Departamento de Ingeniería Industrial
Universidad de Concepción

UNIVESIDAD DE CONCEPCIÓN
FACULTAD DE INGENERÍA
DEPARTAMENTE DE INGENERÍA

Automatización de proceso de cuadraturas en empresa de seguros

Por

Giovanni Francesco Borotto Cerda

Memoria de Título presentada a la Facultad de Ingeniería de la Universidad de Concepción para optar al título de profesional de Ingeniero Civil Industrial

Profesor Guía

Carlos Camilo Navarrete Lizama

Julio 2025

Concepción (Chile)

Abstract

En este informe se presenta una propuesta de automatización de cuadratura y generación de archivos de carga para una empresa financiera con el objetivo de mejorar la precisión y eficiencia de dicho proceso dentro de la industria. Como metodología se usó un sistema de múltiples agentes (MAS), dichos agentes toman decisiones autónomas mediante la integración de un modelo de lenguaje natural (LLM). Se comparó la misma metodología con 4 modelos de inteligencia artificial (IA): deepseek-v3, qwen-plus, deepseek-r1 y qwq-plus, siendo los últimos dos mencionados modelos de razonamiento. Se obtuvo como resultado que los 4 modelos no erraron dentro de las 40 iteraciones realizadas; además, el modelo qwen-plus fue el de menor latencia y deepseek-v3 el de menor costo total. Tras esto se compara el modelo deepseek-v3 contra gemini-2.0-flash con la diferencia que esta vez se aplciará un self-consistency (n=5) a las 40 iteraciones. Se obtuvo como resultado una menor dispersión de los datos para ambos modelos, siendo el modelo de Google mejor en latencia y costos totales.

Tabla de contenidos

Índice de tablas

Índice de ilustraciones

1. Introducción

Una empresa financiera posee varios analistas, los cuales independiente del área en la que son parte, siempre tendrán como responsabilidad "cuadrar" sus procesos de forma diaria, semanal, mensual, trimestral, etc. con el fin de validar que la información que se está manejando es correcta y, en el caso de estar descuadrado, tener la información necesaria para poder gestionar y corregir aquello.

El proceso de cuadrar puede ser agotador debido a que en muchas empresas se hace de una forma muy precaria y desactualizada a los tiempos actuales, dependiendo plenamente de las habilidades del analista en manejo de datos. Adicionalmente, en empresas más establecidas tiene fuerte restricciones a software nuevo y/o a herramientas de programación como Python, esto limita las capacidades de mejora en eficiencia del proceso de cuadratura e impide que el analista logre ejecutar su proceso adecuadamente, generando atraso o demoras en el cumplimiento. Inclusive, algunas cuadraturas son fundamentales ser terminadas lo antes posible ya que son requeridas para cerrar meses o desbloquear otros procedimientos de otros analistas.

En general, el proceso es tedioso no solo por el manejo de datos, sino también porque, en la mayoría de los casos, se deben justificar cada descuadre. Para ejemplificar esto, se encuentra el caso de bonos de reconocimiento, donde suele suceder que este liquida dentro del periodo sin embargo aún no se realiza el pago real de este, por tanto, no sale de la custodia. Esto genera un descuadre ya que el custodio afirma que existe esa cantidad de nominal mientras el sistema interno de la empresa asume que se paga el mismo día de liquidación y por ende ya no existe en custodia. Si uno es nuevo en una empresa, no tiene ninguna pista de intuir de que el sistema funciona de esta manera ya que normalmente el funcionamiento del software es llevado por el equipo de tecnología.

1.1. Objetivos

Dado a la problemática descrita en el punto anterior, a continuación, los objetivos de esta propuesta.

1.1.1. Objetivo general

Se busca en primera instancia, generar un programa/aplicativo que pueda generar cuadraturas para ser usadas en auditorias o controles internos. En segunda instancia, facilitar la justificación o la corrección de descuadres o diferencias durante el proceso de cuadratura.

1.1.2. Objetivos específicos

La propuesta de usar un sistema de múltiples agentes tiene los siguientes objetivos:

- Reducir el tiempo en horas de trabajo del analista versus la forma ya establecida de hacerlo.
- Alcanzar un mayor nivel de precisión y detalle a la hora de cuadrar comparado al procedimiento ya establecido en el mercado.

2. Marco Teórico

2.1 Antecedentes generales

La revolución de los datos y la tecnología abre un gran espacio para que nuevas empresas de vanguardia entraran en los sectores más tradicionales y consolidados, como la banca y las finanzas (Alt et al., 2018). Con la aparición de la competitividad debido al uso de tecnología, los bancos tradicionales se vieron obligados a tomar una decisión: cambiar sus procesos y actualizar su pila tecnológica para incorporar nuevas funcionalidades o seguir con un plan de negocio obsoleto condenado al fracaso (Frame et al., 2018).

Para evitar quedarse atrás, los bancos e instituciones financieras tradicionales empezaron a aplicar prácticas derivadas de las empresas de tecnología financiera (FinTech). Entre estas prácticas y nuevas tecnologías, la inteligencia artificial (IA) destaca como una gran forma de aumentar la eficiencia, descubrir nuevos patrones, desbloquear información y optimizar los recursos de una corporación. El uso correcto de la IA puede traducirse en mejores conocimientos, decisiones mejor informadas y, en general, mejores resultados (Wamba-Taguimdje et al., 2020).

Las soluciones de IA se han introducido en todos los grandes sectores de la economía; un sector que está experimentando una profunda transformación impulsada por la revolución tecnológica en curso es el financiero. Las instituciones financieras, que dependen en gran medida de los macrodatos y la automatización de procesos, se encuentran en una posición única para liderar la adopción de la IA (PwC 2020), lo que genera varios beneficios: por ejemplo, fomenta la automatización de los procesos de fabricación, lo que a su vez mejora la eficiencia y la productividad. Además, como las máquinas son menos propensas a errores de tipificación y a los factores subjetivos, garantiza análisis predictivos y estrategias comerciales con mayor imparcialidad. (Bahoo et al., 2024).

Los factores que ayudan a que el mundo financiero tenga esta posición de ventaja se deben a la gran cantidad de datos que manejan, lo cual permite que estos modelos se evalúen y entrenen de mejor manera. También existe un abanico enorme de potenciales uso como: detección de patrones, predicción de tendencias y evaluación de riesgo. Todo con una rapidez y precisión que difícilmente otros sectores requieran al mismo nivel que una compañía financiera.

Asimismo, el auge de los modelos extensos de lenguaje (LLM, por su sigla en inglés) han transformado múltiples sectores, desde la tecnología hasta la salud. Desde el lanzamiento de ChatGPT en 2022, los LLMs han demostrado su capacidad para automatizar tareas, mejorar la atención al cliente y generar contenido con precisión (Díaz, 2025). Empresas de todo el mundo han adoptado esta tecnología para optimizar procesos y aumentar la productividad, lo que ha impulsado una ola de innovación sin precedentes (Hanano & Rizzo, 2025). Sin embargo, su implementación requiere una evaluación constante y consideraciones éticas para garantizar su uso responsable (Díaz, 2025).

Un factor que hizo que estos modelos fueran adoptados rápidamente por la población global es su capacidad de interacción mediante lenguaje natural, haciendo la tecnología más accesible para personas sin conocimientos técnicos. Su rapidez para procesar grandes volúmenes de información y su flexibilidad para adaptarse a distintos contextos permiten automatizar tareas complejas de forma eficiente. Además, no dependen de reglas predefinidas, lo que los hace altamente adaptables a situaciones cambiantes. Esto los convierte en herramientas complementarias al trabajo de los programadores, optimizando procesos, potenciando la creatividad y ofreciendo soluciones dinámicas y fáciles de usar.

A pesar de que los creadores de distintos LLMs declaren que estos modelos son imparciales, la literatura sugiere que los LLMs muestran sesgos raciales, de género, religiosos y políticos (Motoki et al., 2023). En la última década, ha habido una fuerte tendencia al alza; donde el ritmo de crecimiento y el grado de penetración de la adopción de la IA en el ámbito se han convertido en objeto de un número cada vez mayor de artículos de investigación. de artículos de investigación (Bahoo et al., 2024), lo que se debe principalmente a la irrupción de la IA Generativa.

Bahoo et al. (2024) identificaron que, aunque el uso de la IA se concentra principalmente en la banca y los servicios financieros, las investigaciones existentes han explorado su aplicación en una amplia variedad de industrias. Esto evidencia el vasto potencial de la IA y cómo puede beneficiar prácticamente a cualquier sector. Entre los sectores destacados se encuentran “Commodities”, “Energy

and utilities", "IT industry" y "Wholesale and Retail", los cuales suman 27 publicaciones de las 110 revisadas en la última década.

2.2 Antecedentes sobre procesos de ETL

Nwokeji et al. (2018) investigaron distintos enfoques de Extracción, Transformación y Carga de datos (ETL en su sigla en inglés), publicados anteriormente por otros autores. Algunos son:

Modelamiento Conceptual:

Diseño visual del proceso mediante diagramas. Ayuda a la claridad y a reducir potenciales errores al desarrollar. A pesar de que la aplicación del modelamiento conceptual al ETL ofrece algunas ventajas, como la generación automática de código (Akkaoui et al., 2013), el énfasis excesivo en ello, en detrimento de otros enfoques suscita preocupación. Por ejemplo, aunque los enfoques de modelización conceptual parecen útiles en la actualidad, no hay no dirección clara de la investigación ni pueden aplicarse para desarrollar soluciones eficaces para abordar el futuro aumento exponencial de la complejidad, el volumen y la heterogeneidad de los datos. (Nwokeji et al., 2018)

Sistema de múltiples agentes (MAS):

Arquitectura basada en la colaboración de varios agentes o componentes autónomos que trabajan juntos para realizar las tareas asociadas con la extracción, transformación y carga de datos. Son capaces de simular las interacciones sociales y el trabajo en equipo en el mundo real, mejorando la adaptabilidad y la eficiencia generales mediante procesos descentralizados de toma de decisiones e intercambio de información (Li et al., 2024).

Nwokeji et al. (2018) concluye que los enfoques de soluciones ETL están sobreenfocados en modelamiento conceptual, descuidando soluciones emergentes e innovadoras como es el machine learning.Esto se ve reforzado por otros autores como Casters et al. (2010), quienes afirman que un 45% aproximado de las soluciones de ETL se sigue realizando, utilizando programas/scripts codificados a mano. Además, expresan las desventajas de este enfoque que son:

- Propensión a errores
- Lentitutd en términos de tiempo de desarrollo
- Dificultad de hacer mantención mientras el código va escalando en complejidad
- Falta de metadatos
- Falta de coherencia en el registro/gestión de errores

La posible automatización en la medición de los procesos reduce los costos relacionados a tiempo, dinero, reduce la observación, elimina la revisión de pocos procesos, y maximiza la revisión de todos los procesos en todas las áreas del banco, todo esto puede aumentar la eficiencia del personal de ingeniería de procesos.

Frameworks como Azure Data Factory (ADF) han mejorado, en varios factores, la eficiencia de una compañía tras ser implementados. La literatura habla de un aumento del 50% en velocidad de procesamiento de datos, reducción de un 30% en costos, reducir el ratio de error en un 80% (Rapolu, 2023).

También reconoce la importancia de desarrollar procesos en la gestión de datos para obtener ahorros financieros, perfección operativa y desarrollar la calidad de los datos. La naturaleza compleja y el desarrollo de volúmenes de datos se lleva a cabo a través de las empresas y las herramientas útiles como ADF juegan un papel importante para manejar el fondo basado en datos. Bajo la asistencia de los tipos complejos de efectos, estructuras y aplicaciones de la industria, la investigación adicional tiene que ser desarrollado y por lo tanto explica las características reconocidas. Así, durante el éxito de la organización, los investigadores pueden manejar el complejo papel de reconocimiento de la automatización de datos (Rapolu, 2023).

