



Universidad Internacional de La Rioja
Escuela Superior de Ingeniería y
Tecnología

Máster Universitario en Inteligencia artificial

MarIA: Asistente de compra personalizada basado en lenguaje natural

Trabajo fin de estudio presentado por:	Bermejo Collado, Fidel Martínez Ferreiro, Susana Salcedo Puyo, Eduardo
Tipo de trabajo:	Desarrollo Software
Director/a:	Almudena Ruiz Iniesta
Fecha:	11 de septiembre de 2024

Resumen

El objetivo de este Trabajo Fin de Estudios, presentado en la Universidad Internacional de La Rioja, es desarrollar MarIA, un sistema de inteligencia artificial que simula un *personal shopper* para el sector de la venta minorista online. MarIA se basa en técnicas de *machine learning*, técnicas de procesamiento del lenguaje natural (PLN) y en *Large Language Models* (LLM) para proporcionar recomendaciones personalizadas de productos de moda mediante la interacción en lenguaje natural con el usuario. MarIA se compone de dos módulos: uno de recomendación, que utiliza algoritmos de recomendación basados en grafos, y otro de interacción, que emplea técnicas de procesado de lenguaje natural y LLMs para generar respuestas adecuadas al contexto y las preferencias del usuario. Este trabajo muestra la evolución constante de los sistemas de recomendación, que han pasado de los enfoques tradicionales basados en filtrado colaborativo o basado en contenido a los enfoques más avanzados basados en redes neuronales, que son capaces de capturar las características latentes de los usuarios y los productos, así como de modelar las relaciones complejas entre ellos. Además, este estudio destaca la relevancia de proporcionar atención personalizada a los clientes, un desafío significativo en el sector mencionado. Especialmente, el auge de las recientes técnicas de LLM, que facilitan la generación de diálogos fluidos y adaptados a cada usuario, considerando sus necesidades y preferencias, permiten abordar este objetivo.

Palabras clave: Sistema de recomendación, Procesamiento del Lenguaje Natural, *LLM*, Red neuronal

Abstract

The goal of this final project, presented at the International University of La Rioja, is to develop MarIA, an artificial intelligence system that simulates a personal shopper for the online retail sector. MarIA is based on machine learning techniques and Large Language Models (LLM) to provide personalized fashion product recommendations through natural language interaction with the user. MarIA consists of two modules: a recommendation module, which uses graph-based recommendation algorithms, and an interaction module, which employs natural language processing techniques and LLMs to generate responses tailored to the user's context and preferences. This work demonstrates the continuous evolution of recommendation systems, which have moved from traditional approaches based on collaborative filtering or content-based filtering to more advanced approaches based on neural networks that can capture the latent features of users and products, as well as model the complex relationships between them. Furthermore, this study highlights the importance of providing personalized attention to customers, a significant challenge in the mentioned sector. Specifically, the rise of recent Large Language Model (LLM) techniques, which facilitate the generation of fluid and user-adapted dialogues by considering their needs and preferences, allows for achieving this objective.

Keywords: Recommendation system, Natural language processing, LLM, Neural network

Índice de contenidos

1.	Introducción	1
1.1.	Motivación	3
1.2.	Planteamiento del trabajo	4
1.3.	Estructura del trabajo	5
2.	Contexto y estado del arte	7
2.1.	Contexto del problema	7
2.2.	Tipos de técnicas de Sistemas de recomendación	8
2.2.1.	Filtrado colaborativo	8
2.2.2.	Basado en contenido	10
2.2.3.	Basados en conocimiento	13
2.2.4.	Sistemas de recomendación híbridos	16
2.2.5.	Sistema de recomendación basado en Aprendizaje profundo	16
2.3.	Estado del arte en los SR para el ámbito del <i>retail</i>	18
2.3.1.	A comprehensive review of recommender systems: transition from theory to practice	19
2.3.2.	Cross-Border E-Commerce Intelligent Information Recommendation System Based on Deep Learning	21
2.3.3.	Text Understanding and Generation Using Transformer Models for Intelligent E-commerce Recommendation	24
2.4.	Conferencia de Sistemas de Recomendación ReCSys 2022	25
2.5.	Conclusiones	27
3.	Objetivos concretos y metodología de trabajo	30
3.1.	Objetivo general	30
3.2.	Objetivos específicos	31

3.3.	Metodología del trabajo	31
3.3.1.	Búsqueda Bibliográfica	32
3.3.2.	Selección y Adecuación del <i>Dataset</i>	32
3.3.3.	Selección de Técnicas y Algoritmos	32
3.3.4.	Creación del Sistema de Recomendación de Productos	32
3.3.5.	Integración de Procesamiento de Lenguaje Natural (PLN)	33
3.3.6.	Desarrollo del Sistema de Recomendación Conversacional	33
3.3.7.	Evaluación y Validación del Sistema	33
3.3.8.	Iteración y Mejora Continua	33
4.	Desarrollo específico de la contribución	35
4.1.	Identificación de requisitos	35
4.2.	Descripción del Data Set	36
4.2.1.	Tipos de conjuntos de datos	36
4.2.2.	División del Conjunto de Datos	40
4.3.	Descripción de la herramienta software desarrollada	41
4.3.1.	Análisis exploratorio de los datos	42
4.3.2.	Descripción del módulo de Recomendación	52
4.3.3.	Sistema Encoder-Decoder	57
4.3.4.	Descripción del módulo de usuario: MarIA	63
4.3.5.	Configuración del <i>Chatbot</i> basado en <i>Chainlit</i>	68
5.	Evaluación	73
5.1.	Pruebas Funcionales	73
5.2.	Evaluación por Expertos	75
6.	Conclusiones y trabajo futuro	80
6.1.	Conclusiones	80

6.2. Líneas de trabajo futuro	81
7. Bibliografía	83
Código fuente y datos analizados.....	88

Índice de figuras

Figura 1. <i>Necesidades y expectativas de los clientes</i>	2
Figura 2. <i>Un vistazo al futuro del sector minorista</i>	3
Figura 3. <i>Comprehensive Overview of Graph Neural Network Models across Various Metrics and Use Cases.</i>	21
Figura 4. <i>Diagrama de fases de trabajo</i>	34
Figura 5. <i>Captura datos archivo train_sessions.csv</i>	37
Figura 6. <i>Construcción de los datos de sesión y compras</i>	38
Figura 7. <i>Captura datos archivo “purchases.csv”</i>	38
Figura 8. <i>Ejemplificación de las características particulares de un artículo concreto</i>	39
Figura 9. <i>Captura datos archivo item_features.csv</i>	39
Figura 10. <i>División realizada sobre el conjunto de datos</i>	41
Figura 11. <i>Distribución del número de artículos por categoría</i>	44
Figura 12. <i>Distribución de la cantidad de valores distintos por categorías</i>	44
Figura 13. <i>Número de apariciones de cada valor</i>	45
Figura 14. <i>Distribución del número de categorías por artículo</i>	46
Figura 15. <i>Distribución de la longitud de las sesiones de usuario en el conjunto de datos</i>	46
Figura 16. <i>Número de consultas por franja horaria 2020</i>	47
Figura 17. <i>Análisis de las compras por día de la semana (2020/2021).</i>	48
Figura 18. <i>Análisis de las compras al mes (2020/2021)</i>	49
Figura 19. <i>Top 5 de artículos más comprados por mes</i>	50
Figura 20. <i>Top 5 de artículos más comprados al mes 2020</i>	52
Figura 21. <i>Top 5 de artículos más comprados al mes 2021</i>	52
Figura 22. <i>Configuración de hiperparámetros y resultados obtenidos en MRR</i>	57
Figura 23. <i>Descripción de un artículo tras decodificación</i>	59

Figura 24. <i>Esquema generación de sesiones compatibles con el módulo de recomendación (encoder)</i>	61
Figura 25. <i>Esquema decodificación de resultados a lenguaje natural</i>	63
Figura 26. <i>Flujo de un funcionamiento de un agente conversacional</i>	63
Figura 27. <i>Modelo de Plataforma de Copilot Stack</i>	65
Figura 28. <i>Arquitectura del Módulo de Interacción con el usuario</i>	66
Figura 29. <i>Interfaz web de MarIA basado en Chainlit</i>	68
Figura 30. <i>Ejemplo de conversación con el chatbot MarIA</i>	72
Figura 26. <i>Evaluación dentro de contexto de la moda</i>	74
Figura 27. <i>Evaluación fuera de contexto de la moda</i>	75
Figura 28. <i>Evaluación usuario 1</i>	77
Figura 29. <i>Evaluación usuario 2</i>	78
Figura 30. <i>Evaluación usuario 3</i>	79

Índice de tablas

Tabla 1. <i>Organización del trabajo en grupo.</i>	X
--	---

Organización del trabajo en grupo

Distribución y estructura de la memoria

Tabla 1. Organización del trabajo en grupo.

Organización del trabajo en grupo - Desarrollo de la memoria	
Apartado de la memoria	Responsables
Introducción	Fidel Bermejo, Susana Martínez, Eduardo Salcedo
Contexto y estado del arte	Fidel Bermejo, Susana Martínez, Eduardo Salcedo
Objetivos y metodología de trabajo	Fidel Bermejo, Susana Martínez, Eduardo Salcedo
Marco normativo	Fidel Bermejo, Susana Martínez, Eduardo Salcedo
Desarrollo específico de la contribución	Fidel Bermejo, Susana Martínez, Eduardo Salcedo
Desarrollo específico de la contribución	Fidel Bermejo, Susana Martínez, Eduardo Salcedo
Desarrollo específico de la contribución	Fidel Bermejo, Susana Martínez, Eduardo Salcedo
Conclusiones	Fidel Bermejo, Susana Martínez, Eduardo Salcedo

Fuente: Elaboración propia.

Objetivo del TFE desde el punto de vista de la adquisición de conocimientos

El objetivo de este TFE es mostrar cómo se puede aplicar el análisis de datos, el procesamiento del lenguaje natural y las redes neuronales a un problema real y relevante para el ámbito de concreto de la industria como es la venta minorista. Para ello, se ha diseñado y desarrollado

un sistema que permite extraer información relevante de un set de datos y generar recomendaciones automáticas que facilitan la toma de decisiones. Este TFE supone integrar conocimientos teóricos y prácticos de diferentes disciplinas, que nos han permitido adquirir competencias técnicas, consultas e investigadoras, de gran valor para nuestro desempeño profesional.

Mecanismos de coordinación empleados

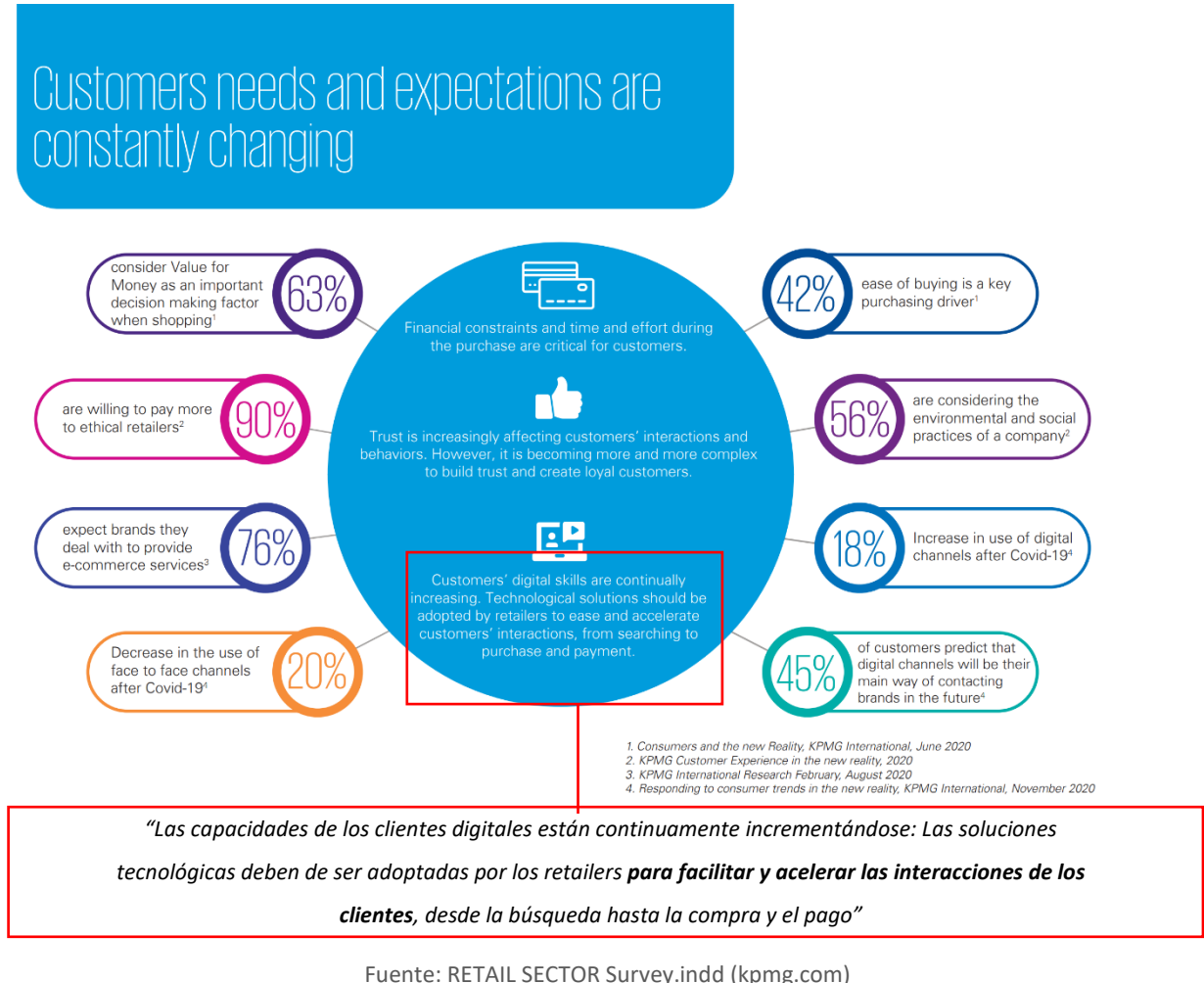
Para el desarrollo del TFE grupal, se ha empleado el uso de la plataforma *Teams*, que ha permitido trabajar de forma colaborativa y sincronizada en los documentos, así como comunicarnos de forma fluida y eficiente. Se han realizado reuniones semanales para compartir el grado de avance, debatir las conclusiones, negociar los retos y ajustar el cronograma. Además, se ha mantenido un alto nivel de calidad en la colaboración, realimentando las ideas de unos con otros, respetando las opiniones y aportando soluciones creativas.

1. Introducción

Dado el contexto competitivo actual, la diferenciación en la venta minorista es retadora desde cualquier perspectiva. Y las estrategias tradicionales de este sector ya no son tan efectivas como antes. En este contexto, la venta minorista (*retailers*) está desarrollando estrategias centradas en el consumidor que les permitan mejorar tanto la satisfacción del cliente como las ventas. (Erik Lindecrantz et al., 2020). Estas estrategias no son ajenas a la adaptación a los cambios en el comportamiento de compra del consumidor (Erik Lindecrantz et al., 2020; Holly Briedis, 2020), la integración de los canales físicos y digitales (Iglesias-Pradas & Acquila-Natale, 2023) (Acquila-Natale et al., 2022) o la cadena de suministro, que permitirían, entre otros, la citada diferenciación con la competencia.

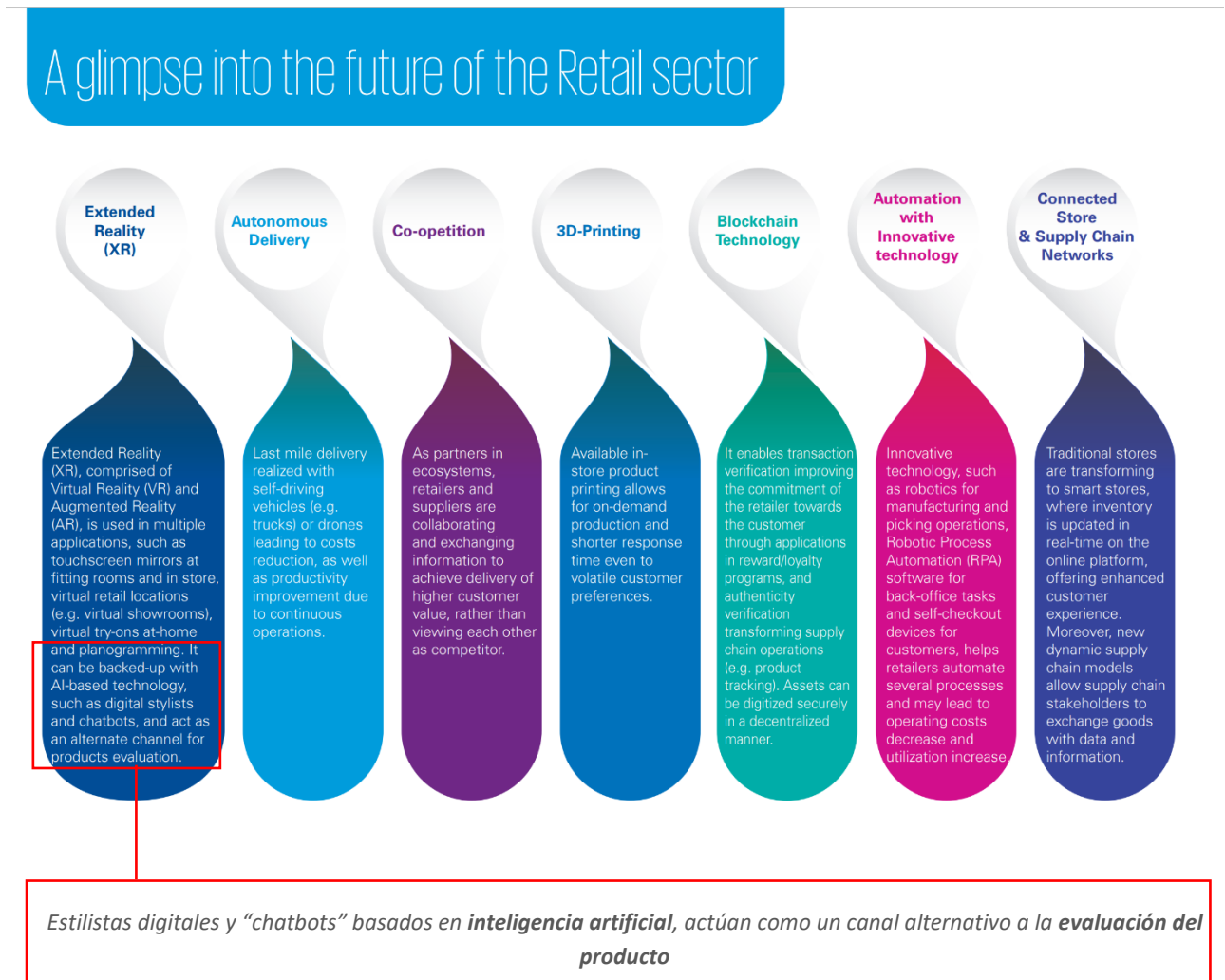
Según un informe de KPMG de 2022, las necesidades de los clientes y sus expectativas están constantemente cambiando, así como las capacidades digitales del usuario consumidor, por lo que los *retailers* deben adoptar soluciones tecnológicas que faciliten y aceleren las interacciones de sus clientes, como son la búsqueda de productos, la compra o el pago asociado, tal y como se ilustra en la siguiente figura, (KPMG, 2022):

Figura 1. Necesidades y expectativas de los clientes



Estos retos requieren de una estrategia innovadora y flexible que aproveche las oportunidades que ofrece la tecnología y el análisis de datos. KPMG en su informe, sitúa a la Inteligencia Artificial (IA) y la realidad extendida (*Extended Reality*) como dos de los vectores para dar respuesta en el futuro a estos retos. En la siguiente figura se puede observar cómo se sitúa a la Inteligencia Artificial como un canal alternativo para la evaluación de productos:

Figura 2. *Un vistazo al futuro del sector minorista*



Fuente: RETAIL SECTOR Survey.indd (kpmg.com)

1.1.Motivación

Dar una respuesta personalizada al consumidor requiere conocer su comportamiento de compra, no obstante, la mayoría de los *retailers* están en las primeras fases de la citada personalización (Erik Lindecrantz et al., 2020). Por otro lado, los grandes minoristas están moderadamente preparados para ofrecer servicios multicanal y omnicanal (Acquila-Natale et al., 2022b). Adicionalmente el Departamento de Ingeniería de Organización, Administración de Empresas y Estadística, indica que durante la crisis de COVID-19, los *retailers* se enfocaron en integrar servicios rápidos y fáciles de implementar, como tarjetas de regalo omnicanal, o

comunicarse a través de aplicaciones de mensajería instantánea como “WhatsApp”, tal y como se menciona en el artículo *“Effects of COVID-19 on Channel Integration and Digital Transformation of Large Clothing and Apparel Retailers in Spain.”* (Acquila-Natale et al., 2022b)

La motivación de este trabajo se inspiró en intentar mejorar la atención personalizada al consumidor, mediante un sistema de recomendación basado en Inteligencia artificial, con el ánimo de una mejora en la comunicación y que el consumidor tenga una experiencia de compra más rica y personalizada. Para ello, se proporcionará a través de la incorporación de un sistema de recomendación que simule a un *Personal Shopper* (PS). Éste asesorará y acompañará al consumidor en su proceso de decisión, a través de lenguaje natural, con el objetivo de mejorar la experiencia de evaluación del producto y su compra asociada.

1.2.Planteamiento del trabajo

El presente trabajo se enfoca en el diseño e implementación de un sistema de recomendación basado en la conversación con el usuario, haciendo para ello uso de técnicas avanzadas de *Machine Learning* y *Large Language Models* (LLM). Presenta varios desafíos como es la creación del catálogo de los productos a recomendar, evaluación de la conversación con el usuario como la integración con el interfaz que interactuará con el usuario. Además de la necesidad de decodificar el *dataset* en el que se ha basado el trabajo, a literales que permitan tener una conversación con sentido y en contexto con el cliente.

A continuación, se enumeran las principales características del trabajo:

- El objetivo último es que la experiencia de atención del usuario *on-line*, sea análoga a la que haría un vendedor en una tienda física, por lo que el PS tendrá un rol de asesor al cliente en la búsqueda del producto que mejor responda a las necesidades del cliente, en este caso del mundo de la moda, según la ocasión, evento, gustos etc.
- El objetivo del PS será extraer a través de la conversación y la información de la sesión del cliente, la información máxima que permita recomendar al cliente la mejor opción dentro del catálogo disponible en la venta online, según su estilo y necesidades.
- El PS tendrá información sobre el comportamiento de los clientes y los patrones de visitas basado en el registro de sesiones.

- Para hacer una recomendación predictiva, será importante conocer los atributos que modelarán el comportamiento de la sesión de consumo del *dataset* utilizado en este trabajo. Esta información será utilizada por el PS para proporcionar recomendaciones de productos personalizados. La recomendación se realizará utilizando técnicas de *Machine Learning* que puedan predecir si la mercancía propuesta, son los que realmente van a gustar al consumidor.

1.3. Estructura del trabajo

Este trabajo se divide en cuatro capítulos principales, además de esta introducción y las conclusiones. A continuación, se describe brevemente el contenido de cada uno de ellos:

- **Capítulo 2:** Se describe el Estado del arte de los sistemas de recomendación. En este capítulo se realiza una revisión de los conceptos y las técnicas fundamentales de los sistemas de recomendación, así como de los principales desafíos y tendencias en este campo. Se clasifican los sistemas de recomendación según los criterios de filtrado, contenido y contexto, y se analizan las ventajas y desventajas de cada uno. Se presentan algunos ejemplos de aplicaciones reales de los sistemas de recomendación en el dominio del comercio electrónico y del *retail*. A continuación, se explica la evolución de los sistemas de recomendación desde los enfoques fundacionales basados en reglas, heurísticas o factorización matricial, hasta los enfoques actuales basados en redes neuronales, que ofrecen mayor flexibilidad, escalabilidad y capacidad de aprendizaje. Se destaca el papel de las redes neuronales basadas en grafos (GNN) como una de las técnicas más prometedoras para modelar las relaciones dinámicas entre usuarios y productos, a partir de la matriz de interacción (URM – *User-Item Rating Matrix*) que refleja la preferencia de los consumidores. Finalmente, se describe el reto de sistemas de recomendación para *retailers* propuesto por ReCSys en 2022, donde los ganadores utilizaron técnicas de GNN para generar recomendaciones personalizadas y relevantes.
- **Capítulo 3:** Metodología y diseño del sistema de recomendación. En este capítulo se explica la metodología y el proceso seguido para diseñar e implementar el sistema de recomendación, basado en técnicas de aprendizaje automático supervisado. Se describen los pasos realizados para la preparación de los datos, la selección de las variables explicativas, la definición de la variable objetivo, la división de los datos en conjuntos de

entrenamiento y prueba, la elección de los algoritmos y los criterios de evaluación del modelo.

- **Capítulo 4:** Desarrollo específico de la contribución: En este capítulo se detalla el origen, la estructura y el contenido de los datos que se han empleado para entrenar y evaluar el sistema de recomendación, así como las operaciones de limpieza, transformación y análisis exploratorio que se han realizado sobre ellos. El conjunto de datos proviene de *Dressipi*, una empresa de tecnología en el sector de la moda que se especializa en proporcionar recomendaciones de moda personalizadas que opera en varios países, y contiene información sobre los productos, las categorías, los precios, las imágenes, las valoraciones, las compras y los clics de los usuarios durante un periodo de 18 meses. Se muestra cómo se ha construido la matriz de interacción entre usuarios y productos, que representa el nivel de interés o satisfacción de los clientes con los artículos que han visto o comprado, y que constituye la base para el aprendizaje del modelo. Además, se describe la construcción del módulo de interacción con el usuario, que es un *chatbot* desarrollado con *Azure Open AI* y *Chainlit*. Este módulo permite el uso de LLMs que proporcionan respuestas dinámicas e integradas con el usuario, mejorando significativamente la interacción y personalización de las recomendaciones.
- **Capítulo 5:** Conclusiones y trabajo futuro. En este capítulo se resume el trabajo realizado y se plantean las posibilidades de mejora y ampliación. Se dedica a las conclusiones, donde se presentan los resultados, las contribuciones y las limitaciones del sistema de recomendación de productos para una tienda online de moda y se perfilan las líneas en el trabajo futuro, donde se proponen algunas líneas de investigación y desarrollo que podrían mejorar el rendimiento y el valor del sistema de recomendación.

