

Facultad de
Ciencias Exactas,
Ingeniería y Agrimensura



Trabajo práctico NLP

2° Semestre - Año 2025

Nicolás Mancini

Introducción

El presente informe documenta el diseño, desarrollo y funcionamiento de un asistente virtual especializado en información de una empresa de electrodomésticos. A lo largo del documento, se describen las herramientas y funciones implementadas que nos permiten ofrecer respuestas precisas y contextualizadas a una amplia variedad de consultas de usuario.

El informe se encuentra dividido en dos ejercicios, por un lado el ejercicio uno que corresponde a un RAG y posteriormente su evolución hacia un Agente Autónomo en el ejercicio dos.

Ejercicio 1: RAG

Fuentes de datos y búsquedas

Como parte del diseño del asistente virtual, resulta fundamental definir cómo se organiza y recupera la información para responder de manera eficiente a las distintas consultas de los usuarios. En este contexto, la arquitectura del sistema se apoya en la idea de que es más eficiente recurrir a la fuente de datos más apta de acuerdo al tipo de pregunta, que utilizar siempre la misma o todas. Esta estrategia permite resolver un amplio rango de preguntas, desde consultas descriptivas hasta análisis complejos de relaciones o métricas numéricas.

Por esta razón, se decide crear tres bases de datos para almacenar de forma independiente información semántica (no estructurada), datos estructurados precalculados y relaciones explícitas entre entidades. Además, Junto con cada fuente se crea una interfaz adecuada que nos permite consultarla.

Base Vectorial: Almacena los siguientes documentos dentro de una única colección unificada:

- resenas_usuarios (carpeta): contiene múltiples archivos de texto con reseñas escritas por usuarios, que reflejan experiencias, opiniones y percepciones sobre los productos.
- manuales_productos (carpeta): agrupa manuales técnicos de productos en formato Markdown, con descripciones, especificaciones, instrucciones de uso y recomendaciones.
- faqs.json: almacena preguntas frecuentes asociadas a productos específicos, junto con sus respuestas, categoría temática, métricas de uso y fechas de publicación.

Si bien, cada uno de estos formatos presenta características diferentes en cuanto a su estructura y densidad de información. Al transformarlos en embeddings, todos pasan a representarse como vectores en un espacio semántico común.

Fragmentación de documentos

Antes de realizar los embeddings, se procede a fragmentar los documentos con el método `chunk_text_recursive`. La elección de este enfoque responde a la necesidad de preservar la coherencia semántica de cada fragmento y reducir el ruido en las búsquedas semánticas. A diferencia de otras técnicas de segmentación, el método recursivo analiza la estructura interna del texto y prioriza cortes en puntos lógicos como encabezados, párrafos o secciones. Esto resulta especialmente importante en estos tipos de documentos, donde una sección completa suele desarrollar una idea específica. De esta forma, evitamos correr el riesgo de generar chunks incompletos o ambiguos que luego degradaría la precisión de las respuestas a la hora de recuperar la información.

Luego, se utiliza `multilingual-e5-small` para la representación semántica de los fragmentos obtenidos. Este modelo de embedding, fue mantenido a lo largo de todo el TP, al ofrecer un equilibrio entre calidad semántica y eficiencia, al estar optimizado para capturar la similitud semántica entre consultas y documentos. Además, es un modelo con soporte multilingüe, que es capaz de representar textos en distintos idiomas dentro de un mismo espacio vectorial, que si bien no es el caso, agrega robustez a nuestro trabajo final.

Luego, se crea la función `busqueda_vectorial` que funciona como interfaz para realizar consultas, algunos ejemplos son:

```
busqueda_vectorial("Cómo reiniciar el dispositivo")

[{'texto': '. Si el problema persiste, contactar servicio técnico --- #',
  'metadata': {'source_type': 'manual',
               'filename': 'manual_P0017_Profesional_Batidora_de_Mano.md',
               'full_path': '/content/manuales_productos/manual_P0017_Profesional_Batidora_de_Mano.md',
               'chunk_id': 11},
  'distancia': 0.21313530206680298}]
```

```

busqueda_vectorial("¿Qué opinan los usuarios de esta cafetera?")

[{'texto': 'Reseña sobre Cafetera (ID P0122) Puntaje: 4/5 Opinión:',
  'metadata': {'producto_id': 'P0122',
    'nombre_producto': 'Cafetera',
    'puntaje': 4,
    'filename': 'resena_R01633.txt',
    'provincia': 'Chaco',
    'usuario': 'Agustina_Flores',
    'fecha': '2025-10-03',
    'source_type': 'resena'},
  'distancia': 0.2095051109790802}]

```

```

busqueda_vectorial("¿Qué voltaje requiere el rallador digital eléctrico?", k=3)

[{'texto': 'El Digital Rallador Eléctrico funciona con 220V. El consumo es de 1300W. Recomendamos usar un estabilizador de tensión.',
  'metadata': {'filename': 'faqs.json',
    'respuesta': 'El Digital Rallador Eléctrico funciona con 220V. El consumo es de 1300W. Recomendamos usar un estabilizador de tensión.',
    'id_faq': 'FAQ00271',
    'categoria': 'Especificaciones',
    'nombre_producto': 'Digital Rallador Eléctrico',
    'vistas': 697,
    'fecha_publicacion': '2025-06-05',
    'pregunta': '¿Qué voltaje requiere?',
    'util': 27,
    'id_producto': 'P0028',
    'source_type': 'faqs'},
  'distancia': 0.11572200059890747},