2.3 Antecedentes en detalle sobre sistema de múltiples agentes (MAS)

Se define como ***agente*** a la entidad que puede percibir su ambiente y actuar en respuesta a este o al cambio de este; además, esta entidad muestra un comportamiento autónomo para el desarrollo del objetivo que se le plantea(Wooldridge, 2002).

De forma más técnica, Wooldridge da las siguientes características de un agente:

- Autónomos: Puede actuar sin la intervención humana y tiene control sobre sus acciones y estados internos.
- Reactivos: Puede responder a los cambios de su ambiente.
- Proactivos: Puede tomar iniciativas para llevar a cabo su objetivo.
- Sociales: Puede interactuar con otros agentes.

Se busca que cada agente se enfoque en una tarea particular del proceso con el beneficio de poder comunicarse entre ellos y soportar errores que un programa se caería, dando una solución más global y robusta.

Cabe destacar que MAS ya era implementado y estudiado previo al boom de la IA y los LLMs. Uno de los *frameworks* previo al uso de LLM es Java Agent DEvelopment Framework (JADE). Según Talib et al. (2016), JADE ayuda a mejorar capacidad de respuesta y la eficiencia del sistema en la fase de preprocesamiento de datos, es decir, para gestionar los valores que faltan durante el proceso de extracción de datos. JADE soporta diferentes estados de los agentes como: comunicación entre agentes, protocolo, comportamiento y ontología.

Por la misma época, Scholz et al. (2017) destaca JADE por ser de código abierto y tener varias funciones listas para usar, que permiten a los desarrolladores centrarse en su tarea principal (el desarrollo de los agentes) y no en problemas técnicos relacionados con la plataforma, como la comunicación entre agentes.

En la actualidad, JADE fue opacado por otros *frameworks* enfocados al uso de LLM. La razón de este cambio es debido principalmente a que estos mantienen los mismos beneficios de los que describe Scholz et al. (2017) para JADE. Por ejemplo, LangChain AI ha presentado LangGraph, un marco de trabajo de código abierto para construir aplicaciones LLM de estado multiactor con operaciones cíclicas, mientras que LangGraph puede facilitar la creación de aplicaciones multiagente (Easin et al., 2024).

Algunas aplicaciones de estos agentes usando el framework LangGraph en la literatura son:

- Agentes para automatización web (Wang & Duan, 2025)
- Agentes para la traducción de varios idiomas (Wang & Duan, 2024a)
- Agentes para el análisis de datos en ambientes de big data (Wang & Duan, 2024b)

En estas publicaciones, Wang y Duan han concluido que el framework de LangGraph destaca por su sistema que admite la recuperación de errores sin fisuras, la retención en memoria y el procesamiento iterativo, esenciales para las aplicaciones de alto riesgo (Wang & Duan, 2024b). Allana el camino para una adopción más amplia de soluciones de procesamiento lingüístico inteligentes y adaptables en campos tan diversos como el comercio electrónico, la educación y la comunicación internacional (Wang & Duan, 2024a).

Además, LangGraph mejora las capacidades del agente almacenando de forma persistente el historial de conversaciones para garantizar la continuidad, introduciendo la supervisión humana para corregir los flujos de trabajo y utilizando la memoria multisesión para mejorar el rendimiento aprendiendo de

los fallos del pasado. Estas funciones crean una experiencia más inteligente y centrada en el usuario (Wang & Duan, 2025).

Fuera de la literatura, tenemos casos reales en producción de empresas a la vanguardia como LinkedIn y Uber. La primera menciona que usa la inteligencia artificial para agilizar la contratación mediante la búsqueda conversacional, el emparejamiento de candidatos y un sistema de agentes jerárquicos basado en LangGraph (Ramgopal et al., 2024). La segunda, con el fin de abordar migraciones de código a gran escala, el equipo de la plataforma de desarrolladores de Uber utilizó LangGraph para construir una red de agentes y automatizar la generación de pruebas unitarias (Smith & Huda, 2024).

2.4 Antecedentes sobre estandarización de datos

Las empresas prosperan con flujos de trabajo eficientes, y uno de los aspectos más críticos implica el mantenimiento de datos estructurados en múltiples sistemas. La estandarización de datos puede ser una tarea ardua, ya sea entre departamentos de una misma empresa o entre toda una organización. El hecho es que cuando existen normas de datos claras y coherentes, todos los departamentos tienen acceso a lo que necesitan para realizar su trabajo sin tener que preocuparse por volver a aprender los formatos, y se pueden evitar los problemas de integridad de los datos (Simplilearn, 2024).

Los datos incoherentes ralentizan de forma global a una empresa, desde las operaciones rutinarias hasta los análisis avanzados, y dejan a las organizaciones expuestas a costosos errores. Sin embargo, para las empresas que dependen de múltiples plataformas SaaS y aplicaciones internas, las discrepancias de datos entre sistemas son casi inevitables. Por eso es imperativa la estandarización de los datos. Al aplicar la estandarización de datos, las empresas pueden eliminar estas discrepancias, agilizar los fallos en los datos y garantizar que cada decisión se base en información coherente y de alta calidad. El resultado no son sólo datos más limpios, sino una mayor eficiencia, información más precisa y resultados más sólidos basados en datos (Elahi, 2024).

Por ejemplo, el estudio de Lee et al. (2024) examina la viabilidad para la estandarización automatizada de datos en tiempo real aprovechando los modelos de lenguaje grandes para mejorar los sistemas de posicionamiento sin fisuras en entornos IoT. Al integrar y estandarizar datos de sensores heterogéneos procedentes de smartphones, dispositivos IoT y sistemas dedicados como los de banda ultraancha (UWB), garantizando la compatibilidad de los datos y mejora la precisión del posicionamiento. Además, obtuvieron como resultado una pérdida cercana a cero y una precisión total tanto en la fase

de entrenamiento como en la de validación, lo que demuestra unas capacidades de aprendizaje sólidas y una normalización de datos fiable en diversas entradas de sensores.

2.5 Antecedentes de barreras de entrada a la innovación

Christensen (2011) explora las complejidades y desafíos que enfrentan las empresas al intentar adaptarse a la innovación disruptiva. Este concepto se refiere a la aparición de tecnologías o modelos de negocio que transforman industrias existentes, poniendo en riesgo a las organizaciones establecidas. Christensen identifica tres principales barreras que dificultan este proceso de adaptación: el enfoque en los clientes actuales, las estructuras organizacionales y la asignación de recursos. Estas barreras representan retos significativos, pero también oportunidades de transformación para las empresas que buscan mantenerse competitivas en un entorno en constante cambio.

Un factor que limita la adaptación es el excesivo enfoque en las necesidades de los clientes actuales. Las empresas suelen priorizar la satisfacción de su base de consumidores existente, relegando las oportunidades que las tecnologías emergentes podrían brindar. Este enfoque puede llevar a subestimar los avances disruptivos, permitiendo que competidores más pequeños ganen terreno. Asimismo, las estructuras organizacionales tradicionales, diseñadas para optimizar la eficiencia de los negocios establecidos, a menudo son inflexibles. Esto dificulta la rápida adopción de innovaciones disruptivas, ya que los procesos internos no están preparados para manejar estos cambios.

Otro obstáculo importante señalado por Christensen es la asignación de recursos. Las empresas tienden a invertir en iniciativas que prometen altos márgenes de rentabilidad, ignorando tecnologías disruptivas que, en una etapa inicial, pueden parecer menos atractivas financieramente. Este enfoque conservador puede limitar el crecimiento a largo plazo y dejar a las organizaciones vulnerables frente a los nuevos actores del mercado. Para superar estas barreras, Christensen propone estrategias como la creación de unidades independientes dedicadas a explorar nuevas tecnologías y mercados emergentes. Estas unidades pueden operar fuera de las restricciones de la estructura principal de la empresa, permitiendo mayor flexibilidad e innovación.

En este sentido, esta memoria de título se enfoca en proponer una metodología con el fin de poder implementar una solución disruptiva para una empresa de seguros consolidada. Además del conocimiento de ETL e IA, que se explica en puntos anteriores, se requiere un plan estratégico para afrontar las barreras al cambio. Actualmente existe una brecha dentro de las soluciones de ETL aplicadas al mundo financiero. Esta es el enfoque mediante el uso de MAS. Además, dentro de estos

agentes, existe un enfoque nuevo orientado al uso de LLMs. Con esta memoria de título se busca indagar en una innovación que poco a poco puede reemplazar a las soluciones establecidas por el mercado actual, ofreciendo una solución flexible y a menor costo.

3. Metodología

La metodología se divide en la obtención, selección y tratamientos de los datos; programación del sistema de múltiples agentes; las métricas a usar para comparar entre modelos.

3.1 Datos

El proceso de cuadraturas del área de custodia y visado requiere principalmente dos datos: El saldo de cuota (o nominales) registrado en el sistema de la empresa versus los nominales registrados por el custodio. Para hacer el cruce entre ambos datos se requiere un identificador único o key, en este caso se usará el nemotécnico, el cual es un identificador único del instrumento finaciero solo a nivel de empresa. Sin embargo, el problema con esto es que, para la mayoría de los casos, los nemotécnicos no son universales, sino que se usan exclusivamente a nivel interno de cada empresa. Por lo que se requiere un dato adicional tipo diccionario que ayude a homologar entre nemotécnicos de distintas empresas.

Con respecto al formato de los datos, estos pueden ser categorizados en planillas Excel y archivos de texto. Las planillas de Excel pueden ser principalmente del formato 2003 (xls) o el formato moderno (xlsx). Para los archivos de texto estos son más diversos; estos pueden ser archivos txt con ancho fijo (FWF) archivos separados por comas (csv), separados por tabuladores (tsv), etc.

Adicionalmente a lo mencionado, estos datos también tienen una diversidad de nombres de columnas. Por ejemplo, para el nemotécnico puede llamarse “id”, “n° operación”, entre otros. Esto también aplica para los nominales que pueden llamarse “saldos”, “saldos cuotas”, entre otros. Además, en algunos casos los datos no cuentan con una cabecera que dé nombre a la columna, como es el caso del Depósito Central de Valores de Chile (DCV).

El proceso de cuadrar consiste en comparar los nominales de cada nemotécnico en el sistema interno de la aseguradora versus el del custodio. En el caso de haber una diferencia real, esta debe ser explicada por su porqué. Además, este proceso puede repetirse en el caso de que el holding tenga varias empresas filiales. En ese caso se debe repetir el proceso por cada empresa filial que use dicho custodio.

Por ejemplo, imaginemos que la tabla 1 es la cartera de una empresa de seguros y la tabla 2 es la cartera de un custodio o contraparte. Un ejemplo de cuadratura es la tabla 3, donde “NOMINAL_1”

es la cantidad nominal de la tabla 1 y “NOMINAL_2” de la tabla 2. Asimismo, la columna “DIFF” es la diferencia absoluta entre ambas columnas de nominales.

