Universidad de la República

Facultad de Ingeniería

Recuperación de Información y Recomendaciones en la Web

Amazon Copilot

Autores:
Juan Pablo Conde
Xavier Iribarnegaray
Juan Pablo Sotelo

Profesores:
Libertad Tansini

25 de junio de 2025

Índice

1. Introducción

Amazon Copilot constituye un sistema diseñado para optimizar la búsqueda y selección de productos en plataformas de comercio electrónico. Basado en un conjunto de datos de productos de Amazon, el sistema implementará técnicas avanzadas de recuperación de información y procesamiento de lenguaje natural para proporcionar una experiencia de compra intuitiva y personalizada.

El proyecto integra tres funcionalidades principales: un sistema de búsqueda híbrido que combina técnicas semánticas y tradicionales para ofrecer resultados más relevantes; un asistente conversacional (Agente de Inteligencia Artificial) que proporciona una interfaz de búsqueda interactiva, refinando requerimientos mediante diálogo natural y utilizando como mecanismo subyacente el sistema de búsqueda híbrido implementado; y un sistema de recomendación que, basándose en los productos seleccionados en el carrito, emplea el mismo agente para identificar artículos similares o complementarios, mejorando así la experiencia de descubrimiento de productos.

2. Funcionalidades del Sistema

En esta sección se presentan las tres principales funcionalidades que se implementarán en Amazon Copilot.

2.1. Búsqueda Híbrida de Productos

Este componente combina técnicas de búsqueda semántica con métodos tradicionales de correspondencia textual para optimizar resultados. Incorpora representaciones vectoriales semánticas (embeddings) para capturar el significado contextual de las consultas, mantiene búsqueda por términos exactos para consultas específicas e implementa filtrado por categorías. El sistema ordena los resultados mediante un modelo que pondera tanto la relevancia semántica como la coincidencia sintáctica tradicional, logrando un equilibrio óptimo que aprovecha las fortalezas de ambos enfoques para mostrar los productos más pertinentes al usuario.

2.2. Búsqueda Conversacional con Agente de IA

El asistente conversacional proporciona una interfaz de búsqueda interactiva en lenguaje natural. Este agente mantiene el contexto durante la conversación, refina progresivamente los requerimientos del usuario mediante preguntas específicas, y utiliza la API de búsqueda híbrida como herramienta subyacente para ofrecer resultados personalizados. Resulta especialmente útil para usuarios sin conocimientos específicos sobre los productos que buscan.

2.3. Sistema de Recomendación con Agente de IA

Este sistema analiza los productos seleccionados en el carrito para sugerir artículos complementarios o alternativos. Opera automáticamente al visualizar el carrito, generando recomendaciones basadas en productos frecuentemente adquiridos conjuntamente y alternativas similares. Presenta explicaciones concisas sobre cada recomendación y adapta las sugerencias dinámicamente según el contenido del carrito y el historial de la conversación con el usuario.

3. Arquitectura del Sistema

Esta sección describe los componentes principales del sistema, sus responsabilidades e interacciones, así como las tecnologías propuestas para su implementación.

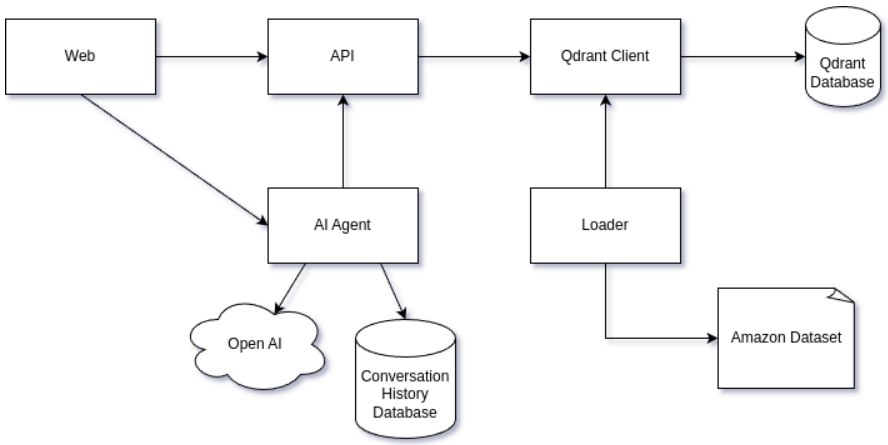

Figura 1: Diagrama de alto nivel de la arquitectura del sistema.

3.1. API

Este componente constituye la capa de servicios que comunica el frontend con los distintos motores de búsqueda y recomendación. Incluye endpoints para listar productos (recibiendo parámetros de filtros y búsqueda) y para obtener información detallada de productos específicos.

3.1.1. Implementación

La API será implementada utilizando FastAPI [3], un framework moderno de Python que permite un desarrollo eficiente y robusto. FastAPI ofrece validación automática de datos mediante Pydantic [4], tipado estático que reduce errores en tiempo de desarrollo, y generación automática de documentación con OpenAPI y Swagger UI. Esta documentación servirá como recurso valioso para la fase de desarrollo del frontend.