```

```

[{'texto': 'El Eco Rallador Eléctrico funciona con 110-220V. El consumo es de 1050W. Recomendamos usar un estabilizador de tensión.',
  'metadata': {'categoria': 'Especificaciones',
    'filename': 'faqs.json',
    'id_producto': 'P0030',
    'nombre_producto': 'Eco Rallador Eléctrico',
    'vistas': 1155,
    'fecha_publicacion': '2025-02-18',
    'util': 64,
    'id_faq': 'FAQ00291',
    'respuesta': 'El Eco Rallador Eléctrico funciona con 110-220V. El consumo es de 1050W. Recomendamos usar un estabilizador de tensión.',
    'pregunta': '¿Qué voltaje requiere?',
    'source_type': 'faqs'},
  'distancia': 0.1867533028125763},

```

```

[{'texto': 'El Digital Rallador Eléctrico de TechHome está diseñado para uso doméstico. Revise el manual del producto.',
  'metadata': {'nombre_producto': 'Digital Rallador Eléctrico',
    'id_faq': 'FAQ00275',
    'vistas': 4648,
    'categoria': 'Problemas Comunes',
    'filename': 'faqs.json',
    'respuesta': 'El Digital Rallador Eléctrico de TechHome está diseñado para uso doméstico. Revise el manual del producto.',
    'id_producto': 'P0028',
    'pregunta': '¿Por qué no funciona correctamente?',
    'source_type': 'faqs',
    'util': 64,
    'fecha_publicacion': '2025-09-05'},
  'distancia': 0.20753705501556396}]

```

Luego de la primera creación de la fuente de datos, nos quedan los siguientes archivos de datos a disponer:

- devoluciones.csv: contiene el registro detallado de devoluciones realizadas, incluyendo información de la venta original, el producto, el cliente, los motivos de la devolución, los montos involucrados y el estado del reembolso.
- inventario_sucursales.csv: registra el estado del inventario por sucursal, incluyendo stock disponible, niveles mínimos y máximos, precios locales, ubicación física y datos del proveedor.

- productos.csv: contiene la información maestra de los productos, con atributos comerciales y técnicos como categoría, marca, precio, stock, especificaciones y descripción general
- tickets_soporte.csv: almacena los tickets de soporte técnico asociados a ventas y productos, incluyendo el tipo de problema, severidad, estado de resolución y validez de la garantía.
- vendedores.csv: contiene los datos del personal de ventas, incluyendo información de contacto, sucursal asignada, nivel, comisión y estado de actividad.
- ventas_historicas.csv: registra el historial de ventas realizadas, con información temporal, productos vendidos, vendedores involucrados, cantidades, precios, descuentos y datos del cliente.

Estos archivos pueden ir tanto para una base de datos tabular como para grafos, por lo cual, se decidió repetir la utilización de algunos archivos en las dos bases restantes, pero almacenando distinta información, con diferentes finalidades.

Base de Datos Tabular: es un Data Frame que contiene información previamente procesada y calculada de los productos, como el total de devoluciones, su stock, la cantidad de sucursales en la cual se encuentra, entre otras métricas agregadas y estadísticas. El objetivo principal de esta base es responder consultas que requieren operaciones estructuradas, como filtros, comparaciones, cálculos aritméticos o análisis estadísticos.

Se crea la función `busqueda_tabular` que funciona como interfaz para realizar consultas, algunos ejemplos son:

```
busqueda_tabular(df_maestro, "Dime el nombre de un producto que valga 283.63 dólares")
```

nombre

0 Licuadora

```
busqueda_tabular(df_maestro, "Dime nombre del producto más caro")
```

nombre

90 Advanced Heladera

```
busqueda_tabular(df_maestro, "Dime las 3 marcas que los productos sean los más economicos")
```

	marca	mean
11	ScreenPro	293.990
13	TechHome	1265.068
16	WashPro	1295.670

```
busqueda_tabular(df_maestro, "Dime la marca más repetida en general")
```

	marca	count
4	CookElite	30

```
busqueda_tabular(df_maestro, "El nombre de los 5 productos con más tickets al soporte")
```

	nombre
72	Profesional Parrilla Eléctrica 3000
201	Advanced Neblinizador
197	Compacto Ventilador de Torre
153	Aire Portátil
114	Vinoteca Pro

Base de Datos de Grafos: La última fuente de información es una base de datos orientada a grafos implementada en Neo4j. En ella se representan entidades y relaciones explícitas obtenidas a partir de todos los archivos previamente ya mencionados que restaban ubicar.

Su uso permite modelar relaciones complejas de manera natural, que resultan más expresivas que las otras bases. Esta fuente se utiliza para responder consultas donde el contexto relacional es clave, como recomendaciones, dependencias o análisis de conexiones indirectas entre entidades.

