# Tecnológico de Monterrey

**ShopMate**


| | |
|---|---|
| Rosa Vanessa Palacios Beltrán | A01652612 |
| Irving Yael López Solis | A01664809 |
| Diego Aguilar Torres | A01657884 |
| Santiago Calderón Ortega | A01663888 |
| Cynthia Amador Santiago | A01737854 |
| David Alberto Padrón Sánchez | A01663806 |
| Katia Geraldine Vidals Estrever | A01657587 |


Noviembre 2025

Inteligencia artificial avanzada para la ciencia de datos II

Profesores:

David Christopher Balderas Silva

Oscar Francisco Fuentes Casarrubias

José Ángel Martínez Navarro

Emmanuel Páez Lopez

Jesús Manuel Vázquez Nicolás

# 1. Resumen ejecutivo

Para el proyecto ShopMate, desarrollamos una solución de compras personalizadas para el sector de retail online basada en inteligencia artificial, con el objetivo de reducir el tiempo de búsqueda de productos y aumentar la tasa de conversión. La solución integró un motor de recomendación basado en embeddings, un backend de IA en Python, un chatbot en Oracle Digital Assistant y una aplicación móvil conectada a servicios de Oracle Cloud.

Partimos de un catálogo de productos de moda almacenado en Oracle Autonomous Database y realizamos un proceso completo de ingesta, limpieza e ingeniería de datos. Optamos por un enfoque moderno de recomendación embedding-based, donde cada producto se representa mediante un vector semántico generado con OCI Generative AI a partir de descripciones enriquecidas. Para este enriquecimiento se incorporaron etiquetas visuales obtenidas con OCI Vision, las cuales se integraron con la información textual del catálogo antes de generar los embeddings.

A partir de estos vectores se construyó un perfil vectorial para cada usuario, calculado mediante el promedio ponderado de los embeddings de los productos que ha comprado. Con esta estructura se habilitaron dos capacidades principales: búsqueda semántica, capaz de interpretar consultas en lenguaje natural, y recomendación personalizada, basada en la similitud entre el perfil del usuario y los productos del catálogo. Ambas funciones se integraron en un motor de búsqueda híbrida (texto + preferencias del usuario) ofrecido a través del backend de IA.

Finalmente, se realizaron pruebas end-to-end que validaron el flujo completo: desde la interacción en lenguaje natural y la navegación por la app hasta la obtención de recomendaciones adaptadas a cada perfil. El proyecto demostró que es posible implementar, con recursos de nube y técnicas modernas de IA, una experiencia de compra más eficiente, contextual y centrada en el usuario.

# 2. Objetivo del Proyecto

Diseñar e implementar una solución de recomendación inteligente para comercio electrónico que integre un modelo de recomendación y un chatbot conversacional basado en texto, utilizando la infraestructura de Oracle Cloud. La solución debe ser capaz de centralizar los datos de clientes, artículos y transacciones, enriquecer el catálogo con información derivada de las imágenes de producto, y ofrecer recomendaciones y búsquedas semánticas en lenguaje natural que mejoren la experiencia de navegación del usuario en la aplicación móvil.

## 2.1 Objetivos específicos

1. **Limpieza y preparación de datos.** Realizar la limpieza y estandarización de los tres conjuntos de datos (clientes, artículos y transacciones), logrando completar o corregir al menos el 90% de los datos faltantes en variables clave, con el fin de construir un catálogo consistente y utilizable por el sistema de recomendación.

2. **Centralización de la información en la nube.** Organizar y almacenar en Oracle Autonomous Database la información depurada de clientes, artículos e interacciones, verificando su accesibilidad mediante consultas de prueba y asegurando que el 100% de las tablas requeridas estén disponibles para el backend.

3. **Construcción del modelo de recomendación basado en embeddings**: Diseñar y entrenar un sistema de recomendación embedding-based que combine información semántica de productos y el historial de compra del usuario, evaluado mediante métricas como Recall@K, NDCG@K, Similitud de Coseno Promedio y Coverage@K. Se considerará exitoso si el modelo supera los baselines de popularidad y aleatoriedad en Recall@K y NDCG@K, mostrando alta coherencia semántica y un buen coverage del catálogo (más del 60% en Top 20).

4. **Implementación del servicio de recomendaciones y búsqueda por texto**: La implementación del servicio de recomendaciones y búsqueda por texto en el backend tiene como objetivo asegurar que el modelo de embeddings proporcione resultados semánticamente relevantes y alineados con la intención del usuario. Para ello, se espera

que el sistema mantenga un promedio de similitud superior a 0.50, una desviación estándar menor a 0.10 entre consultas y promedios de ranking de al menos 0.52 en Top-1, 0.51 en Top-3 y 0.50 en Top-5, reflejando consistencia en los primeros resultados recuperados. Asimismo, se prevé que el porcentaje de consultas por debajo del umbral de similitud baja (<0.30) sea inferior al 5%, garantizando que la mayoría de las búsquedas retornen artículos adecuados.

## 3. Alcances

**Desarrollo de un agente conversacional basado en texto:** Desarrollar un chatbot conversacional que interactúe con el usuario mediante lenguaje natural, capaz de recibir consultas sobre productos, estilos u ocasiones y devolver recomendaciones y resultados de búsqueda relevantes.

**Implementación de Personalización en Tiempo Real:** Tener en cuenta el historial de compra, estilo y disponibilidad de stock actual para ofrecer productos relevantes.

**Integración Completa con Oracle Cloud:** Conectar la solución con sistemas existentes como inventarios, utilizando la infraestructura robusta de Oracle Cloud.

**Optimización del Tiempo de Búsqueda y Mejora en la Conversión de Ventas:** A través de búsquedas inteligentes y un sistema de recomendación eficiente, reducir los tiempos de búsqueda y aumentar la satisfacción del cliente.

