Maestría en

Inteligencia Artificial Aplicada

Trabajo previo a la obtención de título de Magister en

AUTOR/ES:

Paúl Alexander Cortés Barahona

Wilson Gustavo Gamboa Acosta

Allan Alexis Orellana Valenzuela

TUTOR/ES:

Navas Castellaños Andrés Roberto

Desarrollo de un modelo de aprendizaje profundo basado en la arquitectura Two Towers para la recuperación de candidatos y recomendación de productos complementarios en una empresa distribuidora del sector ferretero ecuatoriano

Quito - Ecuador

07 – 2026

REINVENTEMOS EL FUTURO

Certificación de autoría

Nosotros, **Paúl Alexander Cortés Barahona, Wilson Gustavo Gamboa Acosta, Allan Alexis Orellana Valenzuela,** declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido presentado anteriormente para ningún grado o calificación profesional y que se ha consultado la bibliografía detallada.

Cedemos nuestros derechos de propiedad intelectual a la Universidad Internacional del Ecuador (UIDE), para que sea publicado y divulgado en internet, según lo establecido en la Ley de Propiedad Intelectual, su reglamento y demás disposiciones legales.

Firma
Paúl Alexander Cortés Barahona

Firma
Wilson Gustavo Gamboa Acosta

Firma
Allan Alexis Orellana Valenzuela

Autorización de Derechos de Propiedad Intelectual

Nosotros, **Paúl Alexander Cortés Barahona, Wilson Gustavo Gamboa Acosta, Allan Alexis Orellana Valenzuela,** en calidad de autores del trabajo de investigación titulado **Desarrollo de un modelo de aprendizaje profundo basado en la arquitectura Two Towers para la recuperación de candidatos y recomendación de productos complementarios en una empresa distribuidora del sector ferretero ecuatoriano** autorizamos a la Universidad Internacional del Ecuador (UIDE) para hacer uso de todos los contenidos que nos pertenecen o de parte de los que contiene esta obra, con fines estrictamente académicos o de investigación. Los derechos que como autores nos corresponden, lo establecido en los artículos 5, 6, 8, 19 y demás pertinentes de la Ley de Propiedad Intelectual y su Reglamento en Ecuador.

D. M. Quito, Julio 2026

Firma
Paúl Alexander Cortés Barahona

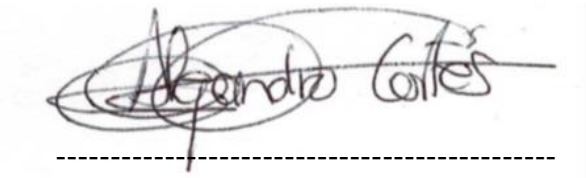
Firma
Wilson Gustavo Gamboa Acosta

Firma
Allan Alexis Orellana Valenzuela

Aprobación de dirección y coordinación del programa

Nosotros **Karla Estefania Mora Cajas y Alejandro Cortés López,** declaramos que: **Paúl Alexander Cortés Barahona, Wilson Gustavo Gamboa Acosta, Allan Alexis Orellana Valenzuela,** son los autores exclusivos de la presente investigación y que ésta es original, auténtica y personal de ellos.

Karla Estefania Mora Cajas
Director/a de la
Maestría en Inteligencia Artificial Aplicada



Alejandro Cortés López
Coordinador/a de la
Maestría en Inteligencia Artificial Aplicada

DEDICATORIA

A nuestras familias, cuyo apoyo incondicional y paciencia hicieron posible la culminación de esta etapa académica. A quienes, con su ejemplo de esfuerzo y perseverancia, nos enseñaron que el conocimiento es la herramienta más poderosa para transformar la realidad.

A los profesionales del sector ferretero ecuatoriano, cuyo trabajo cotidiano sostiene la construcción del país y motivó la pregunta de investigación que dio origen a este trabajo.

Los autores

AGRADECIMIENTOS

Expresamos nuestro más sincero agradecimiento a la Universidad Internacional del Ecuador, por los conocimientos y la formación rigurosa impartidos a lo largo del programa.

De manera especial, agradecemos a nuestro Director de Trabajo de Titulación por su guía metodológica, su disposición permanente y sus observaciones críticas, que elevaron la calidad técnica de esta investigación.

Agradecemos también a la empresa distribuidora del sector ferretero que, en el marco de un acuerdo de confidencialidad, facilitó el acceso a su base de datos transaccional anonimizada, sin la cual este estudio no habría sido posible.

Los autores

RESUMEN

En la distribución ferretera B2B, los clientes no identifican con facilidad los complementos técnicos de sus compras en catálogos de decenas de miles de referencias. El objetivo de este estudio fue establecer en qué medida una arquitectura Two Towers mejora la recuperación de productos complementarios frente a las líneas base de popularidad y coocurrencia. Esta combinación de arquitectura, tipo de comprador y dominio de producto no fue identificada en los estudios incluidos en la revisión sistemática. La investigación adopta un enfoque cuantitativo, de alcance explicativo y diseño cuasiexperimental, con medidas repetidas sobre datos secundarios longitudinales y retrospectivos provenientes de una única fuente. Se utilizaron registros anonimizados de una empresa distribuidora, correspondientes a 5.876.800 transacciones, 14.935 clientes y 20.683 productos, de los cuales 13.840 registraron al menos una interacción. La matriz presentó una esparsidad del 98,78 %. Para evitar fuga de información se aplicó una partición cronológica. El modelo representa clientes y productos en vectores de 128 dimensiones e incorpora negativos in-batch, corrección LogQ y normalización para mantener consistencia entre el entrenamiento y la inferencia. Two Towers obtuvo un Recall@100 de 33,05 %, frente al 21,25 % de la línea base de popularidad y al 24,05 % de coocurrencia, con intervalos bootstrap sin solapamiento. En el top-10, en cambio, las diferencias no fueron concluyentes. La incorporación de LogQ permitió triplicar el Recall@100, mientras que el modelo mantuvo un desempeño de 27,20 % en clientes nuevos. En la prueba de ablación, retirar el producto semilla produjo una variación de 0,65 puntos en Recall@100, valor que se mantuvo dentro del margen de muestreo. **Palabras clave: sistemas de recomendación, recuperación de información, aprendizaje profundo, comercio electrónico entre empresas, distribución ferretera, arquitectura Two Towers.**

ABSTRACT

The aim was to determine how much a Two-Tower architecture improves the retrieval of complements over the popularity and item-to-item co-occurrence baselines, a combination of architecture, buyer, and product domain the systematic review found undocumented.

The study is quantitative, explanatory, and quasi-experimental, with repeated measures over secondary, single-source, retrospective longitudinal data. Anonymized records of a distributor were used: 5,876,800 transactions, 14,935 customers, and 20,683 products (13,840 with at least one interaction; 98.78% sparsity). Under a chronological split against data leakage, the model projects customer and product into a shared 128-dimensional space, with in-batch negatives, LogQ sampling-bias correction, and normalization aligning training with serving.

Two Towers reached a Recall@100 of 33.05%, against 21.25% and 24.05% for the baselines, with non-overlapping bootstrap intervals; differences in the top-10 were inconclusive. It sustained 27.20% for new customers and 0.70% for items with no purchase history, where co-occurrence drops to zero; the LogQ correction tripled Recall@100. Masking the seed product shifted Recall@100 by 0.65 points, within the sampling margin: the dominant signal is customer context, not the queried item.

Keywords: recommender systems, information retrieval, deep learning, business-to-business electronic commerce, hardware distribution, two-tower architecture.

TABLA DE CONTENIDOS

LISTA DE TABLAS

LISTA DE FIGURAS

LISTA DE ABREVIATURAS Y ACRÓNIMOS

Abreviaturas, acrónimos y símbolos empleados

Sigla	Significado
ANN	Approximate Nearest Neighbor (vecino más cercano aproximado)
API	Application Programming Interface
B2B	Business to Business
CF	Collaborative Filtering (filtrado colaborativo)
CTR	Click-Through Rate (tasa de clic)
DCN	Deep & Cross Network
DSSM	Deep Structured Semantic Model
EDA	Exploratory Data Analysis (análisis exploratorio de datos)
ERP	Enterprise Resource Planning
FAISS	Facebook AI Similarity Search
GELU / ReLU	Funciones de activación neuronal
GMV	Gross Merchandise Value
HNSW	Hierarchical Navigable Small World
IVF	Inverted File Index
LogQ	Corrección del sesgo de muestreo (log de la probabilidad)
LOPDP	Ley Orgánica de Protección de Datos Personales (Ecuador)
MLP	Multi-Layer Perceptron (perceptrón multicapa)
MRR	Mean Reciprocal Rank
NCF	Neural Collaborative Filtering
NDCG	Normalized Discounted Cumulative Gain
RUC	Registro Único de Contribuyentes
SKU	Stock Keeping Unit (referencia de producto)
SVD	Singular Value Decomposition (descomposición en valores singulares)
TFRS	TensorFlow Recommenders
t-SNE / UMAP	Técnicas de reducción de dimensionalidad
D	Dimensión del espacio de embeddings compartido (D = 128)

Nota. Las siglas se definen en su primera aparición en el texto.

CAPÍTULO 1: INTRODUCCIÓN

1.1 Definición del proyecto

Los sistemas de recomendación son una pieza clave de la infraestructura digital del comercio. En esencia, filtran volúmenes enormes de información para quedarse con los ítems que tienen más chances de ajustarse a lo que un usuario necesita (Li et al., 2023). En el comercio electrónico y en la distribución, estos sistemas se han vuelto motores directos de ingreso: aumentan el valor del ticket promedio, aceleran la rotación del inventario y mejoran la satisfacción del cliente porque son capaces de adelantarse a necesidades que ni siquiera se han hecho explícitas. A lo largo de los últimos años, los sistemas de recomendación han experimentado una evolución importante. Los primeros enfoques, basados principalmente en reglas heurísticas y técnicas tradicionales de filtrado colaborativo, han dado paso a modelos de aprendizaje profundo capaces de construir representaciones densas de usuarios y productos. Estas representaciones permiten identificar relaciones complejas y patrones no lineales que resultan difíciles de capturar mediante métodos convencionales (Zhang et al., 2019).

En aplicaciones de gran escala, el funcionamiento de un sistema de recomendación suele dividirse en tres etapas principales. La primera busca comprender el contexto y las características asociadas al usuario. Posteriormente, se realiza la recuperación de un conjunto reducido de productos potencialmente relevantes a partir de un catálogo de gran tamaño. Finalmente, estos candidatos pasan por una etapa de ordenamiento más detallada, cuyo propósito es determinar cuáles presentan una mayor probabilidad de resultar adecuados para cada usuario (Covington et al., 2016; Fu et al., 2020). Dentro de este proceso, la recuperación de candidatos adquiere especial importancia debido a la carga computacional que implica analizar catálogos conformados por miles o incluso millones de productos, manteniendo al mismo tiempo tiempos de respuesta reducidos.

A partir de esta problemática, el presente Trabajo de Fin de Máster concentra su análisis en la etapa de recuperación de candidatos dentro de un contexto que ha recibido menor atención en la literatura especializada: la distribución ferretera B2B en Ecuador. Este entorno presenta características particulares frente a sectores como el comercio electrónico de consumo masivo o las plataformas de entretenimiento. En el sector ferretero, las decisiones de compra suelen estar relacionadas con proyectos específicos, se producen con menor frecuencia y pueden representar montos económicos elevados. Además, la relación entre productos responde principalmente a criterios técnicos, funcionales y constructivos, antes que a preferencias individuales del comprador (Meng et al., 2024).

Frente a estas características, se propone desarrollar, implementar y evaluar un sistema de recomendación basado en la arquitectura de aprendizaje profundo Two Towers, también conocida como arquitectura de codificadores duales. Para su entrenamiento se utilizan datos transaccionales reales, previamente anonimizados, provenientes de una empresa distribuidora representativa del sector ferretero ecuatoriano.

El documento se encuentra estructurado en cinco capítulos. El Capítulo 1 presenta el problema de investigación, su justificación, el alcance del estudio, los objetivos planteados y la hipótesis que orienta el desarrollo del trabajo. El Capítulo 2 presenta la revisión de literatura, con el estado del arte y el marco teórico. El Capítulo 3 detalla el desarrollo del trabajo: la metodología de investigación, el diseño del sistema y su implementación técnica. El Capítulo 4 presenta el análisis cuantitativo y cualitativo de resultados, incluyendo experimentos de diversidad, ablación, arranque en frío y simulación de servido. El Capítulo 5 expone las conclusiones y recomendaciones derivadas del estudio.

1.2 Caso de Estudio y Problema de Investigación

1.2.1 Contexto del caso de estudio

El proyecto se llevó a cabo en una distribuidora representativa del sector ferretero ecuatoriano que trabaja bajo un modelo de negocio B2B, con rutas de distribución terrestre que cubren varias ciudades del país. La empresa maneja un catálogo de más de veinte mil productos activos, que incluye desde insumos básicos de construcción como cemento, arena y agregados, hasta herramientas especializadas, materiales eléctricos, tuberías y accesorios de plomería. Su cartera de clientes está compuesta en su mayoría por ferreterías minoristas y contratistas independientes, cuyas compras giran en torno a las necesidades concretas de obras de construcción y mantenimiento.

La dimensión de los datos analizados permite evidenciar la relevancia operativa del caso de estudio. La base consolidada comprende 5.876.800 transacciones correspondientes a un periodo de 27 meses, entre enero de 2024 y marzo de 2026. En este conjunto se identifican 14.935 clientes únicos y un catálogo conformado por 20.683 productos. Sin embargo, pese a la cantidad de información disponible, actualmente la empresa no dispone de un mecanismo de recomendación que se adapte a las características y comportamiento de compra de cada cliente. Las sugerencias existentes responden principalmente a criterios generales, como la popularidad de determinados productos o las promociones comerciales disponibles, sin considerar el historial particular de los compradores ni las posibles relaciones técnicas y funcionales entre los artículos adquiridos.

Dentro del contexto ecuatoriano, el sector ferretero y de materiales de construcción cumple un papel importante debido a su estrecha relación con la actividad constructiva y con las diferentes actividades económicas que se desarrollan alrededor de ella. La distribución mayorista se apoya, en gran medida, en una estructura logística basada en rutas terrestres mediante las cuales se realizan visitas periódicas a ferreterías, distribuidores minoristas y contratistas. Esta dinámica presenta diferencias frente a los modelos de comercio electrónico ampliamente estudiados en

investigaciones internacionales, donde gran parte de la interacción entre compradores y vendedores ocurre a través de plataformas digitales.

En este tipo de distribución, la ubicación geográfica y la continuidad de la relación comercial tienen una influencia directa sobre el comportamiento de compra. Los clientes se encuentran asociados a determinadas rutas y zonas de atención, mientras que la frecuencia y composición de sus pedidos pueden variar de acuerdo con las obras, proyectos y necesidades que atienden en cada periodo. Como resultado, las transacciones almacenadas contienen información valiosa sobre los patrones de consumo y las relaciones existentes entre distintos productos. El aprovechamiento de estos datos, que hasta el momento no han sido utilizados en toda su capacidad analítica, representa una oportunidad para incorporar técnicas de aprendizaje automático que permitan mejorar la generación de recomendaciones y, al mismo tiempo, aportar valor a los procesos comerciales de la empresa.

1.2.2 Planteamiento del problema

El problema principal se presenta en la dificultad que tienen los clientes, entre ellos maestros de obra, propietarios de ferreterías y contratistas, para reconocer qué productos pueden complementar de manera adecuada una compra determinada. Por ejemplo, cuando un cliente adquiere cemento o tubería de PVC, es posible que también requiera accesorios o insumos adicionales, como cinta de teflón, codos para desagüe o pegamento para PVC. Sin embargo, debido a la amplitud del catálogo, estos productos no siempre son identificados durante el proceso de compra, aun cuando puedan ser necesarios para completar un trabajo o una instalación.

Esta situación puede ocasionar que los pedidos no cubran por completo las necesidades del cliente, lo que posteriormente puede traducirse en compras adicionales, interrupciones en la ejecución de una obra o retrasos en determinadas actividades. Para la empresa, este comportamiento también representa una oportunidad comercial que no está siendo aprovechada plenamente, especialmente en lo relacionado con la venta cruzada, ya que existen productos

técnicamente relacionados que podrían ser sugeridos de acuerdo con las características de cada pedido y con el comportamiento histórico de compra.

Desde el punto de vista técnico, abordar este problema implica considerar tres condiciones particulares del dominio. La primera es la altísima esparsidad de la matriz de interacción cliente-producto, que llega al 98,78 %, y que vuelve inaplicables los métodos clásicos de factorización. La segunda es la elevada heterogeneidad de las variables: identificadores de cliente, ciudades, rutas, marcas, familias comerciales, precios, geolocalización. La tercera es la necesidad de operar sobre un catálogo de más de veinte mil SKU con latencias que sean compatibles con la operación diaria, lo que deja fuera a los modelos de ranking completo con interacción temprana. Dicho esto, el problema de investigación se plantea como la necesidad de contar con un modelo de recuperación que aprenda relaciones de complementariedad técnica a partir de patrones de co-compra y que al mismo tiempo sea preciso, robusto frente a la esparsidad y computacionalmente viable.

1.2.3 Pregunta de investigación

¿En qué medida un modelo de aprendizaje profundo basado en la arquitectura Two Towers mejora la recuperación de productos complementarios, medida con Recall@K (Recall@100 como métrica primaria de contraste) y el MRR como métrica complementaria, frente a los métodos clásicos de popularidad global y co-ocurrencia ítem-ítem, dentro del contexto de una distribuidora ferretera ecuatoriana que maneja un catálogo de más de veinte mil productos y una matriz de interacción con una esparsidad superior al 98 %?

1.3 Justificación e Importancia

La relevancia del trabajo se apoya en tres dimensiones que están conectadas entre sí: la académica, la tecnológica y la empresarial.

En el plano académico, el estudio busca aportar evidencia en un contexto que ha sido poco abordado en la literatura revisada. El sustento de esta afirmación se encuentra en la revisión

sistemática presentada en la sección 2.1 y desarrollada con mayor detalle en el apéndice E. En este apartado se especifican las fuentes consultadas, las cinco cadenas de búsqueda empleadas, los criterios definidos para incluir o excluir estudios y el procedimiento utilizado para extraer y organizar la información relevante.

Los resultados obtenidos muestran que una parte importante de las investigaciones revisadas se orienta hacia escenarios de consumo masivo, comercio electrónico y plataformas digitales de contenido. Considerando el alcance y los criterios definidos para la revisión, no se encontraron trabajos que aplicaran de manera específica la arquitectura Two Towers al ámbito de la distribución ferretera B2B en Ecuador. En consecuencia, el aporte original de esta investigación no se fundamenta en afirmar que no existen estudios relacionados, sino en implementar y evaluar esta arquitectura dentro de un sector y una realidad geográfica que han sido poco abordados en la literatura disponible.

Desde la perspectiva tecnológica, la selección de la arquitectura Two Towers respondió a un proceso de comparación entre distintas alternativas. Los enfoques tradicionales, como el filtrado colaborativo y la factorización de matrices, presentan ventajas en términos de eficiencia y facilidad de implementación; sin embargo, tienen limitaciones para representar relaciones no lineales entre clientes y productos, especialmente cuando los datos presentan un alto nivel de dispersión. Además, su capacidad para incorporar información adicional sobre usuarios, productos o contexto suele ser más limitada.

Por otra parte, existen modelos de mayor complejidad, como DeepFM o Deep & Cross Network (DCN), que permiten representar interacciones más elaboradas entre las variables. Aunque estos modelos ofrecen una mayor capacidad expresiva, su aplicación directa en la etapa de

recuperación resulta menos conveniente cuando se trabaja con catálogos amplios, ya que sería necesario evaluar individualmente una gran cantidad de combinaciones entre clientes y productos.

En este escenario, la arquitectura Two Towers ofrece una alternativa adecuada para equilibrar capacidad de representación y eficiencia computacional. Su diseño permite generar por separado las representaciones vectoriales de clientes y productos, de manera que los embeddings correspondientes a los productos pueden calcularse previamente y almacenarse en una estructura de búsqueda. Posteriormente, mediante técnicas de búsqueda aproximada de vecinos más cercanos, conocidas como ANN, es posible recuperar con rapidez aquellos productos cuya representación sea más cercana a la del cliente consultado. Este enfoque permite reducir considerablemente el costo de búsqueda y alcanzar tiempos de respuesta apropiados para sistemas de recomendación que operan sobre catálogos de gran tamaño (Fu et al., 2020; Yi et al., 2019).

Desde el punto de vista empresarial, la incorporación de un sistema de recomendación personalizado puede generar beneficios en diferentes áreas de la operación comercial. Entre ellos se encuentra la posibilidad de incrementar el valor promedio de los pedidos mediante recomendaciones de productos técnicamente complementarios, mejorar la rotación de determinados artículos del inventario y aprovechar patrones de compra que actualmente permanecen ocultos dentro del histórico transaccional. De esta forma, el uso de los datos disponibles puede convertirse en un apoyo adicional para la toma de decisiones comerciales y para la generación de nuevas oportunidades de venta. Del lado de los clientes, el beneficio se traduce en ahorro de tiempo y en menos vueltas por compras que quedaron incompletas.