Se crea la función `busqueda_grafos` que funciona como interfaz para realizar consultas, algunos ejemplos son:

```
busqueda_grafos("Mostrame los productos superiores a 100 dolaraes dentro de la categoria cocina")
```

	nombre	id_producto
0	licuadora	P0001
1	licuadora	P0002
2	plus licuadora pro	P0003
3	compacto licuadora	P0004
4	licuadora	P0005
...
136	elite deshidratador	P0143
137	plus deshidratador 2024	P0144
138	deshidratador	P0145
139	ultra deshidratador	P0146
140	turbo deshidratador	P0147

141 rows x 2 columns

```
busqueda_grafos("¿Qué productos están relacionados con la categoría Cocina?")
```

	nombre	id_producto
0	licuadora	P0001
1	licuadora	P0002
2	plus licuadora pro	P0003
3	compacto licuadora	P0004
4	licuadora	P0005
...
142	elite deshidratador	P0143
143	plus deshidratador 2024	P0144
144	deshidratador	P0145
145	ultra deshidratador	P0146
146	turbo deshidratador	P0147

147 rows x 2 columns

```
busqueda_grafos("Dime todos los productos vendidos por el vendedor Benjamín Suárez")
```

	nombre	id_producto
0	olla arrocera plus	P0081
1	profesional planchita de pelo plus	P0295
2	deluxe procesadora	P0010
3	profesional procesadora 2024	P0011
4	waflera	P0070
...
104	premium centrífuga	P0264
105	plancha seca 2024	P0285
106	tender eléctrico 3000	P0269
107	advanced ionizador plus	P0226
108	deluxe lavarropas automático 2024	P0229

109 rows x 2 columns

```
busqueda_grafos("Muestra las categorías hijas de Cocina")
```

	subcategoria
0	preparación
1	cocción
2	refrigeración
3	pequeños electrodomésticos

```
busqueda_grafos("Dime todas las cafeteras que tenemos disponibles")
```

	nombre	id_producto
0	deluxe cafetera 2024	P0120
1	premium cafetera	P0121
2	cafetera	P0122

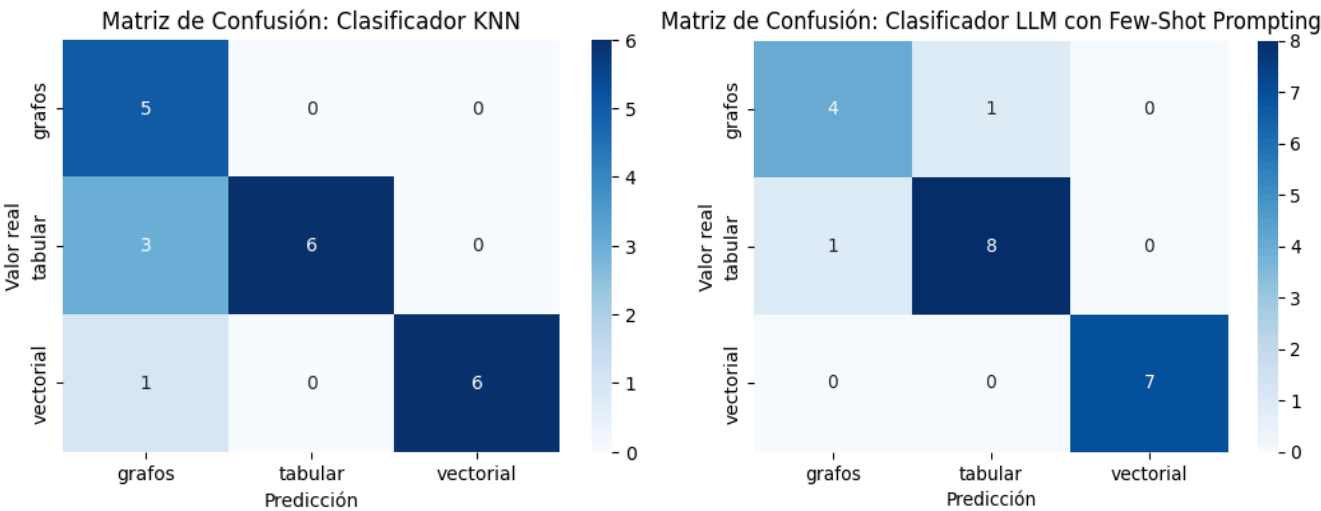
Clasificación de intención

Para determinar qué fuente de datos debe utilizarse en cada consulta, se debe obtener la intención, a continuación se comparan dos formas:

Clasificador KNN (K-Nearest Neighbors): Identifica la intención de una consulta comparando el embedding de la pregunta con ejemplos previamente etiquetados en el espacio vectorial.

LLM con Few-Shot Prompting: Identifica la intención de una consulta, a partir de unos pocos ejemplos proporcionados en el prompt, con los cuales puede inferir patrones y comprender el contexto de nuevas consultas sin necesidad de entrenamiento adicional. Es especialmente útil en situaciones de alta ambigüedad o cuando la intención del usuario depende de matices semánticos que los métodos basados únicamente en distancias vectoriales podrían pasar por alto.

Para decir el modelo se realiza una matriz de confusión y un reporte de sus métricas.



A partir de los gráficos se puede notar que la matriz de confusión del LLM muestra una diagonal principal bien definida, lo que indica una alta capacidad de discriminación, junto con una mejor precisión a la hora de predecir la clase grafos. En cambio, el modelo KNN, presenta una mayor dispersión de errores y no suele predecir correctamente la clase grafos ya que aproximadamente el 33 % de las consultas tabulares son clasificadas incorrectamente como grafos.

