado a métricas objetivo. La línea punteada indica el promedio de la muestra experimental. Fuente: Elaboración propia.

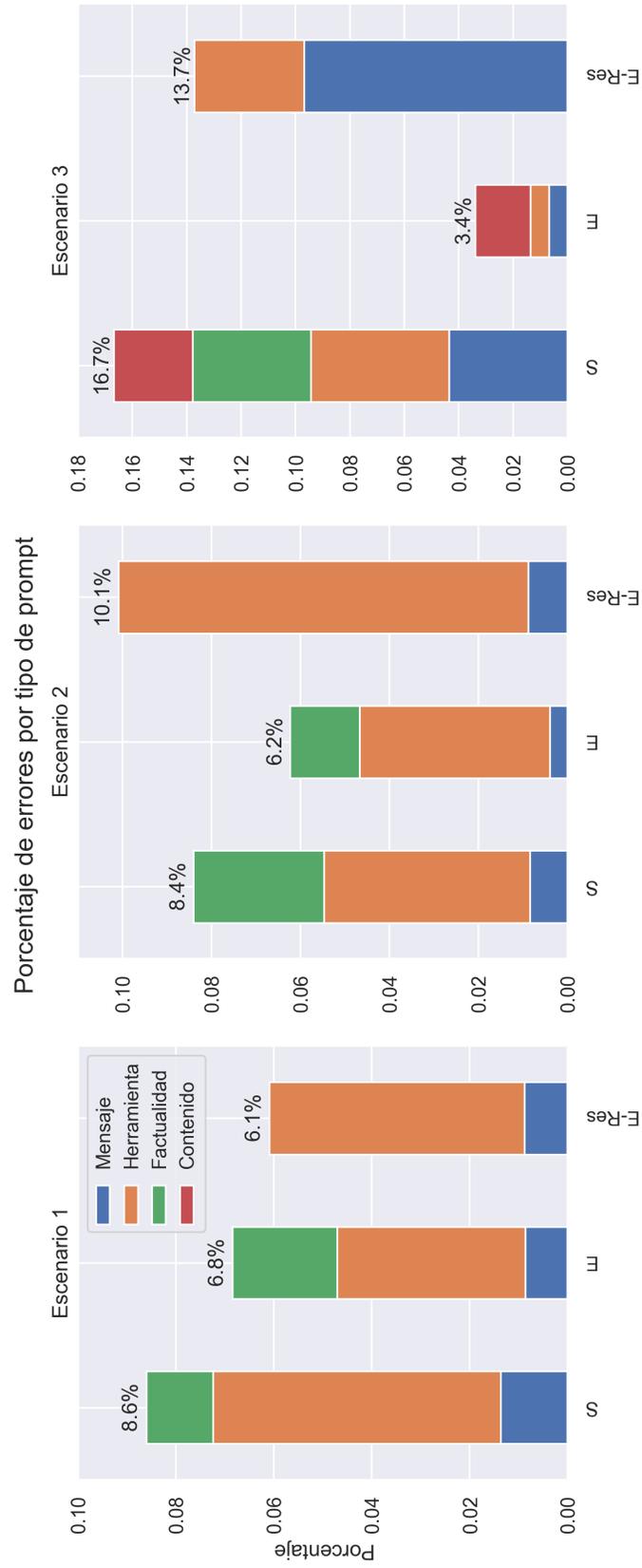


Figura H.5: Promedio de variables de duración de interacción según puntaje asignado a métricas objetivo. La línea punteada indica el promedio de la muestra experimental. Fuente: Elaboración propia.

H.3. Test de Shapiro-Wilk sobre métricas objetivo

Tabla H.7: Resultados obtenidos con test de Shapiro-Wilk para métricas por interacción según prompt. Los p-valores indican que se rechaza la hipótesis nula en todos los casos, luego los grupos no pertenecen a una distribución normal. Fuente: Elaboración propia.