1.4 Alcance del Proyecto

El proyecto consiste en diseñar, entrenar y validar de forma offline un modelo Two Towers orientado a la fase de recuperación de candidatos, tomando como base los datos transaccionales históricos y anonimizados de la empresa. El alcance contempla:

- Diseñar, entrenar y validar en condiciones offline el modelo de recuperación sobre el catálogo completo de productos ferreteros.
- Utilizar los datos transaccionales históricos, los perfiles anonimizados de clientes y el catálogo completo de productos.
- Construir pares de co-compra analizando patrones de co-ocurrencia temporal con ventanas de 30 días, orientados a recomendar productos por complementariedad técnica.
- Evaluar el modelo mediante las métricas de recuperación Recall@K y de ranking Mean Reciprocal Rank (MRR).
- Comparar el modelo frente a un baseline de popularidad global y un modelo clásico de filtrado colaborativo ítem-ítem basado en co-ocurrencia de compras.
- Realizar experimentos complementarios de diversidad y cobertura, ablación de características, robustez ante arranque en frío y simulación de servido con indexación ANN.

El proyecto no cubre el despliegue en producción ni la integración con los sistemas operativos de la empresa, como el ERP o el CRM. Tampoco abarca el entrenamiento en línea (online learning), la evaluación con pruebas A/B sobre usuarios reales, la implementación de un ranking fino posterior a la recuperación, ni la inclusión de datos multimodales como imágenes o descripciones textuales extensas. El contexto geográfico se limita a las operaciones de la empresa en Ecuador, con una importante concentración en la región Sierra.

1.5 Objetivos

1.5.1 Objetivo general

Desarrollar y evaluar un modelo de aprendizaje profundo basado en la arquitectura Two Towers, orientado a la recuperación de candidatos y a la recomendación de productos complementarios en una distribuidora del sector ferretero ecuatoriano.

1.5.2 Objetivos específicos

1. Reunir, integrar y poner a punto el dataset transaccional de la empresa, organizando las variables de entrada para la torre de usuario (historial, ubicación, ruta) y la torre de ítem (categoría, marca, precio, peso), además de generar pares de entrenamiento positivos a partir de co-ocurrencias temporales en ventanas de 30 días.
2. Construir y poner en marcha las dos redes neuronales (torre de usuario y torre de ítem) con TensorFlow Recommenders, definiendo capas de embedding para las variables categóricas de alta cardinalidad, capas densas para las variables numéricas y una función de pérdida basada en entropía cruzada con muestreo negativo in-batch y corrección LogQ.
3. Entrenar y afinar el modelo mediante el ajuste de hiperparámetros, apoyándose en una validación temporal de reserva (partición cronológica entrenamiento-validación) para medir su capacidad de generalización.
4. Medir el rendimiento en la recomendación de productos complementarios con Recall@10, @50, @100 y MRR, adoptando el Recall@100 como métrica primaria del contraste contra los baselines de popularidad global y co-ocurrencia ítem-ítem.
5. Explorar de forma cualitativa los embeddings aprendidos usando técnicas de reducción de dimensionalidad (t-SNE, UMAP) y matrices de similitud, para comprobar que las representaciones reflejan relaciones semánticas coherentes con el conocimiento del dominio ferretero.

6. Documentar los resultados obtenidos, las principales limitaciones identificadas y las posibles líneas de mejora, con el propósito de contar con una base técnica que pueda ser utilizada en futuras investigaciones y como referencia para una eventual implementación del sistema en un entorno productivo.

1.6 Hipótesis

En la recuperación de candidatos sobre el catálogo de la distribuidora ferretera (más de 20.000 SKU y una matriz de interacción con esparsidad superior al 98 %), evaluada sobre los pares de co-compra del primer trimestre de 2026, la arquitectura Two Towers obtiene un Recall@100 superior al de las líneas base de popularidad global y de co-ocurrencia ítem-ítem, con intervalos de confianza bootstrap al 95 % que no se solapan.

CAPÍTULO 2: REVISIÓN DE LITERATURA

2.1 Estado del Arte

El estado del arte presentado a continuación se construyó a partir de una revisión sistemática de la literatura. Como resultado de la búsqueda inicial se obtuvieron 855 registros, los cuales fueron evaluados mediante una primera revisión de títulos y resúmenes. Después de este proceso de selección, se conservaron 38 estudios para la etapa de extracción y análisis de información.

El procedimiento seguido se describe con mayor detalle en el apéndice E, donde se especifican las fuentes consultadas, las cinco cadenas de búsqueda utilizadas, los criterios definidos para la inclusión y exclusión de documentos y el esquema empleado para organizar la información obtenida. Esta documentación busca garantizar que el proceso pueda ser reproducido y que los resultados alcanzados puedan ser revisados o contrastados en trabajos posteriores.

De manera complementaria, se realizaron búsquedas específicas sobre conceptos teóricos que no dependen directamente del sector analizado. Entre ellos se incluyen la factorización de matrices, el uso de retroalimentación implícita y los métodos de búsqueda aproximada de vecinos cercanos. Para estos temas se priorizaron trabajos seminales y revisiones ampliamente reconocidas, con el fin de establecer una base conceptual sólida para el desarrollo de la investigación.

2.1.1 Contexto general de los sistemas de recomendación

Los sistemas de recomendación permiten seleccionar información relevante dentro de grandes volúmenes de datos, con el objetivo de mostrar al usuario aquellos ítems que tienen mayor probabilidad de responder a sus necesidades (Bobadilla et al., 2013; Li et al., 2023). Tradicionalmente, el área se divide en tres enfoques principales: filtrado colaborativo, filtrado basado en contenido y modelos híbridos. Estos métodos enfrentan problemas comunes como la

esparsidad de las interacciones, el arranque en frío y la escalabilidad (Ricci et al., 2022). En entornos industriales, los sistemas suelen organizarse en tres etapas: interpretación del contexto, recuperación de candidatos y ranking final (Covington et al., 2016; Fu et al., 2020). De estas, la recuperación representa uno de los principales retos computacionales.

El aprendizaje profundo incorporó nuevas capacidades frente a los métodos tradicionales, entre ellas la generación de representaciones densas, el modelado de relaciones no lineales y la integración de distintas fuentes de información dentro de un mismo proceso de entrenamiento (Zhang et al., 2019). Sin embargo, Batmaz et al. (2019) señalan también algunas limitaciones, como un mayor costo de entrenamiento, menor interpretabilidad y una dependencia importante de la cantidad de datos disponibles.

El contexto analizado en este trabajo presenta características distintas a los escenarios más estudiados. Las compras suelen estar relacionadas con las necesidades de una obra, tienen menor frecuencia y un valor promedio más alto, mientras que la relación entre productos responde principalmente a criterios técnicos y constructivos, antes que a gustos o preferencias personales (Meng et al., 2024).

2.1.2 Filtrado colaborativo y factorización de matrices

El filtrado colaborativo fue uno de los enfoques predominantes durante más de una década después del Netflix Prize. Dentro de esta línea, Sarwar et al. (2001) formalizaron el filtrado ítem-ítem, que calcula la afinidad entre productos a partir de su coocurrencia en las compras. Debido a su bajo costo computacional y a su rendimiento estable, este método continúa siendo una referencia frecuente como línea base; Su y Khoshgoftaar (2009) presentan una revisión de esta familia de métodos y de sus principales limitaciones. Por su parte, la factorización de matrices representa las interacciones usuario-producto mediante vectores de menor dimensión y estima preferencias no observadas a partir del producto interno entre dichas representaciones (Koren et al., 2009).

Esa formulación nació para valoraciones explícitas. El registro transaccional de un distribuidor presenta una característica distinta: contiene compras, es decir, retroalimentación implícita, donde la falta de interacción no necesariamente representa rechazo, sino ausencia de evidencia. Hu et al. (2008) abordaron este escenario diferenciando la preferencia binaria del nivel de confianza asignado a cada observación, formulación que se ha convertido en una referencia para el análisis de datos de compra. En el ámbito ferretero, además, la matriz de interacción supera el 98 % de esparsidad, lo que afecta el desempeño de los métodos de factorización y agrava el problema de arranque en frío para clientes y productos nuevos (He et al., 2017; Papso, 2023).

La incorporación de redes neuronales al filtrado colaborativo tomó fuerza con Neural Collaborative Filtering (He et al., 2017), que reemplazó el producto interno por una red multicapa capaz de modelar relaciones no lineales. Sin embargo, su principal limitación se encuentra en el costo computacional, ya que NCF mantiene una complejidad O(n×m) durante la inferencia, lo que dificulta su uso para recuperar candidatos en catálogos extensos. Aun así, la superioridad de los modelos neuronales no es concluyente. Rendle et al. (2020) muestran que una factorización correctamente ajustada puede igualar o superar a NCF bajo condiciones similares de calibración, mientras que Steck (2019) obtiene resultados competitivos mediante un autocodificador lineal de una sola capa. Una tendencia similar se observa en los modelos basados en grafos: He et al. (2020) demostraron con LightGCN que la propagación entre vecinos, sin transformaciones lineales ni funciones de activación, puede mejorar el rendimiento del filtrado colaborativo.

2.1.3 Aprendizaje profundo en recomendación: DeepFM, DCN y modelos secuenciales

Wu et al. (2023) describen la evolución desde el filtrado colaborativo tradicional hacia modelos neuronales que incorporan información adicional. Dentro de esta línea surgieron arquitecturas orientadas a la predicción de la tasa de clic. Guo et al. (2017) propusieron DeepFM, que combina Factorization Machines para representar interacciones de bajo orden con una red profunda encargada de capturar relaciones de mayor complejidad. Posteriormente, Wang et al.

(2021) desarrollaron DCN-V2, que incorpora una red de cruce explícita para modelar interacciones de distintos órdenes. Su principal limitación práctica es que tanto DeepFM como DCN-V2 funcionan como modelos de ranking completo, por lo que requieren evaluar de manera conjunta cada combinación usuario-ítem. Para un catálogo de más de veinte mil SKU, generar recomendaciones obligaría a hacer un forward pass por cada producto, algo que en tiempo real resulta prohibitivo (Xiong et al., 2025). En paralelo, los modelos secuenciales como GRU4Rec (Hidasi et al., 2016) y SASRec (Kang y McAuley, 2018), este último construido sobre el mecanismo de atención de los Transformers (Vaswani et al., 2017), se ocupan de modelar cómo evolucionan las preferencias en el tiempo, y resultan especialmente útiles a la hora de diseñar la torre de usuario.

2.1.4 La arquitectura Two Towers: fundamentación y formulación

La búsqueda de un equilibrio entre expresividad, latencia y escalabilidad fue lo que llevó a la arquitectura Two Towers, conocida también como Dual Encoder o Deep Structured Semantic Model (DSSM). La idea central es separar la representación del usuario y la del ítem en dos redes que trabajan de forma independiente: una torre de consulta procesa las características del usuario y produce un embedding denso, mientras que una torre de candidatos hace lo propio con los atributos del ítem y genera otro vector en ese mismo espacio compartido (Yi et al., 2019). La afinidad entre las dos representaciones se calcula mediante una función de similitud simple, como el producto punto o la similitud coseno, aplicada sobre vectores que pueden ser precomputados previamente.

Uno de los trabajos que consolidó esta arquitectura fue el desarrollado por Yi et al. (2019) en Google para YouTube. En este estudio se formalizó el enfoque y se incorporó una corrección para el sesgo producido durante el muestreo, representando la probabilidad condicional mediante una función softmax aplicada al producto punto de los embeddings. Para facilitar el entrenamiento

a gran escala, los autores utilizaron negativos obtenidos dentro del mismo lote e incorporaron un término de corrección para reducir la influencia excesiva de los ítems más populares. Posteriormente, Yang et al. (2020) identificaron una limitación de este método: los negativos in-batch únicamente provienen de los productos presentes en cada lote, por lo que una parte poco frecuente del catálogo queda fuera de la comparación. Como alternativa, propusieron combinar estos ejemplos con negativos seleccionados de manera uniforme sobre el catálogo completo. Esta forma de trabajar se ha convertido en el estándar de facto en la industria, replicada en casos como Facebook Search (Huang et al., 2020) y Uber Eats (Ling et al., 2023).

La gran ventaja computacional está en el desacoplamiento. Una vez que el modelo está entrenado, los embeddings de todos los ítems se precomputan offline y se indexan para poder buscarlos de forma eficiente. Cuando llega una consulta, solo se ejecuta el forward pass de la torre de usuario y una búsqueda por vecindario aproximado (ANN). Un modelo de ranking completo necesita $O(n \times d^2)$ operaciones, mientras que el Two Towers baja esa complejidad a $O(n \times d)$ para la búsqueda ANN, más $O(d^2)$ por consulta (Xiong et al., 2025). Entre las estructuras de índice más usadas están FAISS (Johnson et al., 2021), que ofrece soporte GPU y estrategias como Flat, IVF y HNSW, y ScaNN (Guo et al., 2020), que emplea cuantización vectorial anisotrópica. Ling et al. (2023) cuentan cómo Uber migró miles de modelos locales por ciudad a un único modelo global Two Towers, con una reducción de veinte veces en el tamaño del modelo y latencias de unos 100 ms sirviendo a cientos de millones de usuarios; la mejora del Recall@500 que ese informe reporta, del 89 % al 93 %, la atribuye a la corrección LogQ.

2.1.5 Variantes del paradigma, cold-start y recomendación de complementariedad

La principal limitación teórica de Two Towers es que las representaciones se calculan de forma aislada durante el encoding. Yang et al. (2025) propusieron HIT (Hierarchical Interaction-Enhanced Two-Tower), que incorpora mecanismos jerárquicos de interacción; desplegado en Tencent, con 400 millones de usuarios, elevó el GMV un 1,98 % y el ROI un 1,55 %desplegado en

el sistema de publicidad de Tencent, que sirve miles de millones de impresiones diarias, elevó el GMV un 1,66 % y el ROI un 1,55 %. En una dirección parecida, Xiong et al. (2025) presentaron FIT (Fully Interacted Two-Tower), que introduce interacciones aprendibles durante el entrenamiento y mantiene el serving desacoplado mediante destilación, con mejoras del 3,5 % en AUC y del 6,1 % en Hit Ratecon mejoras relativas de AUC (RelaImpr) de entre el 4,58 % y el 14,34 % sobre la doble torre estándar en cuatro conjuntos públicos. Para el arranque en frío, Lee y Cho (2023) diseñaron una variante con dos codificadores de usuario complementarios, uno acoplado de forma estrecha y otro más suelto, que se seleccionan dinámicamente con Gumbel-Softmax y consiguen mejoras de hasta un 5,2 % en Recall@100mejoras de hasta un 5,2 % en Recall@150. Liu et al. (2026) revisan ese problema y observan que las soluciones recientes se apoyan cada vez más en señales de contenido y en modelos de lenguaje para suplir la falta de historial; Su et al. (2025) muestran, sobre la propia estructura de doble torre, que fusionar fuentes heterogéneas mejora la representación cuando el historial de interacción es escaso.

En el ámbito de la recomendación de productos complementarios, el trabajo más cercano a esta investigación es el de Osowska-Kurczab et al. (2025), quienes describen el sistema implementado en Allegro. Sus resultados muestran que una misma arquitectura Two Towers puede utilizarse para resolver tres tareas diferentes: identificar productos similares, recomendar productos complementarios y generar recomendaciones orientadas a la inspiración. Para complementariedad reportan un incremento del 1,62 % en CTR, y para la variante de inspiración un 3,12 % en la tasa de llamada a la acción. Yan et al. (2022) y Papso (2023) aportan la mirada del comercio electrónico general. Zhang et al. (2021) modelan de forma conjunta las relaciones de sustitución y de complementariedad sobre un grafo de productos para recomendación secuencial, y muestran que distinguir ambos vínculos mejora la calidad de la recomendación; en un catálogo de obra la diferencia es operativa, porque al comprador le sirve el sellador que acompaña al perfil y no un segundo perfil equivalente. Los resultados de despliegues industriales, como el informe de

ingeniería de Uber (Ling et al., 2023), se usan como contexto complementario, distinto de la literatura científica arbitrada que sustenta las afirmaciones centrales.

2.1.6 Análisis comparativo de paradigmas

La Tabla 1 sintetiza la comparativa de los principales paradigmas frente a las restricciones del dominio ferretero B2B, derivada de la revisión bibliográfica.

Tabla *1*

Comparativa de paradigmas de recomendación frente a las restricciones del dominio ferretero B2B (2016–2025)

Paradigma	Escalabilidad (>20k SKU)	Robustez esparsidad	Features heterogéneas	Complemen - tariedad	Cold start
Filtrado colaborativo ítem-ítem por co-ocurrencia (Sarwar et al., 2001)	Media	Baja	No	Limitada	No
NCF (He et al., 2017)	Baja (O(n×m))	Media	Parcial	Limitada	No
DeepFM / DCN-V2	Baja (ranking)	Media	Sí	Sí	Parcial
Recom. secuencial (SASRec)	Media	Media	Parcial	Indirecta	Parcial
Two Towers (Yi et al., 2019)	Alta (ANN)	Alta	Sí	Sí	Sí (contenido)

Nota. Elaboración propia a partir de la revisión bibliográfica. Sarwar et al. (2001); Covington et al. (2016); He et al. (2017); Guo et al. (2017); Wang et al. (2021); Yi et al. (2019); Yang et al. (2025); Xiong et al. (2025).

La Tabla 1 muestra que Two Towers es la única alternativa que cumple a la vez los cinco requisitos del dominio: escala sobre catálogos de más de veinte mil SKU, tolera la esparsidad, integra features heterogéneas, soporta la tarea de complementariedad y puede implementarse con recursos de nivel académico.

La Tabla 2 resume los resultados empíricos reportados en los estudios más relevantes, que sirven como referencia cuantitativa para el diseño experimental de este trabajo. Los estudios de las Tablas 1 y 2 se seleccionaron por su relevancia directa para la tarea, su publicación en venues arbitrados o su condición de referencia técnica reconocida, y su reporte de métricas comparables (Recall@K, Hit Rate, MRR o NDCG), priorizando los trabajos con evaluación empírica reproducible. Paraschakis et al. (2015) comparan recomendadores top-N desde una perspectiva industrial y encuentran que buena parte de la variación entre resultados publicados proviene de diferencias en el protocolo de evaluación y no de diferencias entre modelos.

Tabla *2*

Comparativa de resultados reportados para arquitecturas Two Towers en despliegues industriales y evaluaciones académicas (2022–2025)

Estudio / Contexto	Mejora principal	Métricas de ranking	Latencia / Escala
Yang et al. (2025), Tencent	GMV +1,66 %; ROI +1,55 %	Relevancia sobre líneas base	Latencia comparable; miles de millones de impresiones diarias
Osowska-Kurczab et al. (2025), Allegro	CTR complem. +1,62 %; CTA inspiración +3,12 %	CTR; GMV por visita; CTA; CVR	40 ms p99; 20k consultas/s
Lee y Cho (2023), CiteULike/MLIMDb	Recall@150 +2,7 % / +5,2 %	Recall@K; MRR	No reporta latencia
Xiong et al. (2025), Datasets públicos	RelaImpr en AUC +4,58 % a +14,34 %	AUC; Logloss; RelaImpr	Serving desacoplado
He et al. (2024), Scientific Reports	HR@20 0,8553 y 0,8812; sin mejora porcentual reportada	Pre@20; Recall@20; NDCG@20; HR@20	Sin evaluación de servido

Nota. Elaboración propia. Yang et al. (2025) y Osowska-Kurczab et al. (2025) reportan despliegues en producción; Lee y Cho (2023), Xiong et al. (2025) y He et al. (2024) evalúan sobre conjuntos públicos. GMV = Gross Merchandise Value; ROI = Return on Investment; CTR = Click-Through Rate; NDCG = Normalized Discounted Cumulative Gain; MRR = Mean Reciprocal Rank; ANN = Approximate Nearest Neighbor; HR = Hit Rate.

2.1.7 Vacíos de investigación

La recomendación en entornos empresariales cuenta con más de dos décadas de investigación. Geyer-Schulz et al. (2003) compararon reglas de asociación con enfoques de compra repetida en un contexto B2B; Shambour y Lu (2015) integraron información de confianza asociada tanto al usuario como al ítem; Vlachos et al. (2016) desarrollaron un sistema interpretable aplicado a los equipos comerciales de IBM; Zhang et al. (2017) propusieron GreedyBoost, un método de ensamble para recomendar productos y servicios entre empresas, validado con datos reales; Hildebrandt et al. (2019) estudiaron relaciones no lineales en catálogos de automatización industrial; Cho et al. (2023) utilizaron grafos de conocimiento para recomendar compradores potenciales; y Saraei et al. (2025) combinaron reglas de asociación con técnicas de factorización en un distribuidor canadiense. En conjunto, estos trabajos coinciden en el tipo de relación comercial B2B, pero difieren tanto en el sector de aplicación como en la arquitectura utilizada.

El sector ferretero y de materiales de construcción también presenta algunos antecedentes, aunque principalmente mediante enfoques tradicionales. Pradel et al. (2011) analizaron datos de compra provenientes de una cadena francesa dedicada al bricolaje. Bagherifard et al. (2018) midieron complementariedad por similitud semántica sobre una taxonomía ponderada de materiales de construcción. Mulyawan et al. (2020) aplicaron FP-Growth a la compra de materiales, y Felix y Tjen (2025) derivaron reglas de asociación sobre una muestra de 3.462 transacciones de un comercio de materiales en Indonesia. Ninguno recurre a recuperación neuronal: todos operan con reglas de asociación, análisis de canasta o filtrado ítem-ítem. La doble torre, por su parte, ya llegó al comercio industrial. Tang (2025) propone una recuperación multimodal en dos etapas con codificadores desacoplados, y es el antecedente más cercano en cuanto a arquitectura.

