Informe Técnico: Prueba HiloTools — Pipeline de Datos

Vhanessa Cardona

21 de agosto de 2025

Índice

1. Introducción

En el contexto actual de las organizaciones modernas, la **gestión y análisis de datos** se ha convertido en un factor crítico para la toma de decisiones estratégicas. La disponibilidad de información en múltiples dominios —ventas, gestión de talento humano y control de inventarios— plantea la necesidad de construir soluciones de *data engineering* que permitan integrar, limpiar y transformar datos heterogéneos en un modelo analítico coherente.

La presente prueba técnica tiene como propósito demostrar la capacidad de diseñar un **pipeline de datos end-to-end**, a partir de tres fuentes primarias en formato Excel: `Ventas`, `RRHH` e `Inventario`. Estos archivos contienen información operacional de una compañía ficticia que requiere ser consolidada en un sistema analítico capaz de ofrecer soporte a áreas clave del negocio, tales como el desempeño de empleados, la eficiencia del manejo de inventarios y la evolución de las ventas.

Para lograr este objetivo, se plantearon los siguientes pasos principales:

1. **Ingesta y preprocesamiento**: diseño de un módulo robusto que permita consumir los tres ficheros Excel, procesando todas sus hojas de cálculo, manejando valores faltantes, diferencias de formato y decimales con coma. Esta etapa asegura que los datos se encuentren en un formato uniforme y confiable para análisis posteriores.

2. **Modelado en estrella**: construcción de un esquema de tipo *star schema*, con una tabla de hechos de ventas y dimensiones que representan clientes, productos, fechas, empleados y almacenes. Este modelo, ampliamente utilizado en entornos de *business intelligence*, garantiza eficiencia en consultas analíticas y claridad en la interpretación de las relaciones entre entidades.

3. **Análisis de Componentes Principales (PCA)**: integración de métricas provenientes de ventas, RRHH e inventario en un conjunto de *features* normalizado, sobre el cual se aplica PCA con el fin de identificar patrones latentes y reducir la dimensionalidad del problema. Este análisis facilita descubrir cuáles variables tienen mayor peso en las dinámicas del negocio y cómo se relacionan entre sí.

El resultado final es un **pipeline reproducible**, empaquetado en un entorno controlado (Docker o Conda), documentado en un repositorio GitHub con estructura modular, y acompañado de un informe técnico. De esta manera, no solo se busca resolver el caso propuesto, sino también demostrar buenas prácticas de ingeniería de datos, claridad en la documentación y capacidad de extraer hallazgos valiosos a partir de información integrada.

2. Ingesta de datos

La etapa de ingesta constituye el **punto de entrada del pipeline**, donde se incorporan y estandarizan los datos provenientes de las distintas fuentes primarias. En este caso, los tres ficheros provistos en formato Excel (`sales_sample.xlsx`, `hr_sample.xlsx` e `inventory_sample.xlsx`) fueron leídos de manera automatizada, procesando cada una de sus hojas de cálculo y consolidando la información en un esquema uniforme.

2.1. Lectura y exploración de ficheros

Para la lectura inicial se utilizó la librería `pandas`, que permite manejar archivos Excel con múltiples pestañas mediante la función `pd.ExcelFile`. De esta forma, se logró acceder a todas las hojas contenidas en cada fichero sin necesidad de conocer previamente su número o nombre. Este enfoque garantiza que el pipeline pueda adaptarse a futuras actualizaciones de las fuentes sin modificaciones sustanciales en el código.

Durante esta etapa se efectuó un análisis exploratorio preliminar con el fin de identificar:

- Columnas relevantes en cada hoja de cálculo.
- Formatos de fecha y presencia de marcas temporales.
- Existencia de duplicados o registros inconsistentes.
- Uso de comas como separadores decimales.
- Posibles valores nulos o incompletos.

2.2. Estandarización de formatos

Uno de los retos principales en la ingesta fue la heterogeneidad en los formatos de datos. Por ejemplo, varias columnas con sufijo `Date` presentaban registros en formato `YYYY-MM-DD HH:MM:SS`. Estas fueron convertidas de manera uniforme a objetos `datetime` de `pandas`, lo que permitió posteriores operaciones de filtrado, agrupamiento y análisis temporal.

De forma análoga, los valores numéricos que contenían comas como separador decimal fueron normalizados al estándar de punto decimal. Esto resultó crucial para evitar errores al momento de realizar cálculos de montos, salarios, inventarios y márgenes de ganancia.