Prompt	resolutivity	hallucination	experience	efficacy_input	efficacy_output
S	2.93e-29	8.48e-28	3.54e-28	2.29e-33	5.56e-36
E	8.62e-30	5.43e-28	1.24e-27	6.29e-33	9.52e-35
E-Res	3.18e-30	4.63e-28	1.58e-28	2.49e-31	1.61e-34

Tabla H.8: Resultados obtenidos con test de Shapiro-Wilk para métricas globales según prompt. Los p-valores indican que se rechaza la hipótesis nula en todos los casos, luego los grupos no pertenecen a una distribución normal. Fuente: Elaboración propia.

Prompt	resolutivity_global	experience_global	promoter_score
S	3.04e-14	1.34e-18	3.59e-13
E	5.34e-13	1.20e-16	1.78e-11
E-Res	8.01e-14	5.88e-16	9.80e-12

H.4. Tests de Levene y Bartlett

Tabla H.9: p-valores obtenidos con tests de Levene y Bartlett para métricas objetivo. Fuente: Elaboración propia.

Variable	Levene	Bartlett
resolutivity	0.002378	0.301647
hallucination	0.049444	0.297582
experience	0.005739	0.091222
efficacy_input	0.009706	0.008189
efficacy_output	0.103456	0.000567
resolutivity_global	0.001219	0.037815
experience_global	0.003132	0.005336
nps	0.010695	0.071106

H.5. Tests T y U de Mann-Whitney para diferencias entre medias

Tabla H.10: p-valores obtenidos para comparaciones entre tipos de prompts con tests T y U de Mann-Whitney para métricas objetivo. Fuente: Elaboración propia.

Variable	Comparación	T-test	Mann-Whitney U
resolutivity	(S, E)	0.016055	0.042147
	(S, E-Res)	0.785991	0.974007
	(E, E-Res)	0.039661	0.063522
hallucination	(S, E)	0.416973	0.659308
	(S, E-Res)	0.321417	0.584194
	(E, E-Res)	0.849477	0.883078
experience	(S, E)	0.073599	0.166014
	(S, E-Res)	0.113554	0.320752
	(E, E-Res)	0.873408	0.806349
efficacy_input	(S, E)	0.603152	0.848033
	(S, E-Res)	0.00459	0.014812
	(E, E-Res)	0.021691	0.026271
efficacy_output	(S, E)	0.254013	0.309583
	(S, E-Res)	0.034633	0.14684
	(E, E-Res)	0.314483	0.638502
resolutivity_global	(S, E)	0.049119	0.095363
	(S, E-Res)	0.101336	0.281908
	(E, E-Res)	0.800677	0.651666
experience_global	(S, E)	0.063001	0.10935
	(S, E-Res)	0.00466	0.023043
	(E, E-Res)	0.306599	0.458436
promoter_score	(S, E)	0.027486	0.028678
	(S, E-Res)	0.019273	0.078096
	(E, E-Res)	0.816756	0.84432

H.6. Regresiones a nivel de interacción

Tabla H.11: Resultados de modelos de regresión para la variable de Satisfacción. Se ha incorporado una fila con los p-valores para las diferencias entre los efectos de los prompts E y E-Res en cada modelo. Fuente: Elaboración propia.