La recomendación de complementarios es la más trabajada de las cuatro dimensiones. McAuley et al. (2015) formalizaron la distinción entre sustitutos y complementarios sobre redes

derivadas de registros de navegación y co-compra; Zhang et al. (2018) y Bibas et al. (2023) propusieron modelos neuronales específicos; Hao et al. (2020) desplegaron P-Companion sobre más de doscientos millones de productos en Amazon; y Kvernadze et al. (2022) mostraron que las representaciones duales entrenadas sobre co-compra capturan complementariedad sin supervisión explícita. En el sector ferretero, la relación entre productos tiene una naturaleza distinta. Un perfil de aluminio, por ejemplo, puede ser compatible con un determinado espesor de vidrio y no con otro, por lo que la complementariedad responde primero a una condición técnica y física antes que a una preferencia del comprador. Sugahara et al. (2024) señalan que las etiquetas obtenidas a partir del comportamiento de co-compra no siempre representan una complementariedad funcional real. Esta diferencia puede ser todavía más relevante en catálogos técnicos y se retoma en la sección 5.3 como una de las limitaciones del presente estudio.

En el contexto regional, Videla-Cavieres y Ríos (2014) ampliaron el análisis de canasta mediante técnicas de minería de grafos sobre 238 millones de transacciones mayoristas en Chile, orientadas a identificar oportunidades de venta cruzada. Alatrista-Salas et al. (2019) analizaron transacciones bancarias, Gehrkue y Baladron (2025) estudiaron una empresa latinoamericana de bebidas y Faz Gallardo et al. (2025) abordaron el comercio minorista en Ecuador. Los 38 estudios seleccionados cubren, de manera parcial, cuatro características relevantes para esta investigación: recuperación neuronal mediante arquitecturas de doble torre, relaciones comerciales B2B, catálogos ferreteros y de materiales de construcción, y uso de datos transaccionales latinoamericanos. Sin embargo, ninguno reúne las cuatro condiciones de forma simultánea. Los trabajos más próximos se relacionan con esta investigación desde perspectivas diferentes: Tang (2025) por el uso de la arquitectura en comercio industrial, aunque sin abordar un catálogo ferretero ni datos de la región; Zhang et al. (2017) por el tipo de comprador, con métodos de ensamble; Videla-Cavieres y Ríos (2014) por el mayoreo latinoamericano, con minería de grafos. Este estudio analiza dicha combinación en un catálogo compuesto por más de veinte mil SKU y

con un nivel de esparsidad superior al 98 %. El apéndice E presenta las búsquedas que respaldan esta afirmación y deja documentado el procedimiento para facilitar su reproducción.

2.2 Marco Teórico

2.2.1 Métricas de evaluación en sistemas de recomendación

La evaluación offline de los sistemas de recomendación arrastra desafíos metodológicos que ya están bien documentados: no se puede medir el impacto real sin pruebas A/B, los datos históricos sufren de sesgo de exposición (solo se observan interacciones con ítems que ya se conocían) y los resultados dependen en buena medida de las métricas que se elijan (Herlocker et al., 2004; Jadon y Patil, 2025). Para la fase de recuperación, las métricas que más interesan son las orientadas al top-K. El Recall@K mide qué proporción de los ítems relevantes del ground truth aparecen entre las K primeras recomendaciones, y es la métrica que mejor se alinea con el objetivo mismo de la recuperación (Cremonesi et al., 2010). El Hit Rate@K, por su parte, mide la proporción de usuarios para los que al menos un ítem relevante figura en el top-K, y tiene una lectura más directa desde el punto de vista del negocio. El MRR valora la posición del primer ítem relevante, capturando así la importancia de colocar los productos pertinentes en las primeras posiciones. El NDCG@K amplía esa mirada incorporando relevancia graduada y un descuento logarítmico según la posición.

La situación: En una ferretería, lo único que importa para medir si un sistema de recomendación funciona es si el cliente compró o no compró el producto que le mostraste. No hay términos medios (como "le gustó un poco" o "lo calificó con 4 estrellas").

Por eso usan estas métricas:

1. Recall@K es la principal. Mide: "De todos los productos que el cliente compró, ¿cuántos de ellos aparecieron entre mis K primeras recomendaciones?"

2. MRR la usan como apoyo. Mide: "¿En qué posición apareció el primer producto que el cliente compró? Mientras más arriba, mejor."

¿Y las otras métricas (NDCG y Hit Rate)?

- NDCG no sirve mucho aquí, porque fue diseñada para casos donde los productos tienen diferentes niveles de importancia (ej: "me encantó", "me gustó", "no me gustó"). Pero como acá solo hay "compró" o "no compró", el NDCG se vuelve igual que el MRR, no aporta nada nuevo.
- Hit Rate (que mide si al menos un producto comprado apareció en las recomendaciones) es exactamente lo mismo que el Recall en este caso. ¿Por qué? Porque cada búsqueda solo tiene un solo producto relevante (el que el cliente compró). Entonces:
    - O el sistema muestra ese producto (y tanto Recall como Hit Rate valen 1)
    - O no lo muestra (y ambos valen 0)

La prueba: Cuando hicieron los cálculos, vieron que el Recall@100 dio 33.16% y el Hit Rate@100 dio 33.33%. Prácticamente iguales, así que no tiene sentido usar los dos.

En resumen: Se quedan con Recall como la medida principal (porque es la más clara para este negocio) y MRR como apoyo (para ver si el producto comprado aparece entre los primeros puestos). Las demás métricas no agregan valor y solo complican.

Ese acuerdo sobre qué medir es más reciente de lo que parece. Cremonesi et al. (2010) mostraron que las metodologías basadas en métricas de error no encajan de forma natural en la tarea top-N: los algoritmos que minimizan el error de predicción no rinden necesariamente como se espera al recuperar, y las mejoras en error con frecuencia no se traducen en mejoras de

exactitud. De ese trabajo salen dos advertencias que condicionan el diseño experimental de esta tesis. La primera es que un algoritmo no personalizado puede superar a varios métodos habituales y acercarse al desempeño de los sofisticados, razón por la cual aquí la popularidad global se evalúa como línea base en lugar de descartarse de antemano. La segunda es que unos pocos artículos de alta rotación bastan para sesgar el resultado, de modo que la construcción del conjunto de prueba pesa tanto como la elección de la métrica. Paraschakis et al. (2015) convergen desde la perspectiva industrial, al atribuir buena parte de la variación entre resultados publicados al protocolo de evaluación antes que a los modelos. La respuesta de este trabajo es someter a los tres modelos al mismo conjunto de consultas y al mismo ground truth, y declarar la incertidumbre mediante intervalos, según detallan las secciones 3.1.5 y 4.1.1.

2.2.2 Fundamentos conceptuales de los componentes del modelo

Varios conceptos transversales sostienen la arquitectura y reaparecen a lo largo del desarrollo técnico.

Un embedding es como una tarjeta de identidad numérica para cada cosa que tenga un nombre (como un cliente, un producto o una marca).

En lugar de usar el nombre o un número cualquiera, le asignamos una lista de números (como coordenadas en un mapa) que lo representan. La idea es que dos cosas parecidas tengan listas de números parecidas, es decir, que estén "cerca" en ese mapa.

Por ejemplo:

Dos productos que se compran juntos tendrán números parecidos.

Dos clientes con gustos similares también.

En este trabajo, cuando hay muchas categorías (por ejemplo, miles de productos), cada una aprende su propia lista de números mientras el sistema va mejorando. Para eso usan una "tabla de

búsqueda" que va ajustando esos números poco a poco, como cuando vas corrigiendo un dibujo hasta que quede bien.

El perceptrón multicapa (MLP) es como una cadena de montaje con varias estaciones (capas). Cada estación recibe información, la procesa y se la pasa a la siguiente. Todas las estaciones están conectadas entre sí.

La función ReLU es como un filtro que solo deja pasar los números positivos. Si algo es negativo, lo convierte en cero. Esto le permite a la máquina entender cosas que no son rectas ni simples, como patrones complicados. Además, ayuda a que la máquina no se "canse" ni se olvide de lo que va aprendiendo.

El Dropout es como un entrenamiento con distracciones. Durante el aprendizaje, apagas al azar algunas de las estaciones de la cadena. Así la máquina no se vuelve dependiente de una sola estación y aprende a resolver el problema de varias maneras. Esto evita que memorice de más (sobreajuste) y que falle con datos nuevos.

La normalización es como poner todos los números en la misma página. Si tienes datos muy grandes (por ejemplo, edades de 0 a 100) y otros muy pequeños (como puntajes de 0 a 1), la máquina le daría mucha importancia a los grandes. Para evitarlo, reescalas los números para que todos tengan un promedio de cero y una variación parecida. Así ningún dato domina a los demás y la máquina aprende de forma más justa.

Softmax

Imagina que tienes 10 cantantes y les das un puntaje del 1 al 10. Softmax convierte esos puntajes en porcentajes que suman 100%. Así, el que sacó 9 puntos no tiene 90% de probabilidad de ganar, sino un porcentaje ajustado que resalta al mejor y reduce a los peores. Es como "repartir el pastel" de manera que los mejores se llevan la mayor parte.

Entropía cruzada

Es como un juez que compara tus porcentajes (lo que predices) con los resultados reales (quién realmente ganó). Si tu predicción fue mala (le diste 90% al que quedó último), la entropía cruzada te da un "castigo" grande. Si fue buena, el castigo es pequeño. Por eso se usa para entrenar modelos: el modelo ajusta sus puntajes para que el castigo sea cada vez menor.

Búsqueda por vecindario aproximado (ANN)

Supón que tienes un millón de canciones y quieres encontrar las 10 más parecidas a una que te gusta. Revisar una por una sería muy lento. ANN usa trucos para encontrarlas rápido, aunque no siempre sean las exactamente más parecidas, sino "casi" las más parecidas.

Índice plano (búsqueda exacta)

Es revisar CANCIÓN POR CANCIÓN todas las canciones. Es perfecto pero lentísimo si hay muchas.

Índice invertido (IVF)

Es como dividir las canciones por géneros (rock, pop, etc.). Primero decides a qué género pertenece tu canción y solo buscas dentro de ese grupo. Es más rápido, pero si tu canción está en el borde entre géneros, podrías perder alguna parecida.

HNSW (el mejor balance)

Es como tener varios mapas: uno muy general (solo continentes), uno más detallado (países) y uno muy detallado (calles). Para buscar, empiezas en el mapa general, saltas al detallado y así encuentras rápido las canciones más parecidas sin revisar todas. Es el que mejor equilibra velocidad y precisión, especialmente cuando usas producto punto (una forma de medir parecido).

En resumen: Softmax y entropía cruzada sirven para que el modelo aprenda a hacer buenas predicciones, y ANN (especialmente HNSW) sirve para que esas predicciones se encuentren rapidísimo entre millones de opciones.

CAPÍTULO 3: DESARROLLO

3.1 Metodología de Investigación y Diseño del Sistema

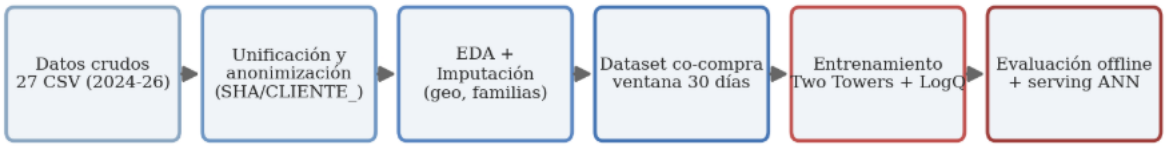
Este trabajo se apoya en un enfoque cuantitativo y responde a una finalidad aplicada: no persigue generar teoría nueva, sino resolver un problema operativo concreto mediante la construcción y evaluación de un artefacto computacional. Su alcance es explicativo, porque contrasta hipótesis sobre el efecto que cada decisión de diseño produce en el desempeño de la recuperación, y no se limita a describir el comportamiento del sistema.

El diseño es cuasi-experimental con medidas repetidas. Existe manipulación deliberada de las variables independientes, a saber, la arquitectura del recomendador, la presencia o ausencia de la corrección LogQ y la disponibilidad de cada bloque de características de la torre de consulta. Lo que no existe es asignación aleatoria de unidades a condiciones, por dos razones. La primera es que todas las consultas de validación atraviesan todas las condiciones, de modo que la comparación es intra-sujeto y no entre grupos independientes. La segunda es que la partición entre entrenamiento y validación la fija el calendario y no el azar, decisión deliberada que la sección 3.1.1 justifica como defensa contra la fuga de información. Por su dimensión temporal el estudio es longitudinal retrospectivo: se apoya en registros históricos de fuente secundaria que cubren de enero de 2024 a marzo de 2026 y mide sobre una ventana posterior a la de entrenamiento. La unidad de análisis es el par positivo de co-compra, según se define en la sección 3.1.1.

Sobre esa base, el trabajo desarrolla un artefacto computacional, el modelo de recuperación, y lo evalúa de forma empírica offline frente a líneas base, siguiendo un protocolo de validación temporal que replica las condiciones de un entorno productivo. La Figura 1 resume lo mencionado.

Figura *1*

Flujo metodológico del proyecto, desde los datos crudos hasta la evaluación offline y la simulación de servido.

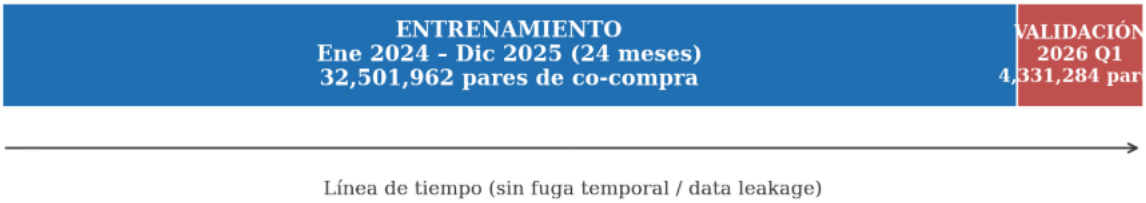


3.1.1 Enfoque metodológico y partición temporal contra fuga de datos

En los sistemas de recomendación B2B transaccionales, una división aleatoria tradicional, como la típica partición 80/20, invalida la evaluación a causa del sesgo temporal y la fuga de información. En el sector ferretero, los proyectos de construcción y las compras siguen una línea de tiempo lógica. Por eso se optó por una partición estrictamente temporal, tal como se muestra en la Figura 2. El conjunto de entrenamiento cubre desde enero de 2024 hasta diciembre de 2025, es decir, 24 meses, y logra capturar tanto las dinámicas estacionales como las tendencias multianuales. El conjunto de validación, en cambio, corresponde al primer trimestre de 2026 y permite medir la capacidad real del modelo para generalizar. La unidad de análisis es el par positivo de co-compra dentro de una ventana temporal, que se interpreta como una sesión implícita de compra técnica.

Figura 2

Partición temporal del dataset sin fuga de información: entrenamiento (2024–2025) y validación (2026 Q1).



3.1.2 Formulación matemática del espacio de recuperación dual

El problema se modela como una tarea de clasificación multiclase a gran escala, donde el catálogo de productos constituye el espacio de clases. La torre de consulta f_θ mapea el vector de entrada del cliente y su contexto x a un embedding denso continuo:

$$u(x) = f_\theta(x) \in \mathbb{R}^D,\ D = 128 \quad (3.1)$$

De forma simétrica, la torre de candidatos g_φ mapea los atributos estructurados del producto y a un embedding de ítem en el mismo espacio compartido:

$$v(y) = g_\emptyset(y) \in \mathbb{R}^D,\ D = 128 \quad (3.2)$$

El puntaje de afinidad entre una consulta y un candidato se modela mediante el producto punto escalar de ambos embeddings:

$$s(x, y) = \langle u(x), v(y) \rangle = u(x)^T V(y) \quad (3.3)$$

Durante el análisis cualitativo y la recuperación se emplea además la similitud coseno, que normaliza el producto punto por las magnitudes de los vectores:

$$sim_{cos}(x, y) = \frac{u(x)^T v(y)}{\|u(x)\| [\![v(y)]\!]} \quad (3.4)$$

La probabilidad condicional de que la consulta x recupere el candidato y se modela como una softmax sobre el catálogo C, siguiendo la formulación de Yi et al. (2019):

$$P(y|x) = \frac{\exp(s(x,y))}{\sum_{j \in C} \exp(s(x,y_i))} \quad (3.5)$$

En inferencia, el problema se reduce a recuperar los top-K candidatos con mayor similitud respecto de la consulta del cliente, operación que se resuelve eficientemente mediante búsqueda por vecindario aproximado sobre los embeddings precomputados del catálogo.

3.1.3 Negativos in-batch y corrección del sesgo de popularidad (LogQ)

Calcular el denominador de la softmax sobre el catálogo completo de 20.683 productos en cada época es en la práctica inviable. Para resolverlo se recurre a los negativos in-batch: las interacciones positivas de otros clientes dentro del mismo minilote se aprovechan como ejemplos negativos implícitos. El inconveniente es que estos negativos se muestrean según la distribución de popularidad de los datos, así que, si no se corrige, el modelo termina con un sesgo de popularidad y tiende a recomendar una y otra vez los productos más populares.

La corrección LogQ (Yi et al., 2019) evita que el modelo beneficie en exceso a los productos que aparecen con más frecuencia durante el entrenamiento. Para lograrlo, antes de calcular las probabilidades finales con la softmax, se resta un valor que refleja qué tan probable era que cada producto hubiera sido seleccionado en el proceso de muestreo, es decir, log P(y). De esta manera, las recomendaciones quedan mejor equilibradas y se reduce el sesgo hacia los productos más comunes.

$$S^{corr}(x,y) = u(x)^T v(y) - \log(P(y)) \quad (3.6)$$

Aplicada sobre el conjunto de negativos del lote B, la probabilidad corregida quedaría:

$$P(y|x) = \frac{\exp(u(x)^T v(y) - \log(P(y)))}{\sum_{j \in \mathcal{B}} \exp(u(x)^T v(y_j) - \log(P(y_j)))} \quad (3.7)$$

En este caso, P(y) denota la frecuencia relativa de cada producto en el conjunto de entrenamiento.

$$\mathcal{L} = -\frac{1}{|\mathcal{B}|}\sum_{(x,y)\in\mathcal{B}} \log(P(y|x) \quad (3.8)$$

3.1.4 Construcción de pares de co-compra

Los ejemplos positivos se construyen a partir del historial de compra de cada cliente: dos productos forman un par positivo cuando un mismo cliente los adquirió dentro de una ventana de 30 días.

$$Fecha_{cand} - Fecha_{query} \leq 30\ días \quad (3.9)$$

La ventana de 30 días responde a que los materiales de una obra rara vez se adquieren en una misma fecha. Para atenuar el peso de los clientes de alto volumen, se limita además a cinco los productos considerados por cliente y día.

3.1.5 Métricas de evaluación

Las métricas conceptualizadas en la sección 2.2.1 se formalizan a continuación. El Recall@K mide la fracción de productos efectivamente co-comprados que el modelo sitúa entre las K primeras recomendaciones:

$$\text{Recall@}K = \frac{1}{|\mathcal{Q}|}\sum_{q\in\mathcal{Q}} \frac{|\text{Top-}K(q)\cap\mathcal{R}_q|}{|\mathcal{R}_q|} \quad (3.10)$$

El Mean Reciprocal Rank (MRR) pondera la posición del primer producto relevante recuperado; su valor crece cuanto más arriba aparece:

$$\text{MRR} = \frac{1}{|\mathcal{Q}|}\sum_{q=1}^{|\mathcal{Q}|} \frac{1}{\text{rank}_q} \quad (3.11)$$

La cobertura de catálogo (Catalog Coverage) es la fracción del catálogo que el modelo recomienda al menos una vez:

$$\text{Coverage} = \frac{|\{j:\exists q,\ j\in\text{Top-}K(q)\}|}{|\mathcal{C}|} \times 100\% \quad (3.12)$$

El coeficiente de Gini mide la concentración de las recomendaciones: un valor cercano a 0 refleja una distribución uniforme sobre el catálogo y uno cercano a 1, la concentración en unos pocos productos:

El contraste estadístico entre modelos se resuelve mediante intervalos de confianza bootstrap al 95 %, obtenidos por remuestreo con reemplazo de las consultas del conjunto de validación, con 1.000 réplicas y percentiles 2,5 y 97,5. Se considera que dos modelos difieren cuando sus intervalos no se solapan; cuando se solapan, la diferencia se declara no concluyente. Aplicado sobre el Recall@100, este procedimiento es el que resuelve el contraste de las hipótesis enunciadas en la sección 1.6: se rechaza la hipótesis nula si el intervalo del modelo Two Towers no se solapa con el de la línea base y su estimación puntual es superior.