3.2. Dataset

Para la información de productos, se utilizará el conjunto de datos de Amazon disponible en Kaggle [8]. Este conjunto de datos ofrece una amplia variedad de atributos por producto, incluyendo:

- Título y descripción detallada
- Categorías y subcategorías
- Precio y disponibilidad
- Valoraciones y número de reseñas
- Imágenes de productos
- Especificaciones técnicas

La riqueza de estos atributos permitirá implementar las modalidades de búsqueda previamente mencionadas.

4. Search

4.1. Loader

Para la carga inicial del conjunto de datos, se desarrollará un módulo de carga consistente en scripts de Python que procesarán el archivo CSV, transformarán los datos en el formato requerido y utilizarán la funcionalidad de almacenamiento del cliente de Qdrant para indexar cada producto.

4.2. Preprocesamiento

El preprocesamiento de datos constituye una etapa fundamental para optimizar la calidad de las búsquedas. Este proceso incluye la limpieza y normalización de textos, la extracción de características relevantes de los productos, y la preparación de los datos para la generación de embeddings.

4.3. Cálculo de Embeddings densos y esparsos (Hybrid Search)

Este módulo constituye el núcleo del sistema de búsqueda vectorial. Se compone de:

- **Base de datos vectorial**: Almacena productos y sus representaciones vectoriales (embeddings densos y esparsos), permitiendo búsquedas por similitud y coincidencia exacta.
- **Cliente de acceso**: Componente que gestiona las operaciones de indexación y consulta a la base de datos. Implementa búsqueda híbrida siguiendo las prácticas recomendadas por Qdrant [7], combinando embeddings densos (para capturar significado semántico) y esparsos (para coincidencias textuales precisas), permitiendo así obtener resultados relevantes tanto para consultas conceptuales como específicas.

La separación de este módulo facilita adaptaciones futuras, permitiendo reemplazar modelos de embeddings o migrar a soluciones alternativas sin impactar al resto del sistema.

4.4. Query

El sistema de consultas implementa algoritmos de búsqueda híbrida que combinan múltiples estrategias para maximizar la relevancia de los resultados. Las consultas se procesan tanto a nivel semántico como sintáctico, permitiendo una recuperación de información más precisa y contextualmente relevante.

4.5. CLI

Se implementará una interfaz de línea de comandos que permitirá interactuar con el sistema de búsqueda de manera directa, facilitando pruebas, depuración y uso programático del sistema.

5. Agent

5.1. Diagrama (flujo)

5.2. Preferencias

El agente de IA mantiene un sistema de preferencias que permite personalizar las interacciones según el perfil y comportamiento del usuario. Estas preferencias se construyen dinámicamente a partir del historial de conversaciones y selecciones de productos.

5.3. Funcionamiento

Este componente implementa la funcionalidad conversacional y de recomendación del sistema utilizando LangGraph [5] y los modelos generativos de OpenAI [6]. Utiliza la API de búsqueda híbrida como herramienta para acceder a los productos, permitiendo una arquitectura donde el diálogo con el usuario y la obtención de información están claramente separados.

Para mantener el contexto de las interacciones, el sistema se integra con una base de datos que almacena el historial de conversaciones, permitiendo referencias a consultas anteriores y facilitando la personalización progresiva de las respuestas.

6. Recommendation

6.1. Estructura (flujo)

6.2. Funcionamiento

El sistema de recomendación analiza los productos seleccionados en el carrito para sugerir artículos complementarios o alternativos. Opera automáticamente al visualizar el carrito, generando recomendaciones basadas en productos frecuentemente adquiridos conjuntamente y alternativas similares.

El sistema presenta explicaciones concisas sobre cada recomendación y adapta las sugerencias dinámicamente según el contenido del carrito y el historial de la conversación con el usuario. Utiliza el mismo agente de IA para proporcionar recomendaciones contextuales y personalizadas.

7. UX/UI

Este módulo define la interfaz gráfica y las interacciones del usuario con el sistema (UI/UX), constituyendo el frontend de la aplicación. A través de esta capa, los usuarios pueden acceder a todas las funcionalidades del sistema de manera intuitiva y eficiente.