2. Contexto y estado del arte

En este capítulo se discute la necesidad de tomar decisiones en un entorno en el que la información es cada vez mayor, subrayando la variedad de canales por la que la información llega hasta todos nosotros. Para abordar este problema, se han desarrollado los sistemas de recomendación, que proporcionan sugerencias personalizadas de productos o servicios a los usuarios, basándose en sus preferencias, comportamientos, necesidades o contextos (Ricci et al., 2015). Estos sistemas tienen una gran importancia en el ámbito del *retail*, ya que pueden mejorar la experiencia de compra de los clientes, aumentar las ventas, la fidelización y la rentabilidad de los negocios.

Se presentan los principales tipos de técnicas de sistemas de recomendación, incluyendo el filtrado colaborativo, el filtrado basado en contenido y el filtrado basado en conocimiento.

Además, se presentan algunas de las últimas innovaciones en los sistemas de recomendación específicamente para el ámbito del *retail*. Éstas incluyen el uso de técnicas de aprendizaje profundo como redes neuronales basadas en grafos (GNN), que mejoran la precisión y efectividad de las recomendaciones al aprender representaciones profundas de los datos de interacción entre usuarios y elementos. Las GNN son un tipo de red neuronal diseñada para trabajar con datos que tienen una estructura de grafo. Un grafo es una colección de nodos y aristas (o enlaces) que conectan estos nodos. En el contexto del *retail*, un grafo podría representar, por ejemplo, la relación entre productos (nodos) y las conexiones entre ellos (aristas), como productos que suelen comprarse juntos.

Se ha incorporado una breve visión de *RecSys*, ya que es la conferencia sobre sistemas de recomendación y el foro internacional más destacado para la presentación de nuevos resultados de investigación, sistemas y técnicas en campo de los sistemas de recomendación.

Por último, se sientan las bases para el desarrollo del sistema de recomendación objeto de este trabajo, y que se discutirá en los capítulos posteriores.

2.1.Contexto del problema

En nuestra realidad actual, se presenta una necesidad continua de toma de decisiones, por ejemplo, elección de películas, libros, tecnología, comida, viajes, regalos, comida, etc., combinada con la desbordante diversidad de información que nos llega a través de los

diferentes entornos y canales digitales (plataformas de *streaming*, tiendas *on-line* y redes sociales, etc.) (Ricci et al., 2015) se hace imperiosa una herramienta que palie esta necesidad y que solucione este problema de una manera ágil y rápida. Así surgen los sistemas de recomendación (SR) para ayudarnos en esta vicisitud, gracias a los algoritmos de Inteligencia Artificial que extraen información de esos canales para ofrecernos productos de nuestro interés y que además se han ido sofisticando tecnológicamente a lo largo de años.

Los SR son aplicaciones que proporcionan sugerencias personalizadas de productos o servicios a los usuarios, basándose en sus preferencias, comportamientos, necesidades o contextos. Estos sistemas tienen una gran importancia en el ámbito de *retail*, ya que pueden mejorar la experiencia de compra de los clientes, aumentar las ventas, la fidelización y la rentabilidad de los negocios. Además, pueden ayudar a los usuarios a descubrir nuevos productos o servicios que les puedan interesar, reduciendo el problema de la sobrecarga de información.

2.2. Tipos de técnicas de Sistemas de recomendación

A continuación, se presentan los principales tipos de SR (Ricci et al., 2015)

- Filtrado colaborativo (CF).
- Filtrado Basado en contenido (CBF).
- Filtrado Basado en conocimiento (KBF).
- Sistemas de recomendación híbridos (HRS).

Se ha completado esta clasificación con los basados en Aprendizaje profundo: Redes Neuronales Convolucionales (CNN) y Redes Neuronales en Grafos (GNN)

2.2.1. Filtrado colaborativo

Este enfoque identifica patrones de comportamiento similares entre usuarios y recomienda elementos que otros usuarios con gustos similares han usado. Esta similitud se obtiene con métricas para conseguir a los vecinos más afines (*Neighbor-based*). Algunas funciones de similitud más habituales son las correlaciones de Pearson, Coseno y Coseno ajustado. (Pinela, 2017)

En este contexto del filtrado colaborativo, existen dos tipos principales:

- Filtrado basado en usuario (*User-based*): Se comparan usuarios similares en función de sus comportamientos pasados de interacción con los elementos. Por ejemplo, si dos usuarios

han calificado de manera similar una serie de películas, es probable que tengan gustos similares. Por lo tanto, si un usuario inicial ha disfrutado de ciertas películas, el SR sugerirá otras películas que usuarios similares al usuario objetivo hayan valorado positivamente.

- **Filtrado basado en elementos (*Item-based*):** En lugar de comparar usuarios, el filtrado basado en elementos se centra en encontrar similitudes entre los propios elementos (por ejemplo, películas, productos). Si dos elementos tienen un historial de interacción similar con los mismos usuarios, es probable que estén relacionados. Por lo tanto, si un usuario ha mostrado interés en ciertos elementos, el SR sugerirá otros elementos relacionados en función de la interacción pasada de los usuarios.

Es importante indicar que, en este algoritmo, las puntuaciones de los usuarios en los productos podrían suponer un problema. Ya que la actualización de esa información ayuda a aprender al algoritmo y ser más preciso en futuras recomendaciones. (Pinela, 2017)

El filtrado colaborativo basado en vecinos es un método que aprovecha las similitudes entre usuarios o elementos para hacer recomendaciones personalizadas. En el enfoque basado en usuario, se comparan usuarios similares, mientras que, en el enfoque basado en elementos, se buscan elementos relacionados. (Jannach et al., 2010)

En el trabajo "*Recommender Systems and Collaborative Filtering*" publicado en la revista "*Applied Sciences*", se proporciona una revisión detallada de los sistemas de recomendación y las técnicas de filtrado colaborativo. Se enfoca en describir las metodologías fundamentales y sus aplicaciones prácticas en diversos contextos:(Ortega & González-Prieto, 2020)

1. Filtrado Colaborativo Basado en Vecinos:

- *User-based*: Este método recomienda ítems basándose en la similitud entre usuarios. Si dos usuarios han calificado ítems de manera similar, es probable que tengan gustos parecidos.
- *Item-based*: En lugar de comparar usuarios, se comparan ítems. Si dos ítems han recibido calificaciones similares de los mismos usuarios, se recomienda un ítem basado en la preferencia del usuario hacia otro ítem.

2. Modelos Basados en Factorización de Matrices: Se utiliza para descomponer una gran matriz de interacciones *usuario-ítem* en productos de matrices más pequeñas, lo que facilita la identificación de patrones latentes en las preferencias de los usuarios y las características de los ítems.

3. **Máquinas de Factorización:** Este enfoque mejora los modelos de factorización tradicionales al incluir variables adicionales como atributos de los usuarios y los *items*, proporcionando recomendaciones más precisas y personalizadas.

El trabajo presentaba un análisis de los retos para este tipo de filtrado:

1. **Escalabilidad:** La capacidad de los sistemas para manejar grandes volúmenes de datos.
2. **Sparsity:** La dispersión de los datos, ya que típicamente los usuarios solo interactúan con una pequeña fracción del total de ítems disponibles.
3. **Cold Start:** La dificultad para recomendar ítems o usuarios nuevos sin suficiente historial de interacciones.

Y las aplicaciones en diversas en diversas industrias, tales como:

1. **E-commerce:** Personalización de recomendaciones de productos.
2. **Entretenimiento:** Sugerencias de películas, música y libros.
3. **Educación:** Recomendaciones de cursos y recursos educativos.
4. **Salud:** Sugerencias de tratamientos y productos médicos basados en las preferencias del paciente.

El artículo concluye que, aunque las técnicas de filtrado colaborativo han demostrado ser efectivas en múltiples dominios, existen áreas que requieren mejoras continuas. La integración de técnicas avanzadas de aprendizaje automático y el manejo adecuado de los desafíos mencionados son esenciales para el desarrollo de sistemas de recomendación más robustos y eficientes.

2.2.2. Basado en contenido

Estos sistemas analizan las características de los elementos (como películas, libros o juegos) y sugieren otros elementos con características similares. Por ejemplo, si un niño disfrutó de una película de *Pixar*, el sistema podría recomendar otras películas de otros estudios del mismo género.

Las ventajas de este tipo de filtrado sobre el colaborativo son las siguientes:

- Independencia de los usuarios (sólo es necesario el usuario activo), no hacen falta valoraciones previas.

- Transparencia: explicabilidad sencilla viendo las características de los productos, a diferencia del de caja negra que puede suponer los vecinos similares.
- Nuevos artículos donde no hace falta puntuaciones previas para recomendarlos.

En contra partida presenta las siguientes desventajas:

- Sobre especialización: encasillamiento ante una novedad, se va a ver muy limitado para recomendar una novedad.
- Usuarios: para un usuario nuevo, si no hay recomendaciones, no va a obtener buenos resultados.
- Análisis de limitación de características: las características de un artículo pueden no ser suficientes para describir el aspecto de un producto, necesarios para la recomendación.

Para estos sistemas es muy importante la manera en la se representa un artículo, sus características descriptivas. Para obtener esta asociación se necesita determinar los términos/características (corpus) más significativos de los artículos, es decir, obtener el conjunto de características y valorarlas numéricamente, que sean los más representativos de los artículos (vectorización de características), para a continuación puntuar/valorar estos términos en nuestros artículos. Estas son las dos técnicas más populares, aunque existen otros menos utilizadas:

1. Bolsa de palabras (BOW): Esta técnica evalúa el número de apariciones de un término específico en los artículos, sin considerar el orden de aparición, únicamente su frecuencia. Existen variantes de BOW que incluyen la consideración de unigramas o n-gramas para capturar el orden en que aparecen los términos en los artículos, proporcionando una representación más contextual del texto. (Yan et al., 2020)
2. *Term-Frequency-Inverse Document Frequency* (TF-IDF): Este enfoque busca mitigar las limitaciones de la técnica BOW al asignar un peso mayor o menor a los términos, según su importancia relativa. De este modo, términos comunes como artículos y preposiciones reciben menos peso en comparación con sustantivos o adjetivos, más significativos en la diferenciación de documentos. TF-IDF utiliza una función matemática de similitud que proporciona un valor decimal para cuantificar la semejanza entre artículos. (Ni et al., 2021)
3. Otras aproximaciones: Entre las alternativas se encuentran las incrustaciones de palabras, como *Word2Vec*, también conocido como *Word Embedding*, que no solo cuantifica la

frecuencia de las palabras, sino que también categoriza las similitudes semánticas entre términos (Di Gennaro et al., 2021). Otra técnica relevante es el modelado de temas, ejemplificado por *Latent Dirichlet Allocation* (LDA), que extrae estructuras semánticas latentes a nivel de los artículos, considerando que cada artículo es una combinación de varios temas y que cada tema está compuesto por una mezcla de palabras (Henderi et al., 2023). Además, existen enfoques híbridos como Lda2Vec, que combina las incrustaciones de palabras con el modelado de temas para proporcionar un análisis más completo y enriquecido.

En este artículo de reciente publicación *A comprehensive Review of recommender Systems: Transitioning from theory to practice* (Julio 2024), (Raza et al., 2024a) se analizan los SR fundacionales y su evolución. En el mismo se refieren a los SR fundacionales como modelos y técnicas tempranas que establecieron los principios y metodologías centrales en el campo de los motores de recomendación. A continuación, se presenta una visión del contenido del artículo sobre CBF:

CBF es una estrategia de recomendación que sugiere elementos a un usuario según los atributos de los elementos y un perfil de las preferencias del usuario, usando medidas de similitud para emparejar las preferencias del usuario con los de estos. En CBF, la recomendación se basa en las características de los elementos $\phi(i)$ y un perfil de las preferencias del usuario $\theta(u)$:

$\phi(i)$ representa el vector de características del elemento.

$\theta(u)$ representa el vector de preferencias del usuario.

La evolución de los modelos de recomendación CBF comienza con métodos tradicionales como modelos de espacio vectorial, modelos probabilísticos y árboles de decisión, basándose en la ingeniería manual de características y cálculos de similitud. Los modelos de espacio vectorial calculan la similitud de los elementos mediante la similitud del coseno, los modelos probabilísticos estiman la probabilidad de preferencia del usuario con análisis estadístico, y los árboles de decisión recomiendan elementos categorizándolos en función de los atributos. Estos modelos sentaron las bases para recomendaciones personalizadas al aprovechar características explícitas de los elementos y preferencias del usuario. Los avances algorítmicos en informática y métodos de aprendizaje automático llevaron a un cambio hacia redes

neuronales sofisticadas que permiten la extracción automática de características y el aprendizaje de patrones de datos complejos. Estos sistemas basados en redes neuronales aprovechan el aprendizaje profundo para analizar las interacciones de los usuarios y las características de los elementos en diferentes modalidades, incluyendo textual (como reseñas, citas y noticias), *streaming* (como música) y datos de imágenes. El objetivo general de estos avances es generar recomendaciones personalizadas alineando los perfiles de los usuarios con las características de los elementos. En este trabajo también se centran en analizar los Desafíos del CBF: En general, el CBF enfrenta desafíos como el problema del arranque en frío, la sobre-especialización (solo sugiere elementos similares a los que el usuario ya ha visto o le han gustado), el costo computacional (que puede aumentar de manera cuadrática o cúbica con el número de usuarios y elementos) y la falta de actualizaciones de los perfiles de usuarios y elementos.

2.2.3. Basados en conocimiento

Estos sistemas de recomendación emplean información específica del usuario y/o del contenido para ofrecer recomendaciones en contextos muy especializados, como la compra de un automóvil, donde las valoraciones son escasas, o de un ordenador, en un entorno de constante evolución tecnológica. La implementación práctica de estos sistemas no es sencilla, ya que requiere un conocimiento profundo del contexto y la información detallada del usuario. El modelo de predicción más adecuado en estos casos son los árboles de decisión, los cuales utilizan reglas de bifurcación para guiar al usuario hacia el resultado óptimo (Burgues, 2018). A continuación, se detallan algunos tipos de sistemas de recomendación basados en este enfoque:

- **Basados en Reglas:** Estos sistemas utilizan reglas de expertos o heurísticas para recomendar elementos. Por ejemplo, un sistema podría recomendar películas de comedia si el usuario ha expresado una preferencia por ese género en el pasado.
- **Basados en Modelos:** Estos sistemas construyen un modelo del usuario y/o del contenido para hacer recomendaciones. Por ejemplo, un sistema podría crear un modelo de los gustos de un usuario basado en sus acciones anteriores y utilizarlo para sugerir nuevas películas, libros o productos.
- **Basados en Casos:** Estos sistemas recomiendan elementos similares a otros que el usuario ha disfrutado en el pasado de manera satisfactoria. Por ejemplo, si un usuario ha

comprado un libro de ciencia ficción, el sistema podría recomendar otros libros del mismo género.

Por ejemplo, un sistema basado en conocimiento podría recomendar libros de ciencia ficción, si el usuario ha expresado un interés en ese género en el pasado, utilizando reglas predefinidas por expertos (explícitamente, especificar -me gusta la ciencia ficción-) o sin ir más lejos, en ropa (Atina & Hartanti, 2022). Este artículo: *“Clothing Product Selection Recommendation System with Knowledge-Based Recommendation Method”* describe un sistema de recomendación para la selección de productos de ropa que utiliza un método de recomendación basado en el conocimiento. El sistema tenía como objetivo mejorar la experiencia de compra de los usuarios mediante la personalización de las recomendaciones de productos. Los puntos clave del estudio son los siguientes:

1. Metodología Basada en el Conocimiento:

- El sistema utiliza una base de conocimientos que contiene información sobre las características de los productos y las preferencias de los usuarios.
- Se implementan reglas de inferencia para combinar el conocimiento experto con los datos del usuario, lo que permite generar recomendaciones más precisas.

2. Ventajas del Sistema:

- El enfoque basado en el conocimiento permite superar algunos de los problemas comunes en los sistemas de recomendación tradicionales, como el problema del arranque en frío (cuando hay poca información inicial sobre el usuario o el producto).
- Proporciona recomendaciones más personalizadas y relevantes al considerar múltiples factores y características específicas de los productos de ropa.

3. Implementación y Resultados:

- Se realizó una implementación práctica del sistema y se llevaron a cabo pruebas para evaluar su efectividad.
- Los resultados mostraron una mejora significativa en la satisfacción del usuario y la precisión de las recomendaciones en comparación con los métodos tradicionales de filtrado colaborativo y basado en contenido.

Este sistema de recomendación puede ser especialmente útil en plataformas de comercio electrónico, donde la personalización de la experiencia del usuario es crucial para incrementar la satisfacción y las ventas.

A continuación, se presenta los componentes del Sistema que fueron usados para este artículo:

1. Base de Conocimientos: La base de conocimientos del sistema incluía información sobre estilos de ropa, preferencias del usuario, datos de tendencias y características específicas de los productos. Esta base de conocimientos se actualizaba continuamente para reflejar las tendencias actuales y las preferencias cambiantes de los usuarios.
2. Método de Recomendación Basado en Conocimiento: A diferencia de los sistemas de recomendación tradicionales que solían utilizar métodos colaborativos o basados en contenido, este sistema utilizaba un enfoque basado en conocimiento. Esto implicaba el uso de reglas y lógica derivadas del conocimiento experto en moda para generar recomendaciones personalizadas.
3. Perfil del Usuario: El sistema creaba un perfil detallado del usuario que incluía sus preferencias personales, historial de compras y comportamientos. Este perfil se utilizaba para adaptar las recomendaciones a las necesidades y gustos específicos del usuario.
4. Integración de Tendencias de Moda: El sistema también incorporaba información sobre las últimas tendencias de moda para garantizar que las recomendaciones no solo fueran personalizadas, sino que también estuvieran alineadas con las tendencias actuales del mercado.

La principal innovación introducida por este trabajo fue el uso de un enfoque basado en conocimiento para la recomendación de productos de ropa. Los sistemas de recomendación convencionales solían depender en gran medida de técnicas de filtrado colaborativo o basado en contenido, que a menudo tenían limitaciones en términos de contexto y personalización profunda. Las Innovaciones clave fueron tres:

1. Las basadas en un enfoque en Conocimiento: El uso de reglas y lógica derivadas de expertos en moda permitió una personalización más precisa y relevante. Este enfoque

proporcionó una comprensión más profunda de las preferencias del usuario y cómo se relacionaban con las tendencias actuales.

2. Actualización Dinámica de la Base de Conocimientos: La base de conocimientos se actualizaba regularmente para incluir nuevas tendencias y preferencias emergentes, lo que aseguraba que las recomendaciones siguieran siendo relevantes y precisas.
3. Perfil del Usuario Detallado: El sistema creaba perfiles detallados que iban más allá de los datos básicos, incorporando información compleja sobre preferencias y comportamientos para mejorar la precisión de las recomendaciones.

2.2.4. Sistemas de recomendación híbridos

Un cuarto tipo de SR, sería el SR híbrido, que combina diferentes técnicas como las mencionadas anteriormente.

- Fusión de Modelos: Estos sistemas combinan múltiples enfoques de recomendación, como filtrado colaborativo y basado en contenido, para mejorar la precisión y la cobertura de las recomendaciones. Por ejemplo, un sistema podría fusionar las recomendaciones de un filtro colaborativo con las de un filtro basado en contenido para ofrecer sugerencias más diversas y precisas.
- Fusión de Resultados: En estos sistemas, las recomendaciones de varios sistemas de recomendación independientes se combinan para generar una lista final de recomendaciones. Por ejemplo, un sistema podría combinar las recomendaciones de un sistema basado en contenido con las de un sistema basado en conocimiento para ofrecer una gama más amplia de opciones al usuario.
- Filtrado a Nivel de Características: Estos sistemas combinan características de diferentes tipos de sistemas de recomendación para mejorar la calidad de las recomendaciones. Por ejemplo, un sistema podría utilizar el filtrado colaborativo para recomendar películas populares entre usuarios similares y luego refinar esas recomendaciones utilizando el filtrado basado en contenido para sugerir películas que comparten características específicas con las preferidas por el usuario. (Jannach et al., 2013)(Burke, 2002)

2.2.5. Sistema de recomendación basado en Aprendizaje profundo

El Aprendizaje profundo, ha supuesto un cambio de paradigma en los SRs a la hora de crear modelos más sofisticados que pueden obtener patrones más complejos en los datos de

interacción entre usuarios y elementos, del que es un exponente las Redes Neuronales Convolucionales (CNNs):(Saxena, 2022)

Las redes neuronales convolucionales (CNNs) son conocidas habitualmente por su efectividad en el procesamiento de datos de imágenes (ámbito médico), pero también se aplican con éxito en SRs. En este contexto, las CNNs pueden aprender representaciones profundas de los datos de interacción entre usuarios y elementos, lo que permite capturar patrones espaciales y temporales complejos.

En este contexto de SRs, las CNNs pueden aplicarse habitualmente de varias maneras:

- **Procesamiento de secuencias:** Cuando los datos de interacción están en forma de secuencias (por ejemplo, secuencias temporales de compras o secuencias de clics de usuario), las CNNs pueden extraer características relevantes a lo largo de estas secuencias, capturando relaciones temporales importantes.
- **Procesamiento de imágenes y texto:** En algunos casos, los elementos en un sistema de recomendación pueden representarse como imágenes o texto (por ejemplo, portadas de películas o descripciones de productos). Las CNNs pueden extraer características significativas de estas representaciones, lo que permite capturar información relevante para hacer recomendaciones.
- **Modelos híbridos:** Se pueden diseñar modelos híbridos que combinen CNNs con otros tipos de arquitecturas de aprendizaje profundo, como redes neuronales recurrentes (RNNs) o redes neuronales totalmente conectadas (DNNs), para aprovechar las fortalezas de cada tipo de arquitectura.

Las CNNs son una herramienta importante en el arsenal de técnicas de aprendizaje profundo para los SRs, ya que pueden aprender representaciones profundas de datos de interacción complejos y heterogéneos no visibles a simple vista. Esto permite construir modelos más precisos y efectivos para una recomendación personalizada. (Benedikt Schifferer, 2021)

Otro tipo de técnica de aprendizaje profundo habitual, serían las GNNs, diseñadas específicamente para trabajar con datos estructurados en forma de grafos. En el contexto de SRs, los grafos pueden representar diferentes relaciones entre elementos como usuarios, elementos recomendados y sus interacciones. (Zhou et al., 2022)

Las GNNs son eficaces porque pueden capturar la estructura y las relaciones entre los nodos en un grafo, lo que las hace ideales para los SRs basados en datos de interacción entre usuarios y elementos. Estas redes pueden aprender a representar nodos en el grafo de manera que los nodos similares (por ejemplo, usuarios con gustos similares) estén cerca en el espacio de representación (de manera similar a las CNNs).

Una de las ventajas clave de las GNNs es su capacidad para propagar y combinar información a lo largo de las aristas del grafo, lo que les permite capturar información contextual y relacional para realizar recomendaciones más precisas.

Las GNNs son una técnica de aprendizaje profundo diseñada para trabajar con datos estructurados en forma de grafos, y en los SRs, pueden aprovechar la estructura de las interacciones entre usuarios y elementos para generar recomendaciones más precisas y personalizadas. (Wu et al., 2021)

2.3. Estado del arte en los SR para el ámbito del *retail*

Una vez realizada la visión anterior, que tenía un enfoque generalista, se ha querido incluir una visión específica del ámbito de este trabajo, que es el de los *retailers*.

La pregunta que se ha planteado es la siguiente: ¿Qué técnicas, aplicaciones y desafíos se han investigado en los sistemas de recomendación en el ámbito de *retail*, en los últimos dos años?

Las fuentes de información que se han utilizado para la búsqueda de literatura han sido las siguientes:

- Bases de datos académicas: *Google Scholar*
- Revistas científicas especializadas:
 - IEEE Transactions on Knowledge and Data Engineering
 - User Modeling and User-Adapted Interaction
 - Journal of Retailing
 - International Journal of Electronic Commerce
 - Electronic Commerce Research and Applications

De las fuentes consultadas se han seleccionado tres trabajos, y de los mismos se ha extraído la siguiente información:

- El nombre del trabajo y el año de publicación
- Algoritmos y técnicas de IA utilizadas
- La aplicación o el dominio de *retail* que se ha abordado
- Los desafíos que se han identificado
- Las principales conclusiones que se han obtenido

2.3.1. A comprehensive review of recommender systems: transition from theory to practice

Aunque los Sistemas de Recomendación fundacionales (CF, CBF y los métodos híbridos) forman el núcleo de muchas soluciones de recomendación personalizadas, su capacidad para abordar desafíos prácticos en sectores como el comercio electrónico, el entretenimiento, las noticias, el turismo, las finanzas, la salud y el *e-learning* es, a menudo, limitada. (Raza et al., 2024), por ejemplo, el CF puede personalizar las experiencias de los usuarios aprovechando los datos de comportamiento en un sistema de recomendación de noticias o un sistema de recomendación de música, pero tiene dificultades con la naturaleza dinámica de las preferencias de los usuarios y la necesidad de recomendaciones en tiempo real.

Las plataformas de comercio electrónico enfrentan el desafío de personalizar la experiencia de compra mediante la recomendación de productos en tiempo real, gestionando grandes volúmenes de datos y adaptándose a las preferencias cambiantes de los consumidores (Raza et al., 2024a). La personalización debe considerar factores como el tiempo, la temporada, la ubicación y la situación actual del usuario. Por ejemplo, las recomendaciones para un nuevo padre que compra productos para bebés serán significativamente diferentes de las de un amante de los libros. Introducir diversidad y novedad en las recomendaciones es crucial para mantener la experiencia fresca y atractiva.

El documento que se presenta es una revisión sistemática de la literatura sobre los sistemas de recomendación en el ámbito del comercio electrónico, con especial énfasis en los últimos avances y desafíos que plantean las técnicas basadas en redes neuronales. En este sentido, las técnicas basadas en redes neuronales han surgido como una alternativa prometedora, capaz de modelar de forma flexible y eficiente las relaciones no lineales y de alta dimensión que existen entre los usuarios y los productos. Entre estas técnicas, se destaca el uso de las redes neuronales de grafos (GNN), que permiten representar y procesar los datos estructurados en

forma de grafos, como los que se encuentran en muchos sistemas de recomendación. Las GNN pueden capturar las características y las dependencias de los nodos y las aristas del grafo, así como propagar la información a través de las diferentes capas de la red. El documento revisa los principales trabajos que han aplicado las GNN a los sistemas de recomendación, analizando sus ventajas, desafíos y oportunidades de investigación futura.

Las GNN pueden abordar de manera efectiva varios desafíos prácticos al modelar relaciones complejas en los datos. En el comercio electrónico, modelos como LightGCN, GC-MC, NGCF y Graph-ICF mejoran la personalización, escalabilidad y eficiencia para la predicción de calificaciones, enlaces y artículos. Estos modelos son capaces de manejar catálogos de productos extensos y grandes bases de usuarios de manera eficiente. En redes sociales, GNNs como GNN-SoR y GraphRec mejoran las predicciones de interacción de los usuarios, aumentando la relevancia del contenido y el compromiso de los usuarios al comprender las dinámicas sociales y las relaciones entre usuarios.