Este criterio es conservador y su lectura debe ser asimétrica. Que dos intervalos calculados por separado se solapen no equivale a que la diferencia entre los modelos sea nula. Un contraste construido sobre la distribución bootstrap de la diferencia es más sensible: los tres modelos se evalúan sobre las mismas consultas y ese diseño pareado descuenta la varianza que comparten. Al enfrentar los intervalos por separado esa información se pierde, y el procedimiento rechaza menos de lo que podría. De ahí la asimetría: cuando los intervalos no se solapan, la diferencia queda firmemente establecida; cuando se solapan, lo que se declara es ausencia de evidencia concluyente bajo este criterio y no evidencia de que ambos modelos rindan igual. El capítulo 4 reporta en esos términos las diferencias de Recall@10 y de MRR frente a la línea base de co-ocurrencia. La pérdida de potencia resulta asumible. La hipótesis se contrasta sobre el Recall@100, donde la distancia entre modelos es amplia y la decisión no depende del procedimiento elegido, y la sección 4.1.1 publica los intervalos completos, de manera que cualquier lector puede rehacer la comparación con un contraste más fino.

$$G = \frac{\sum_{i=1}^{n}(2i-n-1)\,x_{(i)}}{n\sum_{i=1}^{n} x_{(i)}} \qquad (3.13)$$

3.1.6 Diseño y justificación de las líneas base

La validez de una comparación depende de la exigencia de las líneas base: buena parte de las mejoras atribuidas a los modelos neuronales se desvanece al contrastarlos con baselines

clásicos bien ajustados (Ferrari Dacrema et al., 2019). Por eso se eligieron dos referencias que resuelven la misma tarea de recuperación y son de uso extendido: la popularidad global y la co-ocurrencia ítem-ítem.

La popularidad global recomienda a todo cliente los productos más vendidos del catálogo, con independencia de su historial. Pese a su simplicidad, constituye una referencia exigente: en un dominio de cola larga, unos pocos productos de alta rotación (cemento, agregados, varillas) concentran gran parte de la demanda.

La co-ocurrencia ítem-ítem recomienda los productos comprados con mayor frecuencia junto al producto-semilla que el cliente consulta. Es el método clásico de referencia para la complementariedad, popularizado por Amazon (Linden et al., 2003); de sus variantes de puntuación se retuvo la de mejor desempeño (cuaderno 14).

Se evaluó también la factorización de matrices (SVD), pero se descartó porque modela la afinidad cliente-producto de forma global y no la complementariedad condicionada a un producto-semilla, que es el objeto de este trabajo. Todas las comparaciones se realizaron sobre la misma muestra, el mismo ground truth y las mismas métricas, de modo que el contraste es directo.

3.1.7 Herramientas y entorno de desarrollo

La implementación se realizó íntegramente con software de código abierto, sin costos de licenciamiento. El preprocesamiento de los millones de registros se ejecutó con Polars (v1.41), elegido por su eficiencia en memoria y su motor de ejecución paralela sobre operaciones de agrupación y join. El modelado se desarrolló sobre TensorFlow (v2.21) y TensorFlow Recommenders (TFRS, v0.7.2), que proporcionan componentes modulares para la construcción de torres, la tarea de recuperación y las métricas FactorizedTopK. La simulación de servido empleó FAISS para la indexación vectorial (índices Flat y HNSW), y la línea base de co-ocurrencia ítem-

ítem se construyó con Polars y NumPy sobre los pares de co-compra del conjunto de entrenamiento. El análisis cualitativo de embeddings utilizó scikit-learn (t-SNE) y umap-learn (UMAP). El trabajo se organizó en una secuencia reproducible de cuadernos Jupyter (Apéndice A), con control de versiones mediante Git.

3.2 Desarrollo e Implementación Técnico-Operativa

3.2.1 Fuente de datos y consideraciones éticas

La fuente principal es la base de datos transaccional interna (Data Warehouse / ERP) de la compañía distribuidora, seudonimizada conforme a la Ley Orgánica de Protección de Datos Personales del Ecuador (LOPDP). A diferencia de los datasets públicos (MovieLens, Amazon Reviews, Yelp), pertenecientes a dominios de entretenimiento y consumo masivo, los datos del ERP favorecen que el modelo aprenda de patrones de consumo auténticamente locales: la estacionalidad de la construcción en Ecuador, los tipos de obra predominantes, las marcas efectivamente disponibles y las rutas de distribución terrestre. La anonimización se realizó mapeando cada RUC único a un identificador determinista del formato CLIENTE_00001 a CLIENTE_14935, y truncando las coordenadas geográficas a dos decimales (~1,1 km de precisión), medidas que suprimen el identificador tributario y rebajan la precisión de la ubicación.

Ese tratamiento es una seudonimización y no una anonimización, y su riesgo residual se puede medir sobre el propio conjunto. De los 14.935 clientes, 5.777 conservan una coordenada propia, y para el 99,52 % de ellos el par truncado a dos decimales es único: cada uno ocupa en solitario su celda de algo más de un kilómetro de lado. El truncamiento rebaja la precisión, pero en distribución ferretera B2B esa celda suele contener un único negocio, de modo que la coordenada sigue funcionando como identificador. Los atributos de contexto se comportan de otra forma. La ciudad no aísla a ningún cliente, y la combinación de ciudad y ruta deja solo al 0,78 % y en grupos de cuatro o menos al 4,22 %. El riesgo lo carga la geolocalización, no el contexto operativo.

El conjunto de trabajo conserva además la dirección del cliente, presente en la totalidad de los registros y con 12.902 valores distintos, junto a la naturaleza jurídica, el sexo, el estado civil y el cupo de crédito. Ninguna de ellas alimenta a las torres, cuyas variables de entrada son las de la Tabla 7, pero todas sobreviven al preprocesamiento porque este se limitó a sustituir el identificador tributario. La dirección es un identificador directo, no un cuasi-identificador, y su presencia obliga a tratar el conjunto como dato personal seudonimizado. Dos condiciones acotan lo que eso significa en la práctica. El acceso a la base se produjo bajo un acuerdo de confidencialidad con la empresa distribuidora, que restringe el uso de los datos a los fines de esta investigación. Y el repositorio público descrito en el apéndice A publica el código, los artefactos derivados del modelo y las tablas de resultados, no la base transaccional ni la correspondencia entre seudónimos y contribuyentes. Quien tuviera acceso al conjunto completo, en cambio, reidentificaría a los clientes de forma inmediata, y este trabajo no sostiene lo contrario.

3.2.2 Descripción y características del dataset

El dataset se compone de dos fuentes complementarias: un catálogo de productos y un registro histórico de transacciones de venta. La Tabla 3 presenta el resumen general obtenido del análisis exploratorio sobre el dataset unificado y anonimizado.

Tabla *3*

Resumen general del dataset transaccional unificado

Característica	Valor
Tipo de datos	Tabulares estructurados y texto corto (descripciones)
Período cubierto	Enero 2024 – Marzo 2026 (27 meses)
Total de transacciones (líneas)	5.876.800 registros
Clientes únicos (RUC anonimizado)	14.935
Productos únicos en catálogo	20.683 (13.840 con interacciones)
Pares cliente-producto únicos	2.518.215
Esparsidad de la matriz de interacción	98,78 %
Ciudades con operación registrada	8
Rutas de distribución	301
Venta promedio por línea	$46,71 USD (máx. $73.031,46)
Cantidad y ganancia promedio	17,17 unidades; $7,17 USD

Nota. Valores obtenidos del análisis exploratorio de datos (EDA, cuaderno 02_eda.ipynb) sobre el dataset anonimizado. La esparsidad se calcula como 1 − (pares únicos / (clientes × productos con interacciones)). Sobre el catálogo completo de 20.683 referencias asciende al 99,18 %.

La esparsidad se verifica directamente a partir de los conteos del EDA:

$$\text{Esparsidad} = 1 - \frac{2.518.215}{14.935 \times 13.840} = 98{,}78\,\% \qquad (3.14)$$

Las Tablas 4 y 5 detallan las variables disponibles en los registros transaccionales y en el catálogo de productos, junto con su tipo y porcentaje de valores nulos.

Tabla *4*

Variables del dataset de transacciones

Variable	Tipo	Descripción	% Nulos
CIUDAD	Categórica	Ciudad de la transacción	0,00 %
RUC (hash)	Identificador	Código único anonimizado del cliente	0,00 %
ANIO / MES	Numérica	Año y mes de la transacción	0,00 %
RUTA	Categórica	Ruta de distribución asignada	2,65 %
COD_PROD	Identificador	Código único del producto	0,00 %
DESCRIP	Texto	Descripción corta del producto	0,00 %
CANTIDAD / VENTA / COSTO / GANANCIA	Numérica	Magnitudes de la transacción (USD)	0,00 %
MARCA	Categórica	Marca comercial del producto	5,55 %
EAN13	Identificador	Código de barras internacional	6,09 %
LATITUD / LONGITUD	Numérica	Coordenada geográfica (truncada, 2 dec.)	27,09 %

Nota. Porcentajes de nulos calculados sobre el total de 5.876.800 registros.

Tabla *5*

Variables del catálogo de productos

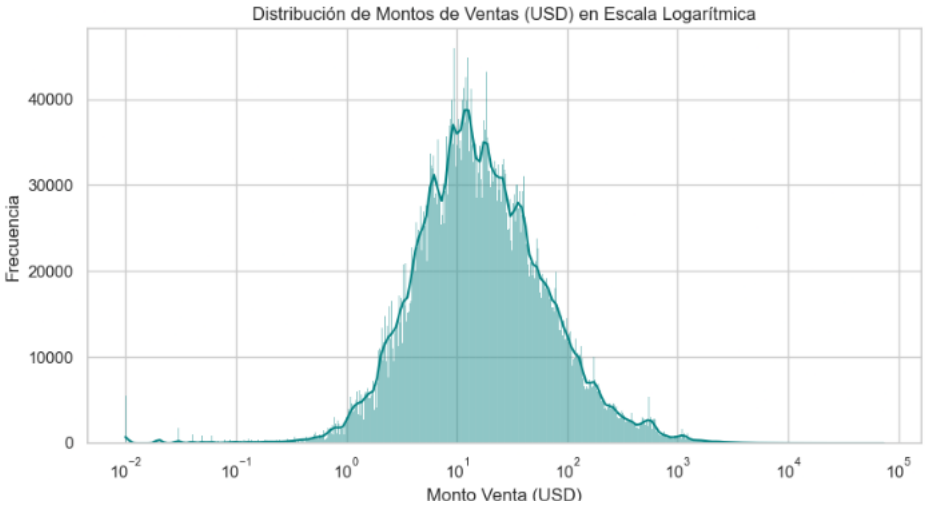
Variable	*Tipo*	*Descripción*	*% Nulos*
id_producto	Identificador	Código único del SKU en catálogo	0,0 %
familia1	Categórica	Categoría de nivel superior	6,72 %
familia2	Categórica	Subcategoría comercial	7,78 %
marca	Categórica	Marca del producto	6,80 %
precio	Numérica	Precio de venta al público (USD)	0,0 %
peso_unitario	Numérica	Peso físico del SKU	0,0 %
descripcion	Texto	Descripción corta del producto	0,0 %
ean13	Identificador	Código de barras EAN-13	50,47 %
categoría	Categórica	Campo heredado (100 % vacío, descartado)	100,0 %

Nota. El catálogo comprende 20.684 registros, de los que resultan 20.683 productos únicos tras depurar un identificador duplicado. El campo categoría está completamente vacío y se sustituye por familia1 y familia2; ean13 se descarta por su alto porcentaje de nulos.

3.2.3 Análisis exploratorio de datos (EDA)

El análisis exploratorio de los datos dejó hallazgos que condicionaron el diseño del modelo. La distribución de las ventas (Figura 3) indica que unos pocos productos, como cemento, agregados y varillas, concentran la mayor parte de las ventas, mientras que la mayoría de los productos se venden con poca frecuencia. Debido a este desequilibrio, fue necesario aplicar la corrección LogQ para evitar que el modelo recomiende casi siempre los productos más vendidos.

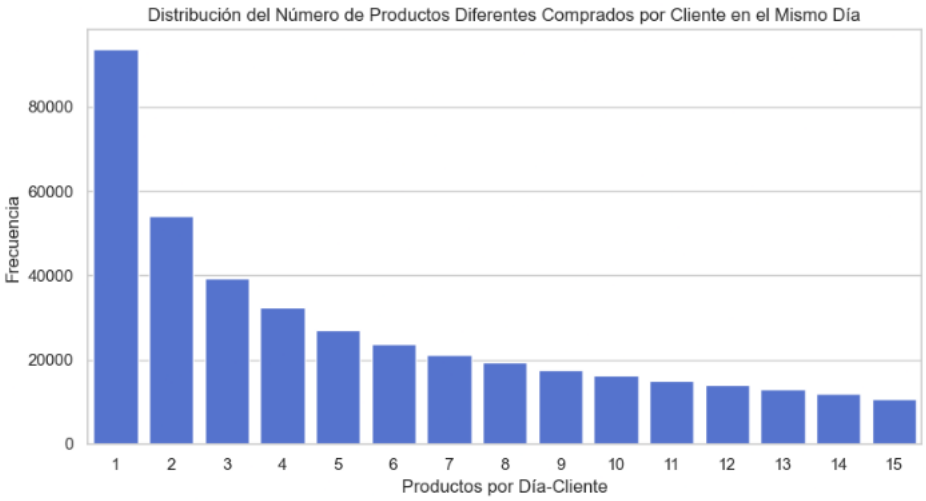
Figura *3*
Distribución de ventas del catálogo, evidenciando el comportamiento de cola larga característico del sector ferretero.



El análisis del ticket diario (Figura 4) mostró que el cliente promedio adquiere 10,94 productos únicos por día (mediana = 6; percentil 75 = 15), lo que confirma una alta co-ocurrencia diaria propicia para el cross-selling técnico y respalda la estrategia de construcción de pares de co-compra.

Figura *4*

Distribución del número de productos únicos adquiridos por cliente y día (ticket diario).



La concentración geográfica es marcada (Tabla 6): Quito representa el 46,5 % de las transacciones, seguida por Cuenca y Guayaquil. Las rutas 3 SUR 2 y 2 SUR 1 son las de mayor volumen. Esta información geográfica y de ruta se incorpora explícitamente a la torre de consulta como señal de contexto.

*Tabla **6***

Concentración geográfica de las transacciones (principales ciudades)

Ciudad	Transacciones (aprox.)	Participación
Quito	2.730.000	46,5 %
Cuenca	736.000	12,5 %
Guayaquil	671.000	11,4 %
Otras cinco ciudades	1.739.800	29,6 %

Nota. Valores aproximados obtenidos del EDA. Rutas principales: 3 SUR 2 (≈438k) y 2 SUR 1 (≈398k).

3.2.4 Preprocesamiento de datos e imputación

El preprocesamiento se realizó con la librería Polars, por su eficiencia en memoria sobre volúmenes de millones de registros. Las acciones principales fueron: (1) resolución de delimitadores inconsistentes y codificación CP1252 (latin-1), incluyendo el manejo de comillas dobles en descripciones que denotan pulgadas, p. ej. 2"; (2) imputación geográfica del 27,08

% de registros sin geolocalizaciónimputación geográfica del 27,09 % de registros sin geolocalización mediante el centroide medio (latitud/longitud) agrupado por ciudad, con fallback global; (3) imputación de RUTA nula como 'DESCONOCIDA' y de variables categóricas del catálogo como 'SIN_MARCA', 'SIN_CATEGORIA' y 'SIN_SUBCATEGORIA'; (4) capping de cinco productos diarios por combinación cliente-fecha, que redujo un 65,3 % las transacciones brutas y estabilizó el dataset frente al sesgo de distribuidores masivos; y (5) generación de las co-ocurrencias en ventana de 30 días mediante joins mensuales optimizados (chunked) para el control de memoria. El resultado fueron 32.501.962 pares de entrenamiento (2024–2025) y 4.331.284 pares de validación (2026 Q1), con cero valores nulos en las columnas críticas.

El umbral de cinco productos responde a la fuerte asimetría del ticket diario: la mediana es de 6 productos distintos por cliente-día, pero la cola se extiende hasta un percentil 99 de 61 y un máximo de 956, y el 54,1 % de las combinaciones lo supera. Las líneas recortadas se concentran así en los compradores de cola pesada (mayoristas cuya operación no responde al patrón de compra proyectual del modelo), a costa de acotar la representatividad de los clientes de alto volumen.

Más allá del análisis de nulos, se auditó la calidad del conjunto unificado mediante una rutina reproducible de validación (15_data_quality_audit): no se detectaron filas exactamente duplicadas; la integridad referencial es alta, con solo un 0,49 % de líneas cuyo SKU no figura en el catálogo; la identidad contable GANANCIA = VENTA − COSTO se cumple en el 100 % de los registros; y la totalidad de las coordenadas no nulas se ubica dentro del rango geográfico del Ecuador continental e insular. La auditoría también registró un 1,26 % de líneas con margen negativo (VENTA < COSTO), un 0,28 % con venta no positiva y 1.070 duplicados por combinación cliente-fecha-producto (0,02 %). Los dos primeros son consistentes con devoluciones, ajustes y promociones. Ninguno de los tres afecta la construcción de pares de co-compra, que opera sobre productos distintos por cliente y ventana. El catálogo codifica como cero

los valores faltantes de precio (10,4 %) y de peso unitario (50,7 %), y la variable de categoría general resultó íntegramente nula.

3.2.5 Implementación de la Query Tower

La torre de consulta codifica al cliente y su contexto operativo. Emplea capas StringLookup y Embedding para las variables categóricas (RUC, CIUDAD, RUTA y el producto consulta COD_PROD) y una capa Normalization para las variables continuas de geolocalización (LATITUD, LONGITUD). Los embeddings y las variables normalizadas se concatenan y alimentan una red densa Dense([256, 128]) con activación ReLU y regularización mediante una capa de Dropout cuya tasa se fija en la búsqueda de hiperparámetros de la sección 3.2.8, seguida de una capa de proyección lineal final a la dimensión compartida D = 128. La torre suma aproximadamente 2,25 millones de parámetros (8,59 MB), dominados por las capas de embedding de cliente y producto. La Tabla 7 detalla la dimensionalidad de cada componente.

Tabla 7
Configuración de la Query Tower (codificador del cliente y contexto)

Componente	Variable	Tipo	Dimensión
Embedding	RUC	Categórica (ID cliente)	64
Embedding	CIUDAD	Categórica	16
Embedding	RUTA	Categórica	32
Embedding	COD_PROD (consulta)	Categórica (ID)	64
Normalization	LATITUD, LONGITUD	Numérica continua	2
MLP	Dense + Dropout	Capas ocultas	256 → 128
Proyección	query_projection	Lineal	128

Nota. Activación ReLU en capas ocultas; proyección final lineal. Total ≈ 2,25 M de parámetros.

3.2.6 Implementación de la Candidate Tower

La torre de candidatos codifica los atributos del producto. Utiliza capas StringLookup y Embedding para COD_PROD (ID del SKU), MARCA, FAMILIA1 y FAMILIA2, y una capa Normalization para las variables continuas precio y peso_unitario extraídas del catálogo. La concatenación se procesa mediante una MLP Dense([256, 128]) con Dropout de la misma tasa y

proyección final lineal a D = 128, de modo que ambos embeddings habitan el mismo espacio vectorial. La Tabla 8 resume su configuración.

Tabla *8*

Configuración de la Candidate Tower (codificador de atributos del catálogo)

Componente	Variable	Tipo	Dimensión
Embedding	COD_PROD	Categórica (ID SKU)	64
Embedding	MARCA	Categórica	16
Embedding	FAMILIA1	Categórica	16
Embedding	FAMILIA2	Categórica	16
Normalization	precio, peso_unitario	Numérica continua	2
MLP	Dense + Dropout	Capas ocultas	256 → 128
Proyección	candidate_projection	Lineal	128

Nota. Activación ReLU en capas ocultas; proyección final lineal al espacio compartido de 128 dimensiones.

3.2.7 Pipeline de entrenamiento y serialización

Las dos torres se acoplan mediante la clase tfrs.Model de TensorFlow Recommenders, utilizando la tarea de recuperación (tfrs.tasks.Retrieval) con negativos in-batch y la tabla de corrección LogQ implementada como una StaticHashTable de probabilidades de muestreo. Ambas torres normalizan su salida en L2, de modo que el producto punto entre una consulta y un candidato es su similitud coseno, la misma función con la que el índice vectorial recupera candidatos en servido; los logits quedan entonces acotados y la pérdida incorpora un parámetro de temperatura. La optimización emplea el algoritmo Adam con tasa de aprendizaje de 0,001 y un tamaño de lote de 1.024. La dimensión del espacio compartido (D = 128) y la estructura de las capas densas Dense([256, 128]) siguen los valores de referencia consolidados en la literatura de Two Towers a escala industrial (Yi et al., 2019; Ling et al., 2023). Los hiperparámetros restantes se determinaron mediante la búsqueda sistemática que se describe en la sección 3.2.8. La Figura 5 ilustra la arquitectura completa del sistema, y la Tabla 9 resume los hiperparámetros de entrenamiento.

Figura *5*

Arquitectura del modelo Two Towers: torres de consulta y de candidatos que proyectan a un espacio compartido de 128 dimensiones, acopladas por producto punto con softmax y corrección LogQ.