## 4. Organización del equipo y roles

El equipo de trabajo se organizó en células con responsabilidades claras:

- **Equipo de datos**: responsable de la ingesta, limpieza y modelado del catálogo en Autonomous Database.
- **Equipo de modelo IA/ML**: encargado del diseño, entrenamiento y evaluación del sistema de recomendación.

- **Equipo de backend**: responsable de la implementación de APIs, la integración con OCI y la orquestación del motor de recomendación.
- **Equipo de chatbot**: dedicado a la construcción de entidades, flujos conversacionales e integración con el backend.
- **Equipo de app móvil**: encargado de la interfaz de usuario, flujos de navegación y consumo de APIs.
- **Equipo de pruebas y documentación**: responsable de casos de prueba, verificación de requisitos y consolidación del reporte final.

Esta organización nos permitió paralelizar tareas y especializar esfuerzos sin perder la visión integral del proyecto.

## 5. Planificación de Tareas

En la siguiente liga se encuentra el diagrama de Gantt correspondiente a la planificación del proyecto, donde se visualiza el progreso actual y los objetivos alcanzables en las próximas semanas:
https://docs.google.com/spreadsheets/d/1pmkOyQ18QooqZUsxgDKBNI0lzw-UTuavYhg8MHno5ks/edit?usp=sharing

## 6. Estado del Arte en Personalización en Retail Online

En el contexto del comercio electrónico y los servicios de entretenimiento en línea, muchas plataformas han adoptado sistemas de recomendación personalizados para mejorar la experiencia de usuario, incrementar retención y elevar las conversiones. Los enfoques más efectivos combinan diferentes técnicas de IA: filtrado colaborativo, análisis de contenido, y en algunos casos búsqueda y representación semántica para entender mejor las intenciones del usuario. A continuación se presentan algunos antecedentes relevantes.

## Sistemas de Recomendación Híbridos

Los sistemas híbridos, que combinan filtrado colaborativo y análisis de contenido, están descritos en la literatura como una de las estrategias más robustas para generar recomendaciones personalizadas en entornos con catálogos amplios y usuarios variados.

Empresas líderes han implementado variantes de estos sistemas. Por ejemplo, según una publicación reciente, plataformas como Amazon, Netflix, Spotify y otros servicios, utilizan sistemas de recomendación personalizados basados en la actividad de los usuarios: historial de compras, visualizaciones o consumo, comportamiento de navegación, entre otros.

El sistema de recomendación de este proyecto sigue ese paradigma híbrido: aprovecha tanto las interacciones históricas de los usuarios como la información de los productos (contenido, metadatos, descripciones enriquecidas) para generar recomendaciones personalizadas.

## Búsqueda Semántica: Mejorando la Relevancia de los Resultados

La búsqueda semántica —es decir, la capacidad de interpretar la intención detrás de la consulta, en lugar de hacer una coincidencia exacta de palabras clave— se ha consolidado como una estrategia valiosa en comercio electrónico. Esta técnica mejora la calidad de los resultados, permite manejar sinónimos, variaciones de lenguaje, y satisface mejor la intención del usuario.

El uso de búsqueda semántica facilita que los usuarios encuentren productos aunque no utilicen los términos exactos del catálogo, lo que mejora la usabilidad y la satisfacción.

Aunque no siempre se publican los detalles técnicos completos de los motores internos de cada empresa, esta tendencia hacia búsquedas más inteligentes y semánticas es ampliamente documentada en estudios de IA aplicada al e-commerce.

Nuestro proyecto se inspira precisamente en este enfoque: combinamos embeddings semánticos derivados de descripciones enriquecidas del producto con un motor de recomendación, con el fin de acercarnos a esa relevancia avanzada en las búsquedas.

**Agentes Conversacionales (Chatbots) en Retail**

El uso de chatbots orientados al retail también ha sido explorado en la industria. Por ejemplo, la tienda online Zalando lanzó un chatbot integrado con un asistente virtual para aconsejar a usuarios en sus decisiones de compra.

Este tipo de bots permiten al usuario expresar sus preferencias en lenguaje natural (tipo de prenda, estilo, ocasión de uso, rango de precio, etc.), y recibir recomendaciones guiadas, lo que mejora la experiencia de compra.

En este proyecto, el chatbot implementa esa filosofía: permite consultas en lenguaje natural, aprovecha la información enriquecida del catálogo (texto + etiquetas visuales) y genera recomendaciones mediante el motor híbrido interno.

# 7. Marco teórico

## 7.1 Personalización y sistemas de recomendación en retail online

La personalización en retail online se basó en la capacidad de adaptar el catálogo y los mensajes mostrados a cada usuario en función de su historial, su contexto y sus preferencias implícitas y explícitas. Los sistemas de recomendación se clasificaron de manera general en:

- **Filtrado colaborativo**, que aprovechó patrones de comportamiento de usuarios similares para predecir qué productos podrían interesar a un usuario dado.
- **Recomendación basada en contenido**, que utilizó las características de los productos (categoría, descripción, atributos visuales) para encontrar elementos similares a aquellos con los que el usuario ya interactuó.
- **Modelos híbridos**, que combinaron ambos enfoques para mitigar sus limitaciones individuales, mejorando la cobertura y la calidad de las recomendaciones.

En el caso de ShopMate, nos centramos en un enfoque híbrido que aprovechó tanto las interacciones como la información de los productos.

## 7.2 Búsqueda semántica

La búsqueda tradicional basada en palabras clave resultó limitada ante consultas naturales, ambiguas o largas. La búsqueda semántica se apoyó en representaciones vectoriales (embeddings) de textos, de tal forma que consultas y documentos se proyectaron en un mismo espacio semántico. Esto permitió recuperar productos relevantes aunque la consulta no coincidiera palabra por palabra con la descripción del catálogo.

En nuestro proyecto, empleamos embeddings generados con servicios de IA generativa para representar descripciones de productos y consultas de usuario, de modo que pudiéramos calcular similitudes semánticas y ofrecer resultados mejor alineados con la intención de búsqueda.