Tabla 1. Ejemplo Cartera Empresa de Seguros

NEMO	NOMINAL
MH20200213218175	20.000
MH20214309826894	33.345
MH20292872908790	8.922

Fuente: Elaboración propia

Tabla 2. Ejemplo Cartera de Custodio

NEMO	NOMINAL
MH20200213218175	20.000
MH20214309826894	33.000
MH20252368404902	6.492

Fuente: Elaboración propia

Tabla 3. Ejemplo de Cruce-Cuadratura

NEMO	NOMINAL_1	NOMINAL_2	DIFF
MH20200213218175	20.000	20.000	0
MH20214309826894	33.345	33.000	345
MH20292872908790	8.922	-	8.922
MH20252368404902	-	6.492	6.492

Fuente: Elaboración propia

En algunos casos más complejos, esta diferencia por instrumento puede requerir un traspaso de nominales con contraparte por medio de una carga masiva al DCV. Para esto es importante que ambas partes se coordinen en cuentas nominales traspasar si no puede provocar errores en el DCV. Por lo que ahora si se filtra la tabla 3 reflejaría los instrumentos con sus nominales a traspasar si la diferencia es negativa y recibir en caso contrario.

Entonces, para estos casos de traspaso se requiere un paso adicional de cruce con la información de la contraparte para verificar que ambas partes están de acuerdo con traspasar la misma cantidad de nominales por instrumento. Si este cruce es positivo y no existen diferencias, se debe proceder en crear un archivo de texto el cual ha de ser cargado en DCV con la información del instrumento, nominal, fecha, cuentas DCV de ambas partes y si es recibir o traspasar dicho instrumento, más una clave que hace el cuadre con la contraparte que debe subir el mismo archivo, pero de forma opuesta,

es decir, si uno coloca que es traspaso de cuotas de un fondo mutuo, la otra compañía debe colocar que va a recibir dichas cuotas.

3.2 Agentes

3.2.1 Modelo de lenguaje natural

Se propone usar un sistema de múltiples agentes, usando la metodología ReAct (Reason and Acting) con el fin de que el modelo deje en explícito su razonamiento y las acciones o herramientas que usó para llevar a cabo la instrucción propuesta. Esto va de la mano con el concepto de "Chain of thought" (cadena de pensamiento), donde se busca que el modelo deje una traza de su pensamiento de modo que se descomponga el problema en pasos lógicos y estructurados. También se buscan acciones específicas de parte del modelo; para este caso, se quiere que utilice las herramientas que se dejan a disposición. Adicionalmente, se espera que el agente sea autónomo y, por tanto, itere las distintas herramientas hasta que obtenga, bajo su criterio, la respuesta final.

El enfoque ReAct reduce el error de alucinaciones al combinar razonamiento y acciones, mejora la precisión y relevancia de sus respuestas al utilizar herramientas que el programador incorpora y dan resultados fijos. Además, incorpora mejor interpretabilidad al dejar dichas trazas de razonamiento que son más comprensibles para un humano.

Los LLMs a elegir son Qwen-Plus, QWQ-Plus, Deepseek-v3 y Deepseek-r1. Los dos primeros son de la empresa Alibaba Group. Mientras que los dos últimos son de la empresa del mismo nombre (Hangzhou DeepSeek Artificial Intelligence Basic Technology Research Co., Ltd.). Las razones principales de escoger estos cuatro modelos chinos son por sus bajos precios y la ventaja de usar el entorno OpenAI en las librerías Python (Qwen Team, 2025).

A pesar de que Qwen es un modelo que lleva en el mercado desde 2023 (Chiang, 2023), ha sido un modelo desapercibido por la comunidad científica e ingenieril en occidente. Sin embargo, varios estudios muestran su potencial, en especial en el campo de la lógica y el razonamiento, superando a modelos como ChatGPT 4o y Llama 3.11 (C. Li et al., 2025; Aydin et al., 2025; Yang et al., 2024). Esto hace a Qwen, a priori, un modelo idóneo para agentes ReAct.

Por su parte, Deepseek, modelo de lenguaje que estalla en popularidad en el año 2025 por el lanzamiento de su modelo R1, el cual, la empresa china afirmó que está a la par de ChatGPT o1 de OpenAI para tareas de razonamiento, pero con un costo menor que la empresa americana. Por otro

lado, el modelo v3 no se queda atrás, el cual muestra un balance entre costo y potencia, el cual lo hace eficiente y competitivo. (DeepSeek-Ai et al., 2024)

3.2.1 Prompting

Un prompt es la instrucción que se le da a un modelo de LLM. En el caso de los chatbots, es decir, modelos que interactúan con un humano, estos tienen dos prompts. El primero es la instrucción del sistema o dada por el programador, y el segundo es el texto que envía el usuario a través del chat. Utilizando un prompt prefabricado del autor hwme17, se decide editarlo con el fin de adaptarlo a las necesidades de nuestro sistema. (Anexo 4.1)

El prompt pide que se responda en el siguiente formato hasta que el propio agente considere que tenga la respuesta final:

Question: La tarea o pregunta que debe resolver

Thought: Pensamiento de qué hacer para resolver question

Action: Acción a tomar, normalmente, el uso de una herramienta

Action input: El input usado para dicha herramienta

Observation: Resultado tras usar dicha herramienta

Para el ejemplo de la ilustración 1, que simula un agente que responde a consulta sobre indicadores diarios del mundo financiero para una corredora de bolsa, la pregunta (question) sería “Quiero saber el tipo de cambio CLP/USD” el pensamiento del modelo (thought) sería “Voy a requerir de usar una herramienta”, la acción a tomar (action) sería el nombre de la herramienta para hacer consultas a una base de datos que contiene los vuelos disponibles como “buscar_dolar_observador”, el input de la herramienta a usar (action input) sería el query, pero depende de cómo se programó dicha herramienta; por simplicidad diremos que es la fecha en formato día-mes-año y la observación del modelo al resultado de la herramienta (observation) va ser dicha consulta, para este ejemplo sería:

“fecha, dólar observado

05-07-25, 900”

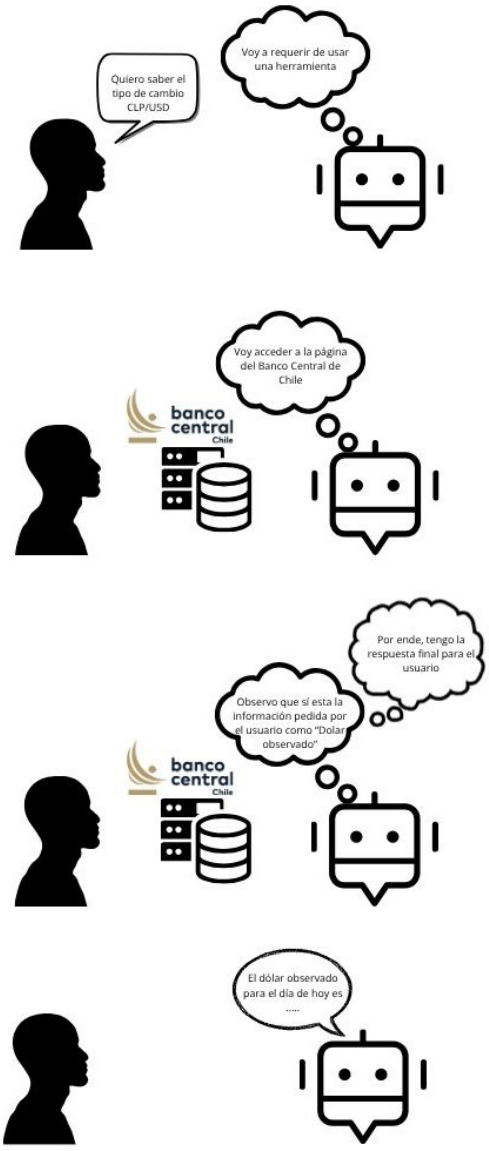

Ilustración 1 Ejemplo de metodología ReAct. Elaboración propia

Si bien esto habla de forma individual de cada agente, con este trabajo se busca además crear una interacción entre varios agentes. De forma macro se proponen crear 3 agentes: Supervisor, Operador e Integrador.

Se busca que el usuario pueda interactuar con lenguaje natural con el agente Supervisor y este sepa entender lo que quiere el usuario y filtre instrucciones nulas o que no van al caso de cuadrar o responder diferencias. Dependiendo de la consulta, puede ir a Integrador, cuyo rol principal es de incorporar nuevos datos al sistema. Por su parte, Operador será el encargado de hacer el cruce de datos, generar las diferencias por nemotécnico o crear el archivo de carga si no hay diferencias.

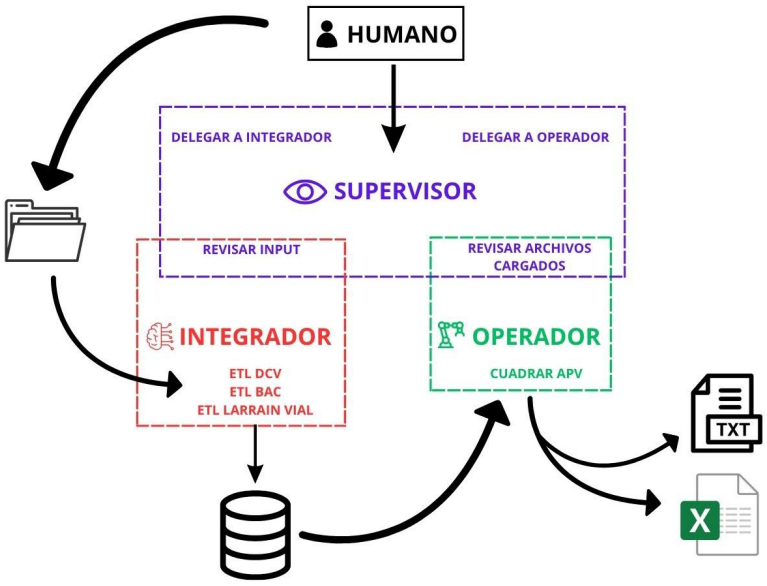

Ilustración 2. Diagrama de Sistema de múltiples agentes para cuadratura. Elaboración propia.

Integrador es quien tiene la mayoría de las herramientas de ETL descritas en el punto 2.2.3, ya que se requiere una limpieza previa de los datos antes de ser integrados al sistema. Una vez que el usuario pida una cuadratura, el Supervisor debe garantizar que los datos estén integrados para cuadrar; si no es así, deberá pedir al usuario cargar los datos para luego ser integrados en el sistema. En el caso de que sí estén los datos, el Supervisor dará la orden al Operador de cargar y cuadrar los datos (Ilustración 2).

3.2.2 Herramientas

Cada agente tendrá ciertas herramientas que le facilitarán su trabajo. Estas en definitivas son funciones de Python las cuales requieren una o varias entradas y se obtendrá una o varias salidas. Además, se les asignará un nombre y descripción, los cuales serán los principales influyentes de que si el agente

use o no dicha herramienta. En el caso de esta memoria de título, las principales herramientas serán de ETL, es decir, de extracción, transformación y carga (guardado) de datos.

Las herramientas de detección de tipo de archivo son las que permiten cargar adecuadamente la información. Aquí se asume que uno conoce el tipo de archivo por contraparte. Por ejemplo, el DCV usa un formato de texto de ancho fijo (.txt)

Las herramientas de extracción son dichas herramientas para extraer la información sobre nemotécnicos y nominales tanto de la contraparte como de la misma compañía de seguro por fecha. Como se explica en el punto 2.1, la dificultad radica en la heterogeneidad de los datos. Para lidiar con esto se utiliza la librería Pandas que cuenta con funciones para leer variedad de archivos a un dataframes (Pandas Dev Team, 2024).

Las herramientas de cuadre hacen el último paso de cruzar la información de dos dataframes, uno de la compañía de seguros y el otro de la contraparte. Luego, se continúa calculando la diferencia de nominales entre ambas compañías. Para el maneo de los datos, se utiliza librerías pandas para cruzar ambas tablas.

La herramienta de guardado es el paso final, donde se guardan las diferencias en un archivo Excel para que luego el usuario tenga acceso a esto y pueda revisar con mayor detalle. Se ocupa la función que viene en la librería pandas, la cual guardará el archivo Excel de forma local.