Esta tabla siguiente, se detalla las características de cada modelo, incluyendo las Características de entrada, año de publicación y características como escalabilidad, interpretabilidad, eficiencia y reproducibilidad (evaluadas como Alta, Media o Baja; el símbolo '—' indica que no hay información disponible para esto). También se enumeran el Conjunto de Datos Utilizado, Métricas de Evaluación, Precisión del Modelo (según la métrica de evaluación de la columna anterior), Tarea de Aprendizaje y Campo de Aplicación:

Figura 3. *Comprehensive Overview of Graph Neural Network Models across Various Metrics and Use Cases.*

Model	Year	Input Data	Scalability, Interpretability, Efficiency, Reproducibility	Dataset	Evaluation Metrics	Model Accuracy	Application
GCN[202]	2015	MovieLens	High, Medium, High, High	MovieLens	MSE	MovieLens: 0.5	e-commerce
GC-MC[189]	2017	MovieLens-1M, MovieLens-10M, Flixster, Douban, Yahoo Music	High, Low, -, High	MovieLens-1M, MovieLens-10M, Flixster, Douban, Yahoo Music	RMSE	MovieLens-1M: 0.832 MovieLens-10M: 0.777 Flixster: 0.941 Douban: 0.734 Yahoo Music: 20.5	e-commerce
NGCF[208]	2019	Gowalla, Yelp2018, Amazon-books	-, Low, Medium, High	Gowalla, Yelp2018, Amazon-books	Recall, NDCG	Gowalla: 0.1569/0.1327 Yelp2018: 0.0579/0.0477 Amazon-books: 0.0337/0.0261	e-commerce
Graph-ICF[193]	2022	MovieLens-1M, Pinterest-20, Yelp	-, Low, Medium, High	MovieLens-1M, Pinterest-20, Yelp	HR, NDCG, MAP	MovieLens-1M: 0.7425/0.4555/0.3721 Pinterest-20: 0.8987/0.5830/0.4873 Yelp: 0.7519/0.4856/0.4033	e-commerce
GNN-SoR[169]	2020	Epinions, Yelp, Flixster	-, -, -, Low	Epinions, Yelp, Flixster	RMSE, MAE, NDGC	Epinions: 0.880/0.791/0.792 Yelp: 0.820/0.871/0.687 Flixster: 0.863/0.859/0.594	Social Network RecSys, e-commerce
GCM[188]	2022	Yelp-NC, Yelp-OH, Amazon-book	-, Low, Medium, High	Yelp-NC, Yelp-OH, Amazon-book	HR@10, NDGC@10	Yelp-NC: 0.1046/0.0557 Yelp-OH: 0.2648/0.1457 Amazon-book: 0.0968/0.0536	e-commerce
GCF-YA[170]	2019	MovieLens-1M, MovieLens-10M, Taobao	-, Low, -, Low	MovieLens-1M, MovieLens-10M, Taobao	HR@10, NDGC@10	MovieLens-1M: 0.7818/0.4873 MovieLens-10M: 0.7642/0.4677 Taobao: 0.3662/0.2491	e-commerce
DGSR[179]	2023	Beauty, Games, CDs	-, Low, -, High	Beauty, Games, CDs	NDCG@10, Hit@10	Beauty: 52.4/35.9 Games: 75.57/55.7 CDs: 72.43/51.22	e-commerce
GraphRec[191]	2019	Ciao, Epinions	-, Low, -, High	Ciao, Epinions	MAE, RMSE	Ciao: 0.7387/0.9794 Epinions: 0.8441/1.0878	e-commerce
KGAT[197]	2019	Amazon-book, Last-FM, Yelp2018	-, High, High, High	Amazon-book, Last-FM, Yelp2018	Recall@20, NDCG@20	Amazon-book: 0.1489/0.1006 Last-FM: 0.0870/0.1325 Yelp2018: 0.0712/0.0867	e-commerce

Fuente: (Raza et al., 2024b)

2.3.2. Cross-Border E-Commerce Intelligent Information Recommendation System Based on Deep Learning

Este trabajo (L. Li, 2022) propone un método de filtrado colaborativo basado en redes neuronales para generar recomendaciones personalizadas a los consumidores en el sector del *retail*. El artículo plantea el problema de la escasez de datos y la diversidad de preferencias de los usuarios, que dificultan la aplicación de los métodos tradicionales de filtrado colaborativo.

El método propuesto utiliza una red neuronal profunda que aprende las características latentes de los usuarios y los productos a partir de los datos de interacción, y que puede adaptarse a diferentes escenarios de *retail*. El artículo presenta los algoritmos utilizados para entrenar y evaluar.

El filtrado colaborativo es una técnica ampliamente utilizada para generar recomendaciones personalizadas a los usuarios en diferentes dominios, como el comercio electrónico, el entretenimiento o las redes sociales. El filtrado colaborativo se basa en la idea de que los usuarios que han tenido preferencias similares en el pasado, tendrán preferencias similares en el futuro, y que los productos que han gustado a un usuario, también le gustarán a otros usuarios con preferencias similares. El filtrado colaborativo utiliza los datos de interacción entre los usuarios y los productos, como las calificaciones, las compras o los clics, para aprender las características latentes de los usuarios y los productos, y para predecir la preferencia de un usuario por un producto que no ha visto o comprado.

El sector del *retail* es uno de los dominios más importantes para la aplicación del filtrado colaborativo, ya que las recomendaciones personalizadas pueden aumentar las ventas, la satisfacción y la fidelidad de los consumidores. Sin embargo, el sector del *retail* también presenta algunos desafíos específicos para el filtrado colaborativo, como la escasez de datos, la diversidad de preferencias de los usuarios, la estacionalidad de los productos, la competencia entre los productos y la necesidad de generar recomendaciones diversas y novedosas. Estos desafíos hacen que los métodos tradicionales de filtrado colaborativo, como los basados en factorización de matrices o en vecindarios, no sean suficientes para capturar la complejidad de las interacciones entre los usuarios y los productos en el sector del *retail*.

En este contexto, el trabajo propone un método de filtrado colaborativo basado en redes neuronales para generar recomendaciones personalizadas a los consumidores en el sector del *retail*. El método utiliza una red neuronal profunda que aprende las características latentes de los usuarios y los productos a partir de los datos de interacción, y que puede adaptarse a diferentes escenarios de *retail*. El método se basa en la hipótesis de que las redes neuronales pueden aprender representaciones más ricas y flexibles de los usuarios y los productos, y que pueden capturar las relaciones no lineales y complejas entre ellos. El método también incorpora mecanismos de atención y diversidad para generar recomendaciones más relevantes y diversas para los consumidores.

El artículo presenta los algoritmos utilizados para entrenar y evaluar la red neuronal, y muestra los resultados experimentales en dos conjuntos de datos reales de *retail*: *Amazon* y *Taobao*. El artículo compara el método propuesto con los métodos basados en factorización de matrices y en redes neuronales convolucionales, y demuestra que el método propuesto supera a los métodos comparados en términos de precisión, cobertura, diversidad y novedad de las recomendaciones. El artículo concluye que el método propuesto tiene una alta aplicabilidad en el sector del *retail*, ya que puede generar recomendaciones personalizadas a los consumidores que se ajustan a sus preferencias y necesidades.

Tipo de filtrado

El tipo de filtrado utilizado en el artículo es el filtrado colaborativo basado en redes neuronales. El filtrado colaborativo basado en redes neuronales es un tipo de filtrado colaborativo que utiliza redes neuronales para aprender las características latentes de los usuarios y los productos a partir de los datos de interacción, y para predecir la preferencia de un usuario por un producto. El filtrado colaborativo basado en redes neuronales se diferencia de los métodos tradicionales de filtrado colaborativo, como los basados en factorización de matrices o en vecindarios, en que las redes neuronales pueden aprender representaciones más ricas y flexibles de los usuarios y los productos, y que pueden capturar las relaciones no lineales y complejas entre ellos. El filtrado colaborativo basado en redes neuronales también se diferencia de los métodos basados en contenido, que utilizan las características explícitas de los usuarios y los productos, en que las redes neuronales pueden aprender las características implícitas de los usuarios y los productos, que pueden ser más relevantes para la generación de recomendaciones.

Aplicabilidad en el mundo del *retail*

El método propuesto tiene una alta aplicabilidad en el mundo del *retail*, ya que puede generar recomendaciones personalizadas a los consumidores que se ajustan a sus preferencias y necesidades. El método puede adaptarse a diferentes escenarios de *retail*, como el online, el offline o el omnicanal, y puede utilizar diferentes tipos de datos de interacción, como las calificaciones, las compras, los clics, los carritos o las visitas. El método puede generar recomendaciones para diferentes tipos de productos, como los de moda, los de electrónica, los de alimentación o los de ocio, y puede tener en cuenta la estacionalidad, la disponibilidad y la competencia de los productos. El método puede generar recomendaciones diversas y

novedosas, que pueden aumentar la satisfacción y la fidelidad de los consumidores, y que pueden impulsar las ventas, el margen y el beneficio de los *retailers*.

Conclusiones

El trabajo propone un método de filtrado colaborativo basado en redes neuronales para generar recomendaciones personalizadas a los consumidores en el sector del *retail*. El artículo plantea el problema de la escasez de datos y la diversidad de preferencias de los usuarios, que dificultan la aplicación de los métodos tradicionales de filtrado colaborativo. El método propuesto utiliza una red neuronal profunda que aprende las características latentes de los usuarios y los productos a partir de los datos de interacción, y que puede adaptarse a diferentes escenarios de *retail*. El artículo presenta los algoritmos utilizados para entrenar y evaluar la red neuronal, y muestra los resultados experimentales en dos conjuntos de datos reales de *retail*. El artículo concluye que el método propuesto supera a los métodos basados en factorización de matrices y en redes neuronales convolucionales, y que tiene una alta aplicabilidad en el sector del *retail*, ya que puede generar recomendaciones relevantes y diversas para los consumidores.

2.3.3. Text Understanding and Generation Using Transformer Models for Intelligent E-commerce Recommendation.

Este artículo de reciente publicación, febrero de 2024 (Xiang et al., n.d.) presenta una introducción a los modelos de lenguaje natural basados en *transformers*, que son una arquitectura de redes neuronales que ha revolucionado el campo del procesamiento del lenguaje natural (PLN) en los últimos años. Los autores explican los conceptos básicos de los *transformers*, sus ventajas sobre otras arquitecturas, y sus aplicaciones en diversas tareas de PLN, como la generación de texto, la comprensión de lectura, la traducción automática, el análisis de sentimientos, la clasificación de textos, la respuesta a preguntas, y la extracción de información.

Los sistemas de recomendación son una herramienta clave para el sector del *retail*, ya que permiten ofrecer a los clientes productos personalizados y relevantes, aumentando así la satisfacción y las ventas. Sin embargo, la mayoría de los sistemas de recomendación actuales se basan en técnicas de filtrado colaborativo o basado en contenido, que tienen algunas

limitaciones, como la escasez de datos, el sesgo de popularidad, la falta de diversidad o la incapacidad de capturar las preferencias dinámicas de los usuarios. El artículo propone el uso de un tipo de filtrado basado en contenido, su modelo Behavior Sequence Transformer (BST) al que añaden la capacidad de los Transformers para capturar las relaciones semánticas y contextuales entre los ítems y los usuarios, y generar recomendaciones más precisas y personalizadas.

El artículo muestra que los algoritmos basados en *Transformers* superan a los métodos más tradicionales, a través de una comparativa del modelo BST, con otros modelos de recomendación DIN y WDL, (modelos de recomendación basados en filtrado colaborativo). Además, destaca la aplicabilidad de los *transformers* en el mundo del *retail*, donde los sistemas de recomendación son una herramienta clave para aumentar las ventas, la satisfacción, y la fidelidad de los clientes. Los *transformers* pueden ayudar a los *retailers* a ofrecer recomendaciones más relevantes, personalizadas, y explicables a los clientes, y a adaptarse a sus cambios de preferencias y comportamientos.

Por último, concluye que los *transformers* son una arquitectura de vanguardia para el procesamiento de lenguaje natural, que ofrece una gran flexibilidad y potencial para resolver diversas tareas y problemas. Se concluye sugiriendo que los *transformers* seguirán evolucionando y mejorando, y que se necesitan más investigaciones para explorar sus límites, desafíos, y oportunidades.

2.4. Conferencia de Sistemas de Recomendación ReCSys 2022

Para concluir el estado del arte, se incluye a continuación una visión de la conferencia ACM sobre Sistemas de recomendación (RecSys) dado que es el foro internacional más destacado para la presentación de nuevos resultados de investigación, sistemas y técnicas en el amplio campo de los sistemas de recomendación. RecSys reúne a los principales grupos de investigación internacionales que trabajan en sistemas de recomendación, junto con muchas de las empresas líderes mundiales activas en el comercio electrónico y otros dominios adyacentes¹. Se ha convertido en la conferencia anual más importante para la presentación y discusión de la investigación en sistemas de recomendación. Para el presente trabajo, se va a

¹ [RecSys 2022 \(Seattle\) - RecSys – RecSys \(acm.org\)](https://www.acm.org/recsys)

centrar la atención en la decimosexta Conferencia anual sobre Sistemas de recomendación (Association for Computing Machinery, New York) celebrado en Seattle en el 2022, un encuentro internacional en el que se analiza cada año, las tendencias y nuevas investigaciones a nivel internacional de los sistemas de recomendación. Se va a poner el foco en esta penúltima celebración, la del 2022, (existe otro más actual en el 2023 celebrada en Singapur) por que fue más orientada a las recomendaciones de moda en ese año, que es ideal para el tema que ocupa este proyecto.

Además de conferencias, foros de discusión, estado del arte, talleres, etc., hay un programa competitivo en el que diferentes grupos internacionales de trabajo e investigación ponen en práctica las últimas técnicas y tendencias para resolver un posible reto empresarial en concreto. Para resolver el reto propuesto el año 2022 por *Dressipi*², el grupo ganador que resolvió el reto, lo llevó a cabo con una solución basada en GNN (Redes neuronales basadas en grafos) (Zzh et al., 2022) como principal técnica de aprendizaje automático.

El equipo ganador del desafío RecSys de 2022 fue el formado por investigadores de la Universidad de Zhejiang, la Universidad de Fudan y Alibaba Group, bajo el nombre de Zzh. Su solución se basó en una red neuronal basada en grafos (GNN) que modelaba tanto las interacciones de los usuarios con los artículos de moda como las relaciones entre los artículos y sus atributos. La GNN constaba de dos componentes principales: un módulo de codificación de grafos y un módulo de decodificación de grafos. El módulo de codificación de grafos tomaba como entrada el grafo de sesiones de usuario, que conectaba los nodos de usuario y artículo con aristas ponderadas por el tiempo y el tipo de interacción, y el grafo de atributos de artículo, que conectaba los nodos de artículo con nodos de atributo (como categoría, color, estilo, etc.) con aristas ponderadas por la frecuencia y la importancia. El módulo de codificación de grafos aprendía una representación vectorial de baja dimensión para cada nodo mediante la propagación de mensajes entre los nodos vecinos. El módulo de decodificación de grafos tomaba como entrada las representaciones vectoriales de los nodos y predecía la probabilidad de compra de cada artículo candidato para un usuario dado. El módulo de decodificación de grafos utilizaba una capa de atención para ponderar la importancia de los diferentes atributos de los artículos y una capa de agregación para

² <https://dressipi.com/>

combinar la información de los artículos y los usuarios. La solución de Zzh logró un rendimiento superior al de los otros participantes del desafío, demostrando la eficacia de las GNN para capturar las complejas dependencias y preferencias que subyacen a las sesiones de compra de moda.

El reto que *Dressip* propuso se basó en un conjunto de que tenía más de un millón de sesiones: Este *dataset* proporcionaba una base para explorar cómo las sesiones de navegación y las características descriptivas de los productos pueden influir en las decisiones de compra finales. El Desafío RecSys 2022, se centró en la tarea de predecir la compra final de un usuario dentro de una sesión de navegación. Este desafío pone a prueba la capacidad de los sistemas de recomendación para interpretar señales complejas de comportamiento de usuario y preferencias implícitas, lo cual es crucial para el diseño de recomendadores más precisos y personalizados. La construcción del *dataset*, que incluía etiquetas descriptivas asignadas por expertos en moda, ofrecía una oportunidad única para que los sistemas de recomendación incorporen conocimiento de dominio en sus predicciones. Las estadísticas del *dataset* revelan la diversidad y la riqueza de las sesiones de compra, lo que subraya la importancia de considerar una amplia gama de factores al desarrollar algoritmos de recomendación.

2.5.Conclusiones

Los SR han avanzado en gran medida desde los primeros modelos de la década 2010 al 2020, hasta ahora en la actualidad se ha experimentado un auge sin precedentes con las redes neuronales basadas en grafos (GNN)(Zhou et al., 2022) y sus tipos (Convolucionales, Auto-Codificadores, Recurrentes, Con Puertas)(Lops Pasqualeand de Gemmis, 2011; Ruiz et al., 2020; Saxena, 2022; Daniel et al., 2023) que se han convertido en la más innovadora aproximación para los SR con un conjunto de algoritmos aplicables según su escenario.

Los datos de entrada del SR son transformados a una forma de grafo, los nodos representando a los artículos y las aristas representando las relaciones. Demostrando un gran potencial para representar y analizar relaciones complejas entre el par producto/cliente para utilizarlas para incrementar la precisión y efectividad de los SR donde se aplica. (Zhou et al., 2022; Zzh et al., 2022)

Los SR todavía se enfrentan a desafíos, como el sesgo en los datos y la sobre personalización. Se espera que estos sistemas integren técnicas más avanzadas de inteligencia artificial, como el procesamiento del lenguaje natural (Redes Neuronales de Grafos con Puertas -GGNN-)(Q. Li et al., 2022) y la comprensión del contexto, para ofrecer recomendaciones aún más precisas y relevantes.

Otros escenarios destacables se centran en los sistemas tradicionales de filtrado colaborativo que se enfrentan al desafío significativo de escenarios de inicio en frío, donde hay pocas interacciones usuario-ítem para aprender.

Las técnicas más recientes han intentado abordar estos retos utilizando información de modalidad (como texto o imágenes de ítems) con codificadores de modalidad pre entrenados y LLMs. (Sanner et al., 2023). En concreto *All-round LLM-based Recommender System* (A-LLMRec), busca ser eficiente en escenarios tanto fríos como cálidos. La idea principal es permitir que un LLM aproveche directamente el conocimiento colaborativo de un sistema de recomendación basado en filtrado colaborativo preentrenado (CF-RecSys). Esto se logra mediante la utilización conjunta de la capacidad emergente del LLM y los *embeddings* de alta calidad de usuario/ítem entrenados por el CF-RecSys.(Hua et al., 2023)

En definitiva, y como queda de manifiesto, por los trabajos examinados, los modelos unitarios por si solos no logran el grado de efectividad definitiva por los SRs, para los cuales existen una variedad de diferentes líneas de investigación a lo hora de abordar este tema tan dinámico, en constante evolución. Todas las líneas siguen un patrón de combinación de modelos y técnicas (*boosting*/hibridación, árboles de decisión, por un lado y redes neuronales Transformers con aprendizaje auto-supervisado) de *machine learning*. Tómese como ejemplo un examen a los trabajos ganadores del RecSys 2023(Basso et al., 2023; Lu & Volkovs, 2023), que plasman de una manera práctica las múltiples líneas de trabajo para este reto de aplicación/ámbito empresarial, en un campo de aplicación cambiante y evolutivo de los SRs.

Aunque de lo investigado se desprende que la combinación de modelos y el uso de LLM es la actual tendencia, el trabajo a desarrollar en este TFM tiene que ser enfocado con una estrategia sensiblemente diferente, ya que los LLM están diseñados principalmente para trabajar con texto, y nuestro *dataset* no contiene texto en el sentido tradicional (como

descripciones de productos o reseñas). En su lugar, tenemos características y datos de sesiones categorizadas, que conducen a las técnicas desarrolladas en los próximos capítulos

3. Objetivos concretos y metodología de trabajo

Este tercer capítulo es el puente entre el estudio del dominio y la contribución a realizar. Se expondrán el objetivo general, a continuación los objetivos específicos y finalmente la metodología de trabajo.

3.1. Objetivo general

A continuación, se presenta el objetivo general, para ello se ha seguido una formulación SMART:

- **Específico:** Desarrollar un sistema basado en inteligencia artificial llamado MarIA, un *personal shopper*, que permita a los consumidores interactuar mediante lenguaje natural y simule la interacción con un vendedor humano, proporcionando recomendaciones personalizadas de productos dentro del catálogo de moda disponible en línea.
- **Medible:** El éxito del sistema se medirá a través de la métrica *Mean Reciprocal Rank* (MRR), que es adecuada para evaluar la calidad de las recomendaciones en sistemas de recuperación de información y clasificación multiclase. MRR calcula la media de los recíprocos de la posición del primer resultado relevante en una lista ordenada de recomendaciones, proporcionando un valor que refleja la eficacia con la que el sistema identifica los productos de mayor interés para el consumidor. Un valor más alto de MRR indica que los productos relevantes se posicionan más cerca del principio de la lista de recomendaciones. Esta medida y la evaluación del sistema se detalla en el apartado 3.3.7.
- **Alcanzable:** El sistema utilizará técnicas de aprendizaje automático para hacer predicciones y mejorar la experiencia del cliente en línea, el usuario deberá poder expresarse en lenguaje natural.
- **Relevante:** El objetivo es asegurar que la experiencia del cliente en línea sea similar a la de una tienda física, con el PS extrayendo la información necesaria a través de la conversación y la sesión del cliente para hacer recomendaciones adaptadas a sus necesidades y estilo. Para ello, el sistema deberá capturar las necesidades dinámicas del usuario, las que surgen y cambian durante el proceso de compra, influenciadas por el contexto, el estado de ánimo, las emociones y las preferencias del usuario.
- **Limitado en el tiempo:** El sistema debería ser desarrollado e implementado dentro de un plazo específico, con actualizaciones y evaluaciones regulares del progreso para asegurar

que cumpla con los objetivos deseados, entregando valor en cada iteración, para ello se seguirá una metodología de *sprints*.

3.2. Objetivos específicos

- Analizar el estado del arte de los sistemas de recomendación en el ámbito del *retail* utilizando fuentes académicas y científicas confiables.
- Desarrollar un sistema de recomendación que mejore la experiencia del cliente permitiéndole actuar con el mismo en un lenguaje natural.
- Analizar los datos disponibles y probar diferentes técnicas de mejora de datos y técnicas de recomendación, así como evaluarlos.
- Diseñar e implementar el módulo de recomendación realizando los experimentos oportunos que permitan elegir el que ofrezca mayor precisión en la recomendación.
- Diseñar e implementar el módulo de interacción con el consumidor estableciendo las preguntas que se emplearán en él. El mismo se fundamenta en el uso de Modelos de Lenguaje de Última Generación (LLM) y arquitectura *transformers*.
- Describir la infraestructura tecnológica subyacente que soporte el sistema de recomendación.

3.3. Metodología del trabajo

En el presente proyecto, se ha llevado a cabo una metodología separada por etapas secuenciales, las cuales se superaban al alcanzar distintos hitos. Al mismo tiempo, la documentación ha sido desarrollada de forma paralela y dinámica, abierta a cambios a medida que se iban logrando avances técnicos y conceptuales.

Se contempla una fase previa: inspirada en las etapas tempranas de CRISP-DM, para dar respuesta al entendimiento del negocio, los datos y la preparación de estos, con objeto de poner al dato en el centro de la estrategia y encontrar el valor en etapas tempranas. Se destacan los aspectos de evaluación de la fuente en el que se propone evaluar la idoneidad del *dataset* actual, como la calidad y fiabilidad de los datos, la capacidad de cruce de la información y la coherencia de la información.

3.3.1. Búsqueda Bibliográfica

Se realizó una exhaustiva búsqueda bibliográfica sobre los dos grandes bloques que componen este trabajo: los sistemas de recomendación y las técnicas de procesamiento de lenguaje natural (PLN). Para ello, se consultaron numerosos documentos académicos, revistas científicas y ponencias en conferencias especializadas como RecSys. Los resultados de esta revisión bibliográfica nos permitieron descubrir las diversas técnicas actualmente utilizadas para resolver problemas similares al nuestro, además de ayudarnos a estructurar e interrelacionar los conocimientos previos.

3.3.2. Selección y Adecuación del *Dataset*

Una vez definido el propósito y el alcance de nuestro trabajo, se inició la búsqueda de un *dataset* adecuado. Se consultaron múltiples conferencias, competiciones y *datasets* públicos, y se realizó un análisis exploratorio de datos (EDA) de los más relevantes. Finalmente, se eligió el *dataset* del RecSys Challenge 2022, cuyos detalles y modificaciones se tratan ampliamente en el apartado 4.3.1.

3.3.3. Selección de Técnicas y Algoritmos

Conociendo las técnicas más relevantes en cada uno de los campos y definida la "materia prima" con la que trabajar, procedimos a seleccionar las técnicas y algoritmos más prometedores para nuestro cometido. Esta selección se basó en la revisión bibliográfica y en la adecuación al *dataset* seleccionado.

3.3.4. Creación del Sistema de Recomendación de Productos

En esta etapa, se desarrolló el modelo de recomendación de ropa y calzado utilizando el *dataset* seleccionado. Este modelo es la base para nuestro sistema de recomendación.

3.3.5. Integración de Procesamiento de Lenguaje Natural (PLN)

Una vez desarrollado el sistema de recomendación básico, se integraron técnicas de procesamiento de lenguaje natural para mejorar la interacción con el usuario. Esto incluyó la decodificación de las características de los artículos (*feature_category_id* y *feature_value_id*) para crear una experiencia de compra más personalizada.

3.3.6. Desarrollo del Sistema de Recomendación Conversacional

El siguiente paso fue desarrollar un sistema de recomendación conversacional. Este sistema permite a los usuarios interactuar de manera natural con el sistema de recomendación a través de lenguaje natural. Se implementaron técnicas avanzadas de PLN para interpretar y responder a las consultas de los usuarios de forma eficiente.