## 7.3 Agentes conversacionales

Los agentes conversacionales evolucionaron desde chatbots basados en reglas hacia asistentes impulsados por modelos de lenguaje y servicios de IA.

En este proyecto, el chatbot se encargó de:

- Entender intenciones de compra y de exploración de catálogo.
- Formular y ejecutar consultas al motor de recomendación usando la búsqueda semántica.
- Presentar recomendaciones y resultados de forma dialogada, guiando al usuario en su proceso de decisión.

## 7.4 Servicios de Oracle Cloud relevantes

La solución se apoyó en diversos servicios de Oracle Cloud Infrastructure (OCI), entre ellos:

- **Autonomous Database** para el almacenamiento estructurado del catálogo y de interacciones.
- **Object Storage** para almacenar datasets y artefactos como modelos entrenados.
- **OCI Data Integration** para soportar procesos de ingesta y transformación de datos.
- **OCI Vision** para la extracción de características visuales a partir de imágenes de producto.

- **OCI Generative AI / NLP** para la generación de embeddings semánticos y el soporte a tareas de lenguaje natural.
- **Oracle Digital Assistant** para la construcción y despliegue del chatbot.

# 8. Exploración, Limpieza y Transformación de Datos

El proceso de preparación de datos inició con la descarga y carga de los tres conjuntos principales (artículos, clientes y transacciones). Una vez almacenados localmente, se realizó una revisión preliminar para conocer la estructura de cada dataset, identificar tipos de datos, valores faltantes, duplicados y características relevantes para su posterior uso en los modelos de recomendación y en los módulos de embeddings.

## 8.1 Exploración inicial

Durante la exploración se evaluó el número de filas y columnas, los tipos de datos presentes y la distribución de variables clave. Esto permitió reconocer columnas categóricas, variables numéricas y atributos con niveles significativos de valores nulos. Asimismo, se aplicaron estadísticas descriptivas y conteos de valores únicos para determinar el nivel de cardinalidad de cada columna y anticipar qué transformaciones serían necesarias.

En el caso de articles, se verificó la integridad de los identificadores, la distribución de grupos y tipos de productos, y el estado de la columna detail_desc, que presentaba valores faltantes. Para customers, se revisó la consistencia de los registros y se detectaron duplicados por customer_id. Finalmente, en transactions, se confirmó la correcta lectura de fechas y la variabilidad del precio.

## 8.2 Manejo del dataset y división en buckets

Para organizar eficientemente las imágenes asociadas a cada article_id, el dataset se dividió en tres buckets de Oracle Cloud, asignados según los dos primeros dígitos del identificador. Los artículos del rango 10–35 se almacenaron en el Bucket 1, los de 36–70 en el Bucket 2 y los de 71–95 en el Bucket 3. Con esta estructura se generó automáticamente la URL de cada imagen siguiendo un formato estándar, lo que permitió un acceso más ordenado y rápido a los recursos. Esta estrategia mejoró la escalabilidad, redujo la latencia de consulta y facilitó la integración visual dentro del sistema de recomendación.

## 8.3 Limpieza de datos

Con base en los hallazgos de la exploración, se realizó una limpieza estructurada:

- Se eliminaron duplicados en el dataset de clientes, asegurando conservar el registro más reciente por identificador.

- Se imputaron valores nulos mediante estrategias apropiadas a cada tipo de dato. Por ejemplo, detail_desc fue rellenado con un texto genérico, mientras que en clientes la edad se imputó con la mediana para mantener estabilidad estadística.

- Se imputaron categorías faltantes como 'UNKNOWN' en los campos de club_member_status y fashion_news_frequency, evitando afectar la consistencia del modelo.

- Se limpiaron columnas que ya no aportaban información o que serían reemplazadas por representaciones más útiles, como identificadores redundantes y códigos internos.

## 8.4 Transformación de datos

Una vez depurados los datasets, se aplicaron diversas transformaciones orientadas a preparar las tablas para el análisis, la generación de embeddings y el entrenamiento del sistema de recomendación.

**Transformaciones principales aplicadas:**

1. **Escalamiento de variables numéricas.** El precio en transactions y la edad en customers fueron estandarizados mediante StandardScaler para evitar que magnitudes elevadas dominen los algoritmos de similitud y recomendación.

2. **Conversión de fechas y generación de variables temporales.** La columna t_dat se transformó a tipo datetime, permitiendo descomponerla en año, mes, día del mes y día de la semana. Estas nuevas variables fortalecen el análisis de patrones de compra y estacionalidad.

3. **One-Hot Encoding para variables categóricas.** En articles, se codificaron múltiples columnas de baja cardinalidad tales como product_type_name, product_group_name, colour_group_name, entre otras. En customers y transactions, también se aplicó OHE

para estandarizar variables como club_member_status, fashion_news_frequency y sales_channel_id.

Tras generar las columnas binarias, se convirtieron explícitamente a valores '1' y '0' para asegurar compatibilidad con Oracle Autonomous Database.

4. **Ingeniería de características para RAG (Retrieval-Augmented Generation).** Se construyó una columna consolidada (description_vector_rag) que integra nombre, grupo, color y descripción detallada del producto. Esta estructura sirve como insumo semántico para la generación de embeddings y mejora la recuperación de información en búsquedas vectoriales.

5. **Eliminación de columnas irrelevantes o redundantes.** Se removieron identificadores internos, descripciones originales, códigos y campos que ya no aportaban valor al pipeline, conservando únicamente las columnas transformadas y útiles.

6. **Generación de URLs de imágenes.** Para cada producto se creó automáticamente una ruta hacia su imagen en el bucket correspondiente, utilizando reglas basadas en los prefijos del article_id. Esta integración enriquece el catálogo y facilita la vinculación visual del modelo.

## 8.5 Resultado final de la transformación

Después de completar el proceso de limpieza y transformación, los tres datasets quedaron estructurados, normalizados y optimizados para su integración en Oracle Autonomous Database. El resultado garantiza datos consistentes, escalables y adecuados para modelos basados en contenido, similitud vectorial y análisis temporal.