KNN

	precision	recall	f1-score	support
grafos	0.56	1.00	0.71	5
tabular	1.00	0.67	0.80	9
vectorial	1.00	0.86	0.92	7
accuracy			0.81	21
macro avg	0.85	0.84	0.81	21
weighted avg	0.89	0.81	0.82	21

LLM con Few-Shot Prompting

	precision	recall	f1-score	support
grafos	0.80	0.80	0.80	5
tabular	0.89	0.89	0.89	9
vectorial	1.00	1.00	1.00	7
accuracy			0.90	21
macro avg	0.90	0.90	0.90	21
weighted avg	0.90	0.90	0.90	21

Las observaciones anteriores se reflejan directamente en las métricas ya que mientras el LLM mantiene una consistencia superior con un F1-score promedio de (0.90), el clasificador KNN desciende a un (0.82), penalizado principalmente por su debilidad en la clase tabular (0.80) y la baja precisión en grafos, a pesar de su buen desempeño en la detección de consultas vectoriales (0.92).

Sin embargo, a pesar de la superioridad de desempeño en las métricas del LLM, se optó por utilizar el algoritmo KNN para clasificar las intenciones. Esta decisión se basa en minimizar el consumo de tokens, sacrificando el margen de robustez del LLM a costa de obtener un modelo más rápido, eficiente y escalable. Dado que el modelo de embeddings ya proporciona una representación semántica rica, KNN logra explotar directamente las distancias en el espacio vectorial sin necesidad de un procesamiento generativo costoso, manteniéndose como una opción altamente competitiva y confiable para la clasificación de intenciones en producción.

Pipeline de Recuperación (Retrieval)

Este proceso se encarga de recuperar la información de la forma adecuada según la intención de la pregunta del usuario, por ello si la intención detectada es para la base de datos tabular o de grafos, se deriva la consulta directamente a sus funciones de consulta sin ningún paso adicional.

Por otro lado, si la intención detectada es para la base de datos vectorial, se realiza un pipeline de recuperación que tiene como objetivo obtener la información más relevante para una consulta del usuario, combinando distintas estrategias de búsqueda y un proceso de reordenamiento semántico para maximizar la precisión de los resultados.

Búsqueda híbrida

En primer lugar, se construye un índice vectorial a partir de todos los documentos fragmentados previamente. Cada fragmento (chunk) se almacena junto con su metadata dentro de un índice que permite búsquedas eficientes.

Para la recuperación inicial se utilizan dos enfoques complementarios:

- Búsqueda vectorial: recupera fragmentos semánticamente similares a la consulta del usuario, comparando embeddings en el espacio vectorial.
- Búsqueda BM25: realiza una recuperación basada en coincidencias léxicas, priorizando términos exactos y su frecuencia dentro de los documentos.

Ambos resultados se combinan mediante un mecanismo de fusión de rankings (Reciprocal Rank Fusion, RRF), lo que permite aprovechar simultáneamente las ventajas de la similitud semántica y la coincidencia textual. De esta forma, se obtiene un conjunto inicial de candidatos más robusto que el generado por un único método de búsqueda.

Re-ranking semántico con Cross-Encoder

Una vez obtenidos los fragmentos candidatos mediante la búsqueda híbrida, se aplica una etapa de re-ranking utilizando un modelo Cross-Encoder basado en Sentence Transformers.

A diferencia de los modelos de embeddings, el Cross-Encoder evalúa directamente cada par (*consulta, fragmento*), asignando un puntaje de relevancia más preciso al considerar la interacción completa entre ambos textos. Este proceso nos permite refinar el orden de los resultados recuperados, con el objetivo de priorizar aquellos fragmentos que mejor respondan a la intención del usuario y penalizar aquellos poco relevantes, por más de que hayan sido recuperados en la etapa inicial. Finalmente, los

fragmentos se ordenan según su puntaje y se seleccionan los *top-k* más relevantes, que serán utilizados como contexto para el modelo de lenguaje.