3.3 Métricas

Con el fin de comparar el rendimiento de estos agentes, se propone comparar en dos aspectos: 1) Costos y 2) Tiempo. También, las métricas son relevantes para explicar y justificar a la empresa de seguros qué LLM es el idóneo para ellos. Adicionalmente, se mide la complejidad computacional de este sistema de múltiples agentes.

3.3.1 Métricas de costos

El principal costo incurrido son los de las consultas API al LLM, básicamente, el modelo de lenguaje natural separa las palabras en “tokens” y son sus variables de entrada. Asimismo, el modelo devuelve un mensaje en “tokens” que luego el usuario lee y entiende al ser en lenguaje natural o humano.

Usando los tokens utilizados de entrada y salida de nuestro modelo, se sacará el costo total por iteración promedio. En otras palabras, se busca presentar un aproximado de lo que costaría pedir al modelo cuadrar desde la primera instrucción hasta que arroje el resultado final.

Se aclara también que el costo en sueldo a programadores queda excluido de las métricas de este trabajo.

3.3.2 Métricas de tiempo

El tiempo es un factor clave en el mundo financiero, donde se tienen que cumplir con normativas en fechas limitadas y están constantemente supervisadas. Por tanto, se busca reducir el tiempo de cuadrar los movimientos, en el caso de las aseguradoras mensualmente, pero, por ejemplo, para las Administradoras de Fondos de Pensiones de Chile (AFP) es diario por motivos de transparencia (Decreto 3500, 1980). Además, el poder cuadrar de forma automatizada y eficiente ayuda a que los analistas a cargo de dichos procesos de cuadratura se puedan enfocar en el control operativo más que en la parte manual del cruce de información, lo cual permite que el factor técnico no sea un limitante de entrada para estos cargos y sea más importante el conocimiento financiero del rubro.

Por tanto, se tomará el tiempo promedio desde que se ingresa el primer prompt de parte del usuario hasta que el modelo en su totalidad dé su respuesta final.

3.4 Experimento

3.4.1 Experimento inicial comparar modelos de chat versus razonamiento

Se procede a iterar 40 veces la siguiente instrucción con 4 modelos de IA chinos (deepseek-v3, qwen-plus, deepseek-r1 y qwq-plus):

“Quiero cuadrar APV para la fecha 0525”.

Adicionalmente, se agrega a la carpeta input los archivos requeridos para la cuadratura. Sin embargo, se altera conscientemente uno de estos archivos con el fin de forzar un error descalce, lo cual el modelo debe detectar y no continuar con el proceso y pedir al usuario corregir o revisar los datos. Lo anteriormente mencionado se seguirá mencionando como la fase 1 del experimento inicial.

Luego de terminar la fase 1 (Ilustración 3) y sin finalizar el chat con el modelo, se da la misma instrucción que en la fase anterior con la diferencia de que ahora se sube el archivo “corregido”, el cual el sistema de múltiples de agentes se espera que vuelva a cargar y complete la cuadratura sin diferencias, por tanto, generando el archivo de carga para el DCV (fase 2).

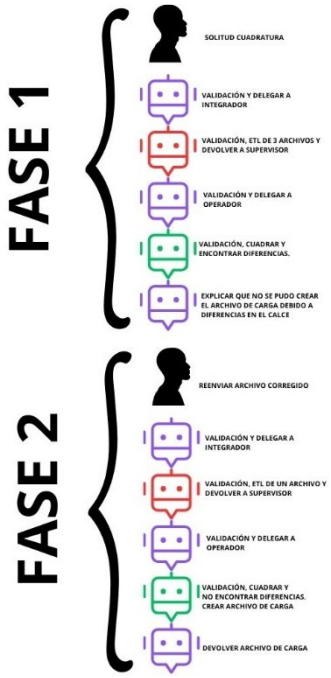

Ilustración 3. Esquema de flujo esperado del MAS. Elaboración propia.

3.4.2 Experimento self-consitency

Se procederá a iterar 200 veces la misma instrucción del punto anterior, esta vez con el modelo deepseek-v3 y gemini-2.0-flash. Tanto la fase 1 como la 2 quedan iguales con la variación que cada 5 iteraciones se elegirá el valor modal de estas (Ilustración 4). Se espera una mejora en la consistencia de los datos con respecto al experimento previo.

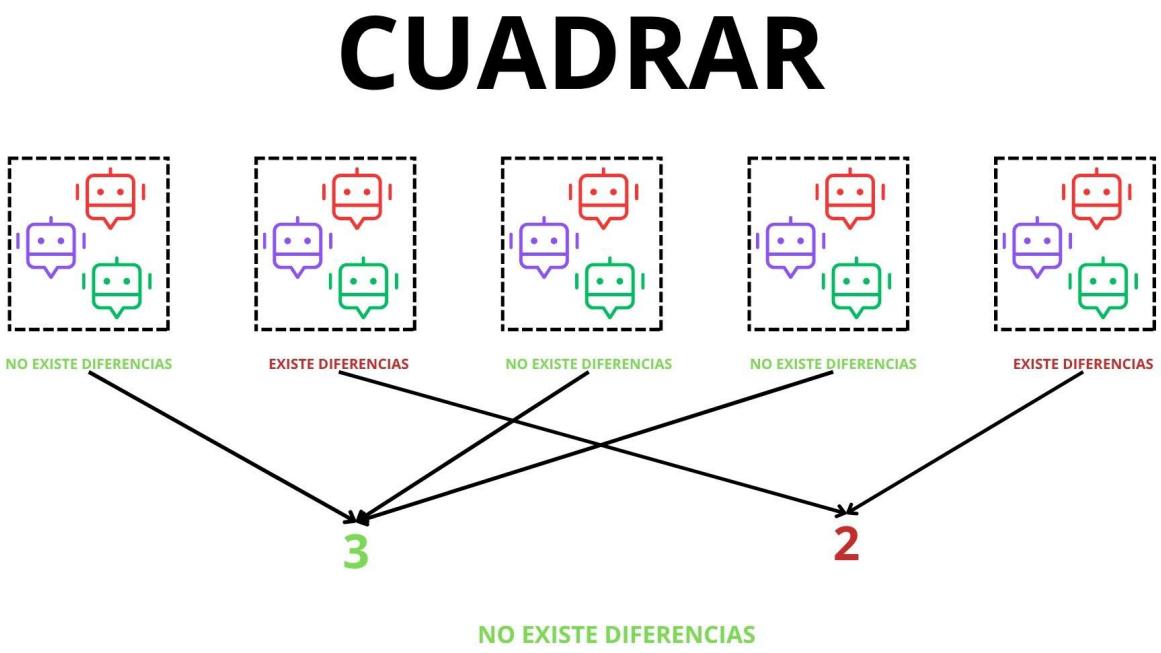

Ilustración 4. Esquema de self consistency n=5. Elaboración propia

4. Análisis de resultados

A continuación, se muestran los resultados de testeo de simulación de un proceso de cuadratura.

4.1 Resultados principales

Lo primero que resalta tras la simulación es ver una precisión del 100% en cada modelo, es decir, independiente del LLMs el MAS se configuró adecuadamente en orden de que el modelo se le facilita la toma de decisiones (Ilustración 5).

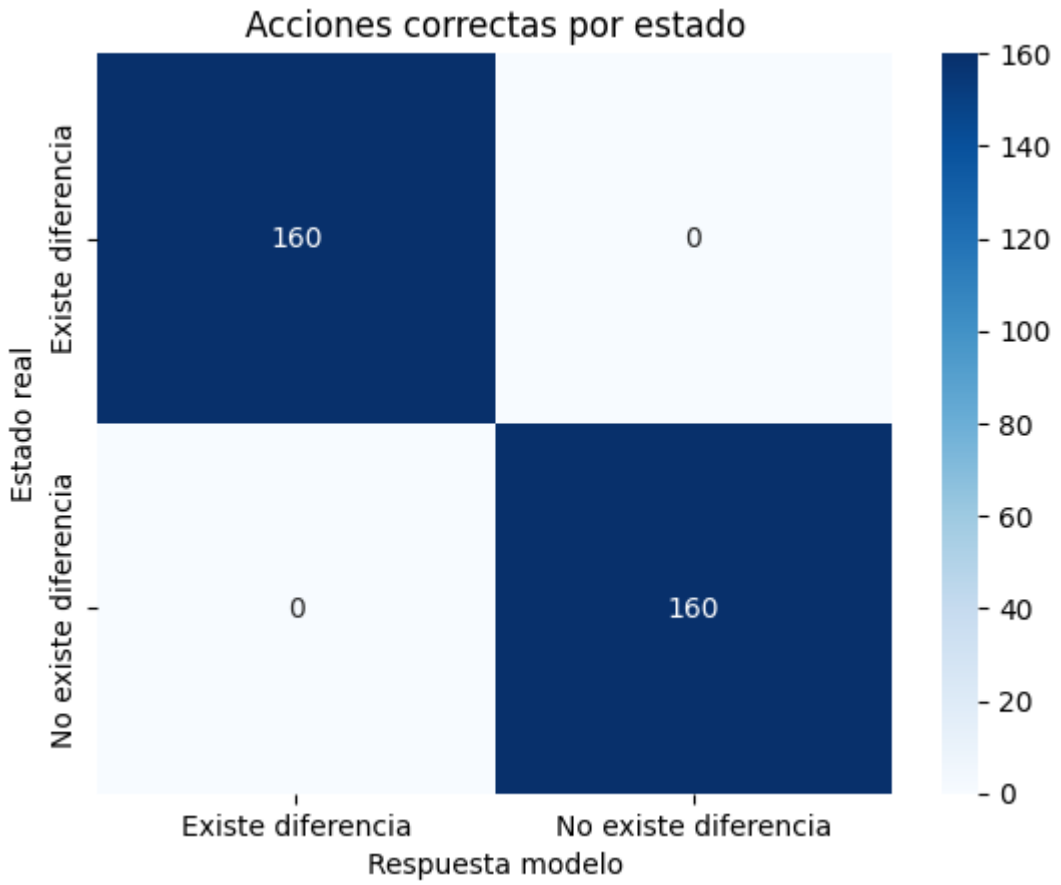

Ilustración 5. Matriz de confusión de MAS Elaboración propia..

A continuación, se describe los resultados que sí hay diferencia por LLM, partiendo con el consumo de tokens, los modelos de chat tanto de Deepseek como de Qwen tienen un prompt de salida (output) significativamente menor y optimizado comparado a sus versiones de razonamiento (Ilustración 6). Sin embargo, esto no aplica para el prompt de entrada (input) el cuál es ligeramente más alto en promedio.