# 9. Visión de la Arquitectura

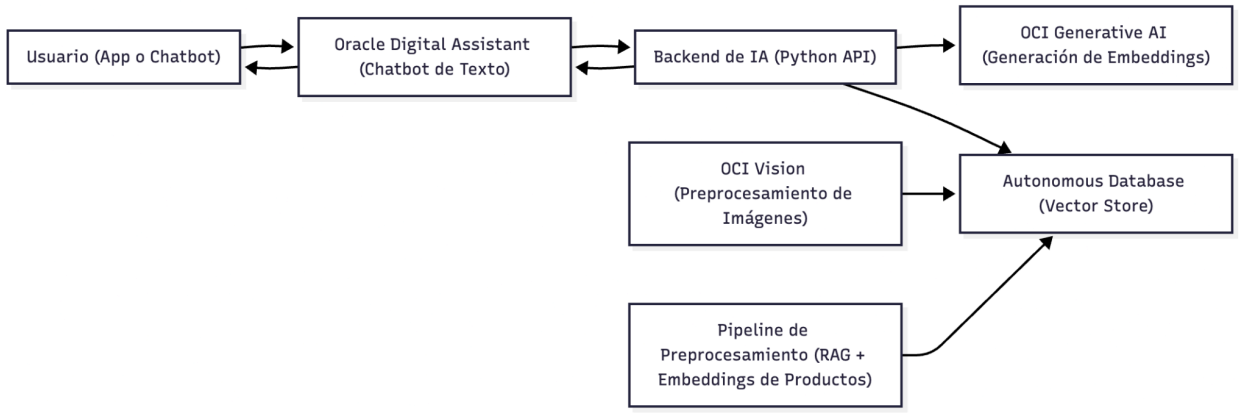


Figura 1. Arquitectura general de la solución ShopMate.

## 9.1 Conexión dentro de la arquitectura

En la Figura 1 se muestra cómo se conectan los componentes principales de ShopMate. El usuario interactúa desde la aplicación o el chatbot, y Oracle Digital Assistant envía cada consulta al Backend de IA. Este backend procesa la solicitud, genera embeddings cuando es necesario mediante OCI Generative AI y consulta la información almacenada en Autonomous Database, que funciona como vector store y repositorio del catálogo. Por otro lado, OCI Vision participa únicamente en el preprocesamiento, generando etiquetas visuales que, junto con el contenido enriquecido del pipeline, se almacenan también en la base de datos. Esta integración permite realizar búsquedas semánticas y recomendaciones personalizadas en tiempo real.

## 10.1 Modelo de Recomendación

El modelo de recomendación implementado en ShopMate se basa en un enfoque moderno centrado en representaciones vectoriales (embeddings). Este enfoque permite capturar relaciones semánticas profundas entre productos, consultas textuales e historial de usuario, habilitando búsquedas inteligentes y recomendaciones altamente personalizadas sin necesidad de entrenar un modelo supervisado.

El sistema se fundamenta en tres pilares tecnológicos:Enriquecimiento semántico del catálogo mediante Retrieval-Augmented Generation (RAG) y etiquetas visuales, generación de embeddings con OCI Generative AI y su almacenamiento en una Vector Store dentro de Autonomous Database, y un motor de búsqueda híbrida que combina intención del usuario y preferencias históricas.

### 10.1.1 Enriquecimiento Semántico del Catálogo (RAG + Vision)

Para mejorar la capacidad del modelo de capturar la semántica de cada producto, se construyó un proceso de RAG interno que fusiona información textual y visual.

1. **Extracción de etiquetas visuales (OCI Vision).** Cada imagen del catálogo fue procesada mediante OCI Vision para obtener descriptores visuales (categorías, colores dominantes, patrones, tipos de prenda).Estas etiquetas se almacenaron en las tablas ARTICLE_VISION_FEATURES y vision_labels_json.
2. **Fusión de información textual y visual** Las etiquetas generadas por Vision se integraron con: descripción original del producto y metadatos estructurados (categoría, grupo, color, departamento), formando un bloque descriptivo ampliado denominado ARTICLE_CONTENT_RAG, que sirve como entrada para los modelos de embeddings.

Este proceso garantiza que los vectores resultantes capturen no solo el texto original del catálogo, sino también características visuales y semánticas útiles para la búsqueda y recomendación.

### 10.1.2 Generación de Embeddings Semánticos

Los embeddings se generaron utilizando el modelo de OCI Generative AI cohere.embed-english-v3.0, un modelo avanzado especializado en semantic embedding.

Los embeddings que representan cada producto del catálogo fueron generados utilizando el modelo cohere.embed-english-v3.0 de OCI Generative AI, un modelo optimizado para tareas de representación semántica de alto nivel. Este modelo produce vectores de 1024 dimensiones, lo que permite capturar relaciones complejas entre atributos textuales y semánticos del contenido enriquecido de cada artículo. Para la inferencia se utilizó el modo de servicio bajo demanda (OnDemandServingMode), permitiendo procesar grandes volúmenes de descripciones sin necesidad de aprovisionar infraestructura adicional. Durante la generación de embeddings, el texto enriquecido fue procesado con la política de truncamiento “END”, la cual asegura que, en caso de superar el límite máximo permitido por el modelo, se recortara únicamente la parte final del texto manteniendo la información inicial más relevante. El proceso se ejecutó en lotes de aproximadamente 90 descripciones por solicitud, optimizando el uso de créditos y garantizando tiempos de respuesta consistentes. Los vectores resultantes fueron almacenados en Oracle Autonomous Database utilizando el tipo de dato VECTOR(1024) en formato float32, lo cual facilita la aplicación directa de funciones nativas como VECTOR_DISTANCE sin necesidad de transformaciones adicionales. Aunque los embeddings no se normalizan al momento de almacenarse, se normalizan dinámicamente durante los cálculos de similitud en el backend, garantizando comparaciones coherentes entre productos, usuarios y consultas en tiempo real.