2.3. Manejo de valores nulos y duplicados

El pipeline se diseñó para ser tolerante a datos incompletos. En lugar de descartar registros, se implementaron estrategias de imputación o marcación explícita de valores faltantes, dependiendo del contexto de cada columna. Los duplicados, en cambio, fueron eliminados tras validar que no aportaban información adicional y que correspondían a errores de carga o repetición.

2.4. Exportación a formato analítico

Una vez normalizados, los datos fueron exportados en formato `parquet`, que ofrece ventajas en términos de compresión, eficiencia de lectura/escritura y compatibilidad con entornos analíticos modernos. Cada hoja de Excel fue almacenada en un fichero `.parquet` independiente, ubicado en la carpeta `/data/processed`. Este diseño modular facilita la trazabilidad de los datos y permite realizar transformaciones adicionales sin necesidad de procesar de nuevo los archivos brutos.

2.5. Reproducibilidad

Finalmente, toda la lógica de ingesta fue encapsulada en scripts dentro de la carpeta `/ingest`, con funciones auxiliares en `utils_ingest.py`. Este módulo garantiza que la ingesta pueda ejecutarse tanto en un entorno local (`conda`) como en un contenedor `Docker`, preservando la reproducibilidad y estandarizando el flujo de datos de extremo a extremo.

3. Modelo en estrella

El modelo en estrella es una técnica de modelado dimensional utilizada en entornos analíticos y de inteligencia de negocios, cuyo objetivo principal es optimizar la consulta de grandes volúmenes de datos. En este proyecto se diseñó un esquema en estrella a partir de los datos de ventas, recursos humanos e inventarios, integrando múltiples dominios en una única vista analítica coherente.

3.1. Diseño del esquema

En el centro del modelo se encuentra la **tabla de hechos de ventas (FactSales)**, que consolida las transacciones individuales con métricas cuantitativas tales como:

- `quantity`: número de unidades vendidas.
- `unit_price`: precio unitario al momento de la venta.
- `discount_percent`: descuentos aplicados.
- `sales_amount`: monto total de la transacción.
- `profit_margin`: margen estimado asociado a la venta.

Rodeando la tabla de hechos se definieron las dimensiones que aportan el contexto necesario para el análisis multidimensional:

- **DimCustomer**: información de clientes (ID, nombre, segmento).
- **DimProduct**: catálogo de productos y categorías.
- **DimStore**: localización física de las tiendas.
- **DimDate**: calendario estandarizado para análisis temporal.
- **DimEmployee**: desempeño y características de empleados.
- **DimDepartment**: estructura organizacional vinculada a RRHH.
- **DimWarehouse**: almacenes donde se controla el inventario.
- **DimCategory**: clasificación de productos según inventarios.

3.2. Esquema resultante

La Figura 1 muestra el diagrama resultante del modelo en estrella, donde se observa la tabla de hechos en el centro y las dimensiones conectadas en torno a ella.