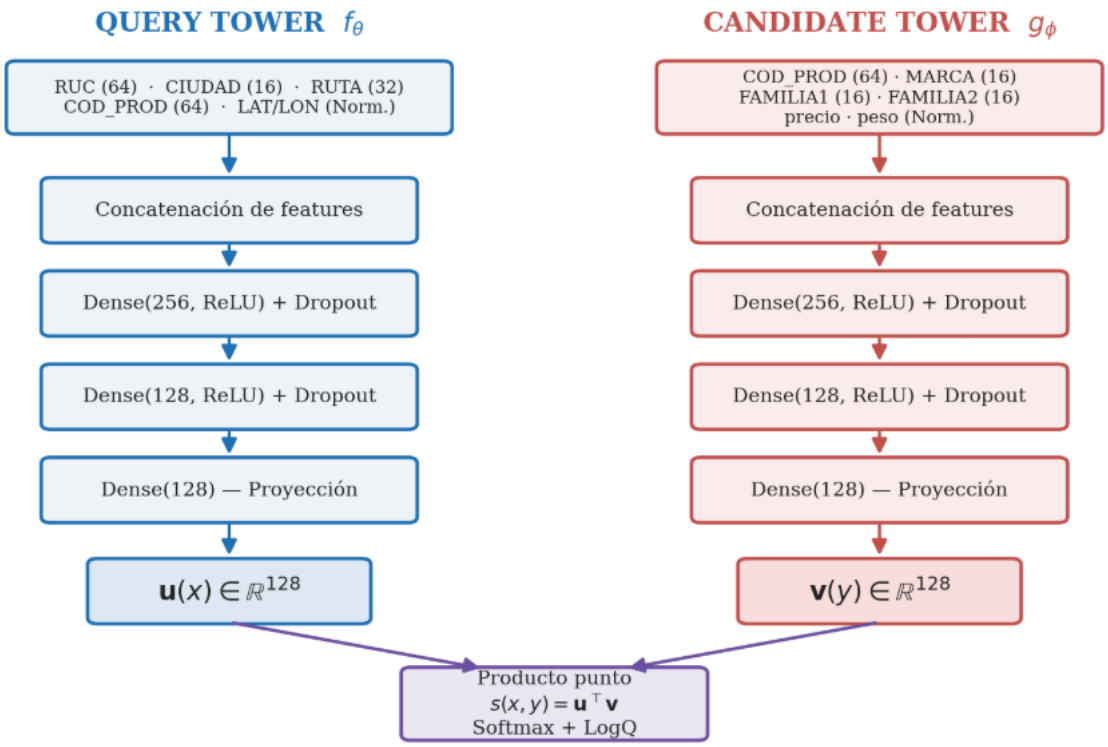


Figura. Arquitectura del modelo Two Towers para recuperación de candidatos

Tabla *9*

Hiperparámetros de entrenamiento del modelo Two Towers

Hiperparámetro	Valor
Dimensión del espacio compartido (D)	128
Optimizador	Adam
Tasa de aprendizaje	0,001
Tamaño de lote (batch size)	1.024
Estrategia de negativos	In-batch negatives
Corrección de sesgo	LogQ (sampling-bias correction)
Función de pérdida	Entropía cruzada categórica (softmax)
Normalización de la salida	L2 en ambas torres (similitud coseno)
Temperatura de la pérdida	0,05
Enmascarado de falsos negativos	Activado
Regularización	Dropout con tasa 0,0
Volumen de entrenamiento	32.501.962 pares (100 %)
Número de épocas	Parada temprana sobre Recall@100

Activación	ReLU (capas ocultas)
Framework	TensorFlow 2.21 + TFRS 0.7.2; Polars 1.41

Nota. Configuración del cuaderno 06_training_logq.ipynb. Los valores de normalización, temperatura, enmascarado de falsos negativos, regularización y número de épocas proceden de la búsqueda documentada en la sección 3.2.8; el criterio de parada es el Recall@100 medido sobre el catálogo completo.

Una vez entrenado el modelo, se precomputan en lote los embeddings de los 20.683 productos del catálogo y se serializan en formato NumPy (candidate_embeddings.npy y candidate_ids.npy) para su indexación ANN offline. Sobre esos artefactos se monta la simulación de servido analizada en el Capítulo 4.

3.2.8 Selección de hiperparámetros

Los hiperparámetros que no provienen de la literatura se determinaron mediante una búsqueda sistemática sobre el 25 % de los pares de entrenamiento, implementada en el script 16_hyperparameter_selection.py. La búsqueda se organizó de forma secuencial por grupos: primero los factores que gobiernan el muestreo de negativos y después los que definen la función de similitud. La configuración resultante se entrenó luego sobre la totalidad de los pares.

La métrica de selección es el Recall@100 medido contra el catálogo completo bajo similitud coseno. Esta elección obedece a la arquitectura de servido y no a los resultados obtenidos: el índice vectorial que recupera candidatos en producción opera sobre embeddings normalizados, de modo que evaluar con una similitud distinta de la que se sirve mediría un sistema que no existe en operación. Por la misma razón, la parada temprana se aplica sobre esa métrica y no sobre la pérdida in-batch, cuyo valor depende de qué otros productos compartieron lote con cada ejemplo y no informa sobre la capacidad de recuperación.

La primera fase contrastó el tamaño de lote, el enmascarado de falsos negativos y la tasa de dropout. Con negativos in-batch el tamaño de lote determina cuántos negativos recibe cada

positivo, y el enmascarado impide que un producto que es positivo para una consulta se emplee como negativo de otra dentro del mismo lote, una situación frecuente cuando la distribución de ventas está tan concentrada como en el sector ferretero. De las ocho combinaciones evaluadas resultó seleccionada la de lote 1.024, enmascarado activo y ausencia de dropout, con un Recall@100 de 20,60 % sobre el subconjunto de selección, frente al 12,85 % de la configuración de partida. Estos valores sirven para ordenar configuraciones entre sí y no son comparables con los del modelo final, entrenado después sobre la totalidad de los pares. Ese resultado precisa el alcance de lo que declaran las secciones 3.2.5 y 3.2.6: la capa de Dropout forma parte de la arquitectura de ambas torres y se conserva en el modelo final, y lo que la búsqueda seleccionó es una tasa de 0,0, de modo que la capa permanece sin descartar unidades durante el entrenamiento. Los prototipos de ambas torres se habían escrito con una tasa por defecto de 0,2, valor que la fase A descartó.

La segunda fase abordó la parametrización de la similitud. Al normalizar la salida de las torres, los logits quedan acotados en el intervalo de menos uno a uno, por lo que el softmax necesita un factor de temperatura para poder discriminar entre candidatos. El barrido lo confirma: sin temperatura el Recall@100 cae a 12,25 %, mientras que con temperaturas de 0,20, 0,10 y 0,05 asciende a 29,00 %, 32,10 % y 32,50 % respectivamente. La temperatura no es, por tanto, un ajuste opcional sino una consecuencia de la normalización. Se adoptó el valor de 0,05, si bien su diferencia con 0,10 no resulta significativa.

La precisión de la propia búsqueda acota qué diferencias pueden interpretarse. Con una muestra de validación de n = 2.000 consultas y una proporción de acierto p próxima al 30 %, el margen de muestreo al 95 % se obtiene por aproximación normal:

$$e = z \cdot \sqrt{\frac{p(1-p)}{n}} = 1{,}96 \cdot \sqrt{\frac{0{,}3305 \times 0{,}6695}{2.000}} = 0{,}0206 \qquad (3.15)$$

El error estándar resulta de aproximadamente un punto porcentual, de modo que las diferencias inferiores a dos puntos no se distinguen del ruido de muestreo. Este cálculo sirve como cota a priori del ruido sobre una proporción única y es independiente del procedimiento bootstrap de la sección 3.1.5, que se emplea para contrastar modelos entre sí sin asumir normalidad. Las configuraciones se seleccionaron atendiendo a esa restricción, y las diferencias que caen dentro de ese margen se declaran equivalentes en lugar de presentarse como mejoras.

3.2.9 Justificación de la selección y descarte de variables

La selección de variables se apoyó en el análisis de nulos del EDA y en la representatividad de cada atributo para la tarea de complementariedad. La Tabla 10 documenta las decisiones de descarte. El campo categoría se eliminó por estar 100 % vacío y fue sustituido por familia1 y familia2; el código de barras ean13 se descartó por su elevado porcentaje de nulos (50,47 % en catálogo), siendo COD_PROD suficiente como identificador; y las variables demográficas individuales (sexo, estado civil, origen de ingresos) se excluyeron por su escasa representatividad en un contexto B2B, donde el cliente es una ferretería o un contratista y no un consumidor final.

Tabla *10*

Variables descartadas y su justificación

Variable	Motivo de descarte
categoría	Campo 100 % vacío; sustituido por familia1 y familia2.
ean13	50,47 % de nulos en catálogo; COD_PROD es identificador suficiente.
sexo, estado civil, origen de ingresos	Baja representatividad en un contexto B2B (cliente institucional).
familia3	97,72 % de nulos; granularidad inutilizable.

Nota. Decisiones derivadas del EDA (cuaderno 02) y de la naturaleza B2B del dominio.

CAPÍTULO 4: ANÁLISIS DE RESULTADOS

Este capítulo reúne los resultados de la evaluación offline del modelo Two Towers. Primero se lo confronta con las líneas base (4.1.2). Luego se examina cómo la corrección LogQ afecta la diversidad y la cobertura del catálogo (4.1.3), y se comprueba de forma cualitativa la semántica de los embeddings aprendidos (4.1.4). Los apartados siguientes reportan la ablación de atributos y el arranque en frío (4.1.5), la viabilidad del servido mediante indexación ANN (4.1.6) y un análisis cualitativo de recomendaciones con sus implicaciones de negocio (4.2.1). El capítulo cierra discutiendo los hallazgos frente a la hipótesis (4.2.2). Todas las métricas se calcularon sobre el conjunto de validación temporal correspondiente al primer trimestre de 2026.

4.1 Pruebas de concepto

4.1.1 Protocolo de evaluación

La evaluación se realizó exclusivamente sobre el conjunto de validación temporal, formado por pares de co-compra del primer trimestre de 2026 que el modelo nunca observó durante el entrenamiento. Cada consulta queda definida por un cliente, su contexto y un producto-semilla. Para cada una se recuperaron los top-K candidatos del catálogo completo (20.683 productos) y se los contrastó con los productos que ese cliente efectivamente co-compró en la ventana de validación, es decir, el ground truth. Las métricas Recall@10, Recall@50, Recall@100 y MRR se promediaron sobre una muestra representativa de 2.000 consultas del conjunto de validación. Ese tamaño fija el margen de muestreo en aproximadamente dos puntos porcentuales al 95 %, criterio que las secciones 3.2.8 y 4.1.5.1 aplican para declarar equivalentes las diferencias que caen por debajo. Los tres modelos se sometieron al mismo conjunto de consultas y al mismo ground truth, de modo que la comparación es directa. Al apoyarse en una partición temporal estricta, este protocolo resulta más exigente y más realista que la validación aleatoria: mide si el modelo anticipa compras futuras, no si reconstruye interacciones ya vistas.

4.1.2 Comparativa frente a las líneas base

El modelo Two Towers con corrección LogQ se comparó contra dos líneas base. La primera es un recomendador de popularidad global, que devuelve los productos más vendidos. La segunda es el método clásico de co-ocurrencia ítem-ítem, que cuenta las co-compras condicionadas al producto-semilla y rellena con popularidad cuando falta señal (sección 3.1.6). La Tabla 11 resume el desempeño de los tres en recuperación y ranking sobre el conjunto de validación.

Tabla *11*

Comparativa de desempeño del modelo Two Towers frente a las líneas base (validación 2026 Q1)

Modelo	Recall@10	Recall@50	Recall@100	MRR
Popularidad global	3,90 %	13,20 %	21,25 %	0,0219
Co-ocurrencia ítem-ítem	6,90 %	16,85 %	24,05 %	0,0389
Two Towers (LogQ)	7,40 %	21,50 %	33,05 %	0,0354

Nota. Resultados de los cuadernos 12_extended_metrics_and_ci.ipynb y 14_cooccurrence_baseline.ipynb (misma muestra y protocolo del cuaderno 07_offline_evaluation.ipynb). El mejor valor corresponde al Two Towers en Recall@10, Recall@50 y Recall@100, y a la co-ocurrencia ítem-ítem en MRR. En Recall@10 y en MRR las diferencias no son concluyentes: los intervalos de confianza bootstrap al 95 % se solapan.

El Two Towers encabeza las métricas de recuperación amplia. Frente a la popularidad global, mejora el Recall@100 en 11,8 puntos, de 21,25 % a 33,05 %, un aumento relativo del 55,5 %, y eleva el MRR un 61,6 %. Frente a la co-ocurrencia ítem-ítem, el método clásico de referencia para complementariedad, la ventaja se encuentra en el presupuesto amplio de recuperación: 9,0 puntos porcentuales más de Recall@100, 33,05 % contra 24,05 %, y 4,65 puntos más de Recall@50. La diferencia en Recall@100 es estadísticamente significativa: el intervalo de confianza bootstrap al 95 % del Two Towers, de 30,90 % a 35,10 %, no se solapa con el de la co-ocurrencia, de 22,30 % a 26,00 %, lo que permite rechazar la hipótesis nula (H0).

En el tramo corto del ranking, en cambio, ninguna diferencia resulta concluyente. El modelo aventaja nominalmente al baseline en Recall@10, 7,40 % contra 6,90 %, con intervalos de 6,25 % a 8,55 % y de 5,80 % a 8,00 %. El baseline conserva una ligera ventaja nominal en MRR, 0,0389 contra 0,0354, con intervalos de 0,0324 a 0,0455 y de 0,0304 a 0,0408. Los dos pares se solapan, de modo que el criterio de la sección 3.1.5 no permite pronunciarse.

El patrón responde al rol de cada método. La co-ocurrencia explota pares frecuentes ya observados y por eso destaca en la cabeza del ranking. El modelo neuronal generaliza mediante representaciones densas y recupera complementos que nunca se co-compraron antes, y así amplía la cobertura del top-100, que constituye el presupuesto operativo de una etapa de recuperación de candidatos. Las Figuras 6 y 7 muestran, respectivamente, la comparativa de Recall@K y de MRR entre los tres modelos.

Figura *6*

Comparativa de Recall@K (K = 10, 50, 100) entre el modelo Two Towers y las líneas base de popularidad global y co-ocurrencia ítem-ítem.

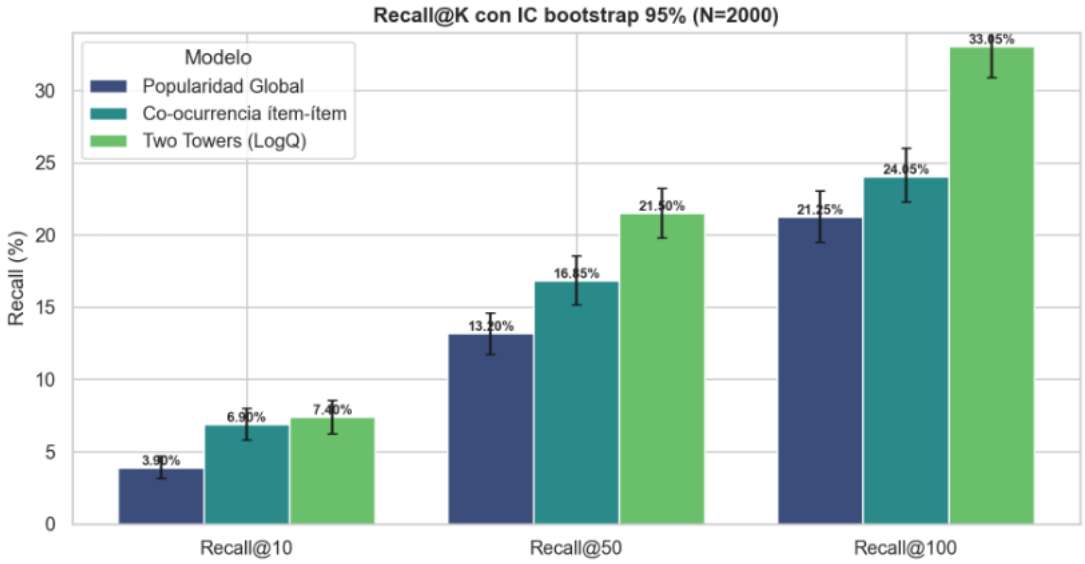
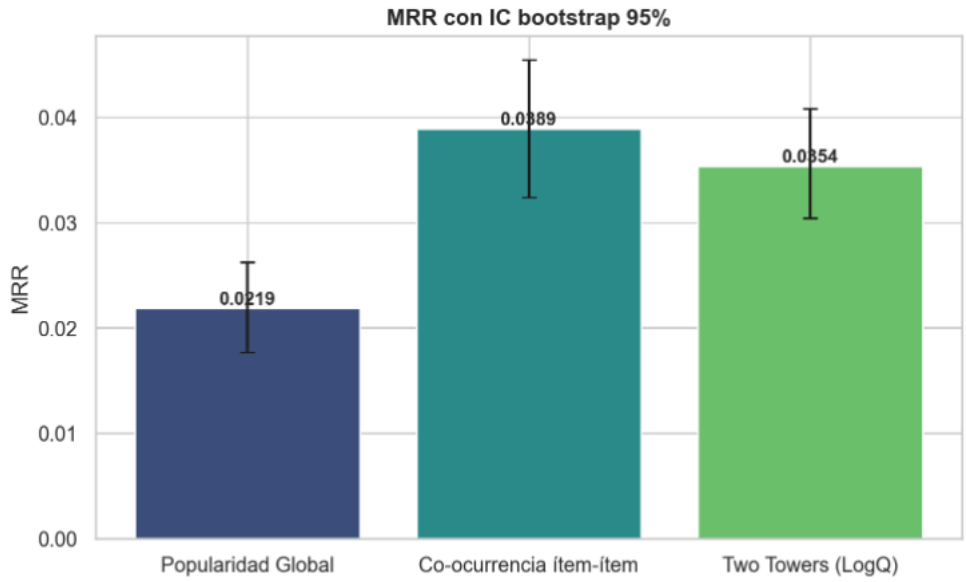


Figura *7*
Comparativa de Mean Reciprocal Rank (MRR) entre los tres modelos evaluados.



La magnitud absoluta de las métricas debe interpretarse en función de la dificultad de la tarea. Se trata de recuperar, entre 20.683 candidatos posibles, los pocos productos que el cliente co-comprará en un futuro, partiendo de un único producto semilla. Recuperar un tercio de los complementos reales dentro de las primeras 100 posiciones (Recall@100 = 33,05 %) es un resultado operativamente valioso para una primera etapa, sobre la que un ranking fino posterior podría afinar las sugerencias.

4.1.3 Impacto de la Corrección LogQ: Diversidad y Cobertura del Catálogo

Para aislar el efecto de la corrección LogQ se entrenó una variante del modelo sin ella, con negativos in-batch puros, y se contrastaron ambas versiones en dos planos: el rendimiento predictivo y la diversidad de las recomendaciones. Ambas variantes se entrenaron sobre la totalidad de los pares y con idéntica configuración, semilla y criterio de parada, de modo que la única diferencia entre ellas es si la pérdida recibe la probabilidad de muestreo de cada candidato.

Toda diferencia observada es, por tanto, atribuible a la corrección y no al volumen de datos ni a la parametrización. La Tabla 12 recoge el rendimiento predictivo; la Tabla 13, las métricas de diversidad, esto es, la cobertura de catálogo y el coeficiente de Gini sobre el top 50.

Tabla *12*

Rendimiento predictivo: modelo con y sin corrección LogQ (validación 2026 Q1)

Métrica	Two Towers con LogQ	Two Towers sin LogQ
Recall@10	7,10 %	1,60 %
Recall@50	22,80 %	6,10 %
Recall@100	33,10 %	10,20 %
MRR	0,0329	0,0082

Nota. Ambas variantes se entrenaron sobre la totalidad de los pares (32.501.962) con idéntica arquitectura, configuración, semilla y criterio de parada, y difieren únicamente en la corrección LogQ. Las cifras proceden de un entrenamiento independiente del que produjo el modelo final, por lo que la variante corregida difiere de la Tabla 11 dentro del margen de variación entre corridas.

Tabla *13*

Métricas de diversidad y cobertura (top-50): modelo con y sin corrección LogQ

Métrica	Two Towers con LogQ	Two Towers sin LogQ
Cobertura de catálogo	20,09 %	62,11 %
Coeficiente de Gini	0,9650	0,6982

Nota. Cobertura = fracción del catálogo recomendada al menos una vez en el top-50. El Gini se calcula sobre la frecuencia de recomendación. Los valores altos indican mayor concentración.