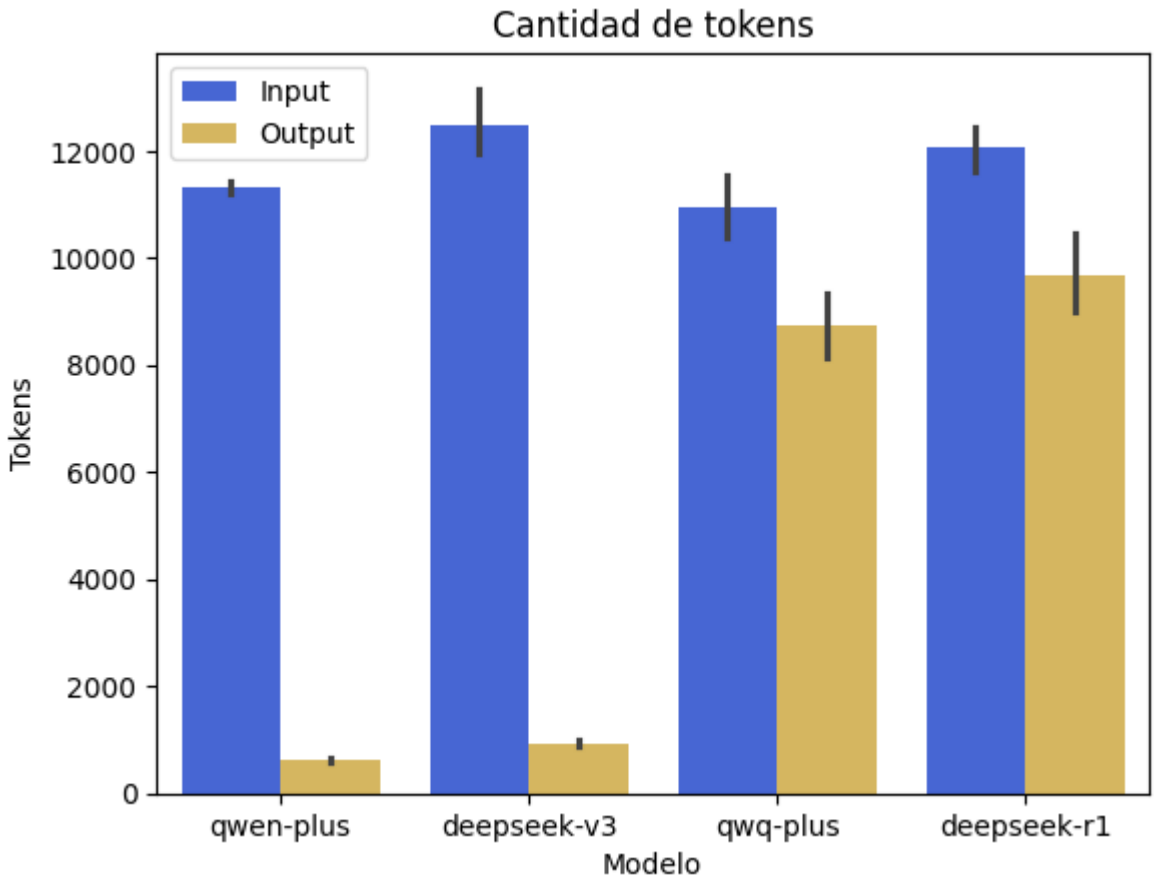

Ilustración 6 Tokens totales por modelo de IA. Elaboración propia.

Al ser más caros los tokens de salida que de entrada en los modelos de IA que probamos, se puede observar que el costo promedio por iteración es más bajo en los modelos de chat que de razonamiento (Ilustración 7). También agregar que, a pesar de que el modelo Deepseek-v3 sea el más económico en costos promedio por iteración, este tiene una mayor dispersión en costos que su contraparte Qwen, por lo tanto, no se puede afirmar que sea una diferencia significativa entre ambos modelos. Esto se ve más claro en la ilustración 9 donde se aprecia una mayor dispersión en los modelos con un enfoque al razonamiento.

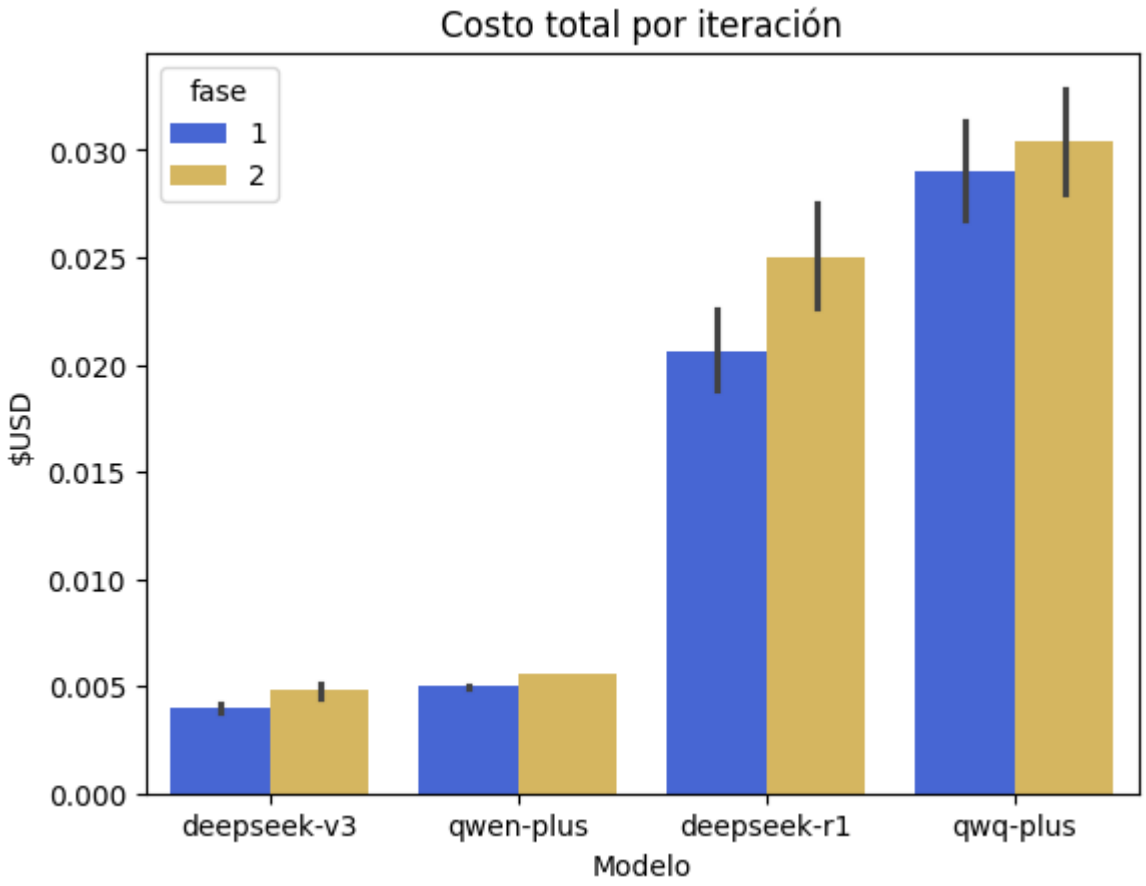

Ilustración 7 Costos totales por iteración. Elaboración propia.

Respecto al tiempo de latencia entre la primera instrucción y la respuesta final del modelo, la cual concluye con el proceso de cuadratura, vemos que el modelo qwq-plus es en promedio el más veloz con unos 35 segundos en promedio (Ilustración 8). De forma complementaria se aprecia en la ilustración 6 una mayor homogeneidad en los modelos convencionales sobre los de razonamiento.

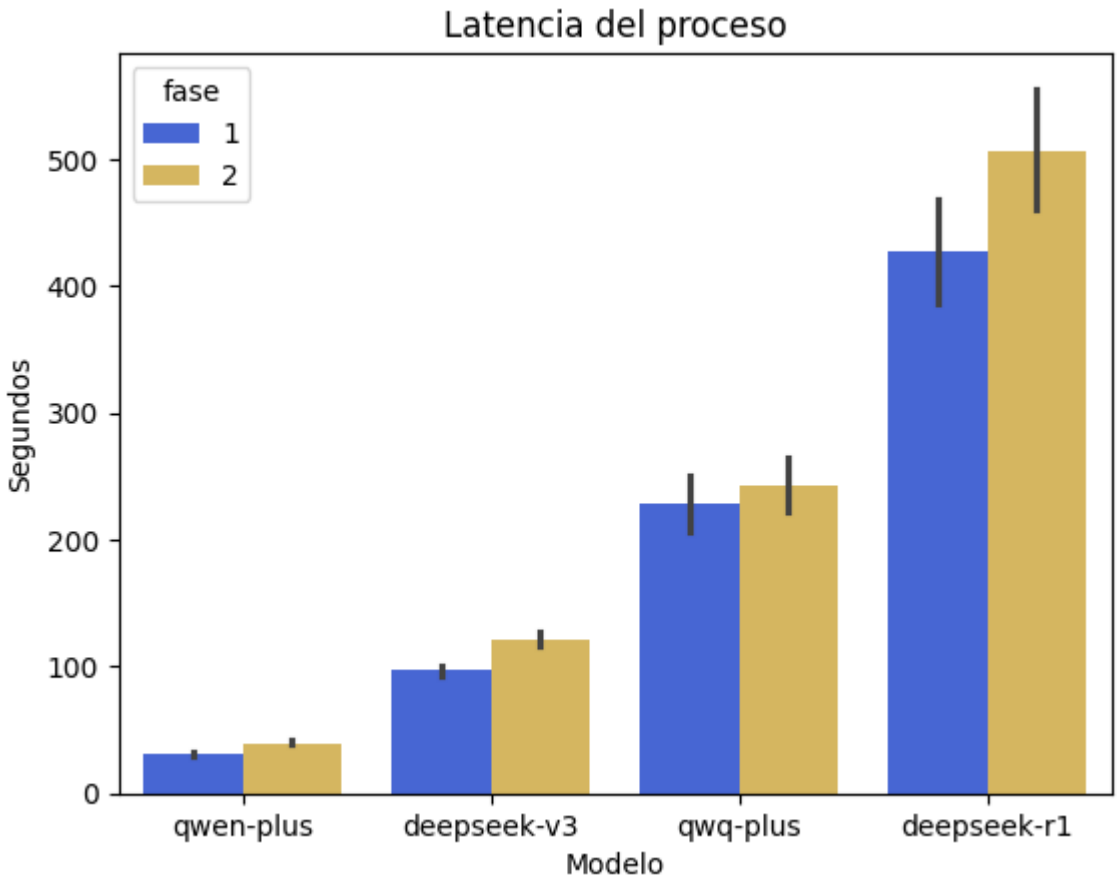

Ilustración 8 Latencia promedio por iteración. Elaboración propia.

Con respecto a fallas o alucinaciones del modelo, ninguno de los 4 modelos presentó alguna en las 40 iteraciones que se realizaron, esto evidencia de que el framework fue construido correctamente, ya que no hubo alteraciones en el comportamiento de estos y se realizó la tarea de cuadratura con éxito, independiente del modelo que se usara. Por ejemplo, para el caso del agente supervisor, este, a través de los 4 modelos, supo delegar correctamente las tareas a los otros agentes, entregando la información clave requerida por ellos. Además, este agente entendía que el usuario estaba pidiendo dos tareas, las cuales dependían una de la otra (cargar los datos y luego cuadrar).

A modo de resumen, las tablas 4 y 5 nos muestran los intervalos de confianza (95%) de la media de costos totales y latencia respectivamente, donde se observa que el modelo qwen-plus muestra la menor dispersión de datos tanto en costos como en latencia. Para el caso de la latencia, este modelo es el claro vencedor, siendo aproximadamente tres veces más rápido que el segundo lugar (deepseek-v3) en promedio. Sin embargo, en cuanto a costos, el modelo deepseek-v3 es el más bajo en costos totales,

aunque se puede apreciar que el costo por iteración en todos los modelos fue menor a 1 USD, por tanto, hay evidencia de poder priorizar el aspecto de latencia versus costos.

Tabla 4 Intervalo de confianza de costos totales (95%).