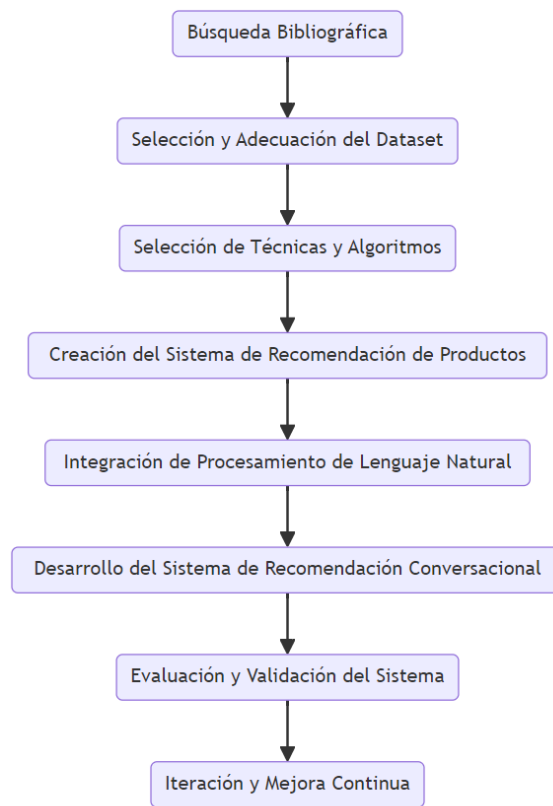
3.3.7. Evaluación y Validación del Sistema

Finalmente, se evaluó el sistema desarrollado. Se utilizaron métricas de evaluación estándar, incluyendo la *Mean Reciprocal Rank (MRR)*, para medir la precisión y efectividad del sistema de recomendación. El MRR es una métrica que evalúa la calidad de las recomendaciones al considerar la posición del artículo comprado en la lista de predicciones. Cuanto más alta sea la posición del artículo correcto, mejor será el puntaje MRR. Además, se realizaron pruebas con usuarios reales para validar la usabilidad y la experiencia de usuario del sistema conversacional.

3.3.8. Iteración y Mejora Continua

Basado en los resultados de la evaluación y el *feedback* de los usuarios, se realizarán iteraciones para mejorar continuamente el sistema.

Figura 4. *Diagrama de fases de trabajo*



Fuente: Elaboración propia

4. Desarrollo específico de la contribución

Este capítulo presenta el desarrollo específico de la contribución realizada en este trabajo, que consiste en una propuesta metodológica para la generación de sistemas de recomendación basados en datos multivariantes. Esta propuesta se divide en dos módulos principales: el de recomendación y el que interactúa con el usuario. A continuación, se identifican los requisitos necesarios para llevarlas a cabo, así como se da una descripción de las actividades realizadas en cada módulo:

4.1. Identificación de requisitos

A continuación, se presentan los requisitos planteados para el desarrollo del trabajo:

1. Definir el problema de recomendación que se quiere resolver, especificando el tipo de datos, los objetivos y las métricas de evaluación.
2. Realizar un análisis exploratorio de los datos proporcionados por el RecSys Challenge, identificando sus características, distribuciones, patrones y anomalías.
3. Preprocesar y limpiar los datos, aplicando las técnicas adecuadas para eliminar el ruido, los valores faltantes, los *outliers* y los duplicados.
4. Realizar el procesamiento de las interacciones que representan las sesiones de usuarios. Esto implica limpiar y transformar datos, desarrollar características adicionales si se requiere, y organizar el conjunto de datos final para entrenarlo, para tener un conjunto estructurado y listo para entrenarlo.
5. El sistema de recomendación debe utilizar una matriz de calificaciones *usuario-item* (URM, *User-Item Rating Matrix*) (Aggarwal, 2016). para representar las interacciones entre las sesiones y los artículos, utilizando calificaciones o también llamados scores implícitas basadas en el comportamiento de los usuarios. La URM debe ser compatible con métodos de filtración colaborativa basados en redes neuronales (GNN) que aprendan factores latentes para las sesiones y los artículos.
6. Diseñar e implementar un algoritmo que tome como entrada la matriz de calificaciones *usuario-item* (URM) y produzca como salida predicciones de interés para cada par *usuario-item*. El algoritmo debe utilizar métodos de grafos para aprender factores latentes que capturen la relación entre usuarios/sesiones e *items*/artículos. El algoritmo debe ser

entrenado y optimizado, ajustando los hiperparámetros, las funciones de pérdida y las técnicas de regularización para mejorar el rendimiento y evitar el sobreajuste.

7. Entrenar y optimizar el modelo, ajustando los hiperparámetros y las técnicas de regularización para mejorar el rendimiento y evitar el sobreajuste.
8. Presentar el modelo, explicando su arquitectura, funcionamiento, ventajas y limitaciones, así como los resultados obtenidos y las posibles mejoras futuras.

4.2.Descripción del Data Set

El RecSys Challenge es una competición anual organizada por la comunidad de sistemas de recomendación, donde investigadores y profesionales del área compiten para desarrollar los mejores algoritmos de recomendación. Cada edición de la competición se centra en un problema específico relacionado con sistemas de recomendación, ofreciendo un entorno competitivo para que los participantes pongan a prueba sus habilidades y técnicas en la resolución de problemas reales del sector.

El objetivo principal de estos desafíos es avanzar en el campo de los sistemas de recomendación mediante la colaboración y la competitividad, promoviendo la innovación y la investigación aplicada. Los participantes tienen la oportunidad de trabajar con *datasets* reales proporcionados por empresas y organizaciones, lo que añade un valor práctico significativo a los resultados obtenidos.

Por estos motivos y dada la naturaleza de nuestro proyecto centrado en el *e-commerce* se decidió utilizar el *dataset* del *Recsys Challenge 2022*.

Este *dataset* consta de 1.1 millones de sesiones de venta en línea en el dominio de la moda, recopiladas a lo largo de un periodo de 18 meses. Las sesiones incluidas son "sesiones de compra", o sea, sesiones que provocaron la compra de al menos un artículo. Los artículos visualizados y comprados incluyen ropa y calzado. Además, este *dataset* contiene datos de contenido para cada uno de los artículos visualizados y comprados, que representan etiquetas descriptivas asignadas a los artículos, tales como podrían ser color, tipo de cuello, longitud de la manga, etc.

4.2.1. Tipos de conjuntos de datos

El *dataset* cuenta con tres tipos de datos que se dividen en:

1. Sesiones
2. Compras
3. Características de los Artículos

A continuación, se describen cada uno de estos tipos de datos:

- Sesiones: Cada sesión representa la actividad de un usuario en un día particular, incluyendo todos los artículos vistos ese día.

Figura 5. Captura datos archivo *train_sessions.csv*

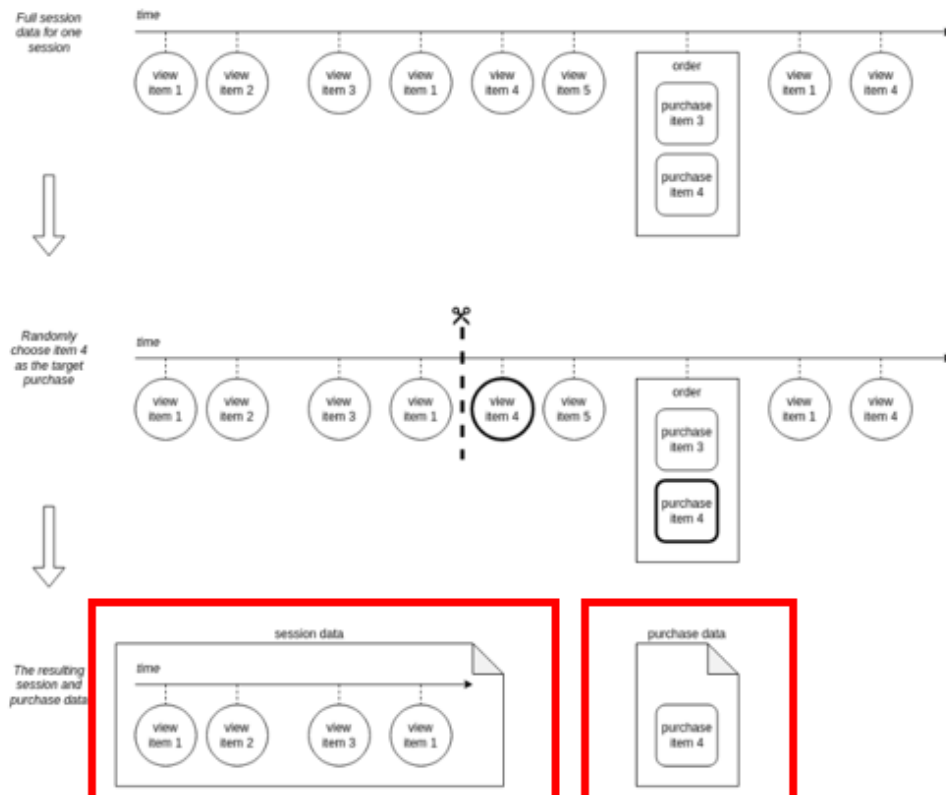
session_id	item_id	date
3	9655	2020-12-18 21:25:00.373
3	9655	2020-12-18 21:19:48.093
13	15654	2020-03-13 19:35:27.136
18	18316	2020-08-26 19:18:30.833
18	2507	2020-08-26 19:16:31.211
18	4026	2020-08-26 19:15:47.232
19	25772	2020-11-02 16:31:18.543
...

Fuente: Elaboración propia

Los campos del conjunto de datos, en el archivo de sesiones, son los siguientes:

- *item_id*: Contiene los ítems que fueron vistos en cada sesión.
- *date*: La columna *date* indica el momento exacto (hasta milisegundos) cuando el *item* fue visto.
- *session_id*: Una sesión corresponde a la actividad de un usuario durante un día. Las sesiones de entrenamiento incluyen todas las vistas de artículos hasta (pero no incluyendo) la primera vista del artículo que fue comprado ese día como se puede observar en la figura anterior.

Figura 6. Construcción de los datos de sesión y compras



Fuente: RecSys Challenge 2022

- **Compras:** Registra el artículo que fue comprado al final de cada sesión. Cada sesión tiene asociada una única compra detallada por *session_id* e *item_id*. “*Purchases.csv*”: contiene las compras realizadas en cada sesión con las columnas: *sesión_id*, *item_id* y *date*.

Figura 7. Captura datos archivo “purchases.csv”

session_id	item_id	date
3	15085	2020-12-18 21:26:47.986
13	18626	2020-03-13 19:36:15.507
18	24911	2020-08-26 19:20:32.049
19	12534	2020-11-02 17:16:45.92
24	13226	2020-02-26 18:27:44.114
28	26394	2020-05-18 12:52:09.764
...

Fuente: Elaboración propia

- **Características de los Artículos:** Detalles como el color, el tipo de cuello o la longitud de la manga están catalogados por números, proporcionando un contexto para cada artículo visto y comprado (*item_id*, *feature_category_id*, *feature_value_id*).

Figura 8. Ejemplificación de las características particulares de un artículo concreto



Fuente: RecSys Challenge 2022

- *item_id*: Un identificador único para cada artículo.
- *feature_category_id*: Un identificador para una categoría de característica específica, como color, material, tipo de prenda, etc.
- *feature_value_id*: Un identificador que representa un valor específico dentro de la categoría mencionada, como rojo cuero, botas, etc...

Figura 9. Captura datos archivo *item_features.csv*

<i>item_id</i>	<i>feature_category_id</i>	<i>feature_value_id</i>
2	56	365
2	62	801
2	68	351
2	33	802
2	72	75
2	29	123
2	16	38
...

Fuente: Elaboración propia

Sin embargo, no existe una relación explícita decodificada entre el *feature_category_id* de cada artículo y un concepto específico, lo que significa que no es conocido si, por ejemplo, el valor *feature_category_id*=7 corresponde a color, estilo o cualquier otra característica. En

otras palabras, el *dataset* no revela qué característica representa cada identificador de categoría y característica.

Aunque esta situación no tiene problemas para modelar un sistema de recomendación basado en sesiones, sí que es un obstáculo para el objetivo del presente TFE el ofrecer una experiencia de compra personalizada, mediante un sistema basado en lenguaje natural. Es esencial conocer estas relaciones para que el usuario pueda interactuar con el sistema de manera natural.

Por ello, tras el análisis exploratorio de datos (EDA) y con el objetivo de generar estas relaciones entre los datos y el lenguaje natural, se llevó a cabo la decodificación de los datos de las columnas *feature_category_id* y *feature_value_id*. Los métodos específicos utilizados para la decodificación se detallan en profundidad en el apartado 4.3.3.

4.2.2. División del Conjunto de Datos

El conjunto original abarca datos de 18 meses completos. De estos, los primeros 17 meses se utilizan para entrenar el modelo y el último mes se reserva para la validación. Para cada sesión de prueba, la entrada para la predicción incluye un porcentaje de las vistas iniciales de la sesión, donde este porcentaje es un valor seleccionado aleatoriamente entre el 50% y el 100%. La longitud máxima de los datos de entrada para una sesión sería hasta, pero sin incluir, la primera vista del artículo que fue comprado. De esta manera, no sabemos en qué punto se cortó cada sesión de prueba.

Para este trabajo, se han utilizado los primeros 16 meses para entrenar el modelo y el mes 17 para la validación, ya que el mes 18 contenía sesiones incompletas diseñadas para aumentar la dificultad de la competición, lo que impedía utilizar algunas de las estrategias propuestas por nuestro equipo, las cuales se detallan en profundidad en el apartado 4.3.2. De este modo, se evita utilizar sesiones incompletas, garantizando que tanto el entrenamiento como la validación de nuestro modelo de recomendación se realicen con sesiones completas.

Figura 10. División realizada sobre el conjunto de datos



Fuente: Recys Challenge 2022

4.3. Descripción de la herramienta software desarrollada

Para la construcción del módulo de recomendación, se ha partido de tres de los modelos de referencia del trabajo (Della Volpe et al., 2022) de un grupo participante de la competición RecSys 2022³ incorporando nuevas adaptaciones y trabajos específicos para los objetivos de este trabajo, entre los que se destaca:

- Método de recomendación *recommend2me* que es la adaptación del modelo de grafos seleccionado para el presente trabajo. En el mismo, además, se ha implementado la conversión del resultado del entrenamiento, a un formato pkl, de la librería pickle⁴ de Python para la serialización de estructuras de datos, convirtiéndolas en un formato de bytes para su almacenamiento o transmisión, y restaurando los objetos a partir de esos bytes. Este formato es el solicitado para la integración con los servicios de LLM en la nube, que serán los usados para el módulo de usuario.
- Orquestador *get_fashion_recomendation* que gestiona todo el flujo de datos entre los tres subsistemas de este trabajo: el *chatbot*, el sistema de encoder-decoder y el sistema de recomendación.
- La desanonimización del dataset a literales específicos del mundo de la moda y la creación del sistema de encoder-decoder, que juntos han hecho posible la integración

³ https://github.com/MaurizioFD/RecSys_Course_AT_PoliMi

⁴ [pickle — Python object serialization — Python 3.12.5 documentation](#)

con el LLM del módulo de usuario. En esta parte ha sido fundamental el uso de matrices dispersas que ha permitido una gestión óptima de la memoria y espacio, ya que en el formato inicial con las matrices densas su tamaño las hace inmanejables y no soportables por la arquitectura subyacente.

- Creación del *Chatbot* que regula la comunicación en lenguaje natural con el usuario.

A continuación, se presenta en detalle el proceso de desarrollo y las características de la herramienta software implementada.

4.3.1. Análisis exploratorio de los datos

Como paso previo esencial para la implementación del modelo, se llevó a cabo un análisis exploratorio de los conjuntos de datos de entrada, incluyendo datos de entrenamiento (consultas y compras), pruebas y artículos. Este análisis se realizó utilizando *Python*, con el entorno de *Jupyter* de *Anaconda*, y *Visual Studio Code* como plataformas de desarrollo y apoyo.

Durante el análisis, se llevaron a cabo diversas verificaciones y estudios que permitieron obtener una visión integral de la naturaleza de los datos:

- Se comprueba la existencia de valores nulos o anómalos en el conjunto de datos, concluyendo que no se encontraron valores de este tipo.
- Se realiza un análisis de los elementos, el número de sesiones y artículos, así como de los valores mínimos y máximos asociados.
- Se comprueba la correspondencia entre los distintos conjuntos de datos proporcionados, verificando si cada sesión de consulta tiene su compra asociada. Esta verificación fue crucial, dado que los datos se proporcionaron en archivos separados para los artículos consultados y comprados.
- Las fechas asociadas al conjunto de datos cubren un periodo de un año y medio, desde enero de 2020 hasta mayo de 2021. Los datos de validación se seleccionaron para un mes dentro de este periodo.
- Se analiza la categorización de las características de los artículos, disponiéndose de una descripción numérica categorizada (*feature_category_id*) y variable para estos

(*feature_value_id*). Se identificaron 73 categorías distintas y 890 valores diferentes, con un conjunto variable de valores numéricos no consecutivos. Por ejemplo, el artículo 1 puede estar descrito por las categorías 3, 12, 24 y 56, mientras que el artículo 2 puede estar descrito por las categorías 3, 17 y 66, con valores distintos dentro de la misma categoría.

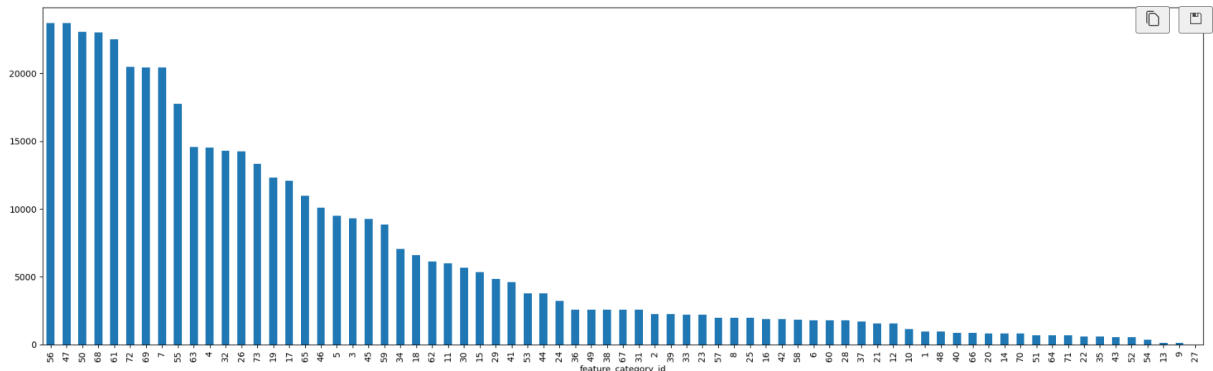
- En el contexto de la representación de atributos de ítems en el conjunto de datos, nos encontramos con una situación donde un artículo i_x puede estar asociado con múltiples pares de categoría-valor, denotados formalmente como $(c_{\{i_j\}}, v_{\{i_j\}})$ y $(c_{\{i_k\}}, v_{\{i_k\}})$. Estos pares forman parte del conjunto de atributos $A(i)$ que describe completamente el artículo i_x . Sin embargo, en ciertos casos, dos o más de estos pares pueden compartir el mismo valor v aunque estén asociados a categorías c diferentes. Esta situación se denomina valores multi-categoría y se expresa matemáticamente como:

$$\exists i_j, i_k: \{ (c_{\{i_j\}}, v_{\{i_j\}}), (c_{\{i_k\}}, v_{\{i_k\}}) \} \in A(i) \wedge c_{\{i_j\}} \neq c_{\{i_k\}} \wedge v_{\{i_j\}} = v_{\{i_k\}}$$

- Se identificaron un total de 14 casos de valores multi-categoría, en los cuales un mismo valor aparece en diferentes categorías. Dado que esta situación introduce un nivel adicional de complejidad en la representación y el procesamiento de los atributos de los artículos, y considerando que estos 14 valores representan una fracción insignificante en comparación con el tamaño total del conjunto de datos, se decidió omitir estos valores multi-categoría. Finalmente, se identificaron un total de 904 parejas únicas $(c_{\{i_j\}}, v_{\{i_j\}})$ de categoría-valor.

Se estudian las distribuciones de los artículos entre sí, y su contextualización en relación con la característica de tiempo de cada sesión. Las siguientes gráficas ilustran la distribución por categoría y varios ámbitos:

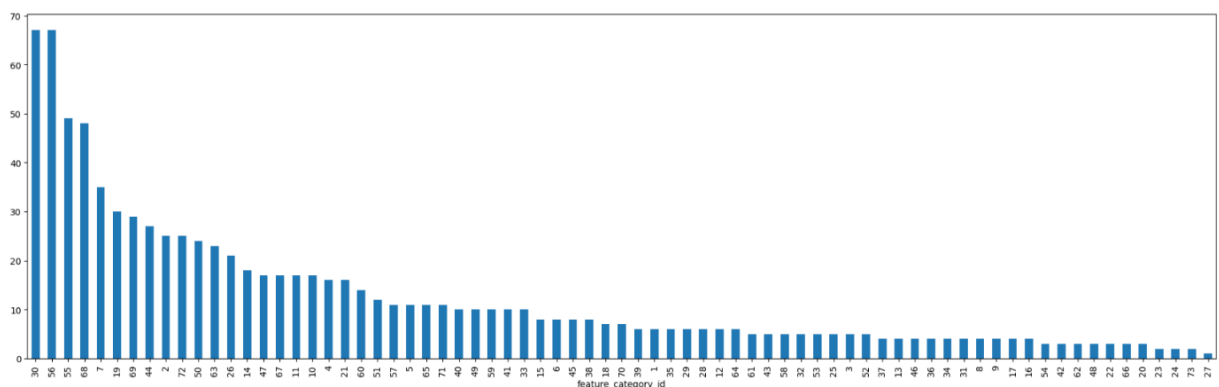
Figura 11. *Distribución del número de artículos por categoría*



Fuente: Elaboración propia.

Se observa que existen algunas categorías que son muy comunes entre la mayoría de los artículos, mientras que otras, en los últimos puestos, son más específicas y están asociadas a productos muy particulares.

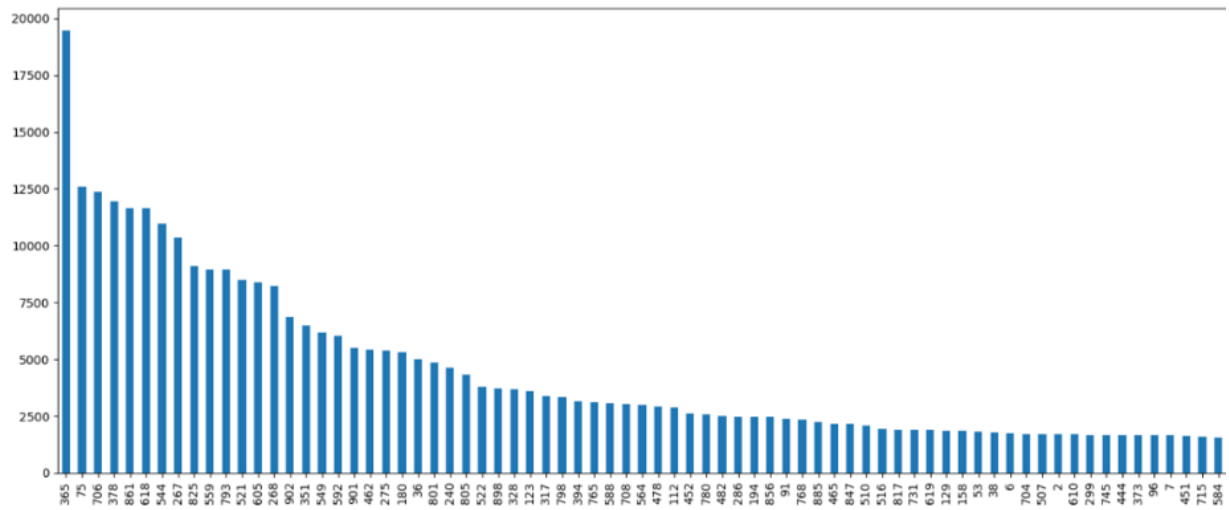
Figura 12. *Distribución de la cantidad de valores distintos por categorías*



Fuente: Elaboración Propia

Como se puede observar en la gráfica, hay un grupo de 12 categorías que tienen más de 20 valores posibles, mientras que las 61 categorías restantes no alcanzan esa cantidad. El número de valores por categoría varía, con la categoría más diversa alcanzando 68 valores, y la menos diversa, solo 1 valor.

Figura 13. *Número de apariciones de cada valor*

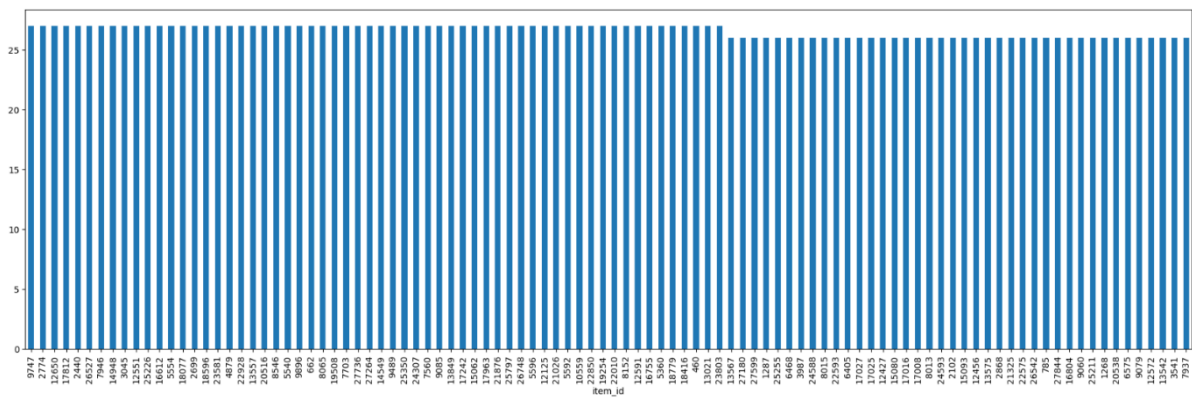


Fuente: Elaboración Propia.

Como se observa en la Figura 14, existe un conjunto de 8 valores que superan las 10.000 apariciones, lo que indica que estos valores específicos son comunes a numerosos artículos. Estos valores podrían representar atributos generales y recurrentes, como por ejemplo el color "blanco," que se encuentra en una gran variedad de productos. En contraste, los valores con menos apariciones parecen corresponder a características más específicas de categorías particulares, como por ejemplo el tipo de tacón *stiletto* que aparecería en un número mucho menor de ocasiones.

Este análisis es fundamental para el desarrollo del sistema de decodificación, ya que permite asociar nombres concretos con los identificadores numéricos de los productos, sus características y valores correspondientes. Esta relación es detallada en la sección 4.3.3, donde se aborda la creación de un sistema que traduce de manera coherente los identificadores numéricos a descripciones textuales, facilitando así la interacción natural con el usuario y mejorando la calidad de las recomendaciones.

Figura 14. Distribución del número de categorías por artículo



Fuente: Elaboración Propia.