A continuación vemos 3 ejemplos.

```
query_pipeline("Dime el nombre de los productos que pertenecen a la subcategoría de pequeños electrodomésticos", knn, mis_retrivers)

('GRAFOS',
 nombre id_producto
0 deluxe cafetera 2024 P0120
1 premium cafetera P0121
2 cafetera P0122
3 pava eléctrica P0123
4 pava eléctrica 2024 P0124
5 max pava eléctrica P0125
6 turbo pava eléctrica 2024 P0126
7 pava eléctrica P0127
8 advanced pava eléctrica P0128
9 exprimidor 2024 P0129
10 deluxe exprimidor x P0130
11 turbo exprimidor P0131
12 exprimidor P0132
13 exprimidor P0133
14 yogurtera P0134
15 profesional yogurtera P0135
16 smart yogurtera P0136
17 yogurtera P0137
18 advanced yogurtera P0138
19 deluxe yogurtera 3000 P0139
20 yogurtera P0140
21 advanced deshidratador P0141
22 profesional deshidratador P0142
23 elite deshidratador P0143
24 plus deshidratador 2024 P0144
25 deshidratador P0145
26 ultra deshidratador P0146
27 turbo deshidratador P0147)
```

```
query_pipeline("Cómo reiniciar el dispositivo", knn, mis_retrivers)

('VECTORIAL',
 [NodeWithScore(node=TextNode(id='b4e1215c-5a15-4041-bae5-94103a4cc6d1', embedding=None, metadata={'id_faq': 'FAQ00044', 'id_producto': 'P0005', 'nombre_producto': 'Licuadora', 'categoria': 'Uso', 'pregunta': '¿Cómo se usa correctamente este producto?', 'respuesta': 'El Licuadora de HomeChef está diseñado para uso doméstico. Revise el manual del producto (código P0005) para más detalles. Ante cualquier duda, contacte a nuestro servicio de atención al cliente.', 'fecha_publicacion': '2025-02-06', 'vistas': 1898, 'util': 11, 'filename': 'faqs.json', 'source_type': 'faqs'}, excluded_embed_metadata_keys=[], excluded_llm_metadata_keys=[], relationships={<NodeRelationship.SOURCE: '1': RelatedNodeInfo(node_id='FAQ00044', node_type='4', metadata={'id_faq': 'FAQ00044', 'id_producto': 'P0005', 'nombre_producto': 'Licuadora', 'categoria': 'Uso', 'pregunta': '¿Cómo se usa correctamente este producto?', 'respuesta': 'El Licuadora de HomeChef está diseñado para uso doméstico. Revise el manual del producto (código P0005) para más detalles. Ante cualquier duda, contacte a nuestro servicio de atención al cliente.', 'fecha_publicacion': '2025-02-06', 'vistas': 1898, 'util': 11, 'filename': 'faqs.json', 'source_type': 'faqs'}, hash='0228bf6ab107fbad99bbff29ced67917aa319bd1cb21a8a97e7b6670e01aaf8e')}), metadata_template='{key}: {value}', metadata_separator='\n', text='Pregunta: ¿Cómo se usa correctamente este producto?\nRespuesta: El Licuadora de HomeChef está diseñado para uso doméstico. Revise el manual del producto (código P0005) para más detalles. Ante cualquier duda, contacte a nuestro servicio de atención al cliente.', mimetype='text/plain', start_char_idx=0, end_char_idx=258, metadata_seperator='\n', text_template='{metadata_str}\n\n{content}'), score=-3.3486753),
 NodeWithScore(node=TextNode(id='857440a7-bb70-4794-88c2-54da138f834d', embedding=None, metadata={'id_faq': 'FAQ01391', 'id_producto': 'P0140', 'nombre_producto': 'Yogurtera', 'categoria': 'Uso', 'pregunta': '¿Cómo se usa correctamente este producto?', 'respuesta': 'El Yogurtera de ChefMaster está diseñado para uso doméstico. Revise el manual del producto (código P0140) para más detalles. Ante cualquier duda, contacte a nuestro servicio de atención al cliente.', 'fecha_publicacion': '2024-12-15', 'vistas': 1348, 'util': 5, 'filename': 'faqs.json', 'source_type': 'faqs'}, excluded_embed_metadata_keys=[], excluded_llm_metadata_keys=[], relationships={<NodeRelationship.SOURCE: '1': RelatedNodeInfo(node_id='FAQ01391', node_type='4', metadata={'id_faq': 'FAQ01391', 'id_producto': 'P0140', 'nombre_producto': 'Yogurtera', 'categoria': 'Uso', 'pregunta': '¿Cómo se usa correctamente este producto?', 'respuesta': 'El Yogurtera de ChefMaster está diseñado para uso doméstico. Revise el manual del producto (código P0140) para más detalles. Ante cualquier duda, contacte a nuestro servicio de atención al cliente.', 'fecha_publicacion': '2024-12-15', 'vistas': 1348, 'util': 5, 'filename': 'faqs.json', 'source_type': 'faqs'}, hash='a71b3f0a33727aa84e75bfde350db15fcd009d4145f0f584db0a8dab0eb6b4b3')}), metadata_template='{key}: {value}', metadata_separator='\n', text='Pregunta: ¿Cómo se usa correctamente este producto?\nRespuesta: El Yogurtera de ChefMaster está diseñado para uso doméstico. Revise el manual del producto (código P0140) para más detalles. Ante cualquier duda, contacte a nuestro servicio de atención al cliente.', mimetype='text/plain', start_char_idx=0, end_char_idx=260, metadata_seperator='\n', text_template='{metadata_str}\n\n{content}'), score=-3.3933167),
```

Modelo de Lenguaje (LLM)

Como modelo de lenguaje se seleccionó Gemini 2.0 Flash, ya que esta versión ofrece un equilibrio entre calidad de respuestas y rendimiento. La versión Flash está optimizada para ofrecer baja latencia y un alto número de consultas por segundo, lo que resulta adecuado para el enfoque del TP.

Durante el desarrollo se evaluaron otras alternativas, como modelos ejecutados localmente de Ollama o de hugging face. Sin embargo, Gemini mostró mejores resultados generales y la ventaja adicional de no requerir infraestructura local, ya que se ejecuta directamente en la nube. De todas formas, su principal desventaja es el límite de tokens disponible en el plan gratuito, lo cual restringe la longitud del contexto y la complejidad de algunas interacciones.

Generación y Conversación

Finalmente, se crea una interfaz que nos permite hacer preguntas y obtener respuestas, a partir de las funcionalidades anteriores. Además, se incorpora una memoria conversacional que almacena los últimos intercambios entre el usuario y el asistente, permitiendo que el modelo tenga en cuenta el contexto previo a la hora de responder.

Cada vez que el usuario envía una consulta:

1. La memoria genera un texto formateado que resume el historial reciente.
2. Se ejecuta el pipeline de clasificación de intención.
3. Según el tipo de consulta, se recupera información desde la base vectorial, tabular o de grafos.
4. La información recuperada se transforma en un formato comprensible para el LLM.
5. Se construye un prompt completo que combina:
 - Instrucciones del asistente.
 - Historial conversacional.
 - Información recuperada.
 - Mensaje actual del usuario.
6. El LLM devuelve la respuesta y luego la agrega a la memoria.