### 10.1.3 Construcción de perfil de usuario.

El sistema genera un perfil vectorial para cada usuario a partir de su historial de compras. Este perfil resume los gustos y preferencias del cliente utilizando exclusivamente la información contenida en los embeddings de los productos adquiridos.

El proceso se desarrolla de la siguiente manera:

- Primero, se identifican los artículos que el usuario ha comprado con mayor frecuencia (Top 20 productos), utilizando la tabla INTERACTIONS_FOR_CF.

- Luego, para cada uno de esos artículos se recupera su embedding correspondiente desde el Vector Store de Autonomous Database.

- Con estos vectores, el sistema construye una representación del usuario mediante un promedio ponderado, donde los productos más recurrentes tienen mayor influencia en la caracterización final.

- Finalmente, este vector se normaliza para asegurar comparaciones consistentes durante la recomendación.

El perfil resultante refleja aspectos relevantes del estilo del usuario, tales como colores preferidos, categorías más compradas, características visuales predominantes y patrones semánticos que se repiten en su historial

Este perfil es fundamental para la etapa de recomendación personalizada, ya que permite evaluar qué tan bien se alinea cada producto con los gustos del usuario.

### 10.1.4 Búsqueda Semántica por Texto

La búsqueda semántica permite interpretar consultas expresadas en lenguaje natural, aun cuando el usuario no utilice términos exactos del catálogo. Para ello:

1. La consulta es convertida a un embedding utilizando el modelo cohere.embed-english-v3.0, con los mismos hiperparámetros aplicados a los productos (1024 dimensiones, truncamiento al final del texto y modo de servicio bajo demanda).

2. Este embedding se compara con los vectores almacenados en la tabla ARTICLE_EMBEDDINGS mediante las funciones nativas de similitud de Autonomous Database.
3. El sistema retorna los productos que presentan mayor cercanía semántica con la intención expresada por el usuario.

Este procedimiento permite manejar sinónimos, consultas vagas o descripciones ambiguas con un nivel de precisión superior al de una búsqueda por palabras clave.

### 10.1.5 Recomendación Personalizada

La recomendación personalizada compara directamente el embedding de cada producto, y el perfil vectorial del usuario.

La similitud entre ambos determina qué tan alineado está un producto con los gustos del cliente. Los artículos que comparten características semánticas con el perfil (como color, categoría o estilo) obtienen puntuaciones más altas y se priorizan en el resultado final.

Este enfoque permite que las recomendaciones evolucionen dinámicamente conforme el usuario interactúa y realiza nuevas compras.

### 10.1.6 Motor de Búsqueda Híbrida (Hybrid Search)

El sistema implementa un motor de búsqueda híbrida que combina dos señales principales para producir resultados más relevantes para el usuario. La primera proviene de la búsqueda semántica, donde la consulta del usuario es convertida en un embedding y comparada con los embeddings de los productos del catálogo para identificar aquellos que se alinean mejor con la intención expresada en el texto. La segunda señal corresponde a la preferencia histórica del usuario, representada mediante su perfil vectorial construido a partir de los productos que ha adquirido previamente.

Durante la generación del resultado final, ambas señales se integran para priorizar artículos que no solo coincidan con lo que el usuario está solicitando en ese momento, sino que también sean compatibles con su estilo general de compra. Esta combinación permite que el sistema responda de manera contextual a cada consulta, mientras mantiene coherencia con los patrones de comportamiento del usuario. La ponderación entre ambas señales es configurable y puede ajustarse según las necesidades del sistema o los experimentos internos de relevancia.

## 10.2 Chatbot Basado en Texto (Oracle Digital Assistant)

El chatbot basado en texto constituye la interfaz principal de interacción entre el usuario y el sistema, permitiendo realizar búsquedas y solicitar recomendaciones mediante lenguaje natural. Su función es recibir las consultas del usuario, estructurarlas a través de intents definidos en Oracle Digital Assistant y enviarlas al backend del sistema, donde se ejecutan los procesos de búsqueda semántica y recomendación personalizada.

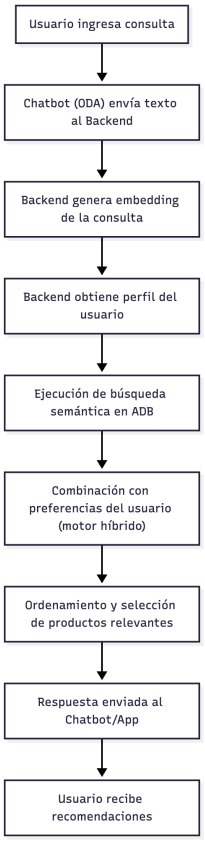

Figura 2. Flujo del motor de recomendación conectado con el chatbot dentro de ShopMate.

### 10.2.1 Procesamiento de Consultas en Lenguaje Natural

Cuando el usuario introduce una consulta escrita, como "¿Me puedes recomendar un vestido elegante para una boda?", el chatbot identifica el intent correspondiente y envía el texto al backend. La interpretación contextual —incluyendo estilo, categoría u ocasión— se lleva a cabo en el backend mediante la generación de embeddings con el modelo cohere.embed-english-v3.0, utilizando los mismos hiperparámetros aplicados a los productos (dimensionalidad de 1024, truncamiento por final de secuencia y servicio bajo demanda).

Este diseño permite que el chatbot mantenga una interacción natural sin requerir que el usuario utilice términos exactos del catálogo.

### 10.2.2 Uso de Información Visual y Textual Preprocesada (RAG + Vision)

Aunque la interacción del usuario con el chatbot es exclusivamente textual, el sistema se apoya en información enriquecida del catálogo generada durante el preprocesamiento. Las imágenes de los productos fueron analizadas mediante Oracle Vision para obtener etiquetas descriptivas, almacenadas en ARTICLE_VISION_FEATURES. Estas etiquetas se integraron con las descripciones textuales originales para construir el contenido enriquecido en ARTICLE_CONTENT_RAG, el cual posteriormente fue vectorizado en el proceso de generación de embeddings.