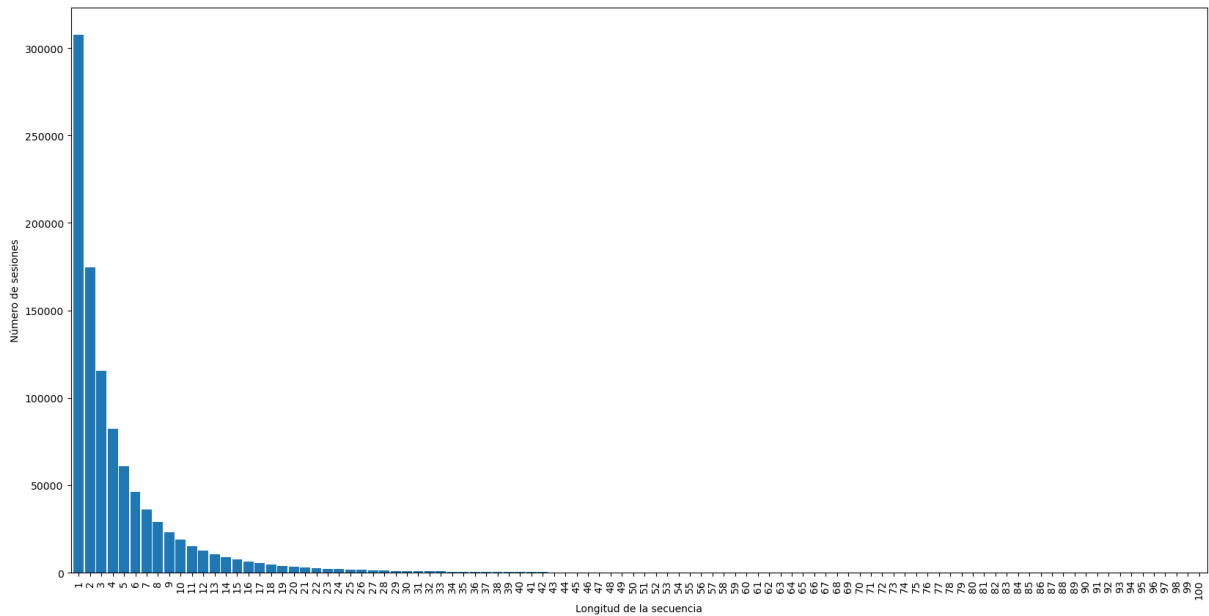
En promedio, el número más frecuente de categorías que describen un artículo son 20. No obstante existe un pequeño conjunto de artículos que cuentan con solo dos categorías.

De los 24.000 artículos disponibles existen un total de 73 artículos que no tienen interacción alguna en ninguna sesión, estos artículos serán filtrados por los algoritmos de recomendación detallados en la sección 4.3.2.

Además, hay una proporción considerable, en la que se vuelve a consultar el mismo artículo de nuevo, alrededor de un 28%, dentro de una misma sesión. Este porcentaje será lo que se denominará “duplicados”.

Es fundamental analizar cuántos artículos son consultados antes de realizar una compra y en qué proporción, ya que este factor es crucial para entender la secuencia temporal de las sesiones de usuario. Los resultados del análisis indican que, en la mayoría de las sesiones del conjunto de datos, aproximadamente 1.000.000, los usuarios toman decisiones rápidamente, siendo más comunes aquellas sesiones en las que se consulta solo un artículo, representando así la moda de la distribución. Además, se observa que la mayoría de las sesiones, más del 90%, se concentran en un rango de consultas que va de 1 a 5 artículos, reflejando así un comportamiento de decisión ágil. Por otro lado, las sesiones que implican un proceso de decisión más lento, en las que se consultan más de 5 artículos y que pueden llegar hasta 100 consultas, representan una mínima concentración dentro del conjunto de datos.

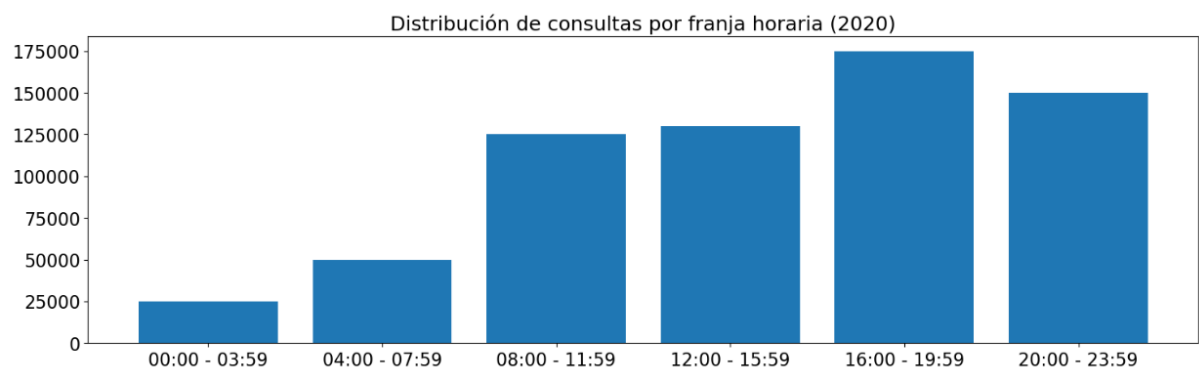
Figura 15. Distribución de la longitud de las sesiones de usuario en el conjunto de datos



Fuente: Elaboración Propia.

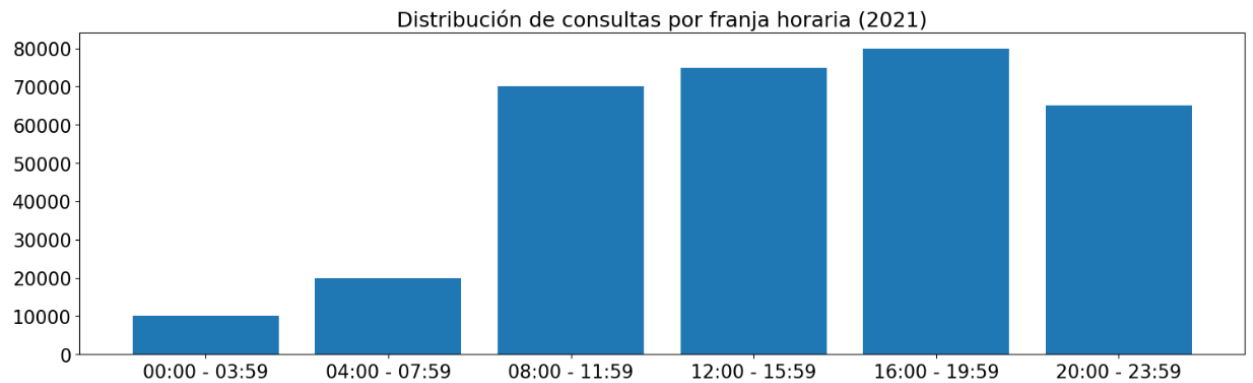
Con intención de encontrar patrones de conducta en función de el horario de consulta, se dividen los días en 6 franjas horarias de 4 horas cada una y se analiza la cantidad de sesiones que ocurren en cada una de ellas por año.

Figura 16. *Número de consultas por franja horaria 2020*



Fuente: Elaboración Propia.

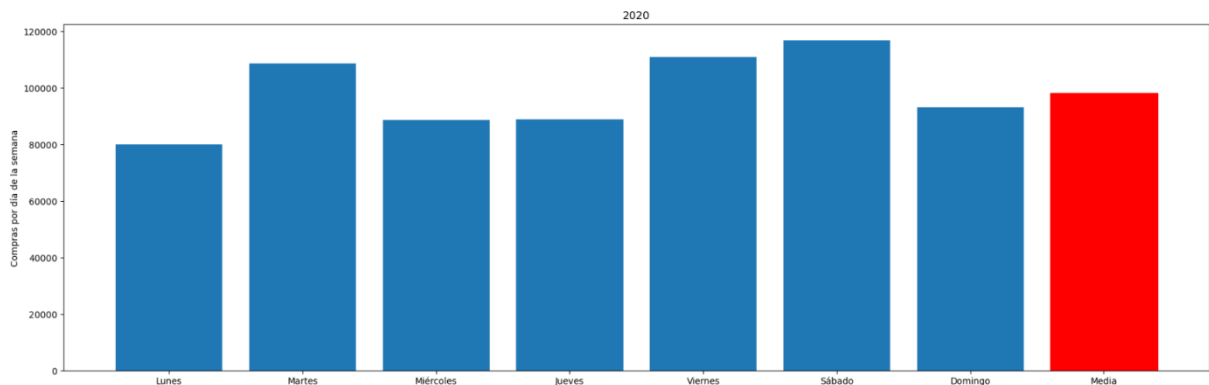
Figura 18. *Número de consultas por franja horaria 2021*

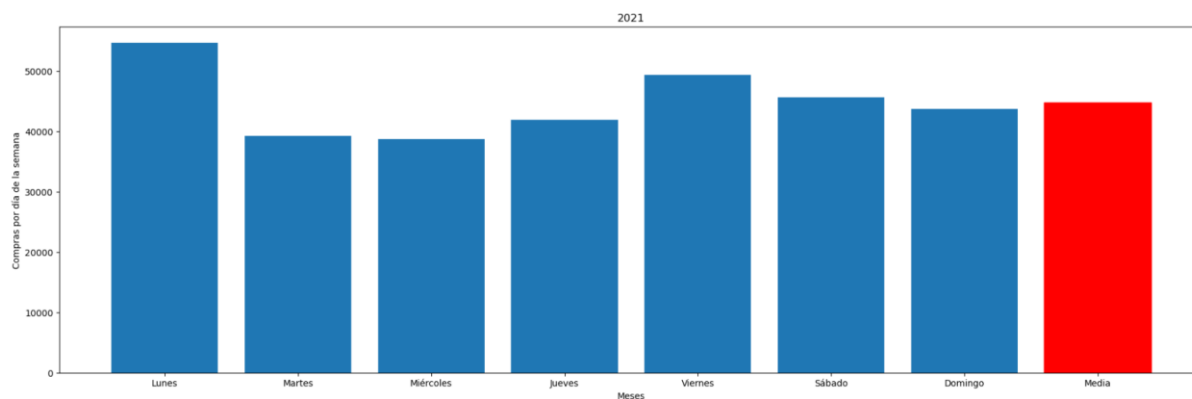


Fuente: Elaboración Propia.

Se observa una notable variación en el número de consultas a lo largo del día, destacándose un incremento significativo a partir de las 8 de la mañana hasta las 12 de la noche. En particular, la franja horaria comprendida entre las 16:00 y las 20:00 horas muestra el mayor nivel de actividad. No se aprecian diferencias significativas en la distribución de consultas al comparar distintos años.

Figura 17. *Análisis de las compras por día de la semana (2020/2021).*

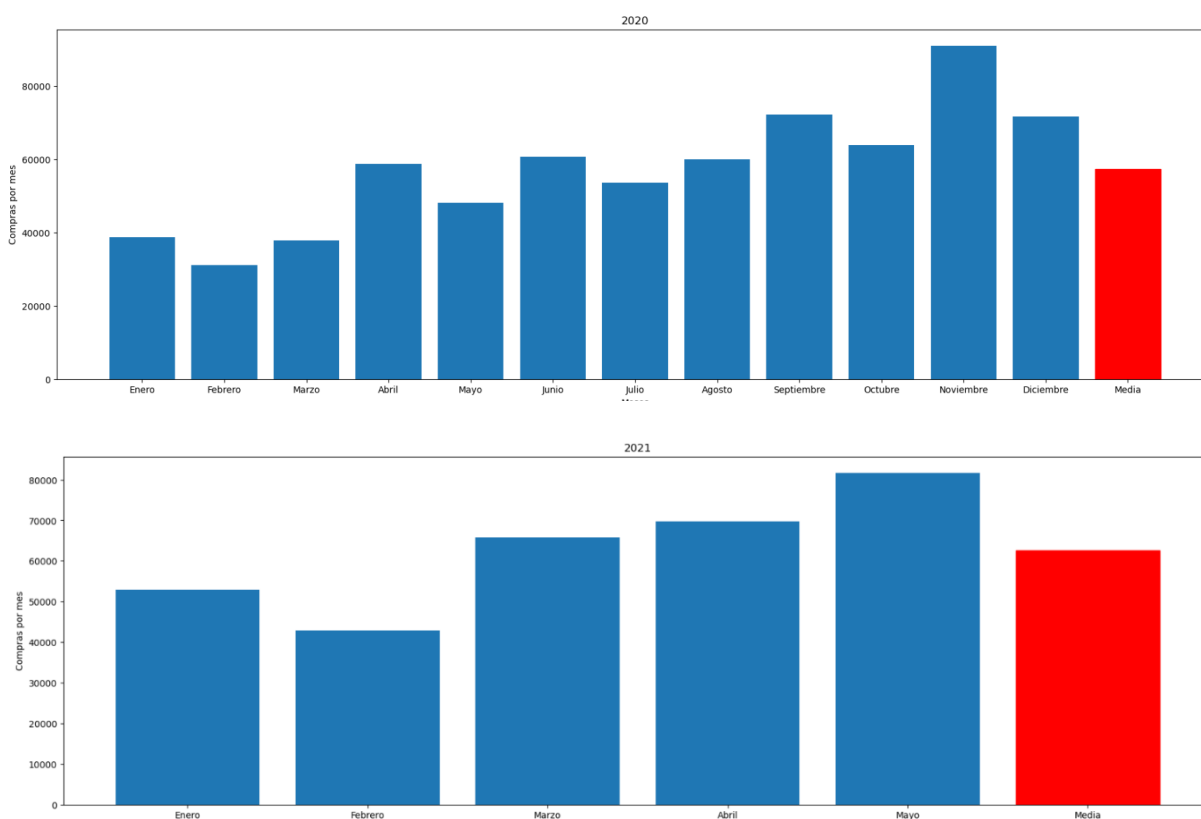




Fuente: Elaboración Propia.

Observando las gráficas por día de la semana, no se aprecian diferencias relevantes entre días de la semana.

Figura 18. Análisis de las compras al mes (2020/2021)



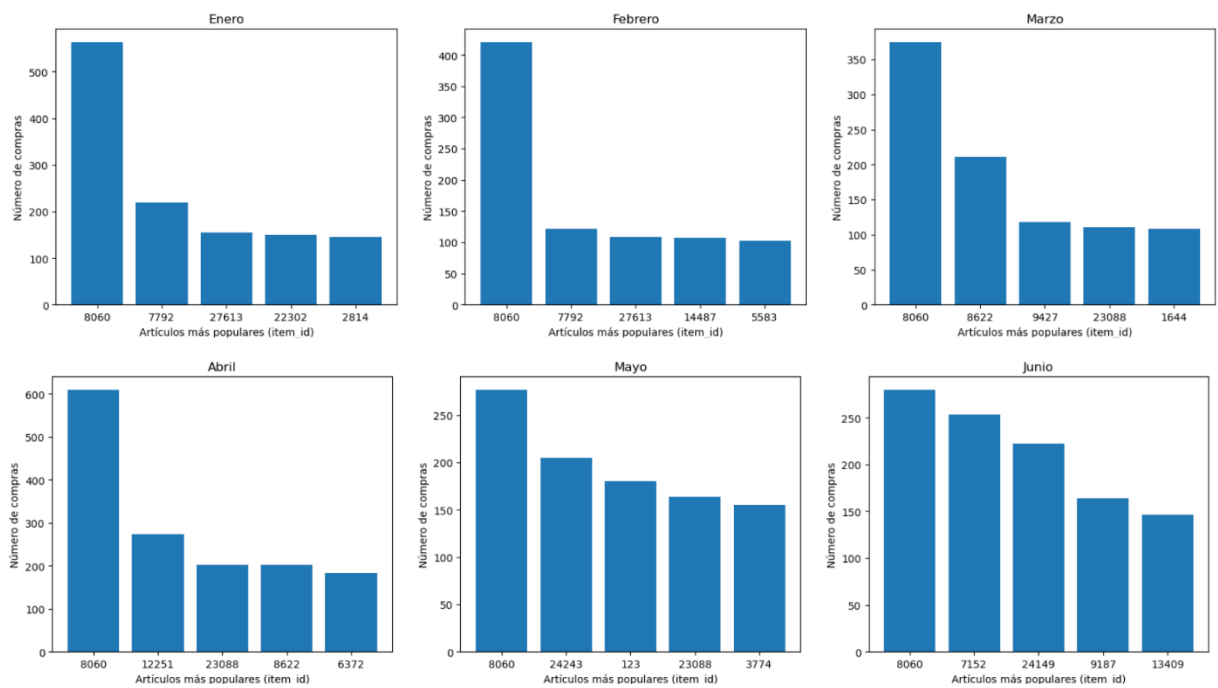
Fuente: Elaboración Propia.

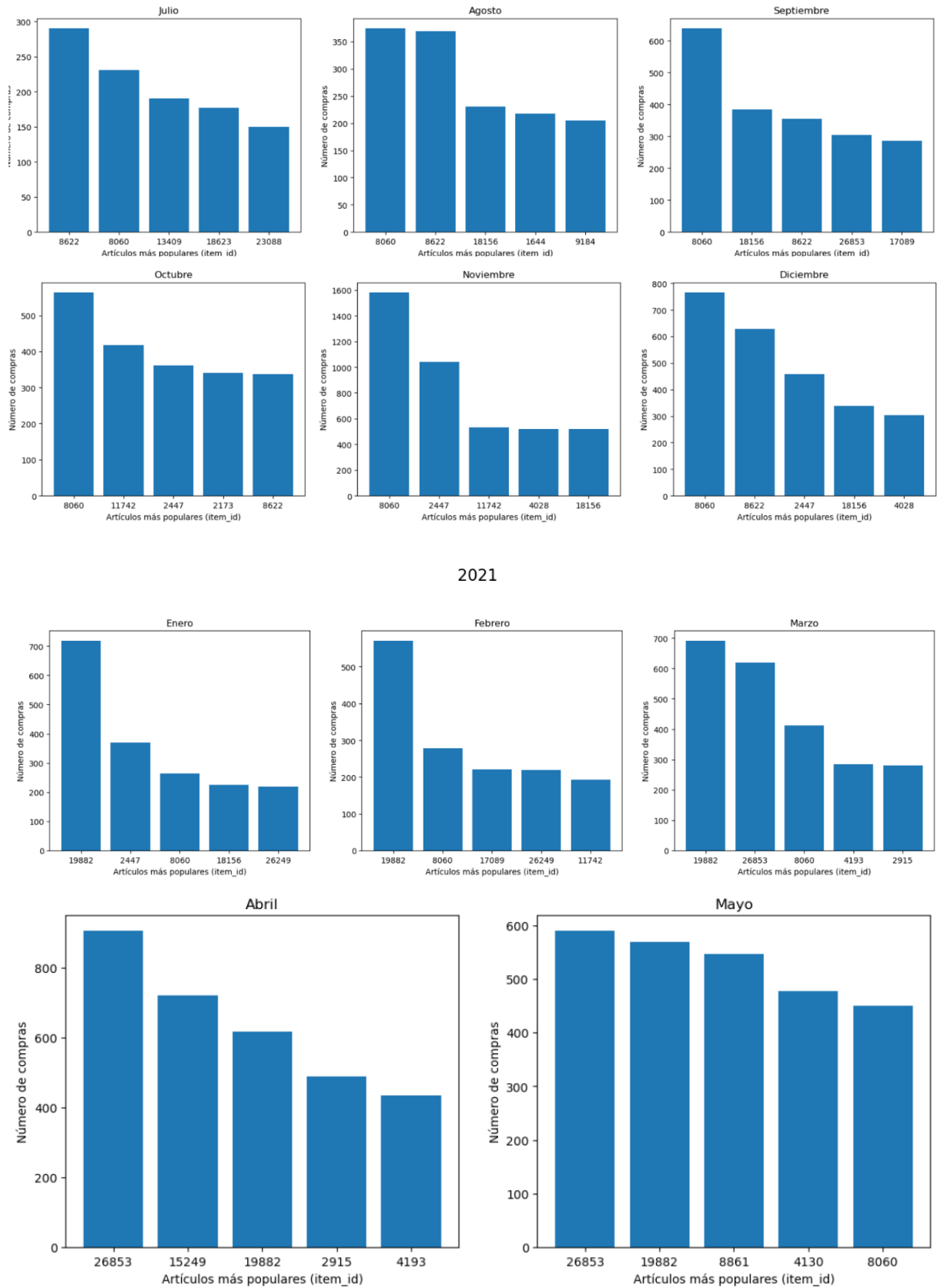
En lo que respecta a las compras mensuales, las gráficas muestran una tendencia creciente a lo largo del año 2020 y la primera mitad del año 2021, con un notable incremento en los últimos meses. Este patrón podría explicarse en parte por eventos clave del comercio estacional como el *Black Friday* y la temporada de compras navideñas, que tienden a concentrar un gran volumen de transacciones en noviembre y diciembre. Asimismo, el

repunte observado en los meses de abril y mayo podría estar relacionado con compras estacionales típicas de la primavera, como la renovación de vestuario debido al cambio de estación, fenómeno común en el sector de la moda. Estos picos sugieren la importancia de los eventos comerciales y las estaciones del año en la planificación y ejecución de estrategias de recomendación para maximizar la interacción y conversión de los usuarios.

Figura 19. *Top 5 de artículos más comprados por mes*

2020





Fuente: Elaboración Propia.

Figura 20. *Top 5 de artículos más comprados al mes 2020*

	Artículo 1	Artículo 2	Artículo 3	Artículo 4	Artículo 5
Enero	8060	7792	27613	22302	2814
Febrero	8060	7792	27613	14487	5583
Marzo	8060	8622	9427	23088	1644
Abril	8060	12251	23088	8622	6372
Mayo	8060	24243	123	23088	3774
Junio	7152	24149	9187	13489	4309
Julio	8622	8060	13409	18623	23088
Agosto	8060	18156	1644	8062	9184
Septiembre	8060	18156	8622	26853	17089
Octubre	11742	2447	2173	8062	8622
Noviembre	8060	2447	11742	4028	18156
Diciembre	8060	8622	2447	18156	4028

Fuente: Elaboración Propia.

Figura 21. *Top 5 de artículos más comprados al mes 2021*

	Artículo 1	Artículo 2	Artículo 3	Artículo 4	Artículo 5
Enero	19882	2447	8060	18156	26249
Febrero	19882	8060	17089	26249	11742
Marzo	19882	26853	8060	4193	2915
Abril	26853	15249	19882	2915	4193
Mayo	26853	19882	8861	4130	8060

Fuente: Elaboración Propia.

Se destaca que el artículo 8060 mantiene una alta popularidad durante el año 2020, situándose entre los más demandados a lo largo de todo el año. Sin embargo, en 2021, se observa una disminución en su demanda, dando paso a la aparición de nuevas tendencias de consumo. Este fenómeno podría explicarse por un factor inherente a la moda, en la que los artículos concretos pierden relevancia a medida que avanza el tiempo, ya que las tendencias cambian y los consumidores buscan novedades. Esto nos sugiere la necesidad de implementar estrategias que permitan anticipar y tener en cuenta la posible pérdida de interés en ciertos artículos porque estos “pasen de moda”, favoreciendo la incorporación de nuevos productos que se ajusten a las preferencias emergentes del público.

4.3.2. Descripción del módulo de Recomendación

En esta sección, se presenta la solución propuesta para realizar recomendaciones de artículos basadas en las sesiones de compra de los usuarios. A continuación, se describen los distintos

métodos empleados previamente para enriquecer los datos y los algoritmos utilizados para completar el módulo de recomendación.

4.3.2.1. Métodos estratégicos de variación de pesos

Con el objetivo de enriquecer y dotar de mayor significado a los datos, se han utilizado técnicas de variación de pesos para las consultas y las compras. Estas técnicas ajustan los pesos de las interacciones para reflejar mejor su importancia relativa en el contexto de la moda, donde las preferencias de los usuarios pueden variar considerablemente con el tiempo y entre diferentes tipos de interacciones.

- Distinción entre tipos de interacciones: Dado que la mayoría de los modelos utilizados en el conjunto de soluciones no estaban diseñados para distinguir entre vistas y compras, se asignó un peso diferencial a las interacciones basadas en la naturaleza de la interacción. Específicamente, se definió un peso $\alpha \in (0,1)$ para las vistas, mientras que las compras se mantuvieron con un peso completo (es decir, $\alpha = 1$ para compras). Este ajuste permite que el modelo capture la diferencia en la importancia entre simplemente ver un artículo y realizar una compra.

$$w_{\text{vista}} = \alpha, \quad w_{\text{compra}} = 1$$

- Decaimiento cíclico: La moda es inherentemente estacional, con tendencias que varían de acuerdo con las estaciones del año. Para reflejar esta estacionalidad, se aplicó un peso que varía cíclicamente con el tiempo. La idea es que las interacciones cercanas a la fecha de referencia, por ejemplo, la fecha de la temporada actual o una fecha importante del calendario de la moda, tienen mayor relevancia que las interacciones alejadas de dicha fecha.

El peso cíclico se modeló mediante una función sinusoidal:

$$w_{\text{cíclico}}(t) = \sin\left(2\pi \frac{t - t_{\text{ref}}}{365}\right)$$

Donde t representa el día del año en que ocurrió la interacción y t_{ref} es la fecha de referencia, típicamente, y en nuestro caso, el inicio del conjunto de pruebas. Esta función asigna un peso más alto a las interacciones que ocurren en días del año cercanos a la fecha de referencia, y un peso más bajo a las interacciones que ocurren en días distantes. De este modo, el sistema tendrá información para recomendar, por ejemplo, abrigos durante

el invierno o trajes de baño durante el verano, basándose en los patrones de comportamiento observados en años anteriores. (Koren, 2009)

- Decaimiento exponencial: Además de la estacionalidad, las preferencias de los usuarios tienden a cambiar con el tiempo, y las interacciones más recientes son generalmente más indicativas de las preferencias actuales. Para capturar este efecto, se aplicó un decaimiento exponencial a las interacciones en función de su antigüedad con respecto a la fecha de referencia. El decaimiento exponencial se modela con la siguiente función:

$$w_{\text{exponencial}}(t) = e^{-\lambda(t-t_{\text{ref}})}$$

Aquí, λ es un parámetro de decaimiento que determina la rapidez con la que disminuye la importancia de una interacción a medida que se aleja temporalmente de t_{ref} .

$$\lambda = \frac{\ln(2)}{\text{abs_half_life}}$$

A su vez, λ tiene una relación inversa con *abs_half_life*, el cual se refiere al tiempo que tarda una interacción en perder la mitad de su relevancia. Es decir, es el número de días (u otra unidad de tiempo) después del cual el valor de la interacción se reduce al 50%. (Koren, 2009)

El ajuste de parámetros de la relación de pesos entre compra y visualización, y el parámetro *abs_half_life* del decaimiento exponencial, se llevó a cabo mediante un proceso de optimización bayesiana. En este proceso, se definió un espacio de búsqueda para estos parámetros con un rango para *abs_half_life* entre 100 y 600 y entre 0 y 1 para la relación de pesos entre vista y compra. Una vez definido el espacio de búsqueda se estableció la función MRR para cuantificar el rendimiento del modelo con el objetivo de maximizarla para encontrar los valores óptimos de w_{vista} y λ antes de proceder a la optimización de los hiperparámetros propios de cada uno de los modelos candidatos.