Modelos	Fase 1 (USD)	Fase 2 (USD)
Deepseek R1	[0.0187 - 0.0226]	[0.0224 - 0.0276]
QWQ Plus	[0.0266 - 0.0315]	[0.0278 - 0.0329]
Deepseek V3	[0.0038 - 0.0042]	[0.0044 - 0.0051]
Qwen Plus	[0.0049 - 0.0050]	[0.0056 - 0.0056]

Tabla 5 Intervalo de confianza de latencia media (95%)

Modelos	Fase 1 (s)	Fase 2 (s)
Deepseek R1	[385 - 468]	[458 - 555]
QWQ Plus	[207 – 251]	[207 – 251]
Deepseek V3	[92.9 - 102]	[116 - 127]
Qwen Plus	[29.3 – 32.1]	[37.8 – 41.6]

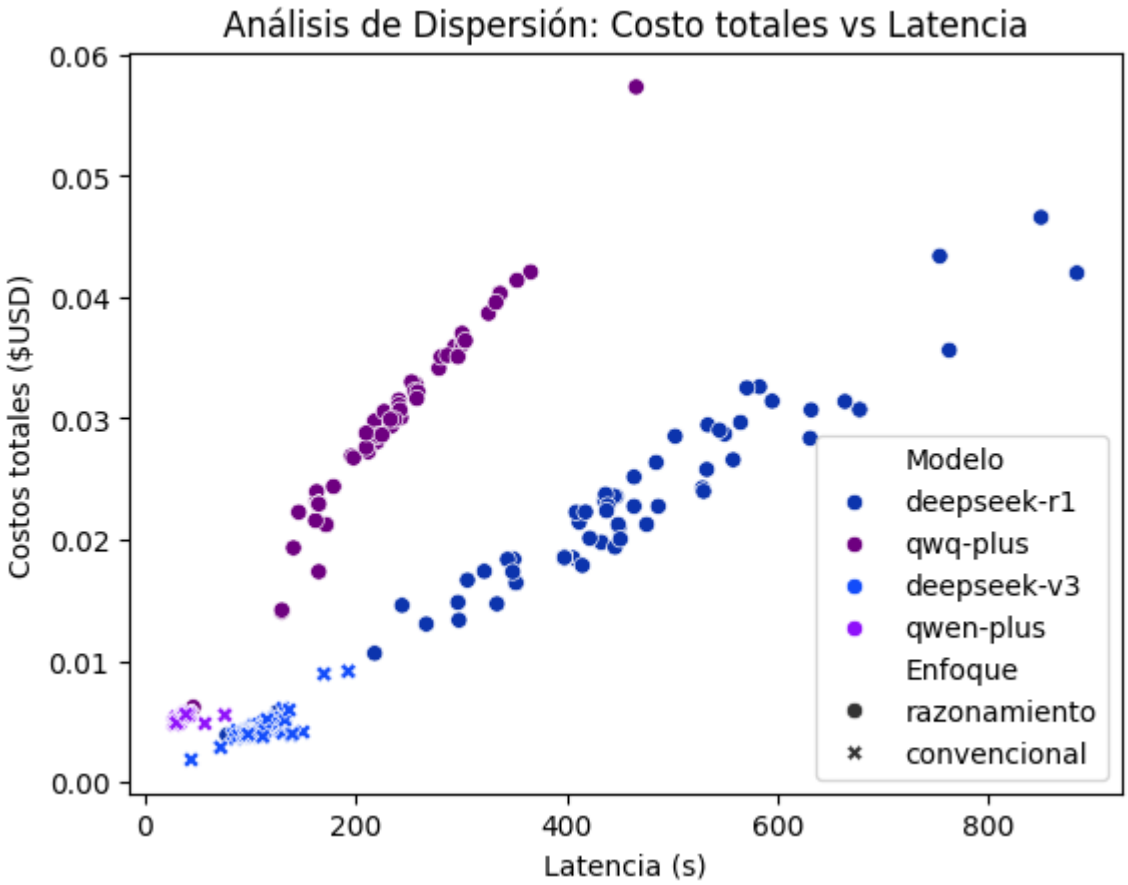

Ilustración 9. Análisis de Dispersión. Costos vs Latencia. Elaboración propia.

4.2 Resultados de self-consistency

Tras la realización de este experimento, se pudo confirmar la solidez del modelo quitando casos aislados y comparándolo con un modelo mucho más competitivo en el mercado como es Gemini-2.0-flash. Para ambos modelos, se obtuvo un 100% de precisión en el resultado final para las 40 iteraciones, es decir, el modelo llega al resultado esperado en cada fase (Ilustración 10).

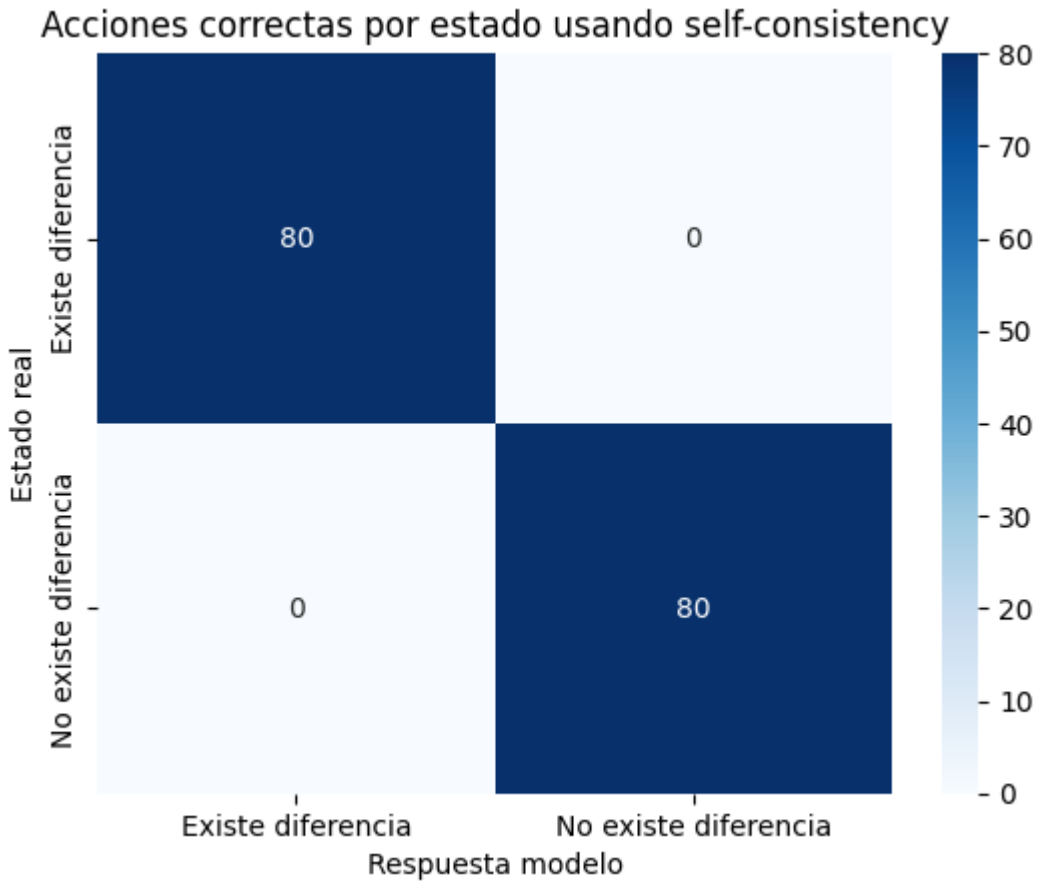

Ilustración 10. Matriz de confusión del MAS Self-consistency. Elaboración propia.

Como se observa en la ilustración 11, usar self-consistency aumentó la cantidad de tokens tanto en input como output 5 veces en promedio, lo cual es lo esperado. También, se aprecia menor uso de tokens por parte del modelo de Google vs Deepseek.

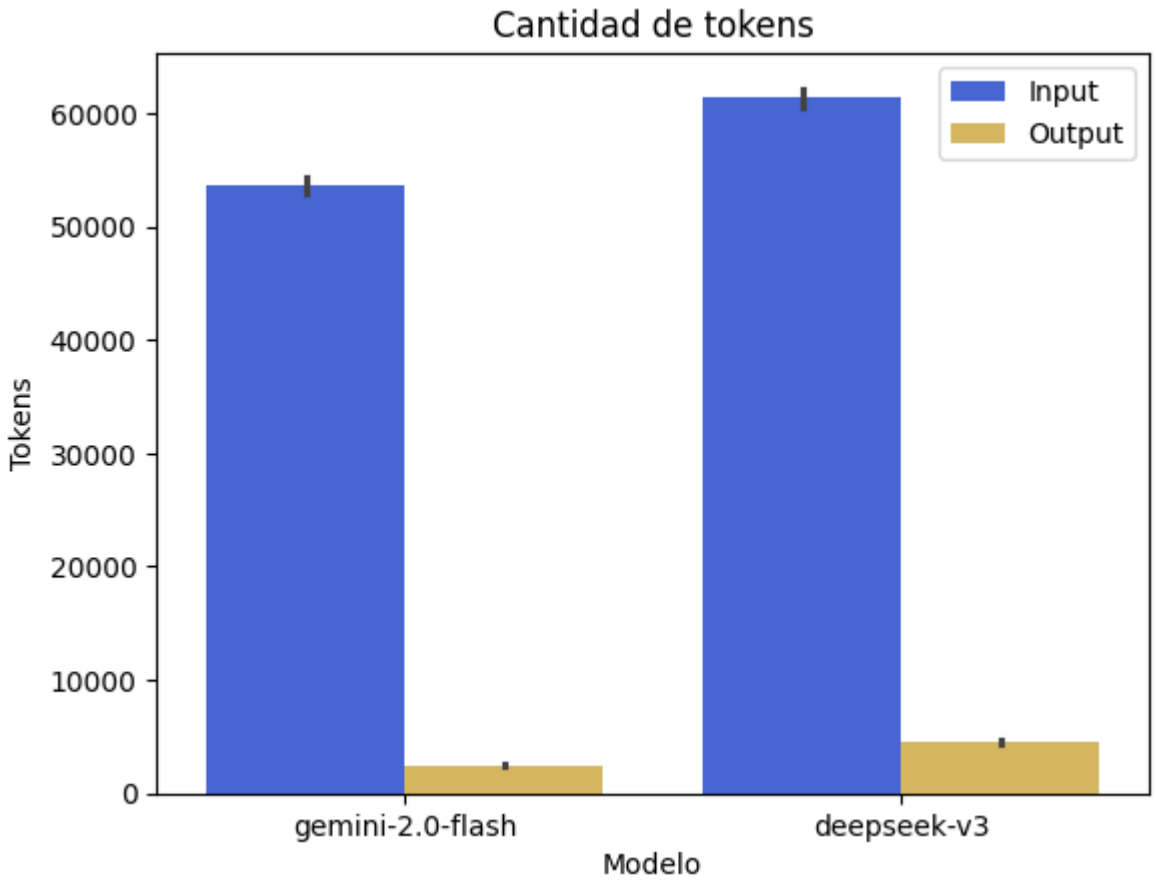

Ilustración 11. Tokens totales por modelo IA con self-consistency. Elaboración propia.

Tras calcular el costo de estos tokens, se observa como el modelo Gemini 2.0-flash tiene un menor costo de uso por iteración, tanto en la fase 1 como en la fase 2 (Ilustración 12). En promedio se ahorra un 33% de costo si se usará el modelo de Google versus el de Deepseek.

