TECNICATURA SUPERIOR EN

Ciencia de Datos e Inteligencia Artificial

Trabajo Práctico ISPC

Exploración, Transformación y Limpieza de Datos

Título del Informe: Proceso ETL para Análisis de Datos de Ventas y Clientes

Materia: Programación II y Base de Datos

Grupo: Data_Retail_ETL

Estudiante: Giovanny Aguilar Rojas

Destinatario: Ramiro Adrian CEBALLES

Carlos Ignacio CHARLETTI

Cohorte: 2025 **Tipo de Proyecto**: Tecnologico

Fecha de Entrega: 17/octubre/2025

2. Resumen Ejecutivo (Executive Summary)

Este informe documenta la implementación de un proceso ETL (Extracción, Transformación y Carga) para unificar y analizar dos conjuntos de datos: ventas transaccionales y datos de clientes. El objetivo principal fue preparar datos de alta calidad en una base relacional (MariaDB) para la extracción de métricas clave sobre el comportamiento de compra.

Metodología Clave: Se utilizó **Python (Pandas)** para la limpieza de datos, la gestión de nulos y la categorización etaria, seguido de la carga a **MariaDB** y la ejecución de consultas **SQL** para el análisis.

Hallazgos Principales (Key Findings):

1. **Pago Dominante:** El método de pago preferido globalmente es **Cash** (Efectivo), representando la mayor parte de las transacciones en casi todos los segmentos.
2. **Segmento de Mayor Valor:** El segmento etario entre **26 y 50 años** es el principal motor de la actividad comercial y volumen de compra.
3. **Valor del Producto:** La categoría **"Cosmetics"** registra el precio promedio más alto entre todos los productos analizados.

Conclusión y Recomendación Principal: El conjunto de datos unificado está listo para modelado predictivo. Se recomienda enfocar inmediatamente las campañas de marketing en el grupo de **26-50 años**, optimizando ofertas que se adapten a la alta preferencia por pagos en efectivo.

3. Introducción

Contexto y Planteamiento del Problema

La información de la empresa se encuentra fragmentada en dos fuentes: datos estáticos de clientes (customer_data.csv) y registros transaccionales de ventas (sales_data.csv). Esta separación dificulta una visión unificada del rendimiento. El problema a resolver es la necesidad de **integrar, limpiar y transformar** estas fuentes para crear un repositorio único que permita un análisis coherente del comportamiento de compra.

Objetivos

- **Objetivo General:** Implementar y documentar de forma transparente el proceso ETL, migrando datos de fuentes dispares a una base de datos relacional para el análisis estratégico.
- **Objetivos Específicos:**
    1. Integrar correctamente la información de ventas y clientes.
    2. Establecer métricas clave (ej. total_price).
    3. Determinar el modo de pago dominante por género y rangos etarios.
    4. Analizar la distribución de precios promedio por categoría de productos.

Alcance y Limitaciones

El alcance del informe cubre desde la ingesta inicial de los archivos CSV hasta la ejecución de consultas SQL descriptivas sobre la base de datos resultante. La principal limitación radica en

que el análisis se limita a datos descriptivos y exploratorios, sin incursionar en el modelado predictivo.

4. Marco Conceptual y Tecnológico

Proceso ETL (Extract, Transform, Load)

- **Extract (Extracción):** Lectura y recopilación de los datos desde las fuentes originales.
- **Transform (Transformación):** Limpieza (gestión de nulos), estandarización (fechas, nombres de columnas) e integración de los datos antes de su destino final.
- **Load (Carga):** Inserción de los datos limpios en la base de datos MariaDB para su persistencia y posterior consulta.

Herramientas y Técnicas Analíticas

- **Software y Librerías: Python** se utilizó como lenguaje principal, apoyado por la librería **Pandas** para la manipulación del DataFrame y **SQLAlchemy/PyMySQL** para la conexión con la base de datos.
- **Base de Datos: MariaDB** (utilizando el entorno XAMPP) como destino final para la persistencia de los datos.
- **SQL (Structured Query Language):** Utilizado para el Diseño de la base de datos (DDL) y la extracción de los resultados (DQL).