4.3.2.2. Selección de modelos candidatos

Para la selección del algoritmo óptimo a implementar en nuestro módulo de recomendación, se evaluaron tres candidatos potenciales, *Embarrassingly Shallow Autoencoder* (EaseR),

Reversed Personalized PageRank with alpha parameter (RP3Alpha) y Reversed Personalized PageRank with beta parameter (RP3Beta).

- EaseR: es un algoritmo de recomendación lineal, diseñado para ser altamente eficiente y escalable. A diferencia de los *autoencoders* tradicionales, EaseR simplifica el proceso al utilizar una única capa lineal, eliminando la necesidad de estructuras complejas y no lineales. Esto permite una rápida convergencia y bajo coste computacional. (Steck, 2019)

Los hiperparámetros de EaseR son:

- *L2_norm*: controla la regularización L2 aplicada al modelo para prevenir sobreajuste.
 - *topK*: controla el número máximo de ítems similares a considerar en las recomendaciones.
 - *normalize_matrix*: indica si la matriz de similitud debe ser normalizada o no.
- RP3Alpha es un algoritmo de recomendación basado en grafos que utiliza caminatas aleatorias con reinicio. Su característica distintiva es el parámetro *alpha*, que controla el grado de dispersión en la red de ítems durante la caminata. Un valor de *alpha* bajo permite al algoritmo explorar más ampliamente, promoviendo la diversidad en las recomendaciones, mientras que un valor alto se enfoca más en los ítems directamente conectados, favoreciendo la precisión. Los hiperparámetros de RP3Alpha incluyen *topK* y *normalize_matrix* (de manera similar a EaseR) y el mencionado *alpha*. (Bernardis, Ferrari Dacrema, & Cremonesi, 2018)
 - RP3Beta es una variante de RP3Alpha que introduce el parámetro beta para ajustar la influencia de la popularidad de los ítems en las recomendaciones. Mientras RP3Alpha se enfoca en la estructura de la red, RP3Beta permite controlar el sesgo hacia ítems populares, promoviendo así la recomendación de ítems menos conocidos en busca de mayor personalización en las recomendaciones. Los hiperparámetros de RP3Beta incluyen los mismos que RP3Alpha más este factor adicional *beta* para controlar la influencia de la popularidad. (Bernardis, Ferrari Dacrema, & Cremonesi, 2018)

El MRR es una métrica de rendimiento que evalúa cuan alto en los resultados aparece la primera respuesta relevante para un conjunto de consultas. Se calcula tomando el recíproco de la posición de la primera respuesta relevante, es decir, el valor inverso de la posición en la que se encuentra. Por ejemplo, si la respuesta relevante aparece en la primera posición, el

recíproco es $\frac{1}{1} = 1$, si está en la segunda posición es $\frac{1}{2} = 0,5$. Luego, el MRR se obtiene promediando estos recíprocos para todas las consultas evaluadas, proporcionando una medida general de rendimiento. (Ricci et al., 2022)

Dado un conjunto de consultas Q donde cada consulta q_i tiene una lista ordenada de resultados, el MRR se define como:

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i}$$

Donde $|Q|$ es el número total de consultas evaluadas y $\{rank_i\}$ es la posición del primer ítem relevante en la lista de resultados para la consulta q_i .

En este estudio, para cada sesión de compra se generó una lista de las 10 mejores recomendaciones (top-10) utilizando cada uno de los algoritmos mencionados. El valor de $rank_i$ corresponde a la posición en la que el primer ítem verdaderamente relevante aparece dentro de estas diez recomendaciones. Si el primer resultado relevante está en la primera posición, su recíproco sería 1, si está en la segunda posición, sería 0.5, y así sucesivamente. Si el ítem relevante no se encuentra dentro del top-10, se considera que $rank_i = \infty$, contribuyendo con un valor de 0 al promedio del MRR.

La optimización de hiperparámetros propios de cada modelo, se llevó a cabo realizando 100 pruebas en cada modelo mediante optimización bayesiana. En cada iteración se prueban diferentes combinaciones de estos hiperparámetros evaluando el rendimiento del modelo mediante la métrica MRR. El algoritmo optimiza este proceso frente a búsquedas aleatorias, ajustando las configuraciones en función de los resultados obtenidos en iteraciones anteriores. Tras las cien pruebas, se selecciona la configuración de hiperparámetros que maximiza el MRR para cada modelo. A continuación, se muestra las configuraciones y resultados concretos.

Figura 22. Configuración de hiperparámetros y resultados obtenidos en MRR

	EASE_R	RP3Alpha	RP3Beta
Vista/Compra	0.2	0.5	0.5
Decaimiento Cíclico	Sí	Sí	Sí
λ	0.00173	0.00347	0.00381
topK	No	195	182
Normalize Matrix	No	Sí	Sí
L2 norm	35.93	-	-
Alpha	-	1.3668	1.067
Beta	-	-	0.260
MRR	0.1552439	0.1509921	0.1571059

Fuente: Elaboración Propia.

Estos resultados indican que **RP3Beta** logró el mejor desempeño en este contexto específico, posicionando más consistentemente ítems relevantes en las primeras posiciones de las recomendaciones. Por ello es el algoritmo elegido para nuestro módulo de recomendación.

4.3.3. Sistema Encoder-Decoder

Un desafío fundamental, como se detalla en la Sección 4.2, es que en el *dataset* las sesiones, los productos y sus características están representados exclusivamente por identificadores numéricos, sin una correspondencia explícita con sus descripciones literales. Esta falta de vinculación semántica impide comprender directamente si un identificador numérico corresponde, por ejemplo, a un bolso, una camiseta o un color específico. Para abordar este problema y asegurar que las recomendaciones sean tanto intuitivas como relevantes, se ha desarrollado un sistema de codificación y decodificación que traduce estos identificadores numéricos en términos comprensibles.

4.3.3.1. Asignación manual de palabras a identificadores numéricos

Debido a la dificultad inherente de asignar de manera coherente y significativa una palabra a cada característica y a cada uno de sus valores correspondientes dentro del *dataset*, surgieron limitaciones para representar semánticamente los artículos. Un ejemplo ilustrativo de esta complejidad es la posibilidad de que un artículo dado pueda tener asignada la característica “tipo de producto” con el valor "camiseta" y, simultáneamente, la característica “material” con el valor "acero inoxidable", lo cual

carece de sentido lógico. Para abordar esta problemática, se llevó a cabo un análisis exhaustivo de los datos, descrito en la Sección 4.3.1. que revelaron que, al asignar palabras a solo 13 de las 73 categorías disponibles y sus respectivos valores, se logra describir más del 50% de las instancias presentes en el archivo *items_features* que describen cada uno de los artículos.

Se creó un diccionario en JSON, en el que se asignaron los literales a los campos categóricos. La creación de los citados literales de categorías y valores, se realizaron, en una primera fase, de forma manual, y después con el soporte de IA generativa hasta completar las 73 categorías y las 904 combinaciones distintas de categoría-valor existentes en el *dataset*.

Este diccionario sirvió como base para el código que permitió mapear las categorías de mayor frecuencia. En él, se codificaron las reglas necesarias para preservar la semántica de los distintos valores dentro de una misma categoría. Además, se abordaron casos especiales, como los relacionados con calzado y accesorios, donde se asignaron únicamente valores específicos para este tipo de productos. Como ejemplo, se muestra el tratamiento aplicado al calzado, en el cual se crearon listas de valores que solo se asignaban si la categoría correspondía a este tipo de producto. De esta forma se aseguraban que estos valores no eran asignados, por ejemplo a un vestido, evitando así incongruencias. A continuación, se presentan algunas de estas listas:

- Tipos de tacón: Diferentes tipos de tacones, como *stiletto*, cuña, bloque, etc.
- Tipos de puntera: Diferentes tipos de punteras, como redonda, puntiaguda, abierta, etc.
- Tipos de Materiales: Materiales utilizados en la fabricación del calzado, como piel sintética y cuero.

Una vez completada la conversión a literales, se constató que todos los casi 24.000 artículos diferentes del *dataset* poseían así al menos una característica descrita. Concretamente, las 13 características principales se nombraron como: “rangos de precio”, “productos”, “estampados”, “talla”, “estaciones”, “características”, “detalles”, “materiales”, “géneros”, “país de fabricación”, “marca”, “ocasión” y “colores”. De esta

manera, el hecho de focalizarse en un subconjunto reducido de categorías permitió una asignación semiautomática de palabras que fue guiada por el código anteriormente descrito, asegurando que la coherencia semántica se mantuviera en las características de cada producto.

Figura 23. Descripción de un artículo tras decodificación

item_id	feature_category_id	feature_value_id	feature_category_literal	feature_value_literal
2	56	365	rangos_de_precio	12-14
2	62	801	-	-
2	68	351	productos	Camiseta
2	33	802	-	-
2	72	75	estampados	Liso
2	29	123	-	-
2	16	38	-	-
2	50	76	talla	M
2	61	462	-	-
2	53	6	estaciones	Primavera
2	7	394	características	Manga larga
2	69	885	detalles	Con botones
2	47	123	materiales	Algodón

Fuente: Elaboración Propia.

El código generado estableció las bases para la categorización y descripción de productos de moda, tanto calzado como accesorios. Al definir características específicas de estos productos, se facilitó la organización y procesamiento de datos relacionados. Este enfoque modular ha permitido trabajar con el chat, obteniendo información valiosa sobre los productos y mejorando la capacidad de realizar recomendaciones personalizadas.

4.3.3.2. Generación de sesiones de interacción para la entrada del recomendador

Una vez que se han decodificado los atributos de los ítems, el siguiente objetivo es generar recomendaciones empleando el módulo de recomendación descrito en la Sección 4.3.2. Dado que la finalidad del proyecto es lograr una interacción efectiva con el usuario a través de un *chatbot* que utiliza lenguaje natural, se enfrenta el desafío de adaptar esta interfaz a la estructura requerida por el módulo de recomendación. Este módulo está diseñado para recibir como entrada sesiones de consultas, las cuales incluyen una secuencia de artículos visualizados antes de la compra del artículo a predecir. Por lo tanto, es necesario transformar la conversación entre el cliente y el *chatbot* en un formato que simule una sesión de usuario.

En esta sección, se detalla el mecanismo de codificación de artículos implementado en nuestro sistema de recomendación de moda, haciendo uso de técnicas de procesamiento de lenguaje natural (PLN).

- Preprocesamiento del texto: Para mejorar la calidad de la codificación, se aplica el *stemming*, que es el proceso de reducir las palabras a su forma raíz o base. Utilizamos el algoritmo *PorterStemmer* para este propósito. (Porter et al., 1980) Adicionalmente, se eliminan el *stop-words*, que son palabras comunes que no aportan significado significativo a la semántica del texto, utilizando la lista de *stop words* en español de la biblioteca *Natural Language Toolkit* (NLTK).⁵
- Vectorización de Palabras: El primer paso en la codificación es la transformación de las descripciones de los artículos y las consultas de los usuarios en vectores de características. Se utiliza la herramienta *CountVectorizer* proporcionada por la biblioteca *scikit-learn*⁶, para crear una representación de bolsa de palabras, donde cada palabra en el vocabulario se convierte en una característica. En esta representación, cada consulta es descrita por un vector en el cual cada dimensión representa una palabra del vocabulario del corpus.

$$X = \text{CountVectorizer.fit_transform}(\text{descriptions})$$

Aquí, X representa la matriz de características, donde cada fila corresponde a un artículo y cada columna a una palabra del vocabulario. (Pedregosa et al., 2011)

- Espacio Vectorial y Similitud de Coseno: Al representar texto en un espacio vectorial, cada artículo o consulta se convierte en un punto en ese espacio, donde la proximidad entre puntos (vectores) puede interpretarse como similitud semántica. La medida de similitud de coseno se emplea para calcular la similitud entre dos vectores.

$$\text{sim}(q, X_i) = \frac{q \cdot X_i}{|q||X_i|}$$

Donde q es el vector de características de la consulta y X_i es el vector de características del artículo i . Esta medida devuelve un valor entre -1 y 1, donde 1 indica vectores idénticos, 0 indica que son ortogonales (sin similitud), y -1 indica direcciones opuestas (Géron, 2019). Se utiliza un umbral de similitud para filtrar las coincidencias más

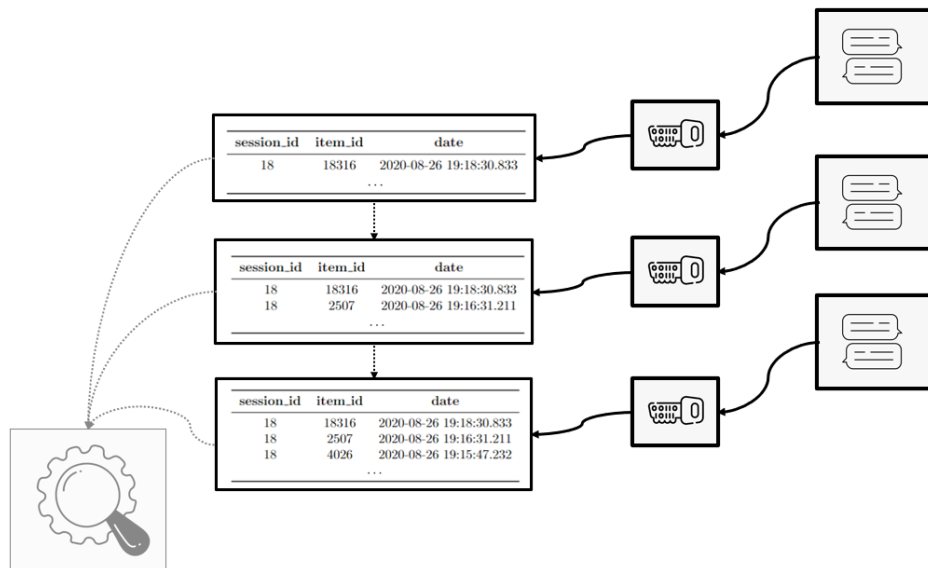
⁵ <https://www.nltk.org/>

⁶ <https://scikit-learn.org/stable/>

relevantes y solo se consideran aquellas similitudes que superan un valor umbral predefinido, en nuestro caso 0.1.

- **Generación sesión ficticia:** Una vez que se identifica el artículo con mayor similitud semántica en relación con la frase en lenguaje natural proporcionada por el usuario, se asocia a este artículo la marca temporal precisa que incluye fecha, horas, minutos, segundos y milisegundos en el momento exacto de la interacción en el chat. Este procedimiento permite crear un registro de consulta que sigue el mismo formato requerido por el módulo de recomendación basado en sesiones. Así, cada interacción del usuario con el *chatbot* añadirá una nueva instancia de artículo con su correspondiente marca temporal a la sesión generada, enriqueciendo progresivamente la sesión ficticia con cada consulta realizada por el cliente.

Figura 24. Esquema generación de sesiones compatibles con el módulo de recomendación (encoder)



Fuente: Elaboración Propia.

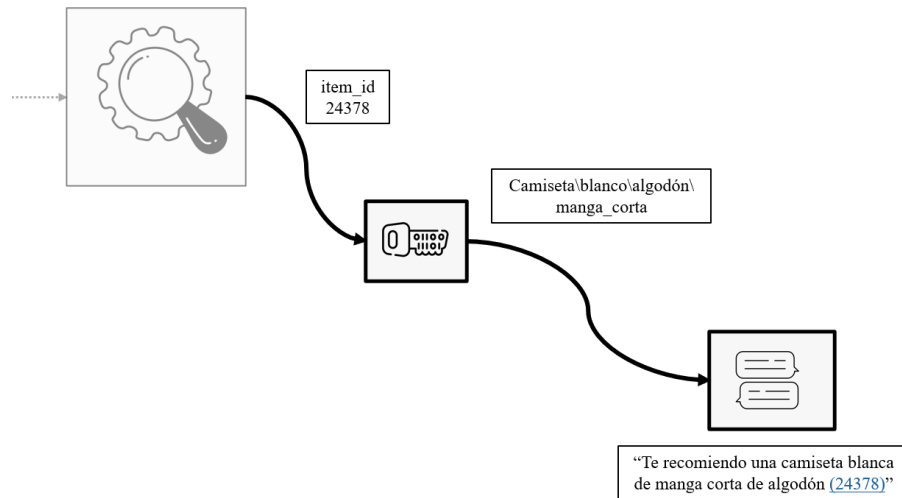
4.3.3.3. Decodificación de resultados

Una vez que el sistema de recomendación genera una lista de identificadores de artículos recomendados, estos se decodifican de vuelta a descripciones de productos utilizando un diccionario de mapeo.

El diccionario de mapeo se construye durante la fase de preprocesamiento nombrada en el apartado anterior y es una tabla de correspondencias que relaciona los identificadores numéricos de artículos (*item_id*) con descripciones textuales (*item_descrip*).

- Se han identificado seis categorías de características que son especialmente relevantes para describir los artículos de manera significativa: "tipo de producto", "material", "características especiales", "color", "ocasión de uso" y "estaciones". Aunque en la recomendación de productos se consideren las 13 características descritas en el apartado anterior, nos centraremos únicamente en estas seis categorías clave para comunicar de manera más natural con el cliente utilizando lenguaje natural.
- Generación de Descripciones: Se genera una descripción textual para cada artículo concatenando los valores de las categorías seleccionadas. Por ejemplo, un artículo con tipo de producto: camiseta, material: algodón, y color: blanco podría tener una descripción como "camiseta de algodón blanco".
- Recuperación de Descripciones: Cuando el sistema de recomendación genera una lista de identificadores de artículos recomendados, cada identificador se busca en el diccionario de mapeo para recuperar su descripción asociada.
- Generación de salida amigable para el usuario: Las descripciones recuperadas se agrupan y presentan al usuario en un formato que es fácil de entender. De esta manera, el *chatbot*, muestra en un mensaje, una descripción en lenguaje natural de los productos recomendados, basándose en las 6 categorías principales que describen cada artículo recomendado. Además, cada recomendación vendrá acompañada del código identificador *item_id* único del artículo en cuestión para poder realizar la compra del artículo concreto.

Figura 25. Esquema decodificación de resultados a lenguaje natural



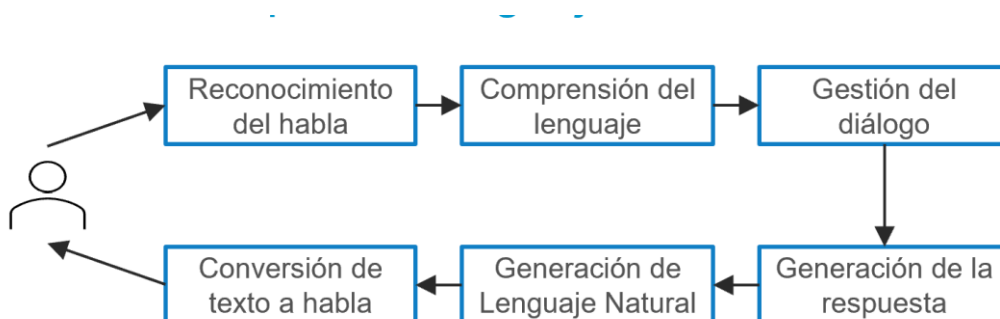
Fuente: Elaboración Propia.

4.3.4. Descripción del módulo de usuario: MarIA

El módulo de usuario, MarIA, es el agente conversacional objeto de este TFE. Actúa como un *personal shopper*, ofreciendo recomendaciones personalizadas de artículos según las preferencias y el perfil del cliente.

El desarrollo de este módulo se ha basado en el siguiente flujo de funcionamiento de un agente conversacional, con especial foco en la Gestión del Dialogo y Gestión de la respuesta.

Figura 26. Flujo de un funcionamiento de un agente conversacional



Fuente: UNIR. Asignatura Procesamiento de Lenguaje natural

Para la construcción del agente conversacional MarIA, se ha seguido un modelo de referencia basado en *Copilot Stack* de *Microsoft Azure* ⁷⁸, que proporciona una base sólida para construir aplicaciones inteligentes mediante el uso de modelos de lenguaje natural pre-entrenados como GPT-4 de *OpenAI* ⁹¹⁰. *Copilot Stack* sirve como un marco conceptual que ha guiado el diseño y desarrollo de este agente conversacional, aunque el despliegue específico se ha realizado utilizando *Chainlit* ¹¹ y otros servicios de *Azure* para la implementación práctica.

Copilot Stack es un marco de referencia que permite enriquecer modelos fundacionales, como GPT-4, con modelos de recomendación personalizados, como es el caso de este TFE. Proporciona una capa de abstracción y una interfaz de programación que facilita la integración de servicios cognitivos de *Azure* con aplicaciones inteligentes. Además, ofrece herramientas para la gestión, monitorización y seguridad durante el ciclo de vida de las aplicaciones.

En este TFE, *Copilot Stack* ha servido como guía conceptual para conectar el modelo de recomendación, desarrollado en capítulos anteriores, con el *chatbot* basado en *OpenAI*, creando una experiencia conversacional fluida y personalizada para los usuarios.

⁷[Understanding The Microsoft Copilot Stack: A Guide For ISVs & Partners Trying To Build Custom Copilots - https://www.linkedin.com/pulse/understanding-microsoft-copilot-stack-guide-isvs-partners-hawkins-lnfzc/](https://www.linkedin.com/pulse/understanding-microsoft-copilot-stack-guide-isvs-partners-hawkins-lnfzc/)

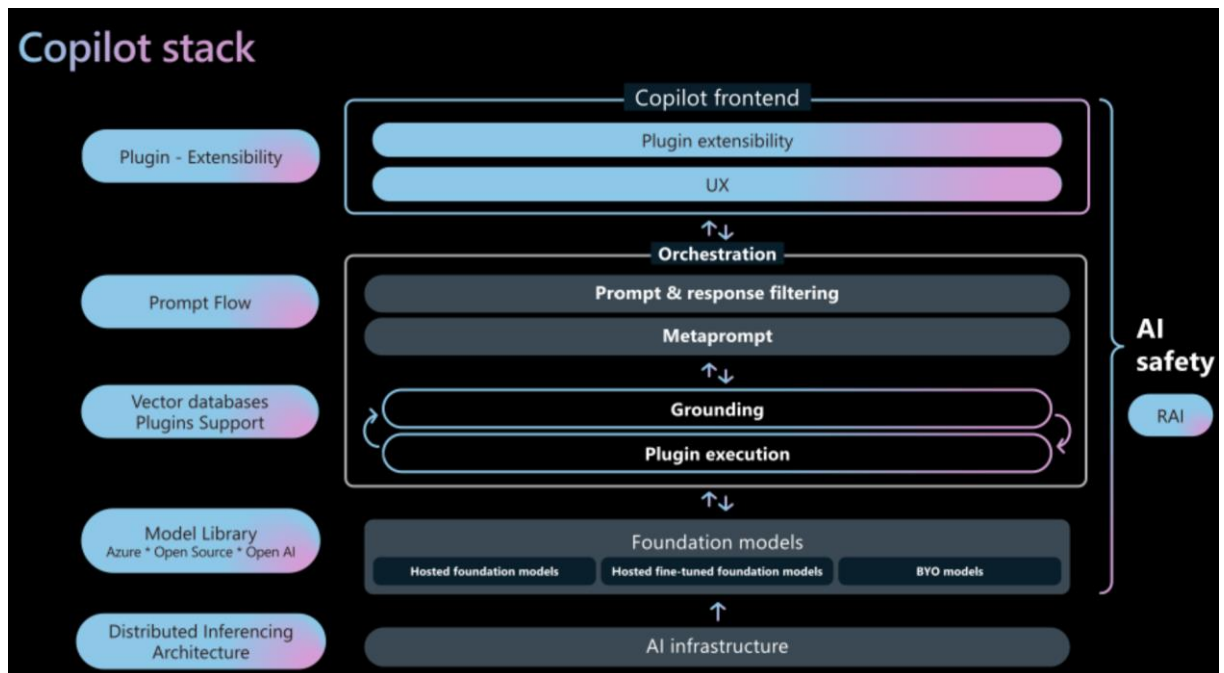
⁸[Copilot | Microsoft Developer - https://developer.microsoft.com/en-us/copilot](https://developer.microsoft.com/en-us/copilot)

⁹[Work with the GPT-35-Turbo and GPT-4 models - Azure OpenAI Service | Microsoft Learn - https://learn.microsoft.com/en-us/azure/ai-services/openai/how-to/chatgpt?tabs=python-new](https://learn.microsoft.com/en-us/azure/ai-services/openai/how-to/chatgpt?tabs=python-new)

¹⁰[GPT-4 | OpenAI - https://openai.com/index/gpt-4/](https://openai.com/index/gpt-4/)

¹¹[Overview - Chainlit - https://docs.chainlit.io/get-started/overview](https://docs.chainlit.io/get-started/overview)

Figura 27. Modelo de Plataforma de Copilot Stack



Fuente: Microsoft

Implementación con *Chainlit* y *Azure*: El despliegue práctico del agente conversacional MarIA se ha realizado utilizando *Chainlit*, integrando los servicios de *Azure Machine Learning* y *Azure OpenAI*. Esta combinación permite una implementación ágil y efectiva, manteniendo los principios del modelo de referencia de *Copilot Stack*.