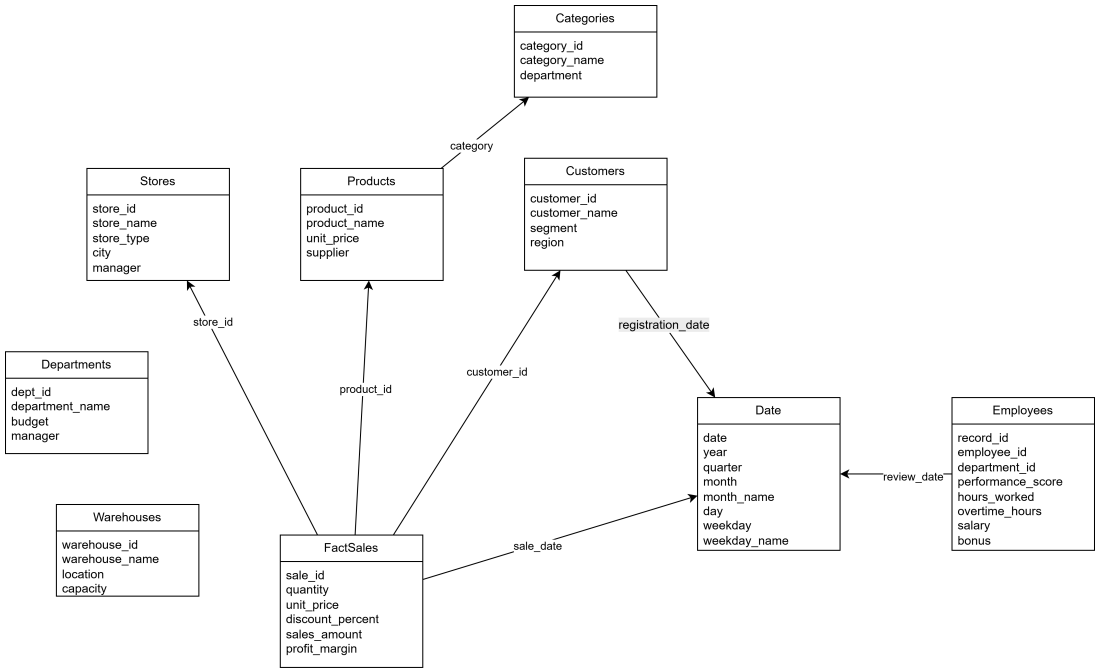

Figura 1: Esquema en estrella integrado a partir de ventas, RRHH e inventarios.

3.3. Ventajas del modelo

El modelo en estrella ofrece múltiples beneficios:

- Simplicidad conceptual: los usuarios pueden comprender fácilmente la relación entre hechos y dimensiones.
- Eficiencia en consultas: optimiza el rendimiento de agregaciones y filtros en grandes volúmenes de datos.
- Flexibilidad analítica: permite análisis combinados, como identificar el impacto de la rotación de empleados en las ventas o la relación entre niveles de inventario y márgenes de ganancia.

4. Análisis de Componentes Principales (PCA)

El análisis de componentes principales (PCA, por sus siglas en inglés) es una técnica estadística de reducción de dimensionalidad que permite identificar patrones latentes en los datos, disminuyendo la redundancia y resaltando las variables que más aportan a la variabilidad del sistema. En este proyecto se aplicó PCA para integrar la información proveniente de ventas, recursos humanos e inventarios, con el fin de detectar relaciones ocultas entre métricas de desempeño, márgenes de rentabilidad y gestión de inventarios.

4.1. Varianza explicada

Se seleccionaron un conjunto de variables cuantitativas relevantes (ventas, descuentos, costos, márgenes, horas trabajadas, entre otras) y se aplicó un PCA con normalización previa. El resultado mostró que las **cinco primeras componentes principales** explican cerca del **80 % de la varianza acumulada**, lo que indica que una proporción significativa de la información contenida en los datos puede ser representada en un espacio de menor dimensión sin pérdida sustancial de contenido.

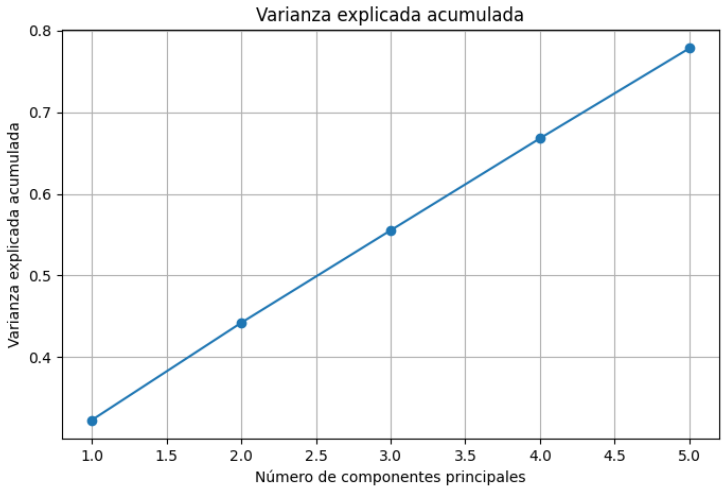

Figura 2: Varianza explicada acumulada por las primeras componentes principales.

4.2. Correlaciones por componente

Para interpretar el significado de cada componente, se analizaron las cargas de correlación entre las variables originales y las componentes principales. Los hallazgos más relevantes fueron:

- **PC1**: fuerte asociación con las variables `sales_amount` y `profit_margin`, lo que sugiere que esta componente captura principalmente la dinámica de rentabilidad y volumen de ventas.
- **PC3**: alta correlación con `discount_percent`, lo cual refleja la influencia de las políticas de descuentos en el comportamiento de las ventas.
- **PC4**: marcada relación con `product_id`, lo que indica que esta componente está asociada a características particulares de los productos (segmentación o tipología).