Este ciclo se repite en cada interacción, permitiendo mantener un ida y vuelta fluido, que aprovecha el contexto de la conversación, a continuación se pueden observar ejemplos:

Primera conversación.

```
Tú: Cuales son las licuadoras de menos de $200?
No tengo información sobre licuadoras de menos de $200. Por favor, reformula tu pregunta con más detalles.

Tú: Hay algún otro producto de menos de $200
Claro que sí. Aquí hay algunos productos de menos de $200:

* **Procesadora KitchenPro:** Categoría Cocina, subcategoría Preparación. Precio: $82.93. Descripción: Equipada con un motor de 1700W, bowl de 2.0L, 15 accesorios intercambiables y panel de control digital.
* **Mixer Pro ChefMaster:** Categoría Cocina, subcategoría Preparación. Precio: $52.7. Potencia de 1650W.
* **Abridor de Latas 2024 CookElite:** Categoría Cocina, subcategoría Preparación. Precio: $156.15. Potencia de 450W.
* **Premium Molinillo de Café HomeChef:** Categoría Cocina, subcategoría Preparación. Precio: $30.02. Potencia de 800W.
* **Compacto Horno Eléctrico TechHome:** Categoría Cocina, subcategoría Cocción. Precio: $97.66. Potencia de 1150W.
* **Pro Sandwichera HomeChef:** Categoría Cocina, subcategoría Cocción. Precio: $62.63. Potencia de 1350W.
* **Plus Panelita Eléctrica KitchenPro:** Categoría Cocina, subcategoría Cocción. Precio: $122.09. Potencia de 1600W.
* **Profesional Olla Arrocera ChefMaster:** Categoría Cocina, subcategoría Cocción. Precio: $144.68. Capacidad de 3.8L y 250W de potencia.
* **Pro Olla Arrocera TechHome:** Categoría Cocina, subcategoría Cocción. Precio: $134.16. Capacidad de 1.5L y 1750W de potencia.
* **Compacto Heladera CookElite:** Categoría Cocina, subcategoría Refrigeración. Precio: $161.81. Capacidad de 550L y 500W de potencia.
* **Cafetera ChefMaster:** Categoría Cocina, subcategoría Pequeños Electrodomésticos. Precio: $173.03. Capacidad de 1.0L y 450W de potencia.
* **Pava Eléctrica 2024 HomeChef:** Categoría Cocina, subcategoría Pequeños Electrodomésticos. Precio: $47.86. Potencia de 900W.
* **Profesional Yogurtera CookElite:** Categoría Cocina, subcategoría Pequeños Electrodomésticos. Precio: $141.44. Potencia de 1250W.
* **Panel Calefactor EcoClima:** Categoría Climatización, subcategoría Calefacción. Precio: $177.18. Potencia de 1350W.
* **Elite Humidificador EcoClima:** Categoría Climatización, subcategoría Purificación. Precio: $120.84. Potencia de 650W.
* **Deluxe Lavarropas Automático 2024 WashPro:** Categoría Lavado, subcategoría Lavado de Ropa. Precio: $191.93. Capacidad de 8kg y 300W de potencia.
* **Lavarropas Automático LaundryTech:** Categoría Lavado, subcategoría Lavado de Ropa. Precio: $184.74. Capacidad de 10kg y 550W de potencia.
* **Lavarropas Semiautomático X CleanMaster:** Categoría Lavado, subcategoría Lavado de Ropa. Precio: $40.45. Capacidad de 8kg y 550W de potencia.
* **Secarropa FreshMash:** Categoría Lavado, subcategoría Secado. Precio: $184.48. Potencia de 750W.
* **Lavavajillas WashPro:** Categoría Lavado, subcategoría Lavado de Vajilla. Precio: $28.22. Capacidad de 8kg y 900W de potencia.
* **Plancha Seca II FreshMash:** Categoría Lavado, subcategoría Planchado. Precio: $189.92. Potencia de 400W.
* **Centro de Planchado CleanMaster:** Categoría Lavado, subcategoría Planchado. Precio: $156. Potencia de 550W.
* **Compacto Planchita de Pelo 3000 FreshMash:** Categoría Lavado, subcategoría Planchado. Precio: $96.12. Potencia de 1900W.

Tú: Más o menos dime que tan bueno es el Lavarropas más económico de esa lista
El Lavarropas más económico de la lista es el Lavarropas Semiautomático X CleanMaster. Es un producto de la categoría Lavado, subcategoría Lavado de Ropa, con un precio de $40.45. Tiene una capacidad de 8kg y 550W de potencia.

Adicionalmente, te puedo decir que el Lavarropas Semiautomático X de CleanMaster está diseñado para uso doméstico. Revisa el manual del producto (código P0236) para más detalles sobre su uso correcto. Ante cualquier duda, t
```

Tú: No tienes alguna descripción sobre ese Lavarropas ?
El Lavarropas Automático LaundryTech tiene un puntaje de 3/5. No tengo la opinión del cliente sobre ese producto.