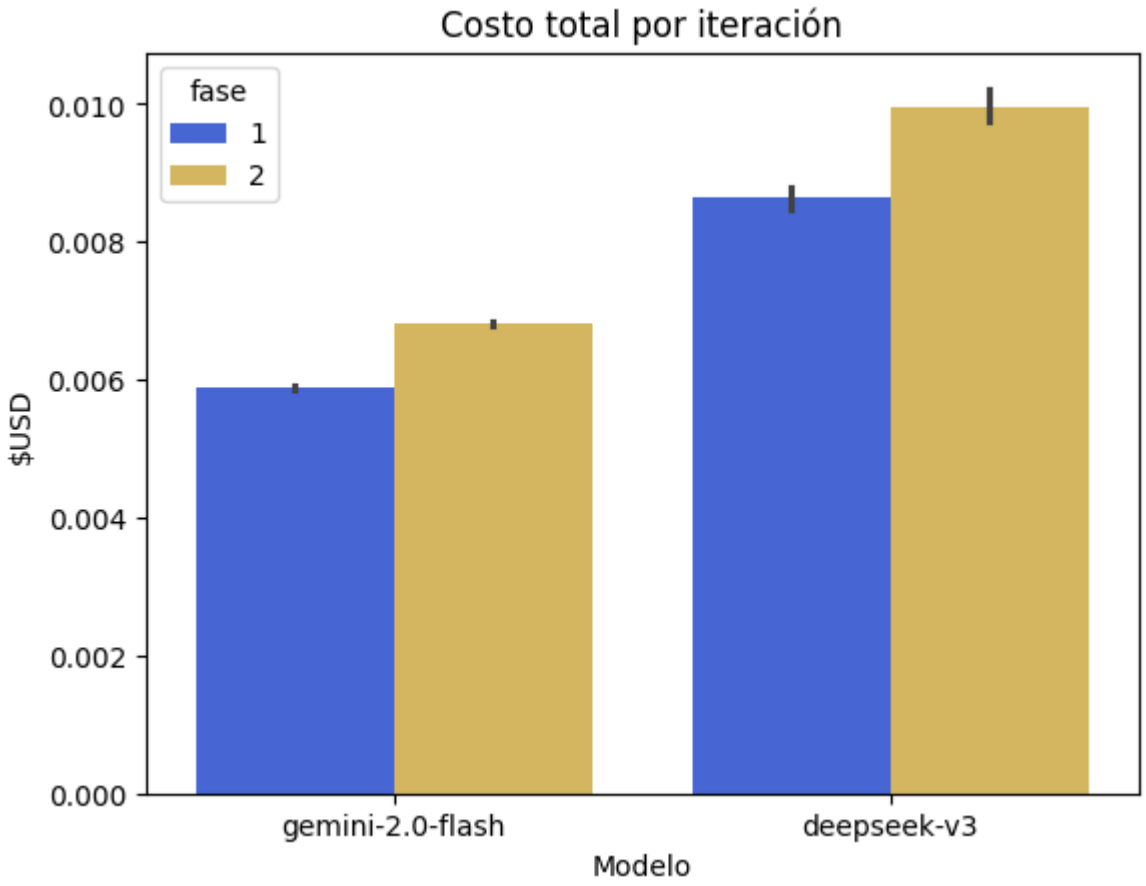

Ilustración 12. Costos totales por iteración usando self-consistency. Elaboración propia.

En el ámbito de latencia es donde más diferencia hubo entre ambos modelos, la ilustración 13 muestra como el modelo Gemini es más de 10 veces más rápido en promedio que el modelo Deepseek, siendo el primero de una latencia promedio de 10 segundos.

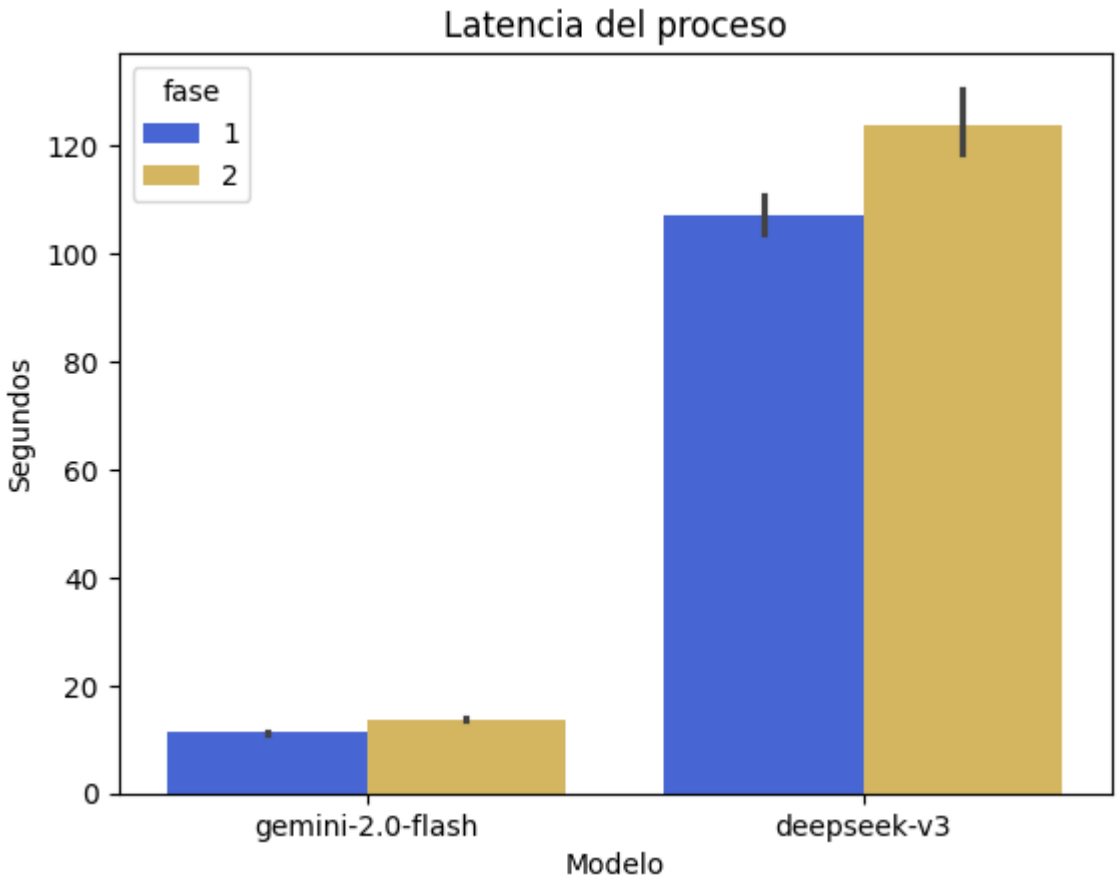

Ilustración 13. Latencia total por iteración usando self-consistency. Elaboración propia.

Se resumen los resultados de la comparativa de los modelos Deepseek V3 y Gemini 2.0 Flash con las tablas 6 y 7. Se evidencia no solo la superioridad del modelo de Google en términos de costos y latencia, sino que también, si se calcula el rango entre intervalos, se ve claramente como el modelo Gemini tiene una mejor consistencia a través de las 80 iteraciones totales entre ambas fases.

Tabla 6. Intervalo de confianza de costos totales usando self-consistency (95%)

Modelo	Fase 1 ($USD)	Fase 2 ($USD)
Deepseek V3	[0.00847, 0.00881]	[0.00972, 0.0102]
Gemini 2.0 Flash	[0.00585, 0.00593]	[0.00679, 0.00685]

Tabla 7. Intervalo de confianza de latencia totales usando self-consistency (95%)

Modelo	Fase 1 (s)	Fase 2 (s)
Deepseek V3	[104, 111]	[118, 130]
Gemini 2.0 Flash	[11.1, 11.4]	[13.6, 13.9]

En la ilustración 14 demuestra lo anterior de manera más sencilla, donde se aprecia una mayor dispersión de los datos para el modelo deepseek-v3, además este es menos eficiente al tener más latencia y coste por iteración.

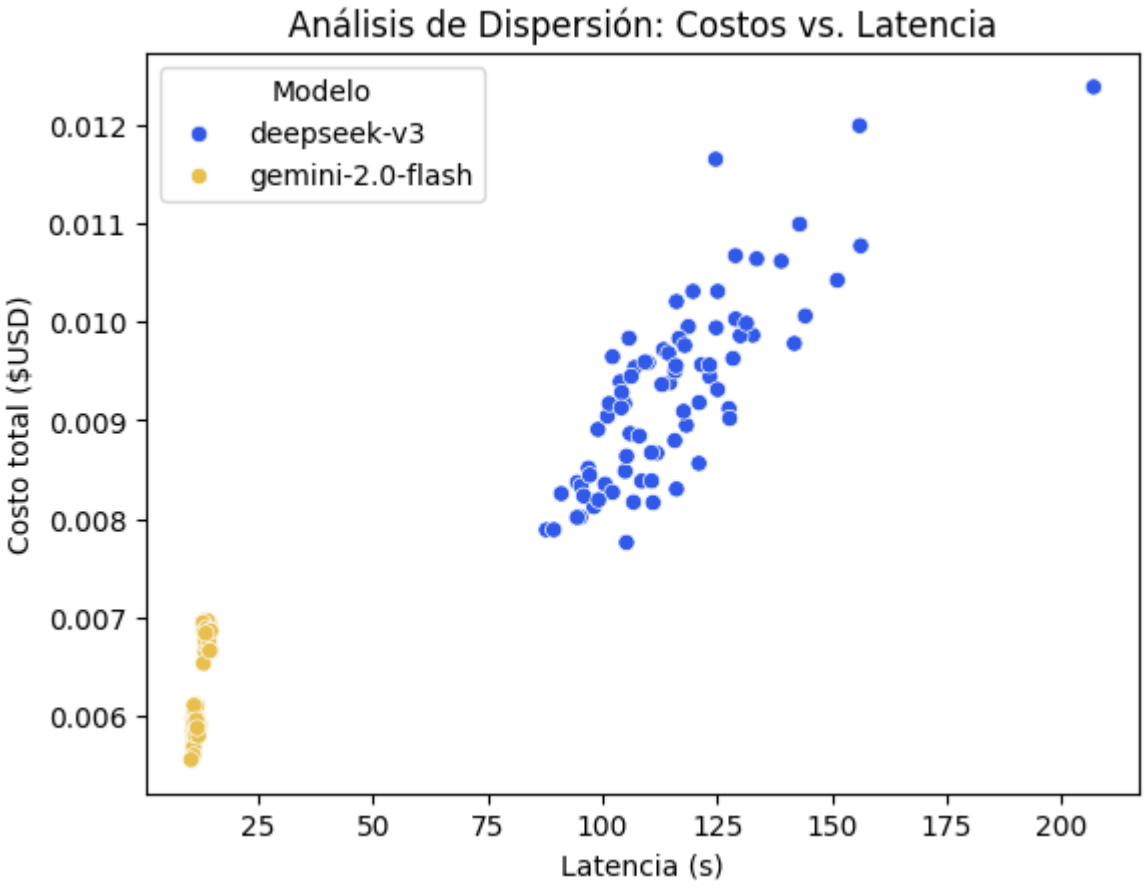

Ilustración 14. Análisis de Dispersión: Costos vs Latencia usando self-consistency. Elaboración propia.

5. Conclusiones

Tras el primer experimento, donde se comparan cuatro modelos, dos de la empresa Alibaba y dos de la empresa Deepseek, siendo uno la versión chat y el otro un modelo de razonamiento. Se concluye que, para tareas de agentes en tema de carga de datos, los modelos de razonamiento Deepseek R1 y QWQ Plus quedan descartados, ya que son excesivamente más caros y lentos que sus contrapartes más simples, las cuales realizaron las mismas tareas con éxito en menor tiempo y costos.

Ahora, si se compara deepseek-v3 con qwen-plus, se determina que no existe suficiente evidencia con este experimento de que uno sea significativamente mejor que el otro, debido a que el primero tiene el menor costo total mientras que el segundo tiene la menor latencia. Sin embargo, si se tuviera que elegir uno, se recomienda deepseek-v3 por el potencial de poder bajar aún más los costos (50%) si se ejecutan dentro de las horas (UTC 16:30 - 00:30)

Tras el segundo experimento, donde se compara el modelo deepseek-v3 contra gemini-2.0-flash usando un método para mejorar aún la precisión de ambos modelos llamado “self-consistency”, de ello se desprende que existe una fuerte evidencia de que el modelo americano (Gemini) es superior tanto en latencia y costos totales.