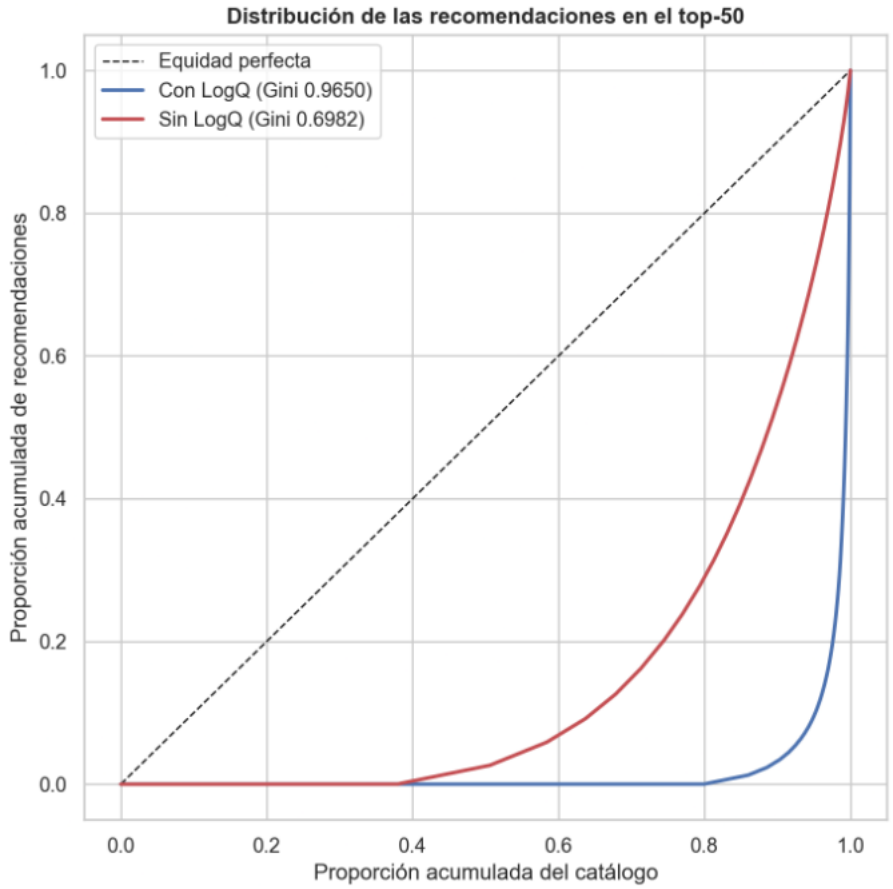
En rendimiento predictivo (Tabla 12), el modelo con corrección LogQ supera ampliamente a la variante sin corrección: triplica su Recall@100 (33,10 % contra 10,20 %) y cuadruplica su MRR (0,0329 contra 0,0082). Las métricas de diversidad (Tabla 13) apuntan en sentido contrario: el modelo sin LogQ cubre tres veces más catálogo (62,11 % contra 20,09 %) y presenta un Gini sensiblemente más bajo (0,6982 contra 0,9650), lo que en apariencia sugeriría recomendaciones más repartidas.

Esa mayor cobertura no puede leerse como una ventaja. Las dos familias de métricas describen fenómenos distintos y deben interpretarse en conjunto: alcanzar una porción mayor del catálogo solo constituye un descubrimiento útil si el modelo conserva la capacidad de recuperar los productos que el cliente efectivamente co-compra. La variante sin corrección recupera menos de un tercio de los complementos reales que recupera el modelo corregido, de modo que su dispersión sobre el catálogo se reparte en su mayor parte entre productos que no fueron adquiridos. La Figura 8 ilustra esta dinámica mediante las curvas de Lorenz de ambas variantes.

El compromiso entre relevancia y cobertura es, por tanto, un efecto atribuible a la corrección y no un artefacto del protocolo experimental, puesto que ambas variantes recibieron presupuesto, configuración y criterio de parada idénticos. Establecer el mecanismo por el cual la ausencia de corrección desplaza las recomendaciones hacia la cola del catálogo excede el alcance de este experimento, que cuantifica el efecto pero no lo descompone; su determinación se plantea como línea de trabajo futuro (sección 5.2).

Figura *8*

Curvas de Lorenz de la distribución de recomendaciones en el top-50 para los modelos con y sin corrección LogQ; la diagonal representa la equidad perfecta.



Se concluye que, dentro del marco de validación temporal de este estudio, la corrección LogQ es indispensable para la calidad de la recuperación. Calibrar el compromiso entre precisión y cobertura (cuánta diversidad conviene ceder a cambio de relevancia) queda como línea de trabajo futuro (ver sección 5.2).

4.1.4 Validación Cualitativa de la Semántica de los Embeddings

Un modelo de recuperación no se juzga solo por sus números. El espacio latente que aprende debe guardar alguna correspondencia con lo que se sabe del dominio. Para examinarlo, los embeddings del catálogo completo se proyectaron a dos dimensiones con t-SNE (Van der Maaten y Hinton, 2008) y UMAP (McInnes et al., 2018), y se calcularon similitudes coseno sobre los embeddings originales entre grupos de productos representativos de tres áreas ferreteras. Las

Figuras 9 y 10 muestran agrupamientos parciales por familia comercial, sin que estos lleguen a constituir regiones netamente separadas en todo el catálogo. t-SNE y UMAP son técnicas de reducción de dimensionalidad no lineales que preservan la estructura local; sus distancias globales y los tamaños de los clústeres en el plano no son cuantitativamente interpretables y dependen de hiperparámetros como la perplejidad. Por ese motivo la lectura visual se contrasta con las medidas de similitud calculadas sobre los embeddings originales.

Figura *9*

Proyección t-SNE de los embeddings del catálogo, coloreada por familia comercial; se observan agrupamientos parciales por categoría.

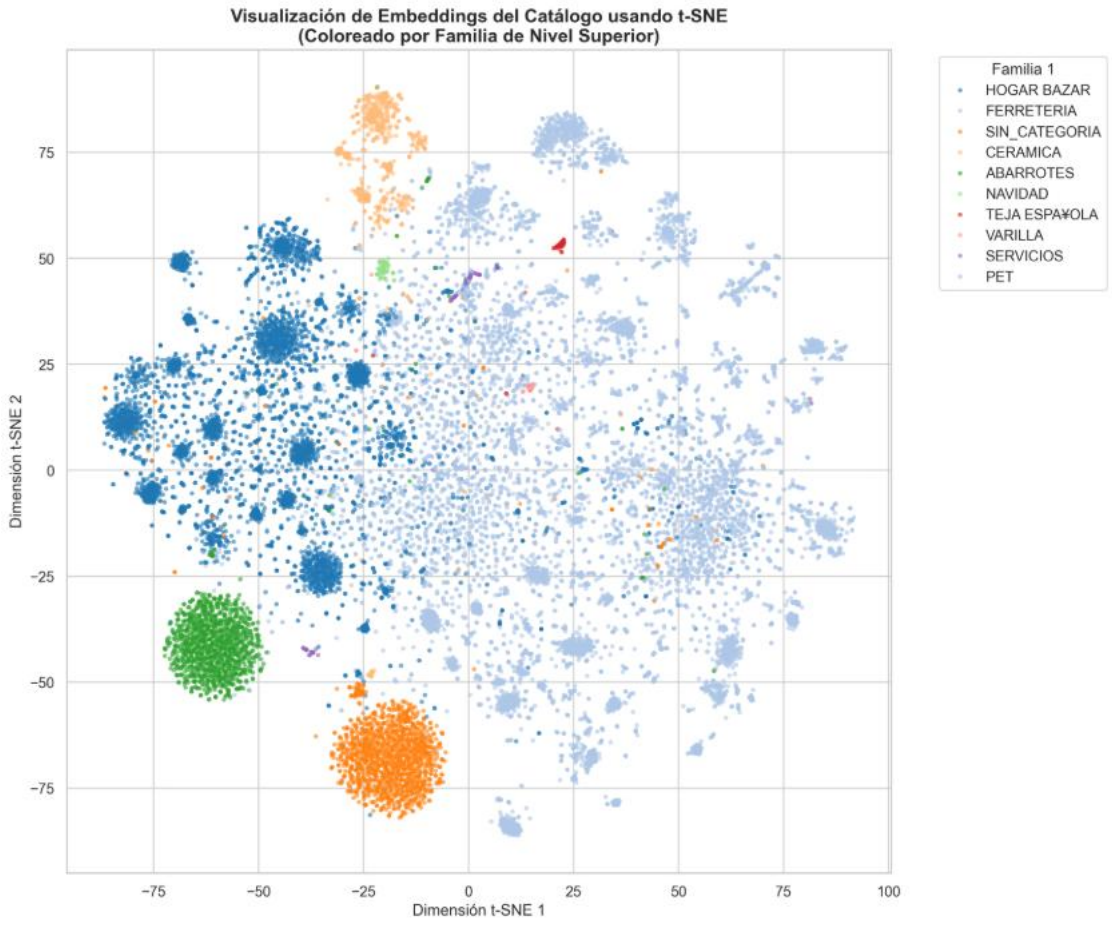


Figura *10*

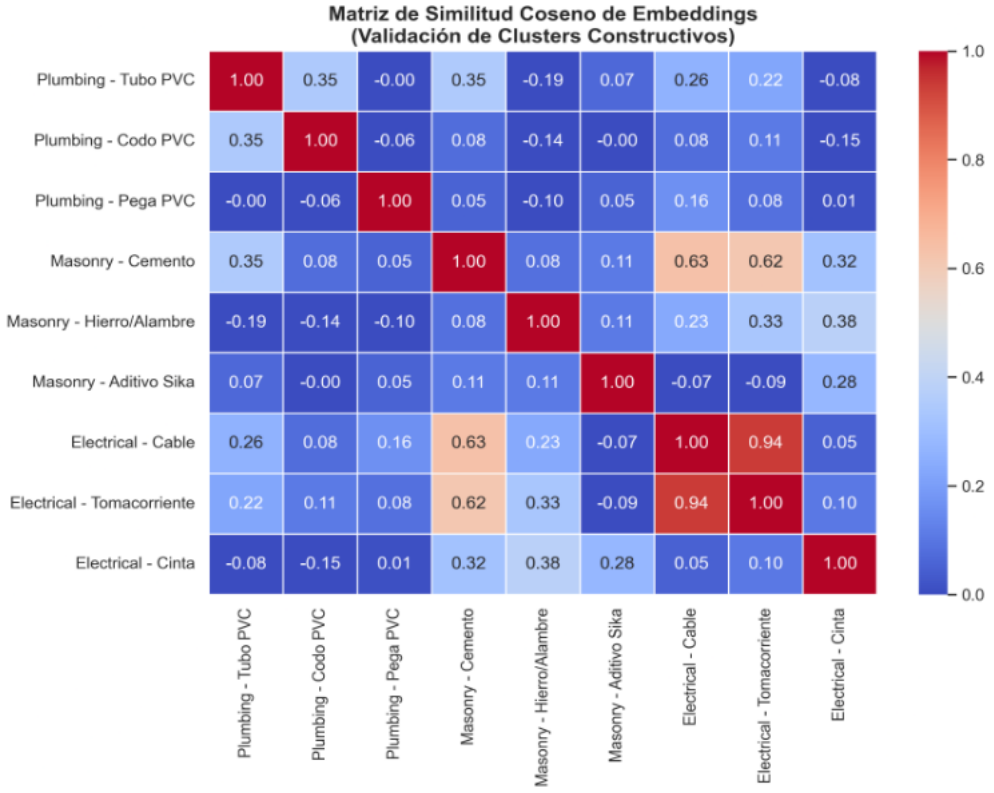
Proyección UMAP de los embeddings del catálogo; la estructura global reproduce parcialmente la separación entre familias constructivas.

Para cuantificar esa lectura se calcularon las similitudes coseno intra e interfamiliar entre productos representativos de tres áreas ferreteras: plomería, obra gris y material eléctrico. La Figura 11 presenta la matriz resultante, y sus valores admiten una interpretación matizada. Material eléctrico es la única de las tres áreas cuya similitud interna (0,3629) supera con holgura el promedio entre áreas (0,1285); plomería (0,0943) y obra gris (0,1001) quedan por debajo de ese promedio, de modo que sus productos representativos no conforman agrupamientos compactos en el espacio latente. Destaca, en cambio, la similitud entre obra gris y material eléctrico (0,2913), la mayor de todas las combinaciones entre áreas. Una explicación plausible se encuentra en la lógica del negocio: durante la cimentación y la obra gris las canastas incorporan de forma recurrente material de canalización eléctrica empotrada, de manera que el modelo aproxima productos que se

adquieren juntos aunque pertenezcan a familias distintas. Debe precisarse el alcance de esta medida: se calcula sobre nueve productos seleccionados mediante expresiones aplicadas a la descripción comercial, y algunos de ellos no son ejemplares típicos de su área, por lo que constituye una comprobación ilustrativa y no una caracterización del espacio latente en su conjunto.

Figura *11*

Matriz de similitud coseno intra e inter-familia para las principales familias constructivas (plomería, obra gris, material eléctrico).



4.1.5 Experimentos de Ablación y Arranque en Frío

4.1.5.1 Estudio de ablación por enmascaramiento de atributos

Para medir cuánto aporta cada bloque de características de la torre de consulta se hizo una ablación por enmascaramiento en inferencia. Se anularon de forma selectiva tres señales: la identidad del cliente (el RUC, sustituido por un token fuera de vocabulario), la geolocalización

(LATITUD y LONGITUD, fijadas en la media normalizada) y el producto-semilla (COD_PROD, sustituido asimismo por un token fuera de vocabulario). La Tabla 14 recoge la degradación que produce cada supresión.

Tabla *14*

Estudio de ablación por enmascaramiento de atributos de la Query Tower

Métrica	Modelo completo	Sin identidad (No RUC)	Sin geolocalización	Sin producto-semilla
Recall@10	7,40 %	5,25 %	1,20 %	7,15 %
Recall@50	21,50 %	15,45 %	6,75 %	20,80 %
Recall@100	33,05 %	23,85 %	14,10 %	32,40 %
MRR	0,0354	0,0265	0,0079	0,0321

Nota. Resultados del cuaderno 09_ablation_and_diversity.ipynb. El enmascaramiento se aplica en inferencia sobre el modelo entrenado completo; no implica reentrenamiento.

El resultado deja un hallazgo claro: la geolocalización es la señal de contexto más determinante. Enmascararla hunde el Recall@100 de 33,05 % a 14,10 %, una caída del 57,3 %, muy por encima de la que provoca eliminar la identidad del cliente (de 33,05 % a 23,85 %, un descenso del 27,8 %). El dato encaja con la estructura del negocio. La ubicación y la ruta de distribución codifican de forma implícita el tipo de zona urbana, periurbana o rural y, con ellas, los patrones de obra dominantes. Ese contexto predice la complementariedad con más estabilidad que la identidad individual de cada cliente.

El tercer enmascaramiento arroja el resultado más relevante para interpretar el sistema. Suprimir el producto-semilla, esto es, la información sobre desde qué artículo se formula la consulta, reduce el Recall@100 de 33,05 % a 32,40 %: una caída de 0,65 puntos, equivalente al 2,0 % en términos relativos, que queda dentro del margen de error del muestreo, estimado en unos dos puntos porcentuales para una muestra de 2.000 consultas. El modelo recupera, en la práctica, un conjunto de candidatos que apenas cambia con el producto desde el que se consulte.

Este hallazgo obliga a precisar qué aprende el sistema. Las señales que gobiernan la recuperación son el contexto geográfico y, en segundo término, la identidad del cliente; no la relación entre el producto-semilla y sus complementos técnicos. El modelo anticipa con buen desempeño qué artículos adquirirá un cliente determinado en una zona determinada, pero no diferencia las distintas necesidades de complementariedad que ese mismo cliente tendría al partir de un saco de cemento o de un rollo de cable eléctrico. La sección 4.2.1 examina las consecuencias de esta observación sobre la lectura cualitativa de las recomendaciones, y la sección 4.2.2 sobre el alcance de la hipótesis contrastada.

4.1.5.2 Comportamiento ante el arranque en frío (cold-start)

También se puso a prueba la robustez del modelo frente a entidades que no vio durante el entrenamiento. En validación se identificaron 776 clientes fríos (9,41 %, con 78.086 consultas asociadas) y 483 productos fríos (7,34 %, con 82.384 consultas). Las Tablas 15 y 16 comparan el Two Towers con el baseline de co-ocurrencia ítem-ítem en ambos escenarios. Conviene recordar que ese baseline es agnóstico a la identidad del cliente por lo que no se degrada ante clientes nuevos, pero depende por completo de haber observado co-compras del producto durante el entrenamiento.

Tabla *15*

Arranque en frío de clientes (cold users): Two Towers frente a co-ocurrencia ítem-ítem

Métrica	Co-ocurrencia ítem-ítem	Two Towers (LogQ)
Recall@10	9,80 %	7,70 %
Recall@50	18,70 %	17,40 %
Recall@100	26,30 %	27,20 %
MRR	0,0575	0,0368

Nota. Subconjunto de 776 clientes no vistos en entrenamiento (78.086 consultas). Cuadernos 10_cold_start_evaluation.ipynb y 14_cooccurrence_baseline.ipynb, de donde procede la columna de co-ocurrencia.

Tabla *16*

Arranque en frío de productos (cold items): Two Towers frente a co-ocurrencia ítem-ítem

Métrica	Co-ocurrencia ítem-ítem	Two Towers (LogQ)
Recall@10	0,00 %	0,00 %
Recall@50	0,00 %	0,40 %
Recall@100	0,00 %	0,70 %
MRR	0,0001	0,0006

Nota. Subconjunto de 483 productos no vistos en entrenamiento (82.384 consultas). Cuadernos 10 y 14.

El arranque en frío de clientes (Tabla 15) matiza las fortalezas de cada enfoque. Para clientes nunca vistos, el Two Towers mantiene un Recall@100 del 27,20 %, apreciablemente por debajo de su desempeño global, aunque próximo al de la co-ocurrencia ítem-ítem (26,30 %), cuyo intervalo bootstrap, de 23,60 % a 29,20 %, contiene esa estimación puntual; el experimento de arranque en frío no calculó intervalos para el modelo neuronal, así que la cercanía se reporta como tal y no como equivalencia. Lo consigue porque las torres pueden ubicar a un cliente nuevo en el espacio latente a partir de su contexto (ciudad, ruta, geolocalización) sin necesitar su historial individual. La co-ocurrencia, al ignorar la identidad del cliente, tampoco se degrada y supera nominalmente al modelo en el tramo corto (Recall@10 de 9,80 % contra 7,70 %). La diferencia de fondo es otra: el modelo neuronal consigue esa robustez sin renunciar a personalizar por contexto, mientras que el baseline entrega el mismo ranking a todo cliente que parta de la misma semilla.

Con productos fríos (Tabla 16) el panorama es duro para ambos. Aun así, el Two Towers conserva una capacidad diferente a cero de recomendación (Recall@100 = 0,70 %) gracias a los atributos de contenido de la torre de candidatos (marca, familias, precio, peso), frente al colapso total de la co-ocurrencia (0,00 %), que se queda sin señal para productos que nunca se co-compraron. Ese 0,70 % significa que, para un SKU recién incorporado y sin historial de co-compra, el sistema apenas recupera el ítem relevante, de modo que no puede sostener por sí solo la recomendación de novedades; en un escenario productivo, esos productos requerirían una

estrategia de respaldo (reglas de negocio, popularidad por familia o exploración controlada) hasta acumular suficientes interacciones.

4.1.6 Simulación del Entorno de Servido: Latencia frente a Precisión (ANN)

Por último se comprobó la viabilidad de despliegue simulando el servido a escala. Se enfrentó la búsqueda exacta (Brute Force, faiss.IndexFlatIP) a la indexación aproximada HNSW de FAISS, variando el parámetro de profundidad de búsqueda efSearch. La métrica de calidad es el solapamiento del top 100 recuperado respecto de la búsqueda exacta. La Tabla 17 y la Figura 12 recogen los resultados.

Tabla *17*

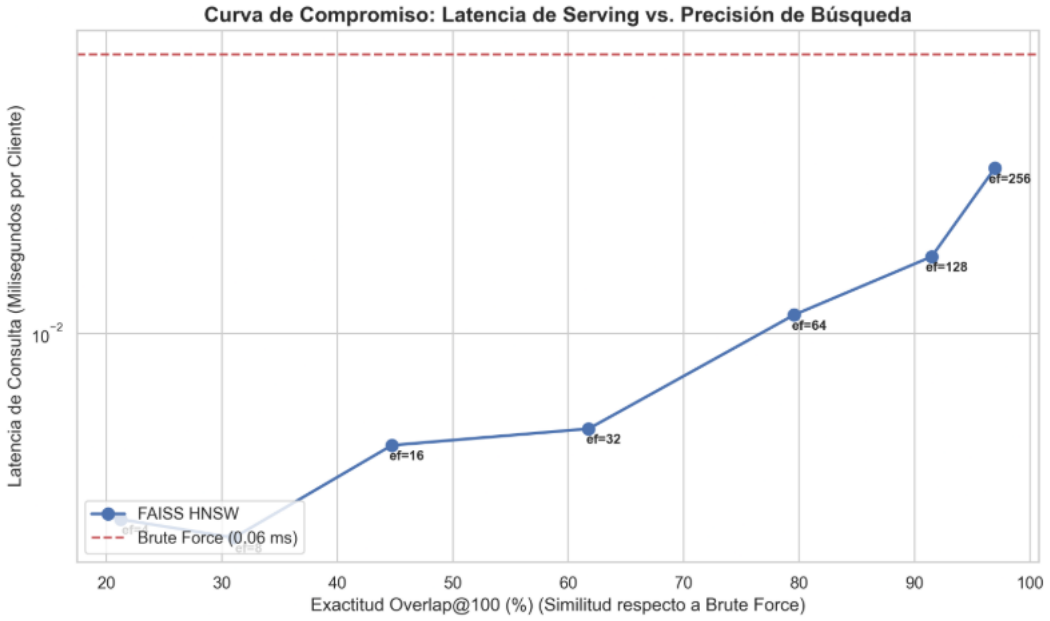
Compromiso latencia-precisión: búsqueda exacta frente a HNSW (FAISS)

Configuración	Latencia por consulta (ms)	Exactitud Overlap@100
Brute Force (exacta)	0,06416	100,00 %
HNSW (efSearch = 4)	0,00290	21,23 %
HNSW (efSearch = 8)	0,00257	31,03 %
HNSW (efSearch = 16)	0,00475	44,73 %
HNSW (efSearch = 32)	0,00530	61,77 %
HNSW (efSearch = 64)	0,01133	79,58 %
HNSW (efSearch = 128)	0,01669	91,48 %
HNSW (efSearch = 256)	0,03009	96,96 %

Nota. Resultados del cuaderno 11_ann_latency_vs_recall.ipynb sobre los 20.683 embeddings precomputados. El índice HNSW se construyó en 0,30 s. Overlap@100 = coincidencia del top-100 frente a la búsqueda exacta.

