TRABAJO PRÁCTICO FINAL

PROCESAMIENTO DE LENGUAJE NATURAL

Integrantes: Cena Lautaro

Fecha limite: 7/12

Año: 2025

Introducción

El presente trabajo desarrolla un sistema completo de Procesamiento de Lenguaje Natural destinado a responder consultas complejas sobre productos, inventario, ventas y devoluciones dentro de un entorno empresarial simulado. La solución combina técnicas de recuperación de información, representación semántica, razonamiento autónomo y análisis de datos, integrando así los contenidos de las Unidades del programa, pero aplicados a un caso práctico concreto.

A lo largo del trabajo se construyeron distintos módulos: una base vectorial para documentos técnicos y reseñas, un conjunto de consultas dinámicas sobre tablas de productos y ventas, un grafo Neo4j que representa relaciones reales entre categorías, subcategorías, productos y motivos de devolución, y una herramienta analítica capaz de generar gráficos directamente desde los datos de ventas. Sobre estas fuentes se entrenó un clasificador de intención que permite determinar qué tipo de operación requiere cada consulta, y finalmente se diseñó un agente con razonamiento ReAct capaz de seleccionar herramientas, ejecutar búsquedas, combinar observaciones y emitir respuestas justificadas en el idioma del usuario.

El objetivo central es mostrar la integración de múltiples técnicas en un sistema que no solo responde preguntas, sino que **razona**, **elige estrategias**, y **fundamenta sus respuestas** apoyándose exclusivamente en datos reales. Este enfoque refleja los principios de los sistemas de diálogo basados en recuperación (Unidad 6, apartado sobre “Arquitecturas híbridas”) y de los agentes autónomos guiados por herramientas (Unidad 7, sección “Agentes basados en ReAct y planificación por pasos”).

Arquitectura general del sistema

El sistema se apoya en cuatro fuentes de información que cubren distintos tipos de consultas y permiten seleccionar dinámicamente la estrategia de recuperación más adecuada. La organización por fuentes facilita el razonamiento del agente y mantiene la coherencia con las limitaciones establecidas en el trabajo práctico, especialmente la prohibición de enviar bases completas al modelo.

La primera fuente corresponde a documentos textuales: manuales técnicos, FAQs y reseñas de productos. Estos textos fueron segmentados en fragmentos y convertidos a embeddings en español, almacenados en FAISS. La recuperación combina similitud vectorial y un componente tipo BM25 para capturar coincidencias léxicas relevantes. Esta arquitectura, recomendada en el apartado “Recuperación híbrida” de la Unidad 6, mejora la cobertura ante consultas ambiguas o descriptivas.

La segunda fuente se basa en datos tabulares provenientes de distintos DataFrames: productos, inventario, ventas y devoluciones. En lugar de incorporar la tabla completa al contexto (algo desaconsejado en los criterios de diseño del material teórico) se implementó un

generador de filtros en lenguaje natural que produce expresiones seguras para Pandas. Esto permite responder preguntas como filtrados por categoría, rangos de precio o disponibilidad sin exponer toda la estructura tabular al modelo. Esta técnica se basa en la sección sobre "Interacción controlada con bases estructuradas" de la Unidad 6.

La tercera fuente es un grafo Neo4j construido con nodos de Producto, Categoría, Subcategoría, Marca y MotivoDevolución. El modelo no genera Cypher directamente, sino que primero identifica parámetros válidos y luego se construye la consulta final únicamente con nodos y relaciones realmente existentes. Este mecanismo evita alucinaciones y está alineado con los lineamientos sobre agentes que operan sobre bases conectadas (Unidad 7, segmento "Consultas guiadas por restricciones del entorno").

Por último, se incorporó una herramienta analítica que opera sobre el DataFrame de ventas. Esta función permite generar agregaciones y gráficos utilizando matplotlib. El ejemplo más relevante es la distribución de los métodos de pago, que el agente puede producir cuando detecta que la consulta requiere un análisis agregado en lugar de una recuperación puntual. La función sigue el principio de "herramientas como extensiones del agente" desarrollado en la Unidad 7, sección dedicada a la integración de capacidades externas.