7.1. Diseño/Navegabilidad

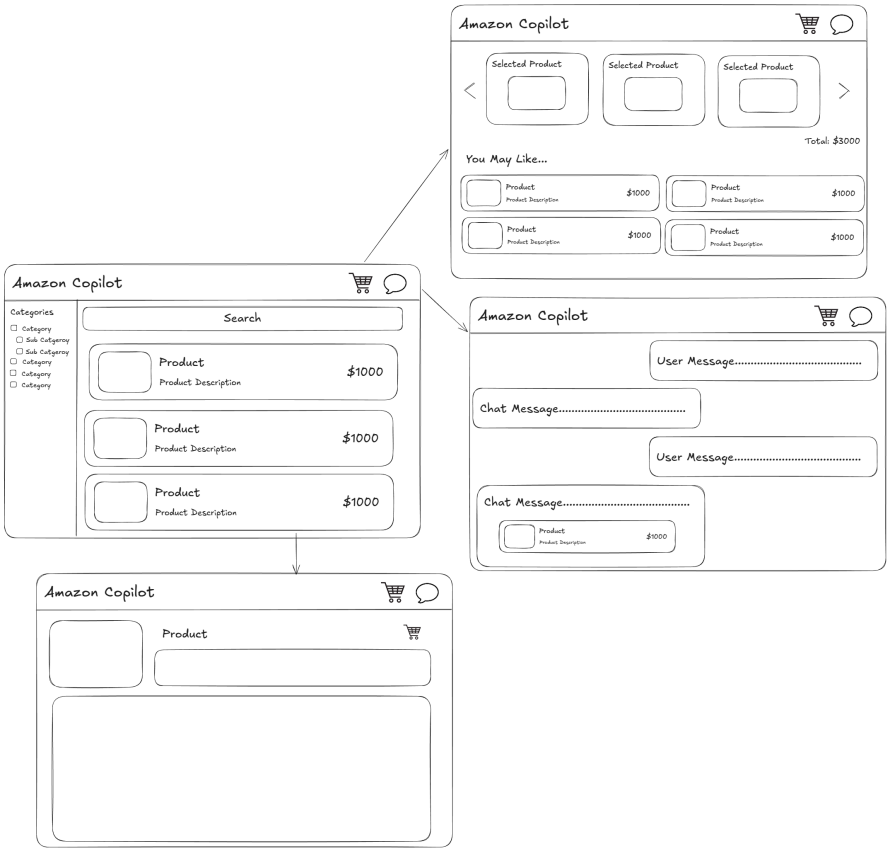

Figura 2: Diseño UX de la interfaz web.

Nota: Este diagrama modela las interacciones del usuario con el sistema. El diseño final de la interfaz de usuario está pendiente de desarrollo.

La interfaz permite al usuario: visualizar un listado inicial de productos con

barra de búsqueda y filtros por categorías; realizar búsquedas que actualizan dinámicamente los resultados; acceder a detalles de productos específicos; gestionar un carrito de compras que muestra recomendaciones relacionadas; e interactuar con el asistente conversacional para búsquedas conversacionales, manteniendo el historial de la conversación.

7.2. Filtros, Pagination

El sistema implementará filtros avanzados que permitan a los usuarios refinar sus búsquedas por categoría, rango de precios, valoraciones y otros atributos relevantes. La paginación eficiente garantizará una experiencia fluida incluso con grandes volúmenes de resultados.

7.3. Estado (Cart & Conversation)

La gestión del estado incluye el mantenimiento persistente del carrito de compras y el historial de conversaciones con el agente de IA. Esto permite una experiencia coherente y personalizada a lo largo de toda la sesión del usuario.

8. Expansión

Esta sección describe las posibles mejoras y expansiones futuras del sistema Amazon Copilot.

8.1. Streaming

La implementación de streaming permitirá respuestas en tiempo real del agente de IA, mejorando significativamente la experiencia del usuario al proporcionar retroalimentación inmediata durante las conversaciones largas.

8.2. Database para registrar actividad de usuario y refinar recomendaciones y búsqueda

La incorporación de una base de datos dedicada al registro de actividad del usuario permitirá:

- Análisis de patrones de comportamiento
- Refinamiento continuo de algoritmos de recomendación
- Personalización avanzada de resultados de búsqueda
- Métricas de rendimiento y satisfacción del usuario

8.3. Embeddings de imágenes

La integración de embeddings de imágenes expandiría las capacidades de búsqueda del sistema, permitiendo:

- Búsqueda visual por similitud de productos
- Comparación automática de características visuales
- Recomendaciones basadas en estilo y apariencia
- Búsqueda multimodal combinando texto e imágenes

9. Conclusiones

10. Referencias

[1] Cursor. (2024). *Cursor: The AI-first code editor*. https://www.cursor.com

[2] Replit. (2024). *Replit: The collaborative browser based IDE*. https://replit.com

[3] FastAPI. (2024). *FastAPI: Framework web de alto rendimiento para APIs con Python*. https://fastapi.tiangolo.com

[4] Pydantic. (2024). *Pydantic: Validación de datos para Python*. https://docs.pydantic.dev/latest

[5] LangGraph. (2024). *LangGraph: Orchestration framework for building stateful, multi-actor applications with LLMs*. https://www.langchain.com/langgraph

[6] OpenAI. (2024). *OpenAI: Creating safe AI that benefits humanity*. https://openai.com

[7] Qdrant. (2024). *Qdrant: Vector Database*. https://qdrant.tech/documentation/search-precision/reranking-hybrid-search

[8] Parab, L. (2023). *Amazon Products Dataset*. Kaggle. https://www.kaggle.com/datasets/lokeshparab/amazon-products-dataset/data?select=Amazon-Products.csv