Gracias a esta capa de enriquecimiento, las búsquedas realizadas desde el chatbot pueden recuperar productos que coinciden semánticamente con la intención del usuario, aun cuando este no mencione características explícitas del artículo.

### 10.2.3 Generación de Respuestas y Recomendaciones

El chatbot devuelve dos tipos principales de resultados, dependiendo del intent activado:

- **Búsqueda semántica:** basada en la similitud entre el embedding de la consulta y los embeddings de los productos almacenados en ARTICLE_EMBEDDINGS, utilizando las

funciones nativas de Vector Store de Autonomous Database.

- **Recomendación personalizada:** generada por el motor híbrido que combina la relevancia semántica de la consulta con la similitud entre los productos y el perfil vectorial del usuario. La personalización se obtiene exclusivamente mediante operaciones vectoriales.

El chatbot presenta los resultados en forma de lista, acompañada de la información descriptiva necesaria para su integración con la aplicación móvil.

## 10.3 Aplicación Móvil.

La aplicación móvil de ShopMate funciona como la interfaz principal de interacción con el usuario final. Su objetivo es ofrecer una experiencia de compra personalizada, sencilla e intuitiva, apoyándose en el modelo de recomendación híbrido y en los servicios de backend desarrollados.

### 10.3.1 Inicio de sesión por identificador de cliente

Al abrir la aplicación, se presenta una pantalla de bienvenida en la que el usuario ingresa su ID de cliente para iniciar sesión.

Este identificador permite vincular la sesión con la información almacenada en la base de datos (historial de compra, preferencias y perfil), de manera que, desde el primer momento, la aplicación pueda mostrar recomendaciones adaptadas a cada persona.

### 10.3.2 Página de inicio con recomendaciones personalizadas

Una vez autenticado, el usuario es dirigido a la pantalla principal, donde se muestra una selección de productos recomendados para ese cliente, generados a partir del modelo de recomendación híbrido.

De esta forma, la aplicación ofrece un escaparate inicial ya filtrado según los gustos y el comportamiento de compra del cliente.

### 10.3.3 Detalle de producto y agregado al carrito

Al seleccionar un producto desde la pantalla principal o desde el catálogo, la aplicación muestra una vista de detalle que incluye: nombre del artículo, grupo o categoría, color, tipo de prenda y descripción o detalle del producto.

En esta misma pantalla se encuentra el botón “Agregar al carrito”, que permite incorporar el producto a la bolsa de compras del usuario.

Posteriormente, desde la sección “Mi bolsa”, el usuario puede visualizar los productos añadidos y gestionar su contenido (por ejemplo, vaciar la bolsa).

### 10.3.4 Catálogo de productos

Además de las recomendaciones iniciales, la aplicación cuenta con una **pantalla de catálogo**, donde los productos se organizan por categorías (por ejemplo: Mujer, Hombre, Niños y niñas, Bebés, Zapatos, Belleza, etc.).

Esta sección permite al usuario navegar por las distintas categorías de forma estructurada y explorar productos más allá de las recomendaciones personalizadas.

De este modo, la aplicación combina exploración libre del catálogo con sugerencias inteligentes.

### 10.3.5 Chat de ayuda con recomendaciones

La aplicación incluye un módulo de chat de ayuda, que simula la interacción con un asistente virtual. En esta sección, el usuario puede recibir recomendaciones basadas en su estilo y preferencias, que se muestran como tarjetas de productos dentro de la conversacióny escribir consultas en texto sobre el tipo de producto que está buscando (por ejemplo: “quiero un vestido amarillo para verano”) y obtener sugerencias relacionadas.

El chat se apoya en el backend de inteligencia artificial para traducir las consultas en lenguaje natural a búsquedas sobre el catálogo y combinar dichas búsquedas con el historial del cliente para ofrecer recomendaciones más personalizadas.

# 11. Métricas de evaluación y resultados

## 11.1 Modelo semántico de embeddings

Para evaluar el desempeño del modelo de búsqueda semántica, se ejecutaron 50 consultas independientes relacionadas con prendas de vestir y se analizaron las similitudes coseno entre los vectores de consulta y los artículos recuperados. El modelo alcanzó un promedio global de similitud de 0.5341, con un mínimo de 0.4270, un máximo de 0.6773 y una desviación estándar de 0.0507, indicando estabilidad en los resultados. Adicionalmente, se midió el rendimiento por ranking, obteniendo un Top-1 promedio de 0.5434, Top-3 de 0.5376 y Top-5 de 0.5341, lo que confirma consistencia en las primeras posiciones. Finalmente, la tasa de fallos fue del 0.00% bajo un umbral de similitud baja (<0.30), evidenciando que todas las consultas retornaron resultados relevantes según la métrica utilizada.

## 11.2 Modelo de Gustos

Para evaluar el desempeño de nuestro modelo de recomendación, utilizamos varias métricas estándar del sector. Estas nos permitieron medir la efectividad del modelo en cuanto a la relevancia de los artículos recomendados, y el grado de personalización ofrecido. A continuación se detallan las métricas aplicadas y los resultados obtenidos en nuestro proyecto:

**1. Similitud de Coseno Promedio**

Para evaluar la coherencia semántica entre el perfil del usuario y los artículos recomendados, se calculó la similitud de coseno promedio del Top 5 para una muestra de 100 usuarios seleccionados aleatoriamente. Esta métrica permite determinar qué tan alineados están los vectores de los artículos recomendados con el vector de gustos del usuario. En esta prueba, el modelo obtuvo un **promedio global de similitud de 87.75%**, acompañado de una **desviación estándar muy baja**, lo que indica que las recomendaciones mantienen una alta consistencia semántica y reflejan con precisión los patrones de preferencia del usuario.