Esta arquitectura multifuente permite que el sistema responda tanto preguntas semánticas como estructuradas o analíticas, combinando recuperación, razonamiento y visualización de forma controlada y trazable.

Clasificador de intención

El sistema incorpora un componente de clasificación de intención cuyo propósito es determinar qué tipo de operación requiere cada consulta del usuario: recuperación documental, filtrado tabular, consulta al grafo o análisis estadístico. Esta etapa funciona como un mecanismo de enrutamiento previo que optimiza el uso de recursos y evita que el sistema aplique herramientas inadecuadas para el tipo de información solicitada.

Para construir el clasificador se elaboró un conjunto de ejemplos representativos de los distintos tipos de consultas. Con este material se entrenó primero un modelo supervisado tradicional basado en TF-IDF, capaz de capturar patrones léxicos asociados a cada categoría de intención. Su desempeño fue correcto para consultas directas, aunque mostraba limitaciones ante preguntas formuladas en lenguaje natural más ambiguo o con variaciones semánticas. A fin de resolver estos casos, se incorporó un clasificador asistido por un modelo de lenguaje, el cual demostró una mayor capacidad para interpretar la intención implícita en consultas abiertas. Por este motivo, el sistema adoptó al clasificador basado en LLM como mecanismo principal y dejó al modelo tradicional como referencia comparativa.

El clasificador distingue cuatro tipos de intención:
1- documental, cuando la respuesta proviene de manuales o textos técnicos;

2- tabular, cuando la consulta involucra atributos estructurados de productos o filtros numéricos.
3- grafo, cuando se consulta por relaciones entre categorías, marcas o motivos de devolución.
4- analítica, cuando el usuario solicita cálculos agregados o la generación de gráficos.

Este enfoque es coherente con los lineamientos de la Unidad 6, sección “Detección de intención y estrategias de respuesta en sistemas de diálogo”, donde se establece la importancia de identificar la intención del usuario antes de seleccionar el módulo de recuperación adecuado. Asimismo, el uso del LLM como clasificador se fundamenta en el apartado de la misma unidad dedicado al “aprovechamiento de modelos de lenguaje en tareas de clasificación contextual”.

Diseño del sistema RAG multifuente

El sistema de recuperación se diseñó siguiendo un enfoque RAG (Retrieval-Augmented Generation) adaptado a un entorno con múltiples fuentes de datos. El modelo no responde utilizando conocimiento interno, sino a partir de información almacenada en documentos, tablas, grafos y registros de ventas. Esta metodología permite respuestas verificables y evita la generación de contenido no respaldado por datos.

La primera fuente corresponde a documentos textuales, como manuales, reseñas y material técnico. Los documentos fueron segmentados y convertidos en embeddings en español; luego se indexaron mediante FAISS. La búsqueda se complementó con un módulo BM25 para capturar coincidencias léxicas y con un reranqueador asistido por un modelo de lenguaje para priorizar los fragmentos más relevantes. Este diseño sigue los principios desarrollados en la Unidad 6, apartado “Recuperación híbrida: combinación de métodos densos y dispersos”, así como en la sección dedicada al “Reranking asistido por modelos de lenguaje”.

La segunda fuente está compuesta por datos tabulares provenientes de diversos DataFrames. Para evitar enviar las tablas completas al modelo (algo desaconsejado en la sección “Limitaciones de contexto y riesgos de sobreexposición de datos” de la Unidad 6) se implementó un generador de filtros seguros que transforma consultas en lenguaje natural en expresiones válidas de Pandas. Esta estrategia permite resolver filtrados, comparaciones y búsquedas estructuradas conservando la integridad de los datos.

La tercera fuente es un grafo Neo4j, que representa relaciones entre productos, categorías, subcategorías, marcas y motivos de devolución. Para mantener la coherencia estructural del grafo, el modelo no genera Cypher libremente: primero selecciona parámetros válidos a partir de opciones extraídas de la base y recién después se construye una consulta segura. Este diseño está alineado con los lineamientos de la Unidad 7, capítulo “Agentes que interactúan

con entornos simbólicos y consultas restringidas", donde se remarca la importancia de limitar la acción del agente a entidades verificables para evitar errores o alucinaciones.