	<i>Variable dependiente: Satisfacción</i>			
	(1)	(2)	(3)	(4)
Intercepto	3.670*** (0.061)	-0.111 (0.068)	-0.001 (0.125)	0.025 (0.133)
prompt.E	-0.149* (0.086)	-0.087** (0.039)	-0.079** (0.039)	-0.079** (0.039)
prompt.E-Res	-0.142 (0.087)	-0.022 (0.040)	-0.008 (0.040)	-0.005 (0.040)
efficacy_input		0.171*** (0.018)	0.174*** (0.018)	0.174*** (0.018)
efficacy_output		0.099*** (0.019)	0.101*** (0.019)	0.100*** (0.019)
hallucination		0.731*** (0.014)	0.720*** (0.015)	0.720*** (0.015)
interaction_time			-0.005 (0.007)	-0.006 (0.007)
n_messages			0.006** (0.003)	0.007** (0.003)
age			-0.002 (0.001)	-0.002 (0.001)
sex.M			0.026 (0.034)	0.027 (0.034)
Efectos Fijos	No	No	No	Si
p-value [E vs. E-Res]	0.943	0.104	0.075	0.066
Observations	1929	1929	1929	1929
R^2	0.002	0.796	0.797	0.798
R^2 ajustado	0.001	0.795	0.796	0.796

Nota:

*p<0.1; **p<0.05; ***p<0.01

Tabla H.12: Resultados de modelos de regresión para la variable de Resolutividad. Se ha incorporado una fila con los p-valores para las diferencias entre los efectos de los prompts E y E-Res en cada modelo. Fuente: Elaboración propia.

	<i>Variable dependiente: Resolutividad</i>			
	(1)	(2)	(3)	(4)
Intercepto	3.406*** (0.068)	-0.091 (0.093)	-0.234 (0.169)	0.146 (0.178)
prompt.E	-0.226** (0.096)	-0.167*** (0.054)	-0.153*** (0.053)	-0.157*** (0.052)
prompt.E-Res	-0.031 (0.097)	0.075 (0.054)	0.131** (0.054)	0.109** (0.054)
efficacy_input		0.151*** (0.024)	0.151*** (0.024)	0.155*** (0.024)
efficacy_output		-0.037 (0.027)	-0.028 (0.026)	-0.030 (0.026)
hallucination		0.841*** (0.020)	0.801*** (0.020)	0.786*** (0.020)
interaction_time			-0.019** (0.009)	-0.024*** (0.009)
n_messages			0.030*** (0.004)	0.020*** (0.004)
age			-0.003** (0.002)	-0.003** (0.002)
sex.M			-0.032 (0.046)	-0.038 (0.046)
Efectos Fijos	No	No	No	Si
p-value [E vs. E-Res]	0.046	0.000	0.000	0.000
Observations	1929	1929	1929	1929
R^2	0.003	0.691	0.703	0.709
R^2 ajustado	0.002	0.690	0.701	0.707

Nota:

*p<0.1; **p<0.05; ***p<0.01

Tabla H.13: Resultados de modelos de regresión para la variable de Correc-
titud. Se ha incorporado una fila con los p-valores para las diferencias entre
los efectos de los prompts E y E-Res en cada modelo. Fuente: Elaboración
propia.

	<i>Variable dependiente: Error (%)</i>			
	(1)	(2)	(3)	(4)
Intercepto	10.780*** (1.169)	9.470*** (3.239)	10.005* (5.157)	4.675 (5.331)
prompt.E	-5.615*** (1.653)	-5.354*** (1.666)	-4.706** (1.993)	-4.965** (1.933)
prompt.E-Res	-0.715 (1.653)	-0.421 (1.670)	-0.505 (1.978)	-0.262 (1.918)
efficacy_input		-0.172 (0.778)	-0.377 (0.835)	-0.541 (0.813)
efficacy_output		0.939 (0.800)	1.245 (0.864)	1.400* (0.839)
hallucination		-0.655 (0.669)	-0.702 (0.801)	-0.508 (0.779)
interaction_time			-0.463 (0.483)	-0.094 (0.495)
n_messages			0.045 (0.156)	0.224 (0.170)
age			-0.034 (0.050)	-0.044 (0.049)
sex.M			-0.250 (1.667)	0.030 (1.618)
Efectos Fijos	No	No	No	Si
p-value [E vs. E-Res]	0.004	0.004	0.036	0.016
Observations	126	126	126	126
R^2	0.100	0.118	0.154	0.220
R^2 ajustado	0.085	0.081	0.047	0.105

Nota:

*p<0.1; **p<0.05; ***p<0.01

H.7. Regresiones a nivel global

Tabla H.14: Resultados de modelos de regresión para la variable de Satisfacción global. Se ha incorporado una fila con los p-valores para las diferencias entre los efectos de los prompts E y E-Res en cada modelo. Fuente: Elaboración propia.