**2. Recall@K**

El Recall@K se utilizó para medir la capacidad del modelo para recuperar artículos relevantes dentro de las primeras K recomendaciones, comparando su desempeño con las estrategias de popularidad y aleatoriedad como baselines. Los resultados obtenidos son los siguientes:

**Para K = 20:**

- Modelo embeddings: Recall@20 = 0.0533
- Popularidad: Recall@20 = 0.0142
- Random: Recall@20 = 0.0000

**Para K = 5:**

- Modelo embeddings: Recall@5 = 0.0253
- Popularidad: Recall@5 = 0.0067
- Random: Recall@5 = 0.0000

Estos resultados muestran que el modelo de embeddings supera significativamente a los baselines de popularidad y aleatoriedad en ambas métricas (Recall@5 y Recall@20), lo que evidencia que el modelo es más efectivo en recuperar artículos relevantes dentro de las primeras recomendaciones.

## 3. NDCG@K (Normalized Discounted Cumulative Gain)

El NDCG@K evalúa no solo si el modelo recupera artículos relevantes, sino también en qué posición los coloca dentro de las recomendaciones. Los modelos con un NDCG alto organizan eficientemente su top-K, priorizando los artículos más útiles para los usuarios.

En nuestro proyecto, la Desviación estándar entre usuarios fue de 2.86%, lo que significa que el modelo organiza de manera eficiente las recomendaciones, con poca variabilidad entre los usuarios. Este bajo nivel de desviación indica que los artículos relevantes aparecen en las

primeras posiciones de manera consistente, mejorando la experiencia del usuario al ofrecerle opciones útiles.

**4. Coverage@K**

El Coverage@K mide el porcentaje del catálogo total que aparece al menos una vez dentro de las recomendaciones generadas y permite evaluar la diversidad del modelo y su capacidad para explorar distintas secciones del catálogo, en lugar de concentrarse únicamente en los artículos más populares. En la evaluación realizada, para el Top 5 se generaron 500 recomendaciones (100 usuarios × 5 artículos cada uno), de las cuales el modelo produjo 364 artículos únicos, lo que corresponde a un 72.8% de coverage respecto a la prueba, y un Coverage@5 del 0.34% respecto al catálogo completo de más de 105 mil artículos. Esto indica que, aunque el alcance sobre el catálogo total es limitado, el modelo muestra una alta diversidad dentro del conjunto reducido de recomendaciones emitidas.

Al ampliar el análisis al Top 20 (2,000 recomendaciones totales), el número de artículos distintos recomendados aumentó a 1,228, lo que representa un 61.4% de coverage respecto a la prueba, y un Coverage@20 del 1.16% sobre el catálogo total. Este aumento significativo refleja que el modelo diversifica de manera más amplia cuando se le permite un conjunto mayor de recomendaciones, logrando explorar una porción más extensa del inventario y mejorando su alcance efectivo dentro del sistema.

## 11.3 Métricas cualitativas

Además de las métricas cuantitativas, se realizaron evaluaciones manuales de calidad para validar el comportamiento del modelo desde una perspectiva cualitativa. En la primera prueba, se seleccionaron varios usuarios y se revisó su historial real de compras; posteriormente, se inspeccionaron visualmente las imágenes de los productos recomendados por el modelo para verificar si existía similitud estética o de categoría con los artículos que el usuario ya había adquirido. En una segunda prueba, orientada a evaluar la coherencia semántica del modelo, se vectorizaron consultas textuales representativas (por ejemplo, búsquedas de ropa con descripciones específicas) y se

revisaron manualmente las imágenes de los artículos presentes en el top de resultados. Este análisis permitió confirmar si las recomendaciones concordaban con el significado y la intención de las búsquedas, proporcionando evidencia cualitativa de la capacidad del modelo para capturar relaciones semánticas y visuales relevantes.

## 12. Desafíos Técnicos

1. **Elección del Modelo Inicial (SVM)**

   En un principio, decidimos implementar un modelo basado en Support Vector Machine (SVM) para la clasificación de las recomendaciones. Sin embargo, nos encontramos con un problema técnico significativo. En nuestros datos, tanto los IDs de los usuarios como los IDs de los productos eran consecutivos (por ejemplo, de 1 a N), lo que generaba una falta de separabilidad en el espacio de características. Esto es porque el SVM, aunque eficaz en tareas de clasificación, requiere una clara separación entre las clases, y en este caso, la estructura de los datos no permitía una distinción clara entre las diferentes clases (usuarios y productos). Esto resultó en un modelo ineficiente que no producía recomendaciones precisas, especialmente cuando se trataba de usuarios y productos nuevos.

   Debido a esta falta de separabilidad y la limitación del SVM para manejar nuestros datos de manera efectiva, decidimos cambiar de enfoque y optar por la generación de embeddings semánticos utilizando OCI Generative AI. Los embeddings permiten representar a los productos y usuarios en un espacio semántico de alta dimensionalidad, donde las relaciones de similitud entre ellos son mucho más evidentes. Este enfoque permitió una mejor comprensión del comportamiento del usuario y las características de los productos, mejorando significativamente la precisión de las recomendaciones.

2. **Limitaciones en la API de OCI Generative AI**

   Durante el proceso de integración con OCI Generative AI, nos enfrentamos a varios

problemas relacionados con la gestión de créditos y limitaciones de uso de la API. En un principio, al generar embeddings semánticos para productos y usuarios, el equipo encontró que los créditos de OCI se agotaron a mitad de camino, lo que afectó la capacidad de algunos miembros del equipo para trabajar de manera continua. Esto generó interrupciones en el proceso de entrenamiento y obligó a un cambio en la estrategia de trabajo.

Ante esta situación, el equipo tuvo que dividirse para trabajar dentro de las cuentas que aún tenían créditos disponibles, lo que resultó en un trabajo más fragmentado. A pesar de esta dificultad, se logró mantener el flujo de trabajo, aunque el proceso se alargó y requirió mayor coordinación entre los miembros del equipo para asegurarse de que todos los datos y modelos estuvieran sincronizados.