Figura *12*

Curva de compromiso entre latencia de consulta (ms) y exactitud de recuperación (Overlap@100) para distintas configuraciones del índice HNSW.



Los números confirman que la arquitectura es operativamente viable. Con efSearch = 128, el índice HNSW recupera el 91,48 % de la exactitud de la búsqueda exacta con una latencia de 0,0167 ms por consulta, y con efSearch = 256 llega al 96,96 %. La curva sigue el comportamiento logarítmico previsto: aumentar el efSearch mejora la exactitud a un costo de latencia marginal. A la escala de este catálogo, no obstante, la indexación aproximada no resulta necesaria: sobre 20.683 candidatos la búsqueda exacta requiere 0,0642 ms por consulta, tres órdenes de magnitud por debajo del umbral de servido considerado en el diseño. El aporte del experimento no es, por tanto, justificar el uso de un índice aproximado en este despliegue, sino caracterizar el compromiso entre latencia y exactitud para catálogos de mayor tamaño. HNSW mantiene latencias sublineales a medida que crece el catálogo, y eso garantiza recomendaciones en tiempo real conforme la empresa amplíe su inventario.

4.2 Análisis de resultados

4.2.1 Análisis Cualitativo de Recomendaciones e Implicaciones Empresariales

Las métricas agregadas cuentan una parte de la historia; el comportamiento del sistema se aprecia mejor en recomendaciones concretas. La Tabla 18 recoge la salida literal del modelo para cinco productos-semilla de áreas distintas, manteniendo fijo el contexto del cliente.

Tabla *18*

Salida del modelo para cinco productos-semilla con contexto de cliente fijo

Producto-semilla (consulta)	Productos recuperados por el modelo
Cemento blanco, funda de 5 kg	Viruta de limpieza n.º 6, sellador catalizado, viruta n.º 5, anticorrosivo negro mate
Tubo PVC a presión, cédula 40, 2″ × 6 m	Viruta de limpieza n.º 6, viruta n.º 5, anticorrosivo negro mate, sellador catalizado
Varilla corrugada 28 mm × 12 m	Viruta de limpieza n.º 6, sellador catalizado, viruta n.º 5, anticorrosivo negro mate
Cable USB 3 en 1	Viruta de limpieza n.º 6, sellador catalizado, viruta n.º 5, anticorrosivo negro mate
Pintura látex, galón	Sellador catalizado en galón, viruta n.º 6, sellador catalizado en litro, anticorrosivo negro mate

Nota. Recomendaciones efectivamente producidas por el modelo (cuaderno 13_topk_qualitative), con RUC, ciudad, ruta y coordenadas constantes; solo varía el producto-semilla. Se muestran las cuatro primeras posiciones de cada lista.

La tabla confirma en el plano cualitativo lo que la ablación de la sección 4.1.5.1 mide en el cuantitativo. Las cinco listas comparten la mayor parte de sus posiciones pese a partir de artículos de áreas distintas, y ninguna guarda una relación técnica evidente con su semilla: para una varilla corrugada el sistema propone viruta de limpieza y sellador catalizado. Una comprobación sistemática sobre cuarenta semillas sorteadas entre los productos observados en el entrenamiento, manteniendo fijo el contexto, arroja un solapamiento medio del 79,1 % entre listas de diez posiciones, con un intervalo de confianza bootstrap al 95 % que va de 75,9 % a 82,1 %, obtenido remuestreando semillas y no pares. Las cuarenta listas se reparten en conjunto veinticinco productos distintos. Si el sorteo se extiende al catálogo completo, incluidos los artículos sin historial de compra, el solapamiento asciende al 86,0 %, porque un producto sin interacciones

aporta todavía menos señal a la consulta. El procedimiento queda documentado en 17_seed_overlap.

El sistema no opera, por tanto, como un recomendador de complementos condicionado al producto consultado, sino como un predictor contextual de la próxima compra del cliente. Esa capacidad posee valor operativo y es la que sustenta su ventaja sobre las líneas base en Recall@100, pero difiere de la que el planteamiento inicial le atribuía. La distinción resulta determinante para el despliegue: el sistema es apto para anticipar el pedido probable de un cliente en su zona, y no para responder a qué acompaña técnicamente a un artículo determinado.

En clave empresarial, y entendido como predictor de la próxima compra, el sistema abre tres propuestas de valor. La primera es el aumento del ticket promedio, mediante sugerencias anticipadas en el momento del pedido. La segunda es la caída de pedidos incompletos y viajes de compra fallidos, con la consiguiente mejora en la experiencia del cliente y en su fidelización. La tercera es una mejor rotación del inventario de media y baja frecuencia, al poner delante del cliente productos relevantes que hoy le pasan inadvertidos. A esto se suma una ventaja estratégica para la captación: como el modelo opera en frío para clientes nuevos (sección 4.1.5.2), puede ofrecer recomendaciones útiles desde la primera interacción, apoyado en el contexto geográfico y de ruta.

4.2.2 Discusión General y Contraste de la Hipótesis

Los resultados contrastan favorablemente la hipótesis en su dimensión central. El Two Towers superó a ambas líneas base en recuperación amplia, Recall@100 de 33,05 %, frente al 21,25 % de la popularidad global y al 24,05 % de la co-ocurrencia ítem-ítem, con intervalos de confianza del 95 % que no se solapan; y no arrojó diferencias concluyentes frente al baseline clásico en el tramo corto del ranking (Recall@10 y MRR). La arquitectura, por tanto, identifica

patrones complejos de co-compra en el dominio ferretero B2B y amplía la cobertura de complementos más allá de los pares ya observados. La validación cualitativa de los embeddings, la robustez ante el arranque en frío de clientes y la viabilidad del servido con indexación ANN apuntan, todas, en la misma dirección: la elección arquitectónica fue pertinente.

Ahora bien, en términos absolutos las métricas son modestas: un Recall@100 del 33,05 % y un MRR de 0,0354 quedan lejos de los valores que reportan los sistemas industriales sobre catálogos de consumo masivo (por ejemplo, el reporte de ingeniería de Ling et al., 2023). Esta brecha no invalida el hallazgo, pero lo sitúa: responde a la esparsidad extrema del dominio (98,78 %) y a la baja frecuencia de la compra ferretera, condiciones distintas a las de esos estudios. La ausencia de diferencia concluyente frente a la co-ocurrencia ítem-ítem en el tramo corto del ranking es un resultado a tener presente: en Recall@10 y en MRR los intervalos se solapan, de modo que la evidencia no acredita ventaja de la red neuronal sobre un baseline clásico y de bajo costo, lo que coincide con la advertencia de Ferrari Dacrema et al. (2019) sobre la dificultad de superar líneas base bien ajustadas. La ventaja del modelo neuronal se concentra, por tanto, en la recuperación amplia y en su capacidad de generalizar a complementos no observados, no en una superioridad uniforme. El arranque en frío de productos (0,70 % de capacidad residual) y el compromiso entre cobertura y relevancia de la corrección LogQ son limitaciones que este trabajo no resuelve.

Corresponde además delimitar con precisión qué quedó contrastado. La ablación de la sección 4.1.5.1 muestra que suprimir el producto-semilla apenas altera el desempeño, mientras que suprimir la geolocalización lo reduce a menos de la mitad. La hipótesis, tal como fue operacionalizada, compara la calidad de la recuperación frente a dos líneas base sobre el mismo conjunto de validación, y en ese contraste el modelo resulta superior de forma estadísticamente significativa. Lo que la evidencia no respalda es la interpretación de que esa superioridad provenga de haber aprendido relaciones de complementariedad técnica condicionadas al artículo

consultado: el sistema aventaja a las líneas base prediciendo qué comprará cada cliente en su contexto, no qué acompaña al producto desde el que se consulta.

Las dos afirmaciones anteriores pertenecen a planos de validez distintos, y mantenerlos separados evita que el respaldo de una se extienda a la otra. La validez de conclusión estadística está acreditada: la diferencia en Recall@100 frente a ambas líneas base es significativa al 95 % y se sostiene bajo el mismo protocolo de evaluación. La validez de constructo queda acotada: la métrica empleada no discrimina entre recuperar complementos técnicos y anticipar la canasta habitual del cliente, de modo que avala el desempeño del modelo como recuperador de candidatos, pero no la lectura de ese desempeño como complementariedad aprendida.

Esta precisión tiene una consecuencia metodológica que excede al caso. El Recall@100 sobre pares de co-compra premia por igual a un modelo que recupera complementos técnicos y a uno que anticipa la canasta habitual de un cliente, porque ambos aciertan sobre el mismo conjunto de productos efectivamente adquiridos. Una evaluación que pretenda medir complementariedad condicionada requiere, en consecuencia, un contraste explícito frente a la variante sin producto-semilla, del tipo que se incorpora en la Tabla 14.

CAPÍTULO 5: CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

Este Trabajo de Fin de Máster desarrolló y evaluó, de forma offline, un sistema de recomendación basado en la arquitectura de aprendizaje profundo Two Towers, orientado a recuperar productos complementarios en una empresa distribuidora del sector ferretero ecuatoriano. A partir de los objetivos planteados se extraen las siguientes conclusiones:

1. Se cumplió el objetivo de construcción del dataset. Se integraron y preprocesaron 5.876.800 transacciones de 14.935 clientes y 20.683 productos, y se generaron 32,5 millones de pares de co-compra con ventanas temporales de 30 días, bajo una partición temporal sin fuga de información que reproduce condiciones de producción.
2. Se diseñó e implementó con éxito la arquitectura Two Towers sobre TensorFlow Recommenders. El modelo integra variables categóricas de alta cardinalidad, variables numéricas y geográficas, y la corrección de sesgo de popularidad LogQ sobre negativos in-batch, y proyecta cliente y producto a un espacio compartido de 128 dimensiones.
3. El modelo superó a las líneas base en la métrica objetivo de la etapa de recuperación. Alcanzó un Recall@100 del 33,05 %, frente al 21,25 % de la popularidad global y al 24,05 % de la co-ocurrencia ítem-ítem, con diferencias estadísticamente significativas al 95 %. En el tramo corto del ranking las diferencias frente al baseline clásico no resultaron concluyentes (Recall@10 de 7,40 % contra 6,90 %; MRR de 0,0354 contra 0,0389). Esto confirma la hipótesis en la tarea de recuperación de candidatos: la arquitectura Two Towers mejora, de forma estadística y operativamente relevante, la recuperación medida sobre pares de co-compra. La sección 4.2.2 delimita el alcance de esa confirmación: acredita el desempeño del modelo como recuperador, no la interpretación de que haya aprendido complementariedad condicionada al artículo consultado.

4. El análisis cualitativo de los embeddings, con t-SNE, UMAP y similitudes coseno, mostró que las representaciones aprendidas reflejan de forma parcial la estructura de familias del catálogo y aproximan productos que se adquieren conjuntamente, como la obra gris y el material eléctrico durante la cimentación. La separación por familia, sin embargo, no es uniforme en todo el catálogo.
5. El modelo mostró robustez ante el arranque en frío de clientes: un Recall@100 del 27,20 % para clientes no vistos, por debajo de su desempeño global y próximo al del baseline de co-ocurrencia, que es agnóstico al cliente por construcción. Esa solidez procede de su naturaleza basada en contenido y contexto, y su ventaja diferencial es conservar la capacidad de personalización. La ablación, por su parte, señaló la geolocalización como la señal más determinante de la torre de consulta y, sobre todo, mostró que suprimir el producto-semilla apenas altera el desempeño: el sistema recupera en función del contexto del cliente antes que del artículo consultado.
6. La simulación de servido con indexación ANN (HNSW/FAISS) probó la viabilidad de despliegue: con efSearch = 128 se recupera el 91,48 % de la exactitud de la búsqueda exacta con una latencia de 0,0167 ms por consulta, y con efSearch = 256 se alcanza el 96,96 % con 0,0301 ms. A la escala de este catálogo la búsqueda exacta ya resulta operativamente suficiente (0,0642 ms por consulta), de modo que el aporte del experimento es caracterizar el compromiso entre latencia y exactitud para catálogos de mayor tamaño.
7. Dentro del alcance de la revisión bibliográfica efectuada no se identificaron aplicaciones previas de la arquitectura Two Towers a la distribución ferretera B2B ecuatoriana, de modo que el trabajo aporta evidencia sobre un vacío de dominio, de tipo de recomendación y geográfico identificado en la literatura.

En conjunto, la evidencia empírica respalda que la arquitectura Two Towers es una solución metodológicamente adecuada y operativamente viable para la recuperación de candidatos en distribución ferretera B2B, si bien la señal que el modelo aprovecha es el contexto del cliente antes que la relación de complementariedad con el artículo consultado. El aporte de esta investigación es de tres órdenes. En el plano conceptual, reformula la complementariedad ferretera como un problema de recuperación en un espacio semántico compartido, gobernado por lógica constructiva y no por preferencia subjetiva. En el plano metodológico, articula un protocolo reproducible de evaluación temporal con líneas base clásicas, contraste estadístico mediante intervalos de confianza bootstrap y control del sesgo de popularidad con la corrección LogQ. En el plano aplicado, demuestra la viabilidad de un recomendador desplegable con recursos de nivel académico sobre un catálogo real de más de veinte mil SKU, al combinar calidad de recuperación, robustez ante la esparsidad y eficiencia de inferencia. La ablación por enmascaramiento aporta, además, un procedimiento sencillo para verificar qué señal sostiene realmente a un recomendador de este tipo, distinción que las métricas de recuperación al uso no revelan por sí solas.

5.2 Recomendaciones y Trabajo Futuro

Las limitaciones detectadas sugieren varias líneas de mejora y de trabajo futuro:

1. Mitigar el cold-start de productos. Conviene enriquecer la torre de candidatos con representaciones textuales de las descripciones e información multimodal, para que los productos nuevos sin historial reciban embeddings más informativos.
2. Descomponer el mecanismo del compromiso diversidad-precisión. La ablación de la sección 4.1.3 cuantifica el efecto de la corrección LogQ sobre la cobertura y el coeficiente de Gini bajo presupuesto igualado, pero no establece por qué su ausencia desplaza las recomendaciones hacia la cola del catálogo. Medir directamente la relación entre la norma

de los embeddings y la popularidad de cada producto permitiría contrastar esa explicación en lugar de inferirla.

3. Incorporar una etapa de ranking fino. La fase de recuperación puede complementarse con un modelo de ranking (DCN-V2 o DeepFM) sobre los candidatos recuperados, para pulir las primeras posiciones y mejorar el MRR.
4. Validar en producción. El paso natural es un despliegue piloto con pruebas A/B que midan el impacto real sobre el ticket promedio, la conversión de cross-selling y la rotación de inventario.
5. Reentrenar y monitorear de forma periódica. Es aconsejable montar un pipeline de regeneración de embeddings para sostener la precisión a medida que ingresan nuevas referencias al catálogo.

5.3 Limitaciones del Estudio

Tras analizar los resultados, conviene tener presentes varias limitaciones. La evaluación es estrictamente offline: aunque la partición temporal reproduce condiciones de producción, no sustituye a una prueba A/B con usuarios reales, de modo que las métricas de negocio (impacto en ticket, conversión) quedan fuera del alcance de esta medición. El ground truth se construye a partir de co-compras observadas, sujetas al sesgo de exposición: no se ven los productos que el cliente habría comprado de haberlos conocido. El alcance se limita a la fase de recuperación, sin una etapa posterior de ranking fino.

La ablación de la corrección LogQ se realizó con presupuesto de entrenamiento igualado, de modo que el efecto sobre la cobertura y la concentración queda establecido; no así el mecanismo que lo produce, que este trabajo cuantifica pero no descompone. La validación cuantitativa del espacio latente se apoya en un conjunto reducido de productos representativos

seleccionados por criterios textuales, por lo que ilustra la estructura de las representaciones sin llegar a caracterizarla para el catálogo completo.

La limitación de mayor alcance es la naturaleza de la señal aprendida: la ablación por enmascaramiento revela que el producto-semilla contribuye en torno al 2 % del Recall@100, dentro del margen de muestreo, de modo que el sistema funciona como predictor contextual de la próxima compra y no como recomendador de complementos condicionado al artículo consultado. Las métricas empleadas no distinguen entre ambos comportamientos, por lo que su lectura exige el contraste adicional que se incorpora en la Tabla 14.

Sugahara et al. (2024) muestran que las etiquetas derivadas del comportamiento de co-compra no siempre corresponden a complementariedad funcional, de modo que una parte de lo que aquí se contabiliza como acierto puede deberse a que dos artículos coinciden en una misma cesta y no a que exista una relación técnica entre ellos. En un catálogo ferretero la distancia entre ambas cosas es mayor que en el comercio minorista general, porque la complementariedad obedece a compatibilidad física y no a hábitos de consumo. Separarlas exigiría una etiqueta construida a partir de fichas técnicas de producto y no del historial de venta, tarea que excede el alcance de este estudio y que la sección 5.2 recoge como trabajo futuro.

El arranque en frío de productos es muy bajo (Recall@100 de 0,70 %), al depender solo de atributos estructurados del catálogo, sin características descriptivas suficientes, lo que impide sostener por sí solo la recomendación de novedades. Y los datos provienen de una única empresa, con fuerte concentración geográfica en la Sierra ecuatoriana, por lo que generalizar a otras empresas o regiones exige validación empírica. Lejos de invalidar los hallazgos, estas limitaciones delimitan su alcance y orientan el trabajo futuro descrito en la sección 5.2.

REFERENCIAS

Alatrista-Salas, H., Hoyos, I., Luna, A., & Nunez-del-Prado, M. (2019). Use of latent factors and consumption patterns for the construction of a recommender system. *IEEE Latin America Transactions, 17*(1), 119–126. https://doi.org/10.1109/TLA.2019.8826703

Bagherifard, K., Rahmani, M., Rafe, V., & Nilashi, M. (2018). A recommendation method based on semantic similarity and complementarity using weighted taxonomy: A case on construction materials dataset. *Journal of Information & Knowledge Management, 17*(1), 1850010. https://doi.org/10.1142/S0219649218500107

Batmaz, Z., Yurekli, A., Bilge, A., & Kaleli, C. (2019). A review on deep learning for recommender systems: Challenges and remedies. *Artificial Intelligence Review, 52*(1), 1–37. https://doi.org/10.1007/s10462-018-9654-y

Bibas, K., Sar Shalom, O., & Jannach, D. (2023). Semi-supervised adversarial learning for complementary item recommendation. En *Proceedings of the ACM Web Conference 2023* (pp. 1804–1812). ACM. https://doi.org/10.1145/3543507.3583462

Bobadilla, J., Ortega, F., Hernando, A., & Gutiérrez, A. (2013). Recommender systems survey. *Knowledge-Based Systems, 46*, 109–132. https://doi.org/10.1016/j.knosys.2013.03.012

Cho, G., Shim, P., & Kim, J. (2023). Explainable B2B recommender system for potential customer prediction using KGAT. *Electronics, 12*(17), 3536. https://doi.org/10.3390/electronics12173536

Covington, P., Adams, J., & Sargin, E. (2016). Deep neural networks for YouTube recommendations. En *Proceedings of the 10th ACM Conference on Recommender Systems* (pp. 191–198). ACM. https://doi.org/10.1145/2959100.2959190

Cremonesi, P., Koren, Y., & Turrin, R. (2010). Performance of recommender algorithms on top-N recommendation tasks. En *Proceedings of the 4th ACM Conference on Recommender Systems* (pp. 39–46). ACM. https://doi.org/10.1145/1864708.1864721

Faz Gallardo, J. V., Guaman Capa, V. A., & Najarro Quintero, R. (2025). E-commerce web system with personalized recommendations based on purchase history for American Eagle in Maná, Ecuador. *Management (Montevideo), 3*, 206. https://doi.org/10.62486/agma2025206

Felix, J., & Tjen, J. (2025). Machine learning in business: Product bundling strategy and customer segmentation via market basket analysis algorithm. *MABIS, 16*(1), 29–37. https://doi.org/10.66003/mabis.v16i01.10550

Ferrari Dacrema, M., Cremonesi, P., & Jannach, D. (2019). Are we really making much progress? A worrying analysis of recent neural recommendation approaches. En *Proceedings of the 13th ACM Conference on Recommender Systems* (pp. 101–109). ACM. https://doi.org/10.1145/3298689.3347058

Fu, Z., Gao, H., Guo, W., Jha, S. K., Jia, J., Liu, X., Long, B., Shi, J., Wang, S., & Zhou, M. (2020). Deep learning for search and recommender systems in practice. En *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM. https://doi.org/10.1145/3394486.3406709

Gehrkue, K., & Baladron, J. (2025). Reinforcement learning-based recommendation system for e-commerce: A case study at a major beverage company. En *2025 44th International Conference of the Chilean Computer Science Society (SCCC)* (pp. 1–6). IEEE. https://doi.org/10.1109/SCCC67219.2025.11420666

Geyer-Schulz, A., Hahsler, M., & Thede, A. (2003). Comparing simple association-rules and repeat-buying based recommender systems in a B2B environment. En *Studies in Classification, Data Analysis, and Knowledge Organization* (pp. 421–429). Springer. https://doi.org/10.1007/978-3-642-18991-3_48