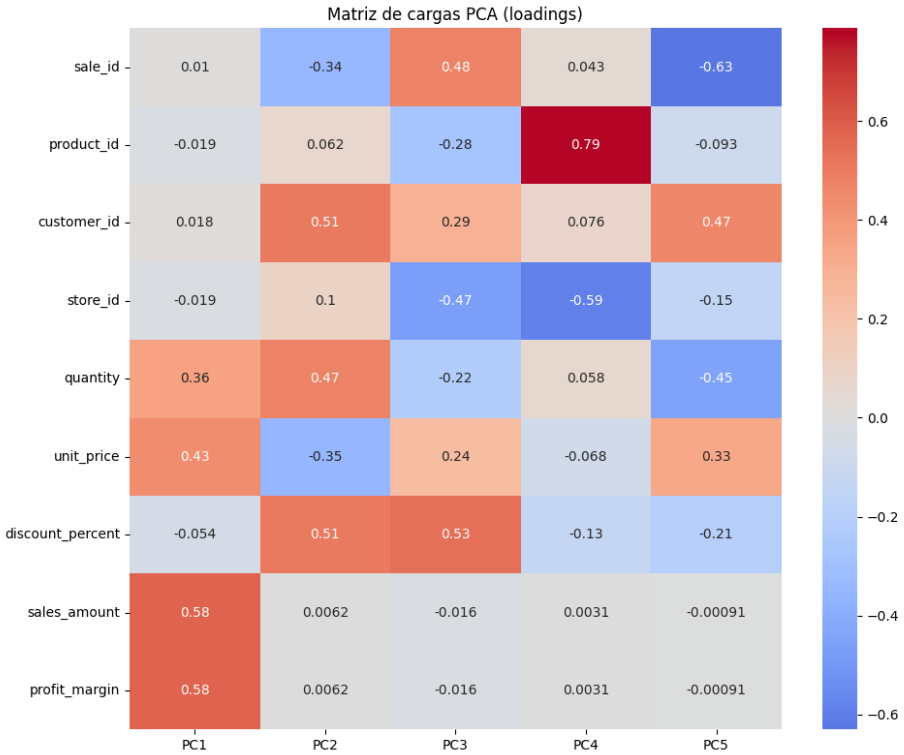

Figura 3: Mapa de calor de correlación entre variables originales y componentes principales.

4.3. Interpretación

El análisis PCA permitió resumir la complejidad del dataset en un número reducido de dimensiones con interpretación económica y de negocio:

- La primera componente representa la **dimensión de rentabilidad**, explicando gran parte de la varianza.
- La tercera y cuarta componentes están más vinculadas a **estrategias comerciales** (descuentos) y a **atributos de productos**.
- En conjunto, las cinco primeras componentes ofrecen una base sólida para segmentar clientes, evaluar desempeño de empleados y planificar inventarios en función de patrones latentes.

5. Conclusiones

El desarrollo de este pipeline de datos permitió demostrar la integración de diversas fuentes heterogéneas (ventas, recursos humanos e inventarios) en un modelo analítico coherente y reproducible. A partir de la experiencia obtenida, se pueden destacar las siguientes conclusiones:

- **Ingesta robusta:** Se diseñó un proceso de ingesta capaz de manejar múltiples hojas de cálculo, formatos inconsistentes, valores nulos y diferencias en el uso de separadores decimales. Esto garantiza una base sólida para futuros procesos de automatización.

- **Modelo estrella integrado:** La construcción del esquema en estrella permitió centralizar la tabla de hechos de ventas (`FactSales`) y vincularla con dimensiones clave como clientes, productos, empleados, almacenes y fechas. Este diseño facilita tanto la escalabilidad como la generación de reportes analíticos.
- **Análisis PCA:** La aplicación de componentes principales reveló que cinco dimensiones explican cerca del 80 % de la varianza en los datos. Se identificaron patrones relevantes, tales como:
  - La **dimensión de rentabilidad** (PC1), asociada a `sales_amount` y `profit_margin`.
  - La **dimensión comercial** (PC3), fuertemente influenciada por `discount_percent`.
  - La **dimensión de producto** (PC4), relacionada con `product_id`.

  Estos hallazgos ofrecen información valiosa para segmentar estrategias de ventas, optimizar inventarios y evaluar el desempeño organizacional.
- **Buenas prácticas:** El proyecto se estructuró bajo una organización modular de carpetas (/ingest, /model, /analytics, /report), promoviendo la reproducibilidad, escalabilidad y mantenibilidad del código en entornos locales y contenedores (Docker).

En síntesis, el pipeline desarrollado no solo responde a los requerimientos de la prueba técnica, sino que sienta las bases para una plataforma analítica más amplia, capaz de adaptarse a nuevos flujos de datos y de evolucionar hacia aplicaciones de inteligencia empresarial y ciencia de datos más avanzadas.

Referencias

[1] Pandas Documentation, `https://pandas.pydata.org/`

[2] Scikit-Learn Documentation, `https://scikit-learn.org/`

[3] Enunciado oficial de la prueba técnica HiloTools.