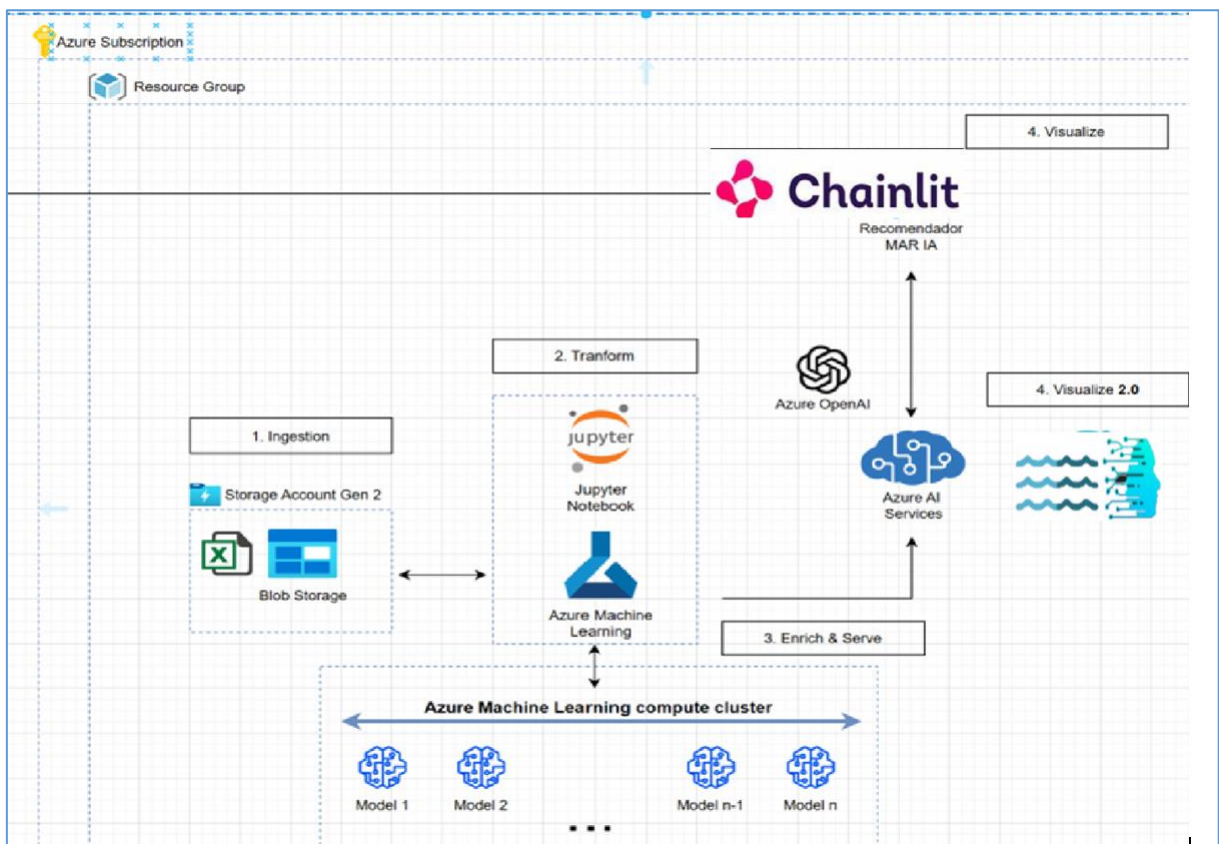
La arquitectura del agente conversacional se basa en una serie de componentes en la nube que permiten la interacción del usuario con el sistema, alineados con el citado modelo de referencia. A continuación, se detallan los servicios empleados:

- *Azure Machine Learning*: Utilizado para crear, entrenar y desplegar el modelo de recomendación personalizado que se adapta a las preferencias y el contexto de cada usuario y en el que se genera el PKL necesario para el módulo de usuario. Para ello se ha serializado el modelo entrenado con los siguientes parámetros:
 - Matriz dispersa del recomendador entrenado.
 - Los valores de las categorías y los ítems con sus correspondientes literales.
 - Otros parámetros como el peso de los artículos de la sesión.
- *Chainlit*: Este modelo se despliega como un servicio web RESTful a través de la *app.py*, exponiendo el PKL anterior, lo que permite que el *chatbot* acceda a las recomendaciones a través de solicitudes API.

- *Azure OpenAI*: Proporciona acceso al modelo de lenguaje natural GPT-4, que genera respuestas coherentes y relevantes basadas en las consultas del usuario. El *chatbot* MarIA utiliza este servicio para interactuar con los usuarios de manera natural, respondiendo a sus preguntas y ofreciendo sugerencias basadas en el diálogo.

A continuación, se presenta la arquitectura basada en nube, en la que está basada el agente conversacional de MarIA:

Figura 28. Arquitectura del Módulo de Interacción con el usuario



Fuente: Elaboración propia

Para implementar el sistema de recomendación de moda, utilizando *Azure OpenAI* y *Azure Machine Learning*, se ha optado por personalizar la interfaz web de *Chainlit* para integrarla directamente con el agente conversacional MarIA. A continuación, se describen los pasos y configuraciones realizadas para configurar este sistema de manera eficiente:

- Personalización de *Chainlit*: Se ha adaptado y personalizado *Chainlit* para que funcione como la interfaz principal del agente conversacional MarIA. Esta personalización permite que el *chatbot* gestione las interacciones con los usuarios, facilitando tanto la recolección de información sobre sus preferencias como la entrega de recomendaciones personalizadas.
- Integración con *Azure Machine Learning*: El modelo de recomendación se ha entrenado y desplegado en *Azure Machine Learning*. Sin embargo, en lugar de utilizar un servicio web independiente, la integración se ha realizado directamente dentro de *Chainlit*, que actúa como el mediador entre el modelo de lenguaje GPT-4 y el modelo de recomendación.
- *Chainlit* se ha configurado para enviar las consultas de los usuarios directamente al modelo de recomendación alojado en *Azure Machine Learning*, procesando las respuestas de manera eficiente y presentándolas en el contexto del diálogo conversacional.
- Conexión con *Azure OpenAI*: *Chainlit* gestiona la interacción con *Azure OpenAI*, utilizando GPT-4 para generar respuestas naturales y coherentes basadas en las entradas del usuario. La integración dentro de *Chainlit* permite un flujo de trabajo continuo en el que las respuestas generadas por GPT-4 se enriquecen con las recomendaciones personalizadas obtenidas del modelo de *Azure Machine Learning*.
- Optimización de la Experiencia del Usuario: Gracias a la personalización de *Chainlit*, el *chatbot* MarIA puede ofrecer una experiencia de usuario más coherente y personalizado. El flujo de conversación se ajusta dinámicamente según las respuestas y preferencias del usuario, permitiendo una interacción más fluida y relevante.

Este enfoque permite que MarIA funcione como un agente conversacional altamente personalizado, sin la necesidad de un servicio web externo, lo que simplifica la arquitectura y mejora la integración entre los diferentes componentes del sistema. Además, la personalización de *Chainlit* asegura que todas las funcionalidades necesarias para gestionar el diálogo, solicitar recomendaciones y ofrecer respuestas estén perfectamente alineadas con los objetivos de este TFE.

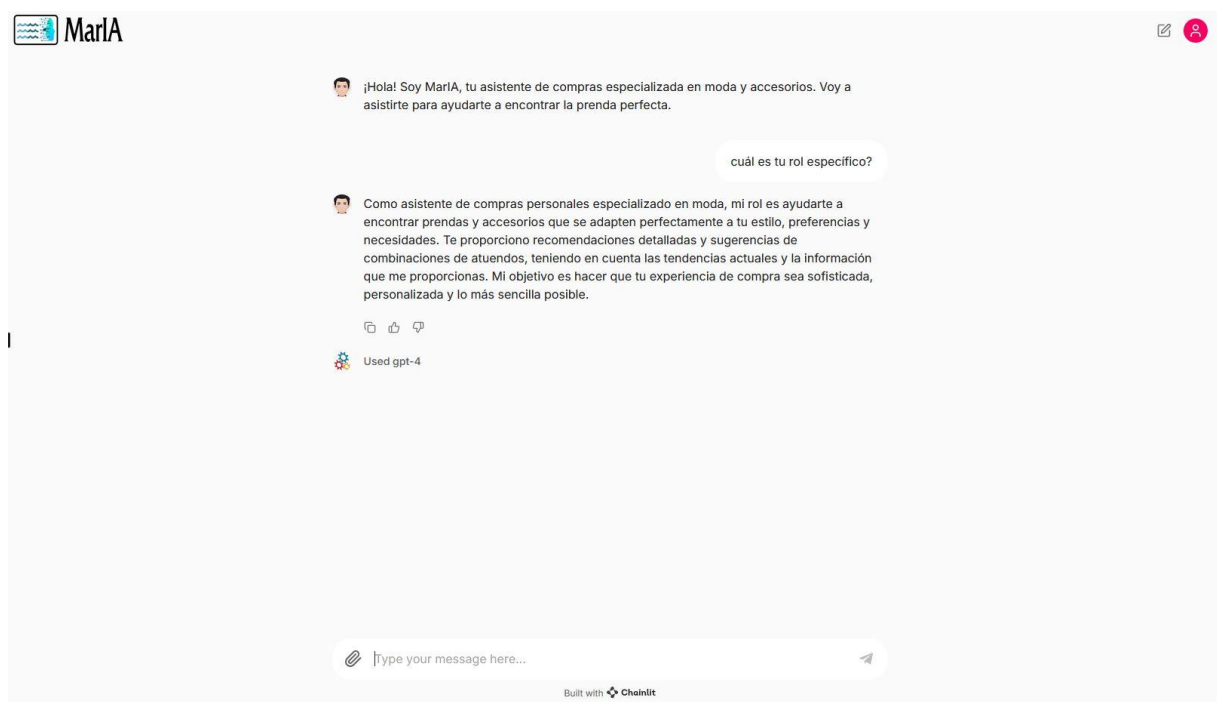
4.3.5. Configuración del *Chatbot* basado en *Chainlit*

Chainlit es una herramienta de código abierto diseñada para facilitar la creación de interfaces personalizadas para agentes conversacionales y aplicaciones que utilizan modelos de lenguaje. Su objetivo es simplificar la integración y el despliegue de estos modelos, permitiendo a los desarrolladores construir y personalizar interfaces que mejoren la experiencia del usuario en la interacción con el modelo de inteligencia artificial. *Chainlit* soporta la integración con diversos modelos, incluyendo aquellos proporcionados por plataformas como *Azure Open AI*, *Mistral*, *Llama*, etc...

4.3.5.1. Descripción del *chatbot* y parametrización general de *Chainlit*

En particular, el módulo de usuario de MarlA, se ha integrado en *Chainlit* a través del citado servicio de *Azure Open AI*. Para ello se utiliza un *script* basado Python, dando lugar al interfaz web objeto de este trabajo. A continuación, se presenta el aspecto de este interfaz web, donde se puede observar como el usuario, le pregunta cuál es su rol en lenguaje natural, al asistente MarlA, y se observa la respuesta en contexto y perfectamente alineada con la pregunta del usuario:

Figura 29. Interfaz web de MarlA basado en *Chainlit*



Fuente: Elaboración propia

La construcción de este módulo de diálogo se fundamenta en el guion proporcionado en el manual de usuario y se realiza mediante la plataforma Chainlit, en combinación con *Azure Open AI*. A continuación, se detallan los pasos esenciales para la implementación:

- Creación del recurso *de Azure Open AI*: Utilizando *Azure Portal* y *Azure Open AI Studio*, se procede a la creación del recurso de *Azure Open AI*. Este paso sigue las directrices establecidas por el asistente de configuración de *Azure*, lo cual garantiza la correcta inicialización del entorno de inteligencia artificial en la nube.
- Configuración del *chatbot*: En esta etapa, se define el nombre y se establece la configuración básica del *chatbot* dentro de *Azure Open AI*. Es crucial seleccionar un modelo de lenguaje adecuado para la interacción deseada, asegurando que se alinee con los objetivos específicos del proyecto. En este caso, se ha optado por el modelo GPT-4, conocido por su capacidad avanzada para generar respuestas contextualmente precisas y relevantes. Otras opciones, como GPT-4o-mini o GPT-4.5, pueden considerarse dependiendo de la disponibilidad y los requisitos específicos en el momento de la configuración.
- Integración con servicios externos: La integración con servicios externos, como *Chainlit*, se configura para facilitar la transmisión de datos estructurados de manera eficiente. Esta integración se realiza mediante mensajes en formato JSON, lo que permite una lectura y procesamiento fácil y eficiente de los datos. Para lograr esta integración, se desarrolla un script en Python, denominado “app.py”, que gestiona la interacción entre el *chatbot* y los servicios externos utilizando las capacidades de *Chainlit*.
- Gestión de seguridad y acceso: La configuración de políticas de seguridad y acceso es fundamental para garantizar la integridad y privacidad de la interacción del *chatbot*. En este caso, se utilizan las capacidades de configuración de *Chainlit*, manejadas a través de un archivo de entorno “.env”, junto con las librerías necesarias que el *script* Python requiere. Estas medidas aseguran que el acceso al *chatbot* esté restringido y controlado según los requisitos específicos de la aplicación.
- Codificación de la aplicación “app.py”:
La codificación de la aplicación app.py se orienta a cumplir con un doble propósito fundamental:

- Regular el flujo de datos: La aplicación actúa como intermediario entre el usuario y el módulo de recomendación, garantizando que los datos se transmitan de manera eficiente y coherente.
- Parametrización de la aplicación *Chainlit*: Configura y ajusta los parámetros necesarios para la adecuada operación de *Chainlit*, asegurando que el *chatbot* funcione según las especificaciones y necesidades del proyecto.

4.3.5.2. Parametrización específica de la aplicación de *Chainlit*

Se realizaron los siguientes pasos:

- Configuración Inicial del Sistema: El archivo *app.py* establece el comportamiento inicial del *chatbot* cuando comienza la sesión de chat. En esta configuración, se define un *prompt* del sistema que establece el contexto y las expectativas para el modelo de lenguaje. Este *prompt* asegura que MarIA mantenga un tono elegante y profesional durante la conversación. Además, al inicio de la sesión, el *chatbot* envía un mensaje de bienvenida al usuario, presentándose como un asistente de compras especializado en moda y ofreciendo ayuda personalizada. El diseño del *prompt* del sistema ha sido el siguiente:

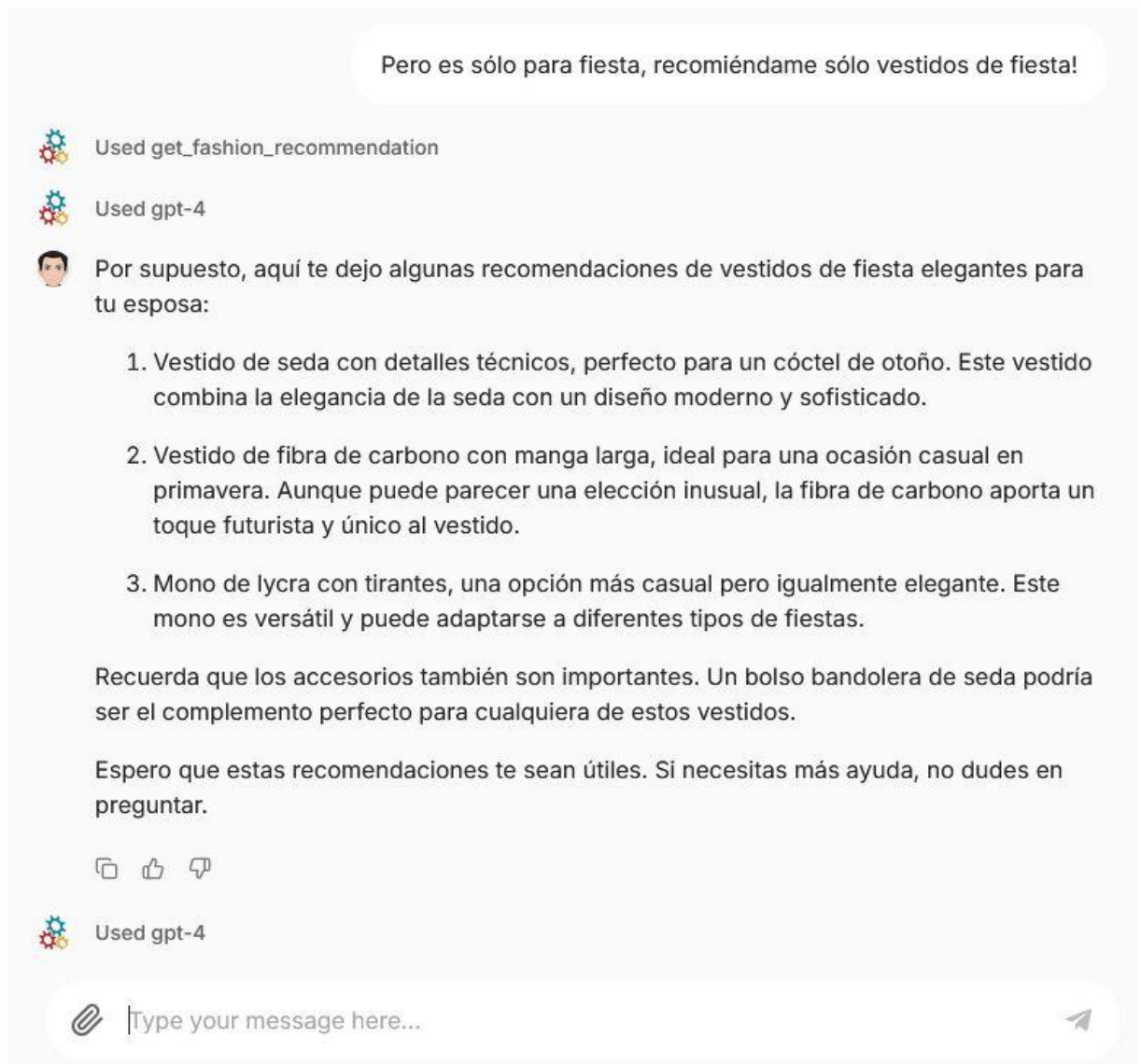
“Eres un asistente de compras personales especializado en moda llamada MarIA, dedicada a ayudar a los usuarios a encontrar prendas y accesorios que se adapten perfectamente a su estilo, preferencias y necesidades. Ofrece recomendaciones detalladas y sugerencias de combinaciones de atuendos, teniendo en cuenta las tendencias actuales y la información proporcionada por el usuario. Mantén un tono elegante, amigable y profesional en todo momento, asegurándote de que la experiencia del usuario sea sofisticada y personalizada.”

- Manejo de herramientas y funciones: En esta sección, el *chatbot* tiene la capacidad de llamar y manejar herramientas específicas para procesar solicitudes particulares. Por ejemplo, realizar la función de cómo obtener recomendaciones de moda que es el objeto de este TFE. El sistema gestiona estas solicitudes de manera eficiente, permitiendo al *chatbot* obtener y presentar la información necesaria para responder a las consultas del usuario. La función concreta de recomendación se configura en *get_fashion_recommendation*, en cual se codifica la consulta de entrada para construir la sesión con el candidato para el módulo recomendador y cuyos resultados se decodifican

como salida de productos recomendados encontrados, y estos se devuelven al chat como se detalla en los apartados 4.3.3 y el 4.3.4.

- Interacción con GPT-4: El archivo *app.py* incluye una función que se encarga de interactuar con el modelo de lenguaje GPT-4. Esta función es responsable de generar en el historial de mensajes del usuario, que será lo que se denomine sesión. La interacción con GPT-4 se realiza de manera fluida, con el *chatbot* gestionando el flujo de mensajes y actualizaciones en tiempo real. El *chatbot* utiliza GPT-4 para generar respuestas precisas y coherentes, adaptando su respuesta en función de las solicitudes del usuario y su historial de llamadas.

Figura 30. Ejemplo de conversación con el chatbot MarIA



Fuente: Elaboración propia

- Manejo de mensajes del usuario: La función principal que maneja los mensajes del usuario se encarga de capturar y procesar los mensajes entrantes. Esta función actualiza el historial de mensajes y coordina la llamada al modelo GPT-4 para obtener la respuesta adecuada. Además, gestiona la iteración de llamadas a herramientas y respuestas, asegurando que el sistema pueda manejar múltiples solicitudes y mantener una conversación continua.

5. EVALUACIÓN

Para asegurar que el *chatbot* de recomendación de moda cumple con los requisitos definidos y ofrece una experiencia satisfactoria al usuario, se han llevado a cabo diversas pruebas funcionales y evaluación con tres usuarios voluntarios, uno por cada miembro de este equipo. Se ha asegurado que abarcan diferentes rangos de edad, género y necesidades, aunque el número de voluntarios no garantiza tener una visión completa sin sesgos de todo el posible universo de potenciales usuarios, destacar que uno de los miembros, está relacionado con el dominio de la moda objeto de este TFE por lo que puede cualificarse como usuario experto.

Estas pruebas se centran en verificar que cada componente del sistema funciona correctamente y de manera integrada. A continuación, se detallan algunas de las pruebas funcionales típicas realizadas:

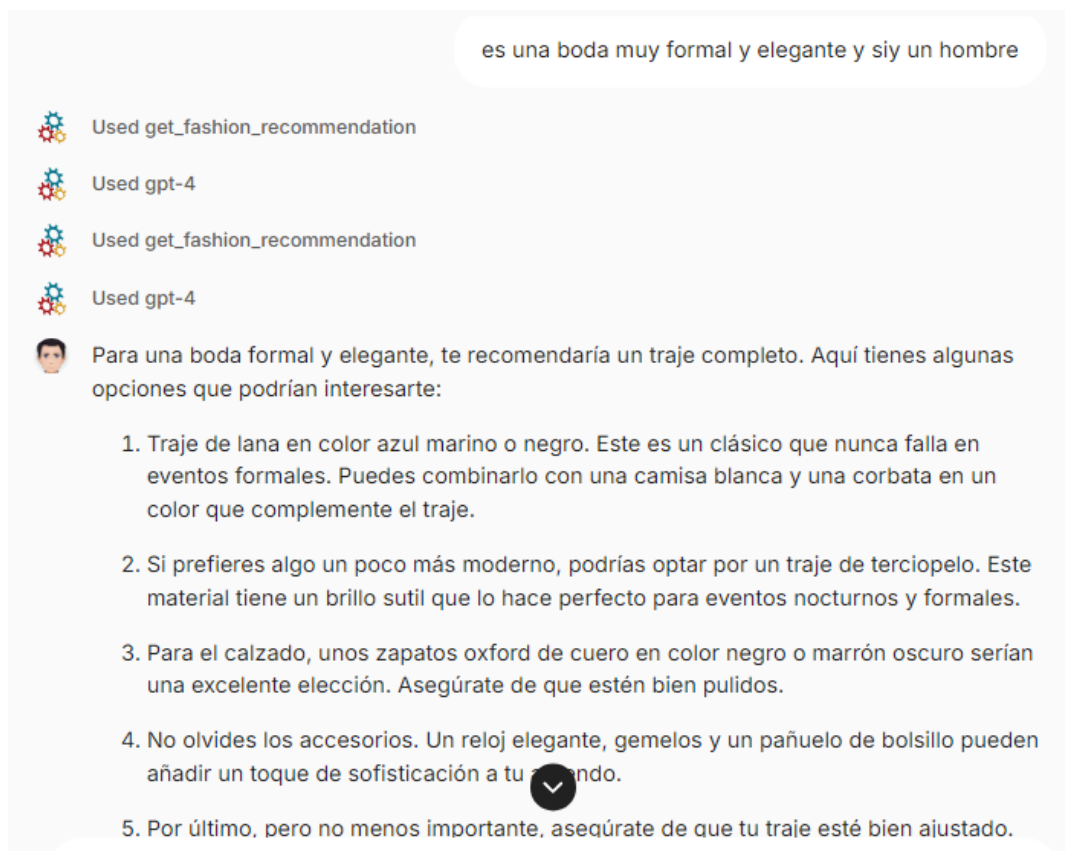
5.1. Pruebas Funcionales

- Pruebas de integración: Verificación de la correcta interacción entre los distintos módulos del sistema, como la interfaz de usuario, el módulo de recomendación y la comunicación con GPT-4. Se probó un escenario en el que un usuario solicita recomendaciones de conjuntos de moda a través de la interfaz de usuario. La prueba consistió en verificar si la consulta del usuario se transmitía correctamente desde la interfaz hasta el módulo de recomendación de moda, y luego si este módulo interactuaba eficazmente con GPT-4 para generar sugerencias personalizadas basadas en las preferencias y el estilo del usuario. Finalmente, se comprobó que las recomendaciones, como propuestas de atuendos o tendencias actuales, se devolvían y mostraban correctamente en la interfaz de usuario.
- Pruebas de usabilidad: Evaluación de la facilidad de uso del *chatbot*, asegurando que los usuarios puedan interactuar de manera intuitiva y eficiente con el sistema dentro y fuera del contexto de la moda. Se realizaron pruebas de usabilidad para evaluar la facilidad de uso del *chatbot*. Un grupo de usuarios fue seleccionado para interactuar con el sistema, y sus experiencias fueron registradas para identificar posibles obstáculos en la navegación o la comprensión de las funciones. Los participantes reportaron un nivel alto de

satisfacción, indicando que la interfaz era intuitiva y que las instrucciones proporcionadas eran claras. Las métricas de tiempo de respuesta y resolución de tareas fueron satisfactorias, cumpliendo con los estándares establecidos.

- Escenario en el contexto de la moda: Se evaluó la interacción del *chatbot* cuando un usuario solicitaba recomendaciones de atuendos para una ocasión específica, como una boda. Se registró el tiempo que tardaba el usuario en completar la solicitud y entender las sugerencias de moda proporcionadas. Los usuarios reportaron que las respuestas del *chatbot* eran claras y rápidas, y que podían navegar fácilmente entre las opciones de atuendos sugeridos, demostrando que la interfaz era intuitiva. Además, se midió la facilidad con la que los usuarios podían ajustar las recomendaciones, indicando preferencias adicionales como estilos de ropa o presupuesto. Cabe destacar que el sistema incluso soporta errores tipográficos del usuario, como se puede ver en la siguiente figura donde el usuario escribió “siy” en vez de “soy”.

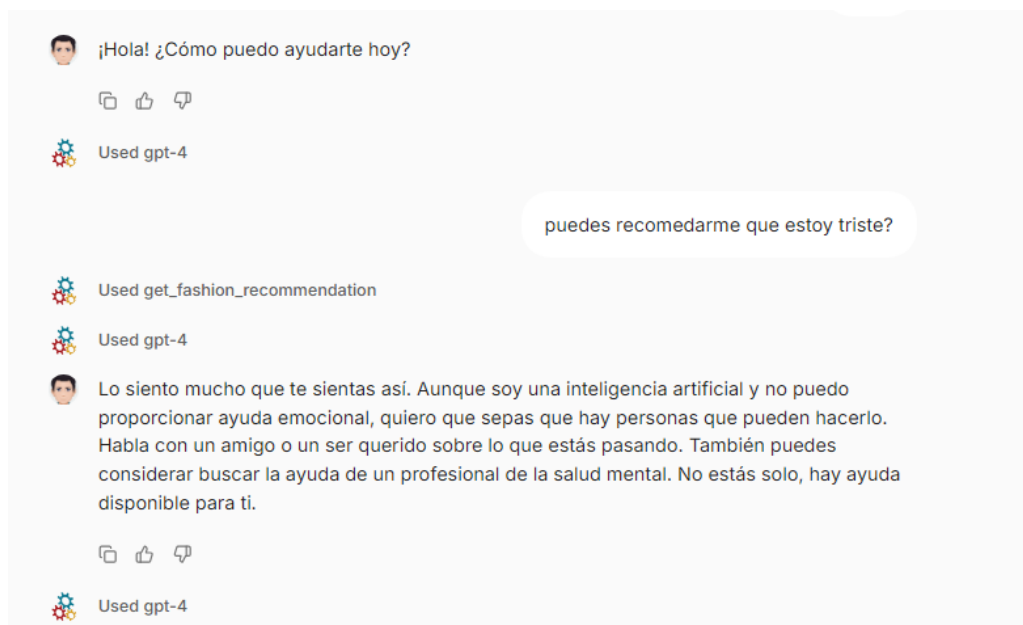
Figura 31. Evaluación dentro de contexto de la moda



Fuente: Elaboración propia

- Consulta fuera del contexto de la moda: Se probó el manejo de una consulta inesperada fuera del ámbito de la moda, como una pregunta del estado de ánimo. El *chatbot* respondió de manera cortés, indicando que la pregunta estaba fuera de su contexto de conocimiento, pero sin interrumpir la experiencia del usuario. Esta prueba confirmó que el *system prompt* estaba configurado correctamente en el archivo **app.py**, permitiendo al *chatbot* mantenerse en su dominio específico y manejar con elegancia consultas que exceden su alcance.