	<i>Variable dependiente: Satisfacción global</i>		
	(1)	(2)	(3)
Intercepto	4.277*** (0.075)	1.313*** (0.192)	1.311*** (0.316)
prompt.E	-0.198* (0.107)	-0.153* (0.085)	-0.140 (0.085)
prompt.E-Res	-0.316*** (0.108)	-0.212** (0.086)	-0.234*** (0.088)
efficacy_input		0.180*** (0.061)	0.186*** (0.061)
efficacy_output		0.192*** (0.062)	0.193*** (0.063)
hallucination		0.381*** (0.045)	0.381*** (0.046)
interaction_time			-0.016 (0.024)
n_messages			-0.006 (0.010)
age			-0.001 (0.003)
sex.M			0.045 (0.075)
p-value [E vs. E-Res]	0.279	0.493	0.291
Observations	643	643	643
R^2	0.014	0.379	0.391
R^2 ajustado	0.010	0.374	0.378

Nota:

*p<0.1; **p<0.05; ***p<0.01

Tabla H.15: Resultados de modelos de regresión para la variable de Resolutividad global. Se ha incorporado una fila con los p-valores para las diferencias entre los efectos de los prompts E y E-Res en cada modelo. Fuente: Elaboración propia.

	<i>Variable dependiente: Resolutividad global</i>		
	(1)	(2)	(3)
Intercepto	3.805*** (0.087)	0.330* (0.197)	0.251 (0.326)
prompt.E	-0.246** (0.124)	-0.189** (0.087)	-0.184** (0.088)
prompt.E-Res	-0.213* (0.125)	-0.104 (0.089)	-0.100 (0.091)
efficacy_input		0.159** (0.062)	0.166*** (0.063)
efficacy_output		0.062 (0.064)	0.066 (0.065)
hallucination		0.706*** (0.046)	0.699*** (0.048)
interaction_time			-0.009 (0.024)
n_messages			0.006 (0.010)
age			-0.001 (0.003)
sex.M			-0.090 (0.077)
p-value [E vs. E-Res]	0.792	0.336	0.365
Observations	643	643	643
R^2	0.007	0.513	0.517
R^2 ajustado	0.004	0.509	0.507

Nota:

*p<0.1; **p<0.05; ***p<0.01

Tabla H.16: Resultados de modelos de regresión para la variable de Correc-titud global. Se ha incorporado una fila con los p-valores para las diferencias entre los efectos de los prompts E y E-Res en cada modelo. Fuente: Elabo-ración propia.

	<i>Variable dependiente: Error global (%)</i>		
	(1)	(2)	(3)
Intercepto	10.743*** (1.096)	11.268*** (4.096)	8.689 (6.266)
prompt.E	-4.933*** (1.550)	-4.586*** (1.570)	-4.592** (1.968)
prompt.E-Res	-1.425 (1.550)	-1.113 (1.576)	-1.579 (1.960)
efficacy_input		-0.460 (1.290)	-1.670 (1.763)
efficacy_output		0.962 (1.146)	2.237 (1.462)
hallucination		-0.876 (0.923)	-0.572 (1.348)
interaction_time			-0.112 (0.714)
n_messages			-0.253 (0.273)
age			-0.017 (0.053)
sex.M			0.444 (1.759)
p-value [E vs. E-Res]	0.029	0.032	0.142
Observations	42	42	42
R^2	0.216	0.276	0.384
R^2 ajustado	0.176	0.175	0.065

Nota:

*p<0.1; **p<0.05; ***p<0.01