Segunda conversación.

```
¡Hola! Soy tu asistente virtual. Pregúntame sobre productos, manuales o inventario.
(Escribe 'salir' para terminar)

Tú: Cual fue la venta cuyo precio total sea el mas alto ?
La venta con el precio total más alto fue la de Aire Split, con un total de 8650.6.

Tú: Dime el nombre de ese cliente ?
No cuento con la información del cliente que realizó la compra de Aire Split con el precio total más alto. ¿Te gustaría que reformule la consulta?

Tú: Cual es el motivo más repetido por cual se realiza una devolución ?
El motivo más repetido por el cual se realiza una devolución es "Compra equivocada", con una cantidad de 345.

Tú: Como puedo usar una Picadora Premium ?
Para usar la Premium Picadora, sigue estos pasos:

1. Verifica que el aparato esté correctamente ensamblado.
2. Conecta a un tomacorriente con conexión a tierra.
3. Prepara los ingredientes o materiales según necesidad.
4. Coloca los ingredientes en el recipiente correspondiente.
5. Cierra la tapa o puerta de seguridad.
6. Selecciona el programa o velocidad deseada.
7. Inicia el aparato presionando el botón de encendido.
8. Monitorea el proceso.

Tú: salir
¡Hasta luego!
```

Ejercicio 2: Evolución del RAG a un Agente Autónomo

Como evolución del asistente, para el segundo ejercicio se da un paso adicional al construir un agente, basado en el paradigma ReAct, con capacidad de uso de herramientas externas. Se mantiene a Gemini flash 2.0 como modelo LMM, para actuar como motor principal de razonamiento, que puede delegar tareas específicas a herramientas especializadas según la necesidad de cada consulta.

Herramientas disponibles

Se definieron las siguientes herramientas, que se agrupan y se ponen a disposición del agente:

- Herramienta documental: recupera texto relevante desde manuales o reseñas utilizando un retriever vectorial y un reranker.
- Herramienta tabular: consulta directamente el DataFrame para obtener datos estructurados como precios o stock.
- Herramienta de grafos: responde preguntas sobre relaciones, compatibilidades o recomendaciones.
- Herramienta de analytics: genera gráficos automáticamente en un nuevo archivo cuando el usuario solicita visualizaciones.

Prompt ReAct personalizado

Se diseñó un prompt ReAct que instruye al modelo sobre cómo debe razonar antes de responder, indicando cómo actuar cuando no encuentra información relevante, responder siempre en el mismo idioma en el que se formula la pregunta, etc. También, se limita la generación de gráficos, al solo producirse cuando el usuario los solicita de manera directa, para evitar que los genere cuando no es esa la intención de la consulta.

Además, el prompt especifica el criterio de selección de herramientas, guiando al modelo para que elija la fuente de información más adecuada según el tipo de datos requeridos en cada consulta. De esta manera, se logra un uso más eficiente de las herramientas disponibles y se mejora la precisión y coherencia de las respuestas generadas por el asistente.

Creación del agente y su interfaz de consulta

Por último, se crea la interfaz de consulta del asistente que se encarga de recibir la pregunta del usuario, invocar al agente y devolver la respuesta generada.

Cada vez que el usuario envía una consulta:

1. El agente recibe el mensaje del usuario y lo inserta en el prompt ReAct junto con las instrucciones del asistente.
2. El modelo analiza la consulta y razona internamente sobre la intención de la pregunta.
3. En función del tipo de información solicitada, el agente decide si es necesario utilizar una herramienta externa.
4. El agente selecciona una única herramienta adecuada según la intención detectada.
5. La herramienta seleccionada recupera la información correspondiente desde su fuente de datos específica.
6. La información recuperada se devuelve al agente en forma de observación (*Observation*) y se integra al razonamiento del modelo.
7. El LLM utiliza dicha información para generar la respuesta final, respetando el idioma original de la consulta y las restricciones del prompt.
8. El agente entrega la respuesta al usuario, completando el ciclo de interacción.

Este diseño modular permite extender el sistema fácilmente, agregando nuevas herramientas o fuentes de datos sin modificar el núcleo del agente.