## 13. Lecciones Aprendidas

1. **La Elección de los Embeddings Semánticos.** La experiencia con el modelo SVM dejó claro que este tipo de modelo no es adecuado para manejar datos con falta de separabilidad, especialmente cuando los IDs de productos y usuarios, ya que son consecutivos. Esto nos enseñó que los embeddings semánticos, que representan productos y usuarios en un espacio de características de alta dimensionalidad, son una opción mucho más adecuada para este tipo de datos. La transición hacia el uso de OCI Generative AI para generar estos embeddings fue clave para mejorar la precisión y personalización de las recomendaciones.

2. **Gestión de Recursos en OCI.** La limitación de los créditos en OCI fue un desafío inesperado, pero nos enseñó la importancia de planificar y gestionar los recursos de manera más eficiente en proyectos que dependen de servicios en la nube. En futuros proyectos, es crucial asegurarse de que todos los miembros del equipo tengan acceso adecuado a los recursos antes de comenzar la generación de embeddings y otros procesos dependientes de la nube.

3. **Escalabilidad y Optimización de Modelos.** Finalmente, una de las lecciones más importantes fue la necesidad de optimizar los modelos y los procesos de entrenamiento de embeddings. La generación de embeddings para grandes volúmenes de datos requiere una infraestructura adecuada para asegurar la escalabilidad del sistema. Aunque OCI Object Storage permitió almacenar los embeddings de manera eficiente, el proceso de entrenamiento y almacenamiento debe ser optimizado para poder manejar grandes volúmenes de datos sin comprometer el rendimiento del sistema.

## 14. Conclusiones

El proyecto ShopMate nos permitió diseñar e implementar una solución integral de compras personalizadas que combinó técnicas avanzadas de inteligencia artificial con servicios de nube y una experiencia de usuario centrada en la conversación. Al iniciar el desarrollo, el desafío principal era la calidad de la información, pero logramos alcanzar un 100% del dataset limpio y estructurado, eliminando inconsistencias en los perfiles de usuario y normalizando los datos transaccionales. Esta base sólida fue el cimiento indispensable para después obtener un modelo híbrido que cumpliera con la compleja tarea de entender tanto el lenguaje humano como los patrones matemáticos de consumo, logrando unir exitosamente la búsqueda semántica con las preferencias personales en una sola recomendación coherente.

Para el usuario final, esta tecnología transforma la compra en línea de una tarea de búsqueda exhaustiva a una conversación natural. El sistema elimina la fricción de navegar por múltiples filtros manuales, ya que el asistente entiende descripciones vagas como "algo elegante para una cena" y las cruza con lo que sabe que al usuario le gusta. Esto genera una sensación de atención personalizada, similar a la de un personal shopper en una tienda física, reduciendo el tiempo de búsqueda y aumentando la satisfacción con los productos encontrados.

Desde la perspectiva empresarial, ShopMate posiciona a la compañía en la vanguardia del retail moderno. La capacidad de ofrecer recomendaciones precisas busca incrementar directamente la tasa de conversión y el valor promedio del ticket de compra. Además, al automatizar el entendimiento del catálogo mediante IA (Vision y Generative AI), se pretende que el negocio reduzca costos operativos manuales y obtenga una herramienta escalable que aprenda y mejore con cada interacción, convirtiendo los datos históricos en un activo estratégico para la retención de clientes.

La validación exitosa de esta arquitectura demuestra que es técnicamente viable unificar la gestión de datos masivos con la inteligencia artificial generativa, brindándonos una base sólida para explorar versiones más avanzadas de la solución en entornos con datos reales y restricciones de negocio más exigentes, abriendo la puerta a futuras implementaciones productivas de alto impacto.

# 15. Trabajo futuro

A partir de los resultados y limitaciones detectadas, se identificaron varias líneas de trabajo futuro:

1. Ampliar y enriquecer los datos de interacción con usuarios reales para fortalecer la componente colaborativa y permitir técnicas más sofisticadas (por ejemplo, modelos secuenciales o contextuales).

2. Explorar modelos de recomendación más avanzados, como redes neuronales profundas para embeddings conjuntos de usuario y producto, o arquitecturas específicas para recomendación secuencial.

3. Mejorar la personalización en tiempo real, incorporando señales de sesión (tiempo de permanencia, scroll, productos observados) en la lógica de recomendación.

**Referencias**

Admin. (2023, July 17). Sistemas de recomendación: personalizando la experiencia del usuario en la era digital. MFAIA. https://mfaia.dia.fi.upm.es/sistemas-de-recomendacion-personalizando-la-experiencia-del-usuario-en-la-era-digital/

Miu. (2025, January 9). Sistemas de recomendación en entornos Big Data - MIUniversity. MIU City University Miami. https://miuniversity.edu/es/blog/sistemas-de-recomendacion-en-entornos-big-data

Modaes. (2017, 5 octubre). Zalando se alía con Google y lanza un asistente virtual para recomendar regalos. Modaes. https://www.modaes.com/empresas/zalando-se-alia-con-google-y-lanza-un-chatbot-para-recomendar-regalos

Caballar, R., & Stryker, C. (2025, 26 noviembre). Motor de recomendaciones. IBM. https://www.ibm.com/mx-es/think/topics/recommendation-engine

Martínez, S. (2025, 14 octubre). Algoritmos de recomendación: tipos, funcionamiento y aplicaciones en empresas. Inesdi. https://www.inesdi.com/blog/algoritmos-recomendacion-cp/

Sánchez Molano, B. (2019). Sistema no supervisado para la recomendación de contenidos educativos basado en un sistema híbrido [Trabajo Fin de Máster, Universidad Internacional de La Rioja]. Repositorio Institucional UNIR.

https://reunir.unir.net/bitstream/handle/123456789/9454/S%C3%A1nchez%20Molano%2C%20Boris.pdf?isAllowed=y&sequence=1