Por último, se incorporó una herramienta de análisis estadístico, que opera sobre el DataFrame de ventas para generar agregaciones y visualizaciones mediante matplotlib. Este recurso extiende el alcance del sistema y permite atender consultas que requieren un tratamiento cuantitativo. Se basa en los criterios desarrollados en la Unidad 7, sección "Herramientas externas como capacidades extendidas del agente", donde se describe cómo los agentes pueden delegar tareas específicas en módulos especializados.

En conjunto, estas cuatro vías de recuperación permiten un sistema RAG robusto, preciso y transparente, capaz de seleccionar dinámicamente la estrategia adecuada según la intención del usuario, tal como lo plantean las secciones avanzadas de integración de fuentes en las Unidades 6 y 7.

Agente ReAct

Una vez desarrollados los módulos de recuperación y el clasificador de intención, el paso siguiente fue integrar todas las capacidades en un agente capaz de razonar y actuar de manera secuencial. Para ello se adoptó el enfoque ReAct, que combina razonamiento en lenguaje natural con la ejecución de herramientas especializadas. Este método permite que el agente decida qué información necesita, seleccione la herramienta adecuada, interprete la observación resultante y formule una respuesta final fundamentada. El procedimiento sigue el esquema conceptual presentado en la Unidad 7, en el capítulo dedicado a agentes basados en razonamiento y acción.

El diseño del agente parte de un conjunto de herramientas, cada una asociada a una fuente de datos distinta: recuperación documental, consultas tabulares, consultas al grafo y análisis estadístico. Para guiarlas se redactó un mensaje de sistema que explica el propósito de cada herramienta, cuándo corresponde utilizarlas y qué tipo de razonamiento debe aplicar el agente antes de ejecutar una acción. Este mensaje de sistema incorpora las reglas centrales del enfoque ReAct, que exigen producir un razonamiento previo en lenguaje natural antes de invocar cualquier herramienta, utilizar únicamente información presente en la observación y responder de forma clara y coherente con la consulta original.

El ciclo completo del agente consta de cuatro pasos. En primer lugar, el agente examina la consulta del usuario y reflexiona sobre qué información necesita y cuál herramienta resulta pertinente para obtenerla. En segundo lugar, ejecuta esa herramienta utilizando como entrada la misma consulta o una versión adaptada para la operación requerida. En tercer lugar, analiza la información devuelta por la herramienta y decide si ya cuenta con elementos suficientes para elaborar una respuesta o si debe realizar una búsqueda adicional. Finalmente, produce una

respuesta que integra el resultado de la observación y el razonamiento previo, manteniendo el idioma y el tono de la consulta inicial.

El uso de este enfoque resultó especialmente adecuado para integrar un sistema multifuente. En preguntas centradas en descripciones o especificaciones técnicas, el agente recurrió al módulo documental. Cuando la consulta requería comparar atributos de productos, identificar rangos de precios o filtrar por categoría, se utilizaron las capacidades tabulares. Las preguntas basadas en relaciones entre objetos, como productos con ciertos motivos de devolución o categorías vinculadas a subcategorías, fueron derivadas al grafo. Por último, cuando la consulta requería análisis cuantitativo, como en el caso de la distribución de métodos de pago, el agente seleccionó la herramienta analítica y produjo tanto una salida textual como un gráfico.

Para evaluar el funcionamiento del agente se realizaron ejecuciones de prueba sobre distintos tipos de consultas. Un ejemplo ilustrativo es el caso en que el usuario solicita un análisis de ventas. El agente identifica la naturaleza estadística del pedido, razona que la herramienta de análisis es la adecuada, la ejecuta y obtiene un resumen completo de los métodos de pago junto con un gráfico en formato de imagen. La explicación final al usuario se basa exclusivamente en esa observación. Ejemplos similares se verificaron para consultas documentales, tabulares y de grafo, confirmando que el agente selecciona and actúa de forma coherente con la intención inferida.