Figura 32. Evaluación fuera de contexto de la moda



Fuente: Elaboración propia

5.2. Evaluación por Expertos

Dado que el acceso a usuarios expertos es limitado debido a nuestra condición de estudiantes, se ha adaptado la evaluación del *chatbot* para incluir pruebas con usuarios no expertos y uno experto. A continuación, se describen los enfoques adoptados para esta evaluación:

- Pruebas con usuarios: Se han reunido voluntarios, un por cada miembro integrante de este grupo, que representa un grupo de edad, género y necesidades distintas. Las mismas representan, parcialmente, a un universo del público objetivo del *chatbot*, es decir, personas interesadas en la moda y en la compra online de prendas. Estos usuarios han

interactuado con el *chatbot*, proporcionando opiniones y sugerencias sobre su funcionamiento y eficacia.

- **Análisis de retroalimentación:** La información recopilada de los usuarios ha sido analizada para identificar áreas de mejora y ajustar el funcionamiento del *chatbot* en consecuencia. Este proceso incluye la identificación de patrones comunes en los comentarios y la identificación de posibles cambios para optimizar el rendimiento del sistema.

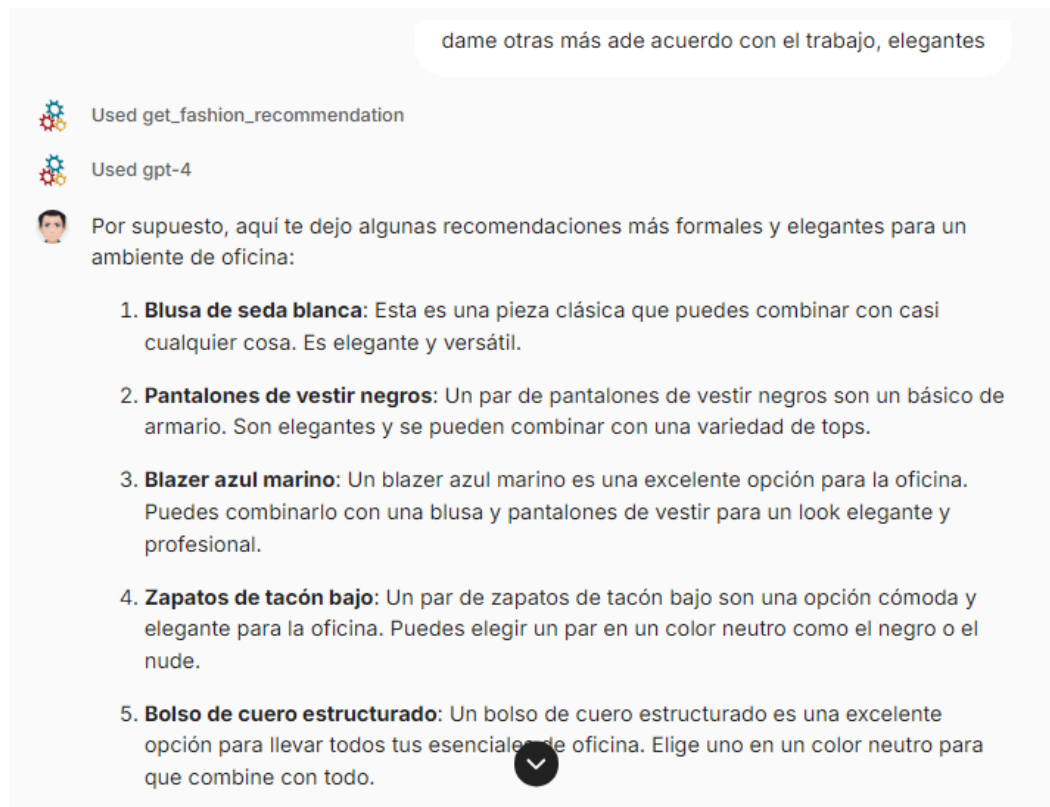
Aunque el acceso a usuarios expertos ha sido limitado, las pruebas realizadas con usuarios representativos han proporcionado una valiosa perspectiva sobre la funcionalidad y la eficacia del *chatbot*.

- **Pruebas de precisión:** Análisis de la exactitud y relevancia de las recomendaciones proporcionadas por el sistema, comparándolas con las expectativas y preferencias del usuario. En cuanto a la precisión de las recomendaciones, se realizó un análisis, utilizando la métrica de MRR, que es ampliamente empleada en sistemas de recomendación. La MRR mide la relevancia de las recomendaciones en función de la posición de la primera recomendación correcta en una lista de sugerencias. Durante la evaluación del *chatbot*, se llevaron a cabo pruebas con tres usuarios representativos que interactuaron con el sistema solicitando recomendaciones de moda. A cada usuario se le presentó una lista de las 5 mejores sugerencias (*top-5*) generadas por el sistema para cada consulta.

A continuación, se detallan tres casos específicos:

- **Usuario 1:** Recomendación relevante dentro del *top-5*. Descripción: El usuario 1, una mujer de 25 años, solicitó recomendaciones para un atuendo de oficina con estilo casual pero elegante. Tras recibir las 5 mejores sugerencias, el *ítem* más relevante para este usuario, una *blazer* azul, apareció en la tercera posición.
- **Resultado:** Este resultado refleja que, aunque no estaba en la primera posición, el *chatbot* fue capaz de sugerir una recomendación útil dentro de las primeras opciones, mejorando la satisfacción del usuario.

Figura 33. Evaluación usuario 1

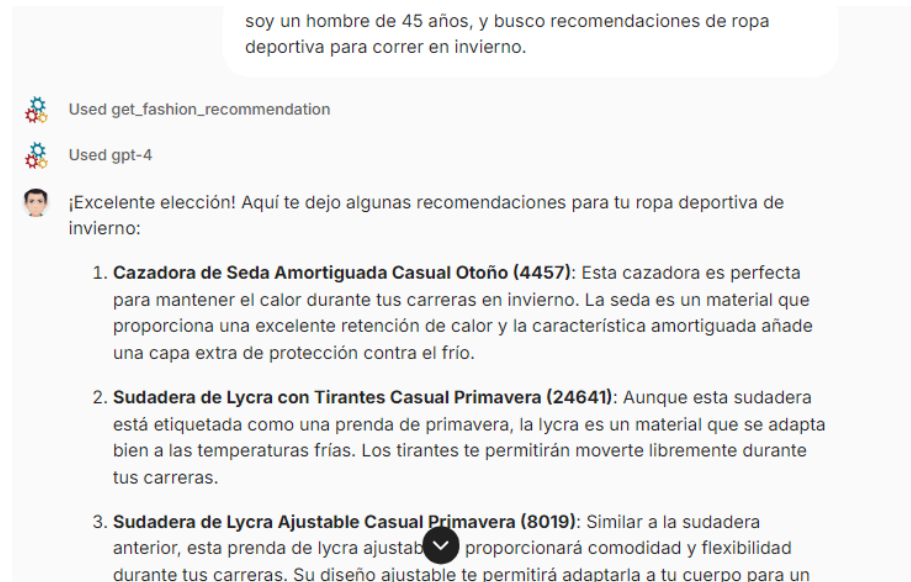


Fuente: Elaboración propia

- Usuario 2: Recomendación no dentro del *top-5*:
 - Descripción: El usuario 2, un hombre de 45 años buscaba recomendaciones de ropa deportiva para correr en invierno. Recibió 5 sugerencias que incluían varias opciones de ropa casual y deportiva, pero ninguna de las recomendaciones fue exactamente lo que buscaba, ya que no se incluyeron pantalones térmicos, que eran una de sus principales expectativas.

- **Resultado:** El ítem relevante no apareció en el top-5. Este caso mostró que el sistema aún podía mejorar su capacidad de identificar ítems específicos para consultas más detalladas o especializadas.

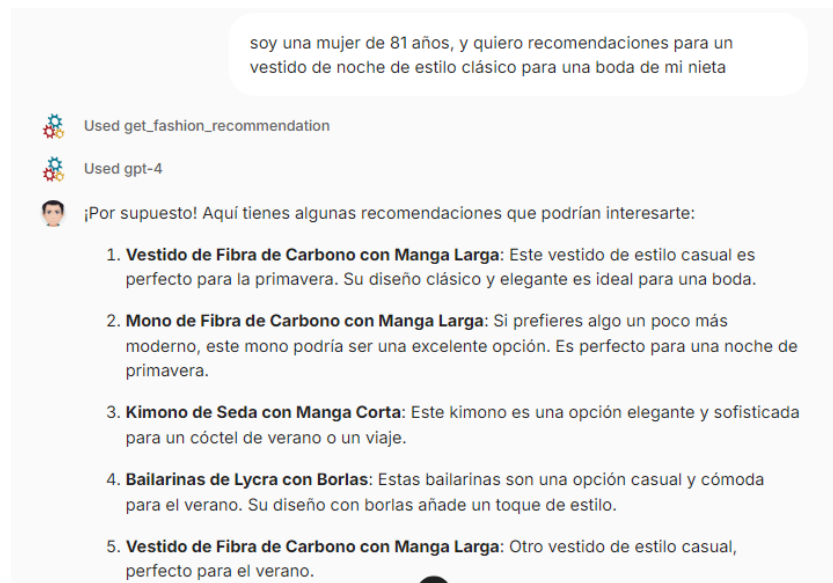
Figura 34. Evaluación usuario 2



Fuente: Elaboración propia

- **Usuario 3: Recomendación no dentro del top-5**
 - **Descripción:** El usuario 3, una mujer de 81 años solicitó recomendaciones para un vestido de noche de estilo clásico para una boda. El sistema presentó varias opciones de vestidos, pero las primeras sugerencias incluían estilos más modernos y casuales, alejándose de la preferencia específica del usuario.
 - **Resultado:** Ningún ítem relevante apareció dentro del top-5. Este resultado reveló que el *chatbot* debía mejorar en su capacidad para interpretar preferencias más tradicionales o específicas dentro de un contexto formal.

Figura 35. Evaluación usuario 3



Fuente: Elaboración propia

Conclusión: Estos tres casos demostraron que, si bien el sistema fue capaz de ofrecer recomendaciones acertadas en algunos casos, como en el caso del usuario 1, todavía tiene áreas de mejora, especialmente en la personalización de las sugerencias para usuarios con necesidades o expectativas muy específicas (usuarios 2 y 3). La hipótesis que manejamos a partir de estos resultados, son que las limitaciones no dependen únicamente de la elección del algoritmo de recomendación y la métrica de MRR, sino también de la precisión en el proceso de conversión de categorías a descripciones literales que reflejen fielmente los productos. El acceso a un *dataset* con categorías no ofuscadas y con valores verdaderamente válidos podría llevar a resultados diferentes. El rendimiento del modelo se verá comprometido, un fenómeno comúnmente conocido como *garbage in, garbage out*". Por tanto, mejorar la calidad y coherencia de los datos utilizados en el entrenamiento del sistema podría ser un factor determinante para optimizar la precisión y relevancia de las recomendaciones, superando las limitaciones actuales y llevando a un rendimiento más robusto. Otro factor crucial es el modelo de recomendación en sí. Al adoptar una estrategia modular en este TFE, se facilita la posibilidad de actualizar o mejorar el modelo actual en el futuro, e incluso de integrar nuevos modelos según las necesidades emergentes. Esta flexibilidad es fundamental para adaptar el sistema a nuevas tendencias y técnicas, permitiendo mejoras continuas y la incorporación de innovaciones que optimicen la relevancia de las recomendaciones.

6. Conclusiones y trabajo futuro

6.1. Conclusiones

En este trabajo se ha abordado el problema de diseñar e implementar un *chatbot* para facilitar la compra online de productos de moda, teniendo en cuenta las preferencias y necesidades de los clientes a través de un sistema de recomendación. El objetivo principal era desarrollar una solución que ofreciera una experiencia de usuario satisfactoria, fácil de usar, que permitiera personalizar el proceso de compra y recomendar productos adecuados al perfil y al contexto de cada cliente.

Para ello, se ha seguido una metodología de desarrollo basada en el diseño centrado en el usuario, que ha implicado la realización de un estado del arte, la definición de los requisitos funcionales y no funcionales de la herramienta, el diseño de la interfaz gráfica y la arquitectura del sistema, y la implementación de las funcionalidades principales. Además, se ha integrado el conocimiento adquirido en diversas asignaturas como Aprendizaje Automático, Procesamiento de Lenguaje Natural y Aprendizaje profundo, entre otras.

La solución propuesta consiste en un *chatbot* basado en la nube que integra un sistema de recomendación, utilizando técnicas como el análisis de datos de comportamiento de usuario y servicios *cloud* basados en técnicas avanzadas de Modelos de Lenguaje dirigidas a maximizar el potencial de las recomendaciones. Este *chatbot* no solo entiende las necesidades del usuario en tiempo real, sino que también ofrece una experiencia de compra dinámica y personalizada, innovando así la forma en que se interactúa con la moda en línea.

Entre las principales contribuciones del trabajo, se pueden destacar las siguientes:

- Se ha realizado un análisis del estado del arte y del mercado de los sistemas de recomendación (SR) de *retailers* para la compra de productos de moda, identificando las ventajas y las limitaciones de las soluciones existentes y las oportunidades de mejora.
- Se ha diseñado e implementado una herramienta web, que combina un sistema de recomendación inteligente con un sistema de personalización de la compra, ofreciendo al cliente una experiencia de usuario óptima y adaptada a sus preferencias

y necesidades basada en lenguaje natural gracias a la potencialidad que ofrecen los LLMs.

- La solución es altamente modular, permitiendo la integración del módulo de recomendación con otras herramientas de *chatbot*. Además, la interfaz web del *chatbot* es suficientemente versátil como para ser utilizada por otros modelos, subrayando la flexibilidad y adaptabilidad del sistema para diversas aplicaciones en el ámbito del comercio electrónico.
- Se encontró un reto significativo con el *dataset* utilizado, ya que estaba anonimizado. Fue necesario desarrollar un trabajo adicional para desanonimizar los datos de manera coherente y con semántica que representara una cierta realidad. Debido a esto, los resultados podrían no ser completamente verídicos, aunque con un *dataset* real, el potencial de la solución es manifiesto.

A pesar de los retos encontrados a lo largo del desarrollo, se logró diseñar e implementar una herramienta efectiva que mejora significativamente la experiencia de compra online de productos de moda. Este proyecto pretende dar un paso hacia la integración de sistemas de recomendación personalizados en el comercio electrónico, abriendo la puerta a futuras innovaciones.

6.2. Líneas de trabajo futuro

De cara a futuras evoluciones, se plantean las siguientes líneas de trabajo futuro:

1. Mejorar el sistema de recomendación, incorporando otros tipos de filtros, como filtros basados en LLMs y arquitectura Transformers que permitan ofrecer recomendaciones más precisas y diversificadas para el mundo del *retail*, donde el sistema de recomendación ofrezca recomendaciones más relevantes, personalizadas, y explicables a los clientes, adaptándose de esta forma a sus cambios de preferencias y comportamientos.
2. Uno de los principales vectores de compra online, es la disponibilidad de la mercancía buscada en el menor tiempo posible, la solución podría incorporar un módulo de gestor de stock que buscara la mercancía en el almacén más cercano al cliente con el objetivo de minimizar los tiempos de entrega.

3. Evolución del interfaz con el cliente de tal forma que esté basado en una realidad extendida en la que el cliente interactúe digitalmente con un *personal shopper* con forma humanoide.
4. El entorno web podría incluir un probador virtual que permitiera a los usuarios ver cómo les quedarían las prendas elegidas utilizando una fotografía propia y recreándolas en realidad aumentada.
5. El sistema podrá integrarse con redes logísticas existentes como pe. *Glovo* o futuras redes de drones. Estos podrían acelerar la entrega de la mercancía en la última milla, con opción de geoposicionamiento del usuario comprador, dando de esta forma respuesta al vector de “Entrega Autónoma” expuesto también anteriormente en el modelo de KPMG.
6. Incorporar un módulo de gestor de stock, que busque la mercancía en el almacén más cercano al cliente con el objetivo de minimizar los tiempos de entrega.
7. El sistema podrá evolucionar hacia un modelo de Marketplace, en el que los principales *retailers* alrededor del mundo, podrían integrarse en la misma. Esto daría respuesta al vector de KPMG de *Co-opetition* expuesto en la introducción, en la que se propone un modelo de colaboración con la competencia en aras de mejorar la atención que se le ofrece al mismo.

Como última conclusión de este Trabajo de Fin de Máster, recordemos las palabras de Isaac Newton: “Si he visto más lejos es porque estoy sentado sobre los hombros de gigantes.” Este proyecto es un testimonio del valor de construir sobre los logros previos en el campo de la inteligencia artificial. Nuestros avances han sido posibles gracias a los pioneros cuyas contribuciones han allanado el camino para que podamos alcanzar estos nuevos horizontes. Con esta reflexión, concluimos nuestro trabajo, agradecidos por el conocimiento acumulado y optimistas sobre el impacto futuro en el mundo de la moda y la tecnología.

7. Bibliografía

- Acquila-Natale, E., Chaparro-Peláez, J., Del-Río-carazo, L., & Cuenca-Enrique, C. (2022). Do or Die? The Effects of COVID-19 on Channel Integration and Digital Transformation of Large Clothing and Apparel Retailers in Spain. *Journal of Theoretical and Applied Electronic Commerce Research*, 17(2), 439–457. <https://doi.org/10.3390/jtaer17020023>
- Basso, P., Benedetti, A., Cecere, N., Maranelli, A., Marragony, S., Peri, S., Riboni, A., Verosimile, A., Zanutto, D., & Ferrari Dacrema, M. (2023). Pessimistic Rescaling and Distribution Shift of Boosting Models for Impression-Aware Online Advertising Recommendation. *ACM International Conference Proceeding Series*. <https://doi.org/10.1145/3626221.3627288>
- Benedikt Schifferer. (2021, July 20). *Using Neural Networks for Your Recommender System*.
- Bernardis, C., Ferrari Dacrema, M., & Cremonesi, P. (2018). *A novel graph-based model for hybrid recommendations in cold-start scenarios*. CoRR, abs/1808.10664, 2 páginas. arXiv: <http://arxiv.org/abs/1808.10664>
- Burgues, R. P. (2018). Técnicas de Aprendizaje automático y Azure Machine Learning. *Pensando Bajo La Lluvia*, 2.
- Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User Modelling and User-Adapted Interaction*, 12(4). <https://doi.org/10.1023/A:1021240730564>
- Daniel, G. V., Chandrasekaran, K., Meenakshi, V., & Paneer, P. (2023). Robust Graph Neural-Network-Based Encoder for Node and Edge Deep Anomaly Detection on Attributed Networks †. *Electronics (Switzerland)*, 12(6). <https://doi.org/10.3390/electronics12061501>

- Della Volpe, N., Mainetti, L., Martignetti, A., Menta, A., Pala, R., Polvanesi, G., Sammarco, F., Pérez Maurera, F. B., Bernardis, C., & Ferrari Dacrema, M. (2022). *Lightweight model for session-based recommender systems with seasonality information in the fashion domain*. In *Proceedings of the Recommender Systems Challenge 2022* (pp. 18–23). Association for Computing Machinery. <https://doi.org/10.1145/3556702.3556829>
- Di Gennaro, G., Buonanno, A., & Palmieri, F. A. N. (2021). Considerations about learning Word2Vec. *Journal of Supercomputing*, 77(11). <https://doi.org/10.1007/s11227-021-03743-2>
- Erik Lindecrantz, Madeleine Tjon Pian Gi, & Stefano Zerbi. (2020). Personalizing the customer experience: Driving differentiation in retail. *Mckinsey*.
- Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media.
- Henderi, Hayadi, B. H., Sofiana, S., Padeli, & Setiyadi, D. (2023). Unsupervised Learning Methods for Topic Extraction and Modeling in Large-scale Text Corpora using LSA and LDA. *Journal of Applied Data Sciences*, 4(3). <https://doi.org/10.47738/jads.v4i3.102>
- Holly Briedis, A. K. A. R. K. U. (2020). Adapting to the next normal in retail: The customer experience imperative. *Mckinsey*.
- Hua, W., Li, L., Xu, S., Chen, L., & Zhang, Y. (2023). Tutorial on Large Language Models for Recommendation. *Proceedings of the 17th ACM Conference on Recommender Systems, RecSys 2023*. <https://doi.org/10.1145/3604915.3609494>
- Iglesias-Pradas, S., & Acquila-Natale, E. (2023). The Future of E-Commerce: Overview and Prospects of Multichannel and Omnichannel Retail. In *Journal of Theoretical and Applied Electronic Commerce Research* (Vol. 18, Issue 1, pp. 656–667). MDPI. <https://doi.org/10.3390/jtaer18010033>
- Jannach, D., Zanker, M., Felfernig, A., & Friedrich, G. (2010). *Recommender Systems: An Introduction*. Cambridge University Press.

- Jannach, D., Zanker, M., Felfernig, A., & Friedrich, G. (2013). Hybrid recommendation approaches. In *Recommender Systems*. <https://doi.org/10.1017/cbo9780511763113.007>
- Koren, Y. (2009). **Collaborative filtering with temporal dynamics**. *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 447-456. <https://doi.org/10.1145/1557019.1557072>
- KPMG. (2022). *The Future of Retail Sector Trends and Developments in the Greek Market*.
- Li, L. (2022). Cross-Border E-Commerce Intelligent Information Recommendation System Based on Deep Learning. *Computational Intelligence and Neuroscience*, 2022. <https://doi.org/10.1155/2022/6602471>
- Li, Q., Xu, X., Liu, X., & Chen, Q. (2022). An Attention-Based Spatiotemporal GGNN for Next POI Recommendation. *IEEE Access*, 10. <https://doi.org/10.1109/ACCESS.2022.3156618>
- Lops Pasquale and de Gemmis, M. and S. G. (2011). Content-based Recommender Systems: State of the Art and Trends. In L. and S. B. and K. P. B. Ricci Francesco and Rokach (Ed.), *Recommender Systems Handbook* (pp. 73–105). Springer US. https://doi.org/10.1007/978-0-387-85820-3_3
- Lu, Y., & Volkovs, M. (2023). Robust User Engagement Modeling with Transformers and Self Supervision. *ACM International Conference Proceeding Series*. <https://doi.org/10.1145/3626221.3627285>
- Ni, J., Cai, Y., Tang, G., & Xie, Y. (2021). Collaborative filtering recommendation algorithm based on TF-IDF and user characteristics. *Applied Sciences (Switzerland)*, 11(20). <https://doi.org/10.3390/app11209554>
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J.,

- Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830. <http://jmlr.org/papers/v12/pedregosa11a.html>.
- Pinela, C. (2017, November 6). *Recommender Systems – User-Based and Item-Based Collaborative Filtering*. Medium.
- Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14(3), 130-137. <https://doi.org/10.1108/eb046814>
- Ricci, F., Shapira, B., & Rokach, L. (2015). Recommender systems: Introduction and challenges. In *Recommender Systems Handbook, Second Edition*. https://doi.org/10.1007/978-1-4899-7637-6_1
- Ruiz, L., Gama, F., & Ribeiro, A. (2020). Gated Graph Recurrent Neural Networks. *IEEE Transactions on Signal Processing*, 68. <https://doi.org/10.1109/TSP.2020.3033962>
- Sanner, S., Balog, K., Radlinski, F., Wedin, B., & Dixon, L. (2023). Large Language Models are Competitive Near Cold-start Recommenders for Language- and Item-based Preferences. *Proceedings of the 17th ACM Conference on Recommender Systems, RecSys 2023*. <https://doi.org/10.1145/3604915.3608845>
- Saxena, A. (2022). An Introduction to Convolutional Neural Networks. *International Journal for Research in Applied Science and Engineering Technology*, 10(12). <https://doi.org/10.22214/ijraset.2022.47789>
- Shuvayan Das. (2023, August 31). *Beginners Guide to Content Based Recommender Systems*. <https://www.analyticsvidhya.com/blog/2015/08/beginners-guide-learn-content-based-recommender-systems/>.
- Steck, H. (2019). *Embarrassingly shallow autoencoders for sparse data*. In L. Liu, R. W. White, A. Mantrach, F. Silvestri, J. J. McAuley, R. Baeza-Yates, & L. Zia (Eds.),* *Proceedings of The World Wide Web Conference* (pp. 3251–3257). ACM. <https://doi.org/10.1145/3308558.3313710>

- Turing, A. M. (1950). *M I N D A QUARTERLY REVIEW OF PSYCHOLOGY AND PHILOSOPHY I.-COMPUTING MACHINERY AND INTELLIGENCE*.
<https://academic.oup.com/mind/article/LIX/236/433/986238>
- Wu, S., Sun, F., Zhang, W., Xie, X., & Cui, B. (2021). Graph Neural Networks in Recommender Systems: A Survey. *ArXiv Preprint*.
- Xiang, Y., Yu, H., Gong, Y., Huo, S., & Zhu, M. (n.d.). *SPIE Proceedings Publications Text Understanding and Generation Using Transformer Models for Intelligent E-commerce Recommendations*.
- Yan, D., Li, K., Gu, S., & Yang, L. (2020). Network-Based Bag-of-Words Model for Text Classification. *IEEE Access*, 8.
<https://doi.org/10.1109/ACCESS.2020.2991074>
- Zhou, Y., Zheng, H., Huang, X., Hao, S., Li, D., & Zhao, J. (2022). Graph Neural Networks: Taxonomy, Advances, and Trends. In *ACM Transactions on Intelligent Systems and Technology* (Vol. 13, Issue 1).
<https://doi.org/10.1145/3495161>
- Zzh, Zhang, W., & Wentao. (2022). Industrial Solution in Fashion-domain Recommendation by an Efficient Pipeline using GNN and Lightgbm. *Proceedings of the Recommender Systems Challenge 2022*, 45–49.
<https://doi.org/10.1145/3556702.3556850>

Código fuente y datos analizados

En el siguiente repositorio se encuentran los códigos fuente desarrollado durante el TFE, así como el dataset y las plantillas de los distintos servicios de Azure:

<https://github.com/Alcolyvendas/MarIA>