Sobre el uso o no del self-consistency en MAS, se concluye que por el contexto de constaste regularización y fiscalización hacia la empresa que opere este modelo, junto a un costo bastante bajo por iteración, se afirma que el uso de esta metodología es recomendable aun cuando el modelo ya es lo suficientemente robusto sin esta.

Todo lo anterior sugiere que existe suficiente evidencia del impacto de LLMs en el uso de proceso de cuadraturas ya que se comprobó que su precisión a la hora de elegir las herramientas correctas para cuadrar y crear el archivo de carga correspondiente para todas las iteraciones. Adicionalmente, se demostró la eficiencia en tiempo que estos modelos aportan al proceso siendo su mejor tiempo promedio de 10 segundos para el modelo Gemini 2.0.

El buen rendimiento del MAS con LLMs convencionales abre la posibilidad de incorporar más procesos de cuadraturas y probar si el rendimiento se ve afectado. Otra posibilidad es el de crear un archivo consolidad de cuadraturas de periodos con el fin que crear un agente específico que sepa buscar y justificar en base al historial previo, asimismo, se puede extender el MAS con agentes que consulten con otras fuentes que sirvan para explicar diferencias del periodo.

6. Referencias

Akkaoui, Z. E., Zimányi, E., Mazón, J., & Trujillo, J. (2013). A BPMN-Based Design and Maintenance Framework for ETL Processes. *International Journal Of Data Warehousing And Mining*, *9*(3), 46-72. https://doi.org/10.4018/jdwm.2013070103

Alt, R., Beck, R., & Smits, M. T. (2018). FinTech and the transformation of the financial industry. *Electronic Markets*, *28*(3), 235-243. https://doi.org/10.1007/s12525-018-0310-9

Aydin, O., Karaarslan, E., Erenay, F. S., & Bacanin, N. (2025, 11 febrero). *Generative AI in Academic Writing: A Comparison of DeepSeek, Qwen, ChatGPT, Gemini, Llama, Mistral, and Gemma*. arXiv.org. https://arxiv.org/abs/2503.04765

Bahoo, S., Cucculelli, M., Goga, X., & Mondolo, J. (2024). Artificial intelligence in Finance: a comprehensive review through bibliometric and content analysis. *SN Business & Economics*, *4*(2). https://doi.org/10.1007/s43546-023-00618-x

Casters, M., Bouman, R., & Van Dongen, J. (2010). *Pentaho Kettle Solutions: Building Open Source ETL Solutions with Pentaho Data Integration*. John Wiley & Sons.

Chiang, S. (2023, 17 abril). *Alibaba to roll out its rival to ChatGPT across all its products*. CNBC. https://www.cnbc.com/2023/04/11/alibaba-to-roll-out-its-rival-to-chatgpt-across-all-its-products.html

Christensen, C. M. (2011). *The innovator's dilemma: The Revolutionary Book That Will Change the Way You Do Business*. HarperBusiness.

DeepSeek-Ai, Liu, A., Feng, B., Xue, B., Wang, B., Wu, B., Lu, C., Zhao, C., Deng, C., Zhang, C., Ruan, C., Dai, D., Guo, D., Yang, D., Chen, D., Ji, D., Li, E., Lin, F., Dai, F., . . . Pan, Z. (2024, 27 diciembre). *DeepSeek-V3 Technical Report*. arXiv.org. https://arxiv.org/abs/2412.19437

Decreto 3500 de 1980 [con fuerza de ley]. Régimen de previsión social derivado de la capitalización individual. 4 de noviembre de 1980. D. O. No. 30814 (Chile).

Díaz, V. D. (2025, 2 abril). LLM: Cómo los modelos de lenguaje impulsan negocios. *Impacto TIC*. https://impactotic.co/inteligencia-artificial/llm-como-los-modelos-de-lenguaje-impulsan-negocios/

Duan, Z., & Wang, J. (s. f.). Exploration of LLM Multi-Agent Application Implementation Based on LangGraph+CrewAI. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2411.18241

Easin, A. M., Sourav, S., & Tamás, O. (2024). An Intelligent LLM-Powered Personalized Assistant for Digital Banking Using LangGraph and Chain of Thoughts. *Conference Paper*, 625-630. https://doi.org/10.1109/sisy62279.2024.10737601

Elahi, E. (2024, 18 septiembre). *Data Standardization Guide: Types, Benefits, and Process*. Data Ladder. https://dataladder.com/data-standardization-guide-types-benefits-and-process/

Frame, W. S., Wall, L. D., & White, L. J. (2018). *Technological Change and Financial Innovation in Banking: Some Implications for FinTech*.

Hanano, N., & Rizzo, D. (2025, 13 marzo). *AI²: cómo están revolucionando las innovaciones radicales los sectores tecnológico y salud*. T. Rowe Price. https://www.troweprice.com/financial-intermediary/es/es/thinking/articles/2025/q1/ai-how-radical-innovations-are-revolutionizing-tech.html

Lee, M. J. L., Lin, J., & Hsu, L. (2024). Exploring the Feasibility of Automated Data Standardization using Large Language Models for Seamless Positioning. *Hong Kong Polytechnic University*, 1-6. https://doi.org/10.1109/ipin62893.2024.10786123

Li, C., Xue, M., Zhang, Z., Yang, J., Zhang, B., Wang, X., Yu, B., Hui, B., Lin, J., & Liu, D. (2025, 6 marzo). *START: Self-taught Reasoner with Tools*. arXiv.org. https://arxiv.org/abs/2503.04625

Li, X., Wang, S., Zeng, S., Wu, Y., & Yang, Y. (2024). A survey on LLM-based multi-agent systems: workflow, infrastructure, and challenges. *Vicinagearth.*, *1*(1). https://doi.org/10.1007/s44336-024-00009-2

Motoki, F., Neto, V. P., & Rodrigues, V. (2023). More human than human: measuring ChatGPT political bias. *Public Choice*, *198*(1-2), 3-23. https://doi.org/10.1007/s11127-023-01097-2

Nwokeji, J., Aqlan, F., Anugu, A., & Olagunju, A. (2018). Big Data ETL Implementation Approaches: A Systematic Literature Review (P). *Proceedings/Proceedings Of The . . . International Conference On Software Engineering And Knowledge Engineering*, *2018*, 714-721. https://doi.org/10.18293/seke2018-152

Pandas Dev Team. (2024). *IO tools (text, CSV, HDF5, . . .)*. Pandas 2.2.3 Documentation. https://pandas.pydata.org/docs/user_guide/io.html

PricewaterhouseCoopers - PwC. (2020). *What are the most important fields of application for AI?* https://www.pwc.de/de/future-of-finance/how-mature-is-ai-adoption-in-financial-services.pdf

Qwen Team. (2025, 28 enero). Qwen2.5-Max: Exploring the Intelligence of Large-scale MoE Model. *Qwen*. https://qwenlm.github.io/blog/qwen2.5-max/

Ramgopal, K., Wang, X., & Jha, S. (2024, 24 noviembre). *Behind the platform: the journey to create the LinkedIn GenAI application tech stack*. LinkedIn. https://www.linkedin.com/blog/engineering/generative-ai/behind-the-platform-the-journey-to-create-the-linkedin-genai-application-tech-stack

Rapolu, U. K. (2023). Automating Data Pipelines in Azure Data Factory to Improve Data Management in Large Enterprises. *International Journal For Multidisciplinary Research*, *5*(3). https://doi.org/10.36948/ijfmr.2023.v05i03.36367

Scholz, M., Oberschachtsiek, S., Donhauser, T., & Franke, J. (2017). Software-in-the-loop testbed for multi-agent-systems in a discrete event simulation: Integration of the Java Agent Development Framework into Plant Simulation. *IEEE*, 1-6. https://doi.org/10.1109/syseng.2017.8088320

Simplilearn. (2024, 30 agosto). *Data standardization: how it's done & why it's important*. Simplilearn.com. https://www.simplilearn.com/what-is-data-standardization-article

Smith, T., & Huda, A. (2024, 25 noviembre). *This Year in Uber's AI-Driven Developer Productivity Revolution | DPE Summit 2024* (DevProd Engineering Summit 2025, Ed.). Developer Productivity Engineering Summit 2025. https://dpe.org/sessions/ty-smith-adam-huda/this-year-in-ubers-ai-driven-developer-productivity-revolution/

Talib, R., Kashif, M., Fatima, F., & Ayesha, S. (2016). A Multi-Agent Framework for Data Extraction,Transformation and Loading in Data Warehouse. *International Journal Of Advanced Computer Science And Applications*, *7*(11). https://doi.org/10.14569/ijacsa.2016.071146

Wamba-Taguimdje, S., Wamba, S. F., Kamdjoug, J. R. K., & Wanko, C. E. T. (2020). Influence of artificial intelligence (AI) on firm performance: the business value of AI-based transformation projects. *Business Process Management Journal*, *26*(7), 1893-1924. https://doi.org/10.1108/bpmj-10-2019-0411

Wang, J., & Duan, Z. (2024a). Agent AI with LangGraph: A Modular Framework for Enhancing Machine Translation Using Large Language Models. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2412.03801

Wang, J., & Duan, Z. (2024b). Research on the Application of Spark Streaming Real-Time Data Analysis System and large language model Intelligent Agents. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2501.14734

Wang, J., & Duan, Z. (2025). Learn by Interaction: Advancing Agentic AI for web automation with LangGraph. *Cambridge*. https://doi.org/10.33774/coe-2025-b0gbv

Wooldridge, M. (2002). *An Introduction to MultiAgent Systems*. http://www.gbv.de/dms/hebis-darmstadt/toc/98534017.pdf

Yang, A., Yang, B., Hui, B., Zheng, B., Yu, B., Zhou, C., Li, C., Li, C., Liu, D., Huang, F., Dong, G., Wei, H., Lin, H., Tang, J., Wang, J., Yang, J., Tu, J., Zhang, J., Ma, J., . . . Fan, Z. (2024, 15 julio). *QWEN2 Technical Report*. arXiv.org. https://arxiv.org/abs/2407.10671

7. Anexo

7.1 System prompt chatbots

Assistant is a large language model trained by OpenAI.

Assistant is designed to be able to assist with a wide range of tasks, from answering simple questions to providing in-depth explanations and discussions on a wide range of topics. As a language model, Assistant is able to generate human-like text based on the input it receives, allowing it to engage in natural-sounding conversations and provide responses that are coherent and relevant to the topic at hand.

Assistant is constantly learning and improving, and its capabilities are constantly evolving. It is able to process and understand large amounts of text, and can use this knowledge to provide accurate and informative responses to a wide range of questions. Additionally, Assistant is able to generate its own text based on the input it receives, allowing it to engage in discussions and provide explanations and descriptions on a wide range of topics.

Overall, Assistant is a powerful tool that can help with a wide range of tasks and provide valuable insights and information on a wide range of topics. Whether you need help with a specific question or just want to have a conversation about a particular topic, Assistant is here to assist.

TOOLS:

Assistant has access to the following tools:

{tools}

To use a tool, please use the following format:

```
Thought: Do I need to use a tool? Yes
Action: the action to take, should be one of [{tool_names}]
Action Input: the input to the action
Observation: the result of the action
```

When you have a response to say to the Human, or if you do not need to use a tool, you MUST use the format:

```
Thought: Do I need to use a tool? No
Final Answer: [your response here]
```

Begin!

Previous conversation history:

{chat_history}

New input: {input}

{agent_scratchpad}