Primera conversación, en español:

```
¡Hola! Soy tu asistente virtual.
Puedo ayudarte con productos, manuales, inventario, relaciones y gráficos.
(Escribe 'salir' para terminar)

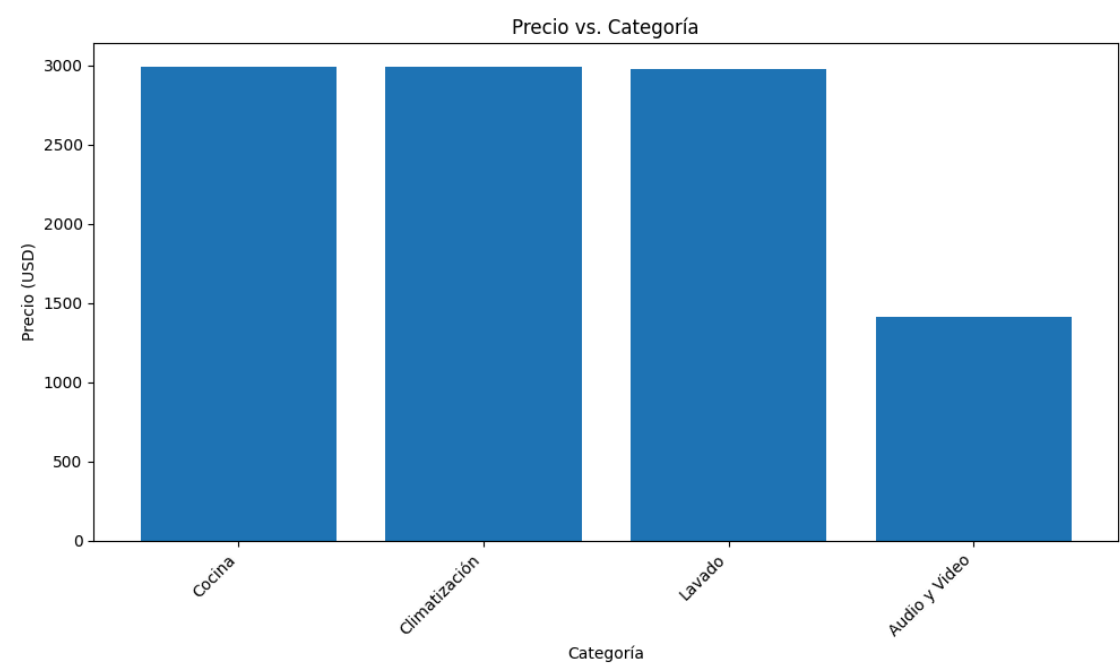
Tú: Estoy haciendo un resumen de la empresa, dime el nombre del el mejor cliente que tenemos, es decir, el que realizo la mayor cantidad de compras
AGENTE: El mejor cliente es laura hernández, con un total de 39 compras.

Tú: Necesito que me digas quien es el mejor vendedor de la empresa, es decir, el que más ventas realizo
AGENTE: El mejor vendedor de la empresa es Antonella Medina, con 492 ventas.

Tú: Finalmente, necesito un gráfico de las categorías de los productos y sus precios.
AGENTE: El gráfico de las categorías de los productos y sus precios ha sido generado y guardado como 'grafico_generado.png'.

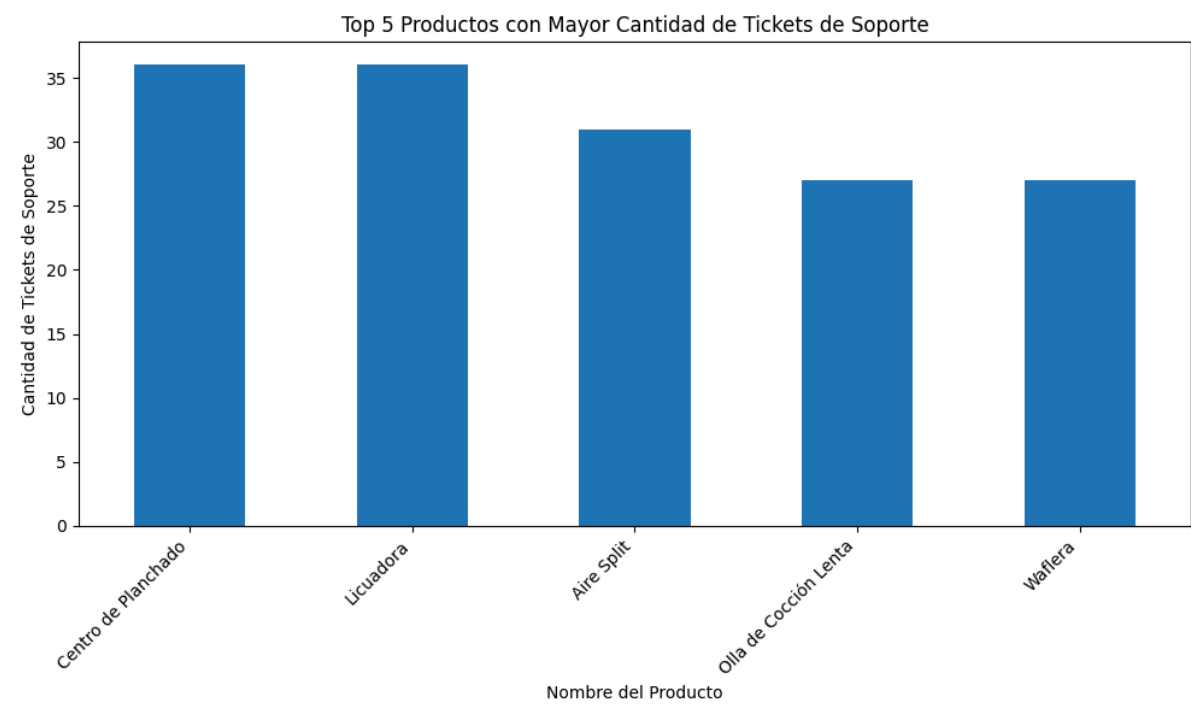
Tú: Cual es el producto más vendido ?
AGENTE: El producto más vendido es la waflera.
```

Gráfico generado.



Ejemplo de otro gráfico.

```
¡Hola! Soy tu asistente virtual.  
Puedo ayudarte con productos, manuales, inventario, relaciones y gráficos.  
(Escribe 'salir' para terminar)  
  
Tú: Necesito un grafico que represente los 5 productos con mayor cantidad de tickers de soporte
```



Manejo de preguntas que no tiene información.

Tú: Dime que películas hizo dicaprio ?

AGENTE: Lo siento, no puedo ayudarte con eso. Mi especialidad son los productos de electrodomésticos.

Segunda conversación, en Inglés.

¡Hola! Soy tu asistente virtual.

Puedo ayudarte con productos, manuales, inventario, relaciones y gráficos.

(Escribe 'salir' para terminar)

Tú: What is the best-selling product?

AGENTE: The best-selling product is waflera (P0070) with a total quantity of 396.

Tú: Which region generated the most sales?

AGENTE: The region that generated the most sales is jujuy.