Guo, H., Tang, R., Ye, Y., Li, Z., & He, X. (2017). DeepFM: A factorization-machine based neural network for CTR prediction. En *Proceedings of IJCAI 2017* (pp. 1725–1731). AAAI Press. https://doi.org/10.24963/ijcai.2017/239

Guo, R., Sun, P., Lindgren, E., Geng, Q., Simcha, D., Chern, F., & Kumar, S. (2020). Accelerating large-scale inference with anisotropic vector quantization. En *Proceedings of the 37th International Conference on Machine Learning* (pp. 3887–3896). PMLR. https://proceedings.mlr.press/v119/guo20h.html

Hao, J., Zhao, T., Li, J., Dong, X. L., Faloutsos, C., Sun, Y., & Wang, W. (2020). P-Companion: A principled framework for diversified complementary product recommendation. En *Proceedings of the 29th ACM International Conference on Information & Knowledge Management* (pp. 2517–2524). ACM. https://doi.org/10.1145/3340531.3412732

He, Q., Li, X., & Cai, B. (2024). Graph neural network recommendation algorithm based on improved dual tower model. *Scientific Reports, 14*(1), 3853. https://doi.org/10.1038/s41598-024-54376-3

He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., & Wang, M. (2020). LightGCN: Simplifying and powering graph convolution network for recommendation. En *Proceedings of SIGIR 2020* (pp. 639–648). ACM. https://doi.org/10.1145/3397271.3401063

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T.-S. (2017). Neural collaborative filtering. En *Proceedings of WWW 2017* (pp. 173–182). ACM. https://doi.org/10.1145/3038912.3052569

Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems, 22*(1), 5–53. https://doi.org/10.1145/963770.963772

Hidasi, B., Karatzoglou, A., Baltrunas, L., & Tikk, D. (2016). Session-based recommendations with recurrent neural networks. En *Proceedings of ICLR 2016*. https://arxiv.org/abs/1511.06939

Hildebrandt, M., Sunder, S. S., Mogoreanu, S., Joblin, M., Mehta, A., Thon, I., & Tresp, V. (2019). A recommender system for complex real-world applications with nonlinear dependencies

and knowledge graph context. En *Lecture Notes in Computer Science* (pp. 179–193). Springer. https://doi.org/10.1007/978-3-030-21348-0_12

Hu, Y., Koren, Y., & Volinsky, C. (2008). Collaborative filtering for implicit feedback datasets. En *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM)* (pp. 263–272). IEEE. https://doi.org/10.1109/ICDM.2008.22

Huang, J., Sharma, A., Sun, S., Xia, L., Zhang, D., Pronin, P., Padmanabhan, J., Ottaviano, G., & Yang, L. (2020). Embedding-based retrieval in Facebook Search. En *Proceedings of KDD 2020* (pp. 2553–2561). ACM. https://doi.org/10.1145/3394486.3403305

Jadon, A., & Patil, A. (2025). A comprehensive survey of evaluation techniques for recommendation systems. En *Computation of Artificial Intelligence and Machine Learning (Communications in Computer and Information Science)* (pp. 281–304). Springer. https://doi.org/10.1007/978-3-031-71484-9_25

Johnson, J., Douze, M., & Jégou, H. (2021). Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data, 7*(3), 535–547. https://doi.org/10.1109/TBDATA.2019.2921572

Kang, W.-C., & McAuley, J. (2018). Self-attentive sequential recommendation. En *Proceedings of ICDM 2018* (pp. 197–206). IEEE. https://doi.org/10.1109/ICDM.2018.00035

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer, 42*(8), 30–37. https://doi.org/10.1109/MC.2009.263

Kvernadze, G., Sudyanti, P. A. G., Subedi, N., & Hajiaghayi, M. (2022). Two is better than one: Dual embeddings for complementary product recommendations. En *2022 IEEE International Conference on Knowledge Graph (ICKG)* (pp. 131–140). IEEE. https://doi.org/10.1109/ICKG55886.2022.00024

Lee, W.-M., & Cho, Y.-S. (2023). A flexible two-tower model for item cold-start recommendation. *IEEE Access, 11*, 146194–146207. https://doi.org/10.1109/ACCESS.2023.3346918

Li, C., Ishak, I., Ibrahim, H., Zolkepli, M., Sidi, F., & Li, C. (2023). Deep learning-based recommendation system: Systematic review and classification. *IEEE Access, 11*, 113790–113835. https://doi.org/10.1109/ACCESS.2023.3323353

Linden, G., Smith, B., & York, J. (2003). Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing, 7*(1), 76–80. https://doi.org/10.1109/MIC.2003.1167344

Ling, B., Barr, M., Kurra, D. D., Zhu, C., & Marcott, N. (2023, 26 de julio). Innovative recommendation applications using two tower embeddings at Uber. *Uber Engineering Blog*. https://www.uber.com/blog/innovative-recommendation-applications-using-two-tower-embeddings/

Liu, C., Jiang, D., Cai, Y., & Li, H. (2026). A review of deep learning and large language models for cold start problem in recommender systems. *WIREs Data Mining and Knowledge Discovery, 16*(1). https://doi.org/10.1002/widm.70068

Malkov, Y. A., & Yashunin, D. A. (2020). Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 42*(4), 824–836. https://doi.org/10.1109/TPAMI.2018.2889473

McAuley, J., Pandey, R., & Leskovec, J. (2015). Inferring networks of substitutable and complementary products. En *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). ACM. https://doi.org/10.1145/2783258.2783381

McInnes, L., Healy, J., Saul, N., & Großberger, L. (2018). UMAP: Uniform Manifold Approximation and Projection. *Journal of Open Source Software, 3*(29), 861. https://doi.org/10.21105/joss.00861

Meng, W., Chen, L., & Dong, Z. (2024). The development and application of a novel e-commerce recommendation system used in electric power B2B sector. *Frontiers in Big Data, 7*, 1374980. https://doi.org/10.3389/fdata.2024.1374980

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems, 26*, 3111–3119. https://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality

Mulyawan, B., Vionelsy, & Sutrisno, T. (2020). Product recommendation system on building materials shopping using FP-Growth algorithm. *IOP Conference Series: Materials Science and Engineering, 1007*, 012144. https://doi.org/10.1088/1757-899X/1007/1/012144

Osowska-Kurczab, A. M., Nazarko, K., Marzec, M., Wojciechowska, L., & Kremeňová, E. (2025). Suggest, complement, inspire: Story of Two-Tower recommendations at Allegro.com. En *Proceedings of the Nineteenth ACM Conference on Recommender Systems*. ACM. https://doi.org/10.1145/3705328.3748135

Papso, R. (2023). Complementary product recommendation for long-tail products. En *Proceedings of the 17th ACM Conference on Recommender Systems* (pp. 1305–1311). ACM. https://doi.org/10.1145/3604915.3608864

Paraschakis, D., Nilsson, B. J., & Holländer, J. (2015). Comparative evaluation of top-N recommenders in e-commerce: An industrial perspective. En *Proceedings of IEEE ICMLA 2015* (pp. 1024–1031). IEEE. https://doi.org/10.1109/ICMLA.2015.183

Pradel, B., Sean, S., Delporte, J., Guérif, S., Rouveirol, C., Usunier, N., Fogelman-Soulié, F., & Dufau-Joel, F. (2011). A case study in a recommender system based on purchase data. En *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 377–385). ACM. https://doi.org/10.1145/2020408.2020470

Rendle, S., Krichene, W., Zhang, L., & Anderson, J. (2020). Neural collaborative filtering vs. matrix factorization revisited. En *Proceedings of the 14th ACM Conference on Recommender Systems* (pp. 240–248). ACM. https://doi.org/10.1145/3383313.3412488

Ricci, F., Rokach, L., & Shapira, B. (2022). Recommender systems: Techniques, applications, and challenges. En *Recommender Systems Handbook* (3rd ed., pp. 1–35). Springer. https://doi.org/10.1007/978-1-0716-2197-4_1

Saraei, T., Benali, M., & Frayret, J. (2025). A hybrid recommendation system using association rule mining, i-ALS algorithm, and SVD++ approach: A case study of a B2B company. *Intelligent Systems with Applications, 25*, 200477. https://doi.org/10.1016/j.iswa.2025.200477

Sarwar, B. M., Karypis, G., Konstan, J. A., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. En *Proceedings of the 10th International Conference on World Wide Web* (pp. 285–295). ACM. https://doi.org/10.1145/371920.372071

Shambour, Q., & Lu, J. (2015). An effective recommender system by unifying user and item trust information for B2B applications. *Journal of Computer and System Sciences, 81*(7), 1110–1126. https://doi.org/10.1016/j.jcss.2014.12.029

Steck, H. (2019). Embarrassingly shallow autoencoders for sparse data. En *The World Wide Web Conference (WWW 2019)* (pp. 3251–3257). ACM. https://doi.org/10.1145/3308558.3313710

Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Advances in Artificial Intelligence, 2009*, 421425. https://doi.org/10.1155/2009/421425

Su, Y., Li, Y., & Zhang, Z. (2025). Two-tower structure recommendation method fusing multi-source data. *Electronics, 14*(5), 1003. https://doi.org/10.3390/electronics14051003

Sugahara, K., Yamasaki, C., & Okamoto, K. (2024). Is it really complementary? Revisiting behavior-based labels for complementary recommendation. En *Proceedings of the 18th*

ACM Conference on Recommender Systems (pp. 1091–1095). ACM. https://doi.org/10.1145/3640457.3691705

Tang, S. (2025). Two-stage multimodal retrieval and multi-task ranking for e-commerce recommendation. En *2025 6th International Conference on Information Science, Parallel and Distributed Systems (ISPDS)* (pp. 126–130). IEEE. https://doi.org/10.1109/ISPDS67367.2025.11390978

Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research, 9*(86), 2579–2605. https://www.jmlr.org/papers/v9/vandermaaten08a.html

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems, 30*, 5998–6008. https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

Videla-Cavieres, I. F., & Ríos, S. A. (2014). Extending market basket analysis with graph mining techniques: A real case. *Expert Systems with Applications, 41*(4), 1928–1936. https://doi.org/10.1016/j.eswa.2013.08.088

Vlachos, M., Vassiliadis, V. G., Heckel, R., & Labbi, A. (2016). Toward interpretable predictive models in B2B recommender systems. *IBM Journal of Research and Development, 60*(5/6), 11:1–11:12. https://doi.org/10.1147/JRD.2016.2602097

Wang, R., Shivanna, R., Cheng, D. Z., Jain, S., Lin, D., Hong, L., & Chi, E. H. (2021). DCN V2: Improved deep & cross network and practical lessons for web-scale learning to rank systems. En *Proceedings of the Web Conference 2021* (pp. 1785–1797). ACM. https://doi.org/10.1145/3442381.3450078

Wu, L., He, X., Wang, X., Zhang, K., & Wang, M. (2023). A survey on accuracy-oriented neural recommendation: From collaborative filtering to information-rich recommendation. *IEEE Transactions on Knowledge and Data Engineering, 35*(5), 4425–4445. https://doi.org/10.1109/TKDE.2022.3145690

Xiong, C., Yu, X., Xu, W., Cheng, L., Yuan, C., & Mo, L. (2025). A learnable fully interacted two-tower model for pre-ranking system. En *Proceedings of SIGIR 2025*. ACM. https://doi.org/10.1145/3726302.3729881

Yan, A., Dong, C., Gao, Y., Fu, J., Zhao, T., Sun, Y., & McAuley, J. (2022). Personalized complementary product recommendation. En *Companion Proceedings of the Web Conference 2022 (WWW '22 Companion)*. ACM. https://doi.org/10.1145/3487553.3524222

Yang, H., Yuan, C., Bai, K., Guo, M., Yang, W., & Zhou, C. (2025). HIT model: A hierarchical interaction-enhanced two-tower model for pre-ranking systems. En *Proceedings of the 34th ACM International Conference on Information and Knowledge Management (CIKM 2025)* (pp. 6201–6208). ACM. https://doi.org/10.1145/3746252.3761501

Yang, J., Yi, X., Cheng, D. Z., Hong, L., Li, Y., Wang, S. X., Xu, T., & Chi, E. H. (2020). Mixed negative sampling for learning two-tower neural networks in recommendations. En *Companion Proceedings of the Web Conference 2020* (pp. 441–447). ACM. https://doi.org/10.1145/3366424.3386195

Yi, X., Yang, J., Hong, L., Cheng, D. Z., Heldt, L., Kumthekar, A., Zhao, Z., Wei, L., & Chi, E. (2019). Sampling-bias-corrected neural modeling for large corpus item recommendations. En *Proceedings of the 13th ACM Conference on Recommender Systems* (pp. 269–277). ACM. https://doi.org/10.1145/3298689.3346996

Zhang, S., Yao, L., Sun, A., & Tay, Y. (2019). Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys, 52*(1), Artículo 5. https://doi.org/10.1145/3285029

Zhang, W., Enders, T., & Li, D. (2017). GreedyBoost: An accurate, efficient and flexible ensemble method for B2B recommendations. En *Proceedings of the Annual Hawaii International Conference on System Sciences*. HICSS. https://doi.org/10.24251/HICSS.2017.189

Zhang, W., Chen, Z., Zha, H., & Wang, J. (2021). Learning from substitutable and complementary relations for graph-based sequential product recommendation. *ACM Transactions on Information Systems, 40*(2), 1–28. https://doi.org/10.1145/3464302

Zhang, Y., Lu, H., Niu, W., & Caverlee, J. (2018). Quality-aware neural complementary item recommendation. En *Proceedings of the 12th ACM Conference on Recommender Systems* (pp. 77–85). ACM. https://doi.org/10.1145/3240323.3240368

APÉNDICES

Apéndice A. Inventario de cuadernos de cómputo (notebooks)

El desarrollo se organizó en una secuencia reproducible de cuadernos Jupyter y scripts de Python, cada uno responsable de una fase del pipeline. Los procesos por lotes de larga duración y reanudables se implementaron como scripts. La Tabla 19 resume su propósito. A ese inventario se suma 17_seed_overlap, incorporado en esta revisión para medir el solapamiento entre listas top-K al variar el producto-semilla, cuyo resultado analiza la sección 4.2.1.

El código fuente completo de estos cuadernos, junto con los datos procesados y las instrucciones de reproducción, está disponible en el repositorio público del proyecto:

https://github.com/aaorella/two_tower_b2b_hardware_retail

Git Tag: v1

Tabla *19*

Inventario de cuadernos del proyecto

Cuaderno	Propósito
01_unification_anonymization	Unificación de 27 archivos CSV y anonimización del RUC.
02_eda	Análisis exploratorio de datos y auditoría de nulos.
03_retrieval_dataset	Construcción de pares de co-compra y partición temporal.
04_query_tower	Implementación y validación de la torre de consulta.
05_candidate_tower	Implementación y validación de la torre de candidatos.
06_training_logq	Entrenamiento del modelo final sobre la totalidad de los pares.
07_offline_evaluation	Evaluación contra baselines (popularidad; la comparativa vigente con co-ocurrencia ítem-ítem se reporta en los cuadernos 12 y 14).
08_embeddings_visualization	Proyecciones t-SNE/UMAP y matrices de similitud.
09_ablation_and_diversity	Ablación de la corrección LogQ: rendimiento, cobertura y Gini.
10_cold_start_evaluation	Evaluación de arranque en frío (clientes e ítems).
11_ann_latency_vs_recall	Simulación de servido HNSW frente a búsqueda exacta.
12_extended_metrics_and_ci	Métricas extendidas (NDCG, HitRate) e IC bootstrap; comparativa con co-ocurrencia.
13_topk_qualitative	Ejemplos de recomendaciones top-K sobre productos semilla reales.

14_cooccurrence_baseline	Baseline de co-ocurrencia ítem-ítem: variantes de puntuación, comparativa y cold-start.
15_data_quality_audit	Auditoría reproducible de calidad de datos (script).
16_hyperparameter_selection	Búsqueda secuencial de hiperparámetros y criterio de parada (script).

Nota. Todos los cuadernos y scripts emplean Python 3.12, TensorFlow 2.21, TensorFlow Recommenders 0.7.2 y Polars 1.41.

Apéndice B. Consideraciones de privacidad y cumplimiento normativo

El tratamiento de datos se realizó conforme a la Ley Orgánica de Protección de Datos Personales del Ecuador (LOPDP). Los identificadores tributarios (RUC) fueron reemplazados por códigos deterministas (CLIENTE_00001 a CLIENTE_14935), y las coordenadas geográficas se truncaron a dos decimales. Ese tratamiento es una seudonimización y no una anonimización: la sección 3.2.1 mide el riesgo residual y comprueba que la coordenada truncada sigue identificando en solitario al 99,52 % de los clientes que conservan una propia, y que el conjunto retiene la dirección del cliente, que es un identificador directo. El acceso a la base de datos se efectuó en el marco de un acuerdo de confidencialidad con la empresa distribuidora.

Apéndice C. Resumen de resultados experimentales

La Tabla 20 consolida los principales resultados cuantitativos obtenidos en los experimentos del Capítulo 4, como referencia rápida.

Tabla *20*

Consolidado de resultados experimentales principales

Experimento	Resultado clave
Comparativa de modelos	Two Towers Recall@100 = 33,05 %; MRR = 0,0354 (mejor en recuperación amplia)
Baseline popularidad	Recall@100 = 21,25 %; MRR = 0,0219
Baseline co-ocurrencia ítem-ítem	Recall@100 = 24,05 %; MRR = 0,0389
Ablación sin geolocalización	Recall@100 cae a 14,10 % (−57,3 %)
Ablación sin identidad (RUC)	Recall@100 cae a 23,85 % (−27,8 %)
Cold-start clientes (Two Towers)	Recall@100 = 27,20 % vs. 26,30 % de co-ocurrencia (sin intervalos para el modelo; ventaja en personalización)
Cold-start productos (Two Towers)	Recall@100 = 0,70 % vs. 0,00 % de co-ocurrencia
Servido HNSW (efSearch = 128)	Overlap@100 = 91,48 %; latencia 0,0167 ms

Nota. Síntesis de los cuadernos 07, 09, 10, 11, 12 y 14. Métricas sobre validación temporal 2026 Q1.

Apéndice D. Notación matemática

La Tabla 21 resume la notación empleada en la formulación matemática del Capítulo 3.

Tabla *21*

Notación matemática empleada

Símbolo	Significado
x	Vector de entrada del cliente y su contexto (consulta)
y	Producto candidato del catálogo
f_θ, g_φ	Torre de consulta y torre de candidatos (con parámetros θ y φ)
u(x), v(y)	Embeddings de consulta y de candidato en ℝ^D
D	Dimensión del espacio compartido (D = 128)
s(x, y)	Puntaje de afinidad (producto punto)
P(y)	Frecuencia relativa (probabilidad de muestreo) del ítem y
B, C	Lote (batch) y catálogo completo de productos
Top-K(q), R_q	Top-K recuperado y conjunto relevante (ground truth) de la consulta q
rank_q	Posición del primer ítem relevante para la consulta q
G	Coeficiente de Gini sobre la frecuencia de recomendación

Nota. Notación coherente con Yi et al. (2019) y la formulación de las ecuaciones 3.1 a 3.15.

Apéndice E. Protocolo de la revisión sistemática de literatura

La revisión que sostiene la sección 2.1 se ejecutó sobre el corpus de Elicit, que indexa alrededor de 138 millones de registros académicos procedentes de Semantic Scholar, mediante recuperación semántica. Se emplearon cinco cadenas complementarias, dirigidas a cubrir por separado cada dimensión del problema: la arquitectura de doble torre para recuperación de candidatos; la recomendación en entornos B2B, mayoristas y de distribución industrial; la recomendación y el análisis de canasta en ferretería y materiales de construcción; los estudios con datos transaccionales de América Latina; y la recomendación de complementarios a partir de co-compra en contextos de aprovisionamiento profesional.

Los criterios de inclusión fueron tres. El trabajo debía proponer, implementar o evaluar un sistema de recomendación, un método de recomendación de productos o un método de asociación sobre canasta de compra. El tipo de comprador, empresa o consumidor final, debía poder recuperarse del texto. Y se privilegiaron los estudios con datos transaccionales reales de una empresa en operación frente a los que emplean únicamente conjuntos de referencia públicos o datos sintéticos. Se excluyeron los trabajos de pronóstico de demanda, fijación de precios, optimización de inventario o predicción de abandono que no producen una recomendación como salida.

El proceso recuperó 855 registros, que se cribaron por título y resumen, y retuvo 38 estudios para extracción. De cada uno se extrajeron la familia de arquitectura, si emplea o no doble torre, el tipo de comprador, el dominio de producto, el país o región de los datos, el origen de los datos, el tamaño del catálogo, el tratamiento de la complementariedad, las métricas reportadas y cualquier afirmación de novedad formulada por los autores. La sección 2.1.7 discute el resultado de esa extracción.

Todas las entradas de la lista final se verificaron por DOI contra Crossref, y las que carecen de DOI se contrastaron contra la página del editor original. Cuando el registro de Crossref no incluía resumen, el dominio de aplicación se comprobó contra el resumen del editor o de Semantic Scholar antes de atribuirlo en el texto. Los informes de ingeniería publicados por empresas se mantienen identificados como literatura gris y se emplean como contexto, nunca como respaldo de afirmaciones centrales.