El diseño final del agente refleja los principios desarrollados en la Unidad 7 sobre agentes basados en herramientas y razonamiento escalonado. En particular, se ajusta a la estructura conceptual presentada en la sección dedicada al método ReAct, donde se explica que la inteligencia del agente no reside en poseer información interna, sino en su capacidad para decidir qué herramienta utilizar y cómo interpretar los resultados para producir una respuesta fundamentada y verificable.

Resultados del sistema conversacional

El sistema conversacional obtenido permite responder una amplia variedad de consultas combinando razonamiento, recuperación de información y análisis de datos. En las pruebas realizadas se observó que el agente no solo logra seleccionar correctamente la herramienta más adecuada para cada consulta, sino que también explica de manera clara el razonamiento detrás de esa elección y fundamenta sus respuestas en observaciones provenientes de datos reales.

Las consultas de tipo documental fueron respondidas mediante fragmentos relevantes de manuales técnicos y reseñas, seleccionados a partir de un proceso híbrido de recuperación y reranking. En las consultas tabulares, el agente produjo filtros válidos que permitieron obtener subconjuntos coherentes de productos o ventas, siempre manteniendo la seguridad y sin exponer las tablas completas. En el caso del grafo, el agente generó consultas estructuradas que utilizaron únicamente nodos y relaciones válidas, lo que evitó la aparición de respuestas

inventadas o inconsistentes. Para los análisis estadísticos, el sistema generó gráficos reales y devolvió resúmenes numéricos precisos, apoyándose exclusivamente en el conjunto de datos de ventas.

Los resultados muestran que el enfoque ReAct permite un control adecuado de cada paso del razonamiento. El agente identifica la intención, elige correctamente la herramienta, ejecuta la acción y analiza la observación resultante antes de elaborar la respuesta. Este proceso coincide con el esquema teórico de la Unidad 7, en la sección dedicada a la planificación por pasos y la toma de decisiones asistida por herramientas externas. Además, el uso de observaciones verificables garantiza que las respuestas finales se mantengan alineadas con los datos del entorno y evita que el modelo genere contenido no sustentado.

En conjunto, el sistema demuestra un funcionamiento estable, transparente y adaptable a distintos tipos de consultas. La separación entre fuentes de información, el clasificador de intención y el agente basado en ReAct permiten una interacción conversacional confiable y suficientemente flexible para responder tanto preguntas descriptivas como estructuradas y analíticas.

Conclusiones

El trabajo permitió integrar distintos contenidos de Procesamiento de Lenguaje Natural en un sistema coherente y funcional, capaz de responder consultas complejas mediante la combinación de razonamiento, herramientas especializadas y recuperación de información. La construcción de un entorno multifuente, junto con la implementación de un clasificador de intención y un agente basado en el enfoque ReAct, mostró cómo los modelos de lenguaje pueden utilizarse de manera controlada para interactuar con datos reales sin perder trazabilidad ni precisión.

La arquitectura desarrollada evitó que el modelo dependiera de conocimientos internos o supuestos implícitos. En su lugar, cada respuesta se apoyó en documentos, tablas, relaciones del grafo o información estadística verificable. Este comportamiento estuvo alineado con las recomendaciones teóricas de las Unidades 6 y 7, donde se enfatiza que los sistemas de diálogo basados en datos deben seleccionar estrategias de recuperación específicas, aplicar razonamiento paso a paso y fundamentar sus respuestas en observaciones transparentes.

La experiencia de desarrollo mostró además que los agentes basados en herramientas introducen un grado significativo de control sobre el proceso conversacional. El agente no solo eligió correctamente qué herramienta ejecutar ante cada pregunta, sino que también explicó su razonamiento y justificó la decisión final, lo que resulta fundamental para evaluar el comportamiento del sistema. La incorporación de una herramienta analítica permitió además extender el alcance del sistema hacia tareas cuantitativas que no pueden resolverse mediante recuperación tradicional, como la generación de gráficos o resúmenes estadísticos.

En conjunto, el proyecto evidencia la utilidad del paradigma ReAct para integrar múltiples fuentes de información dentro de un mismo flujo conversacional, manteniendo transparencia, rigor y adaptabilidad frente a diversos tipos de consultas.