5. Metodología: Procesamiento y Limpieza de Datos

5.1. Extracción (Paso E)

Se cargaron customer_data.csv y sales_data.csv en DataFrames separados. Posteriormente, se realizó la **concatenación** de ambos en el merged_df utilizando la clave común customer_id y un tipo de unión interna (how='inner') para asegurar la existencia de un cliente por cada venta.

```
# 1.1 Cargar los archivos CSV en DataFrames distintos
customer_df = pd.read_csv('customer_data.csv')
sales_df = pd.read_csv('sales_data.csv')
# 1.2 Concatenar ambos DataFrames (usando merge por el campo común 'customer_id')
merged_df = pd.merge(sales_df, customer_df, on='customer_id', how='inner')
```

--DATOS DEL CLIENTE/ customer_df: 99457 x 4

	customer_id	gender	age	payment_method
0	C241288	Female	28.0	Credit Card
1	C111565	Male	21.0	Debit Card
2	C266599	Male	20.0	Cash
3	C988172	Female	66.0	Credit Card
4	C189076	Female	53.0	Cash
5	C657758	Female	28.0	Credit Card

\--DATOS DE VENTAS/ sales_df: 99457 x 7

	invoice_no	customer_id	category	quantity	price	invoice_date	shopping_mall
0	I138884	C241288	Clothing	5	1500.40	05-08-2022	Kanyon
1	I317333	C111565	Shoes	3	1800.51	12-12-2021	Forum Istanbul
2	I127801	C266599	Clothing	1	300.08	09-11-2021	Metrocity
3	I173702	C988172	Shoes	5	3000.85	16-05-2021	Metropol AVM
4	I337046	C189076	Books	4	60.60	24-10-2021	Kanyon
5	I227836	C657758	Clothing	5	1500.40	24-05-2022	Forum Istanbul

DATAFRAME CONBINADOS (VENTAS + CLIENTES)/ merged_df: 99457 x 10

	invoice_no	customer_id	category	quantity	price	invoice_date	shopping_mall	gender	age	payment_method
0	I138884	C241288	Clothing	5	1500.40	05-08-2022	Kanyon	Female	28.0	Credit Card
1	I317333	C111565	Shoes	3	1800.51	12-12-2021	Forum Istanbul	Male	21.0	Debit Card
2	I127801	C266599	Clothing	1	300.08	09-11-2021	Metrocity	Male	20.0	Cash
3	I173702	C988172	Shoes	5	3000.85	16-05-2021	Metropol AVM	Female	66.0	Credit Card
4	I337046	C189076	Books	4	60.60	24-10-2021	Kanyon	Female	53.0	Cash
5	I227836	C657758	Clothing	5	1500.40	24-05-2022	Forum Istanbul	Female	28.0	Credit Card

5.2. Transformación y Limpieza (Paso T)

La transformación fue la etapa crítica del proceso ETL, enfocada en asegurar la calidad, la consistencia y la preparación analítica del *dataset* unificado. A continuación, se detalla la finalidad de cada operación:

1. Normalización de Nombres de Columnas

Propósito: Asegurar la uniformidad en el acceso y manejo de las columnas, facilitando su posterior mapeo a la base de datos SQL.

- **Detalle:** La sentencia df.columns = df.columns.str.lower().str.replace(' ', '_') convierte todos los nombres de columna a minúsculas y sustituye los espacios por guiones bajos (snake_case).

2. Gestión y Eliminación de Valores Nulos

Propósito: Garantizar la integridad de los datos, especialmente en campos críticos para el análisis (como precios y cantidades).

- **Detalle:** El método merged_df = merged_df.dropna() se aplicó para eliminar cualquier fila que contuviera valores nulos en *cualquiera* de sus campos. Esta estrategia asegura que los cálculos posteriores de ingresos y promedios sean precisos y no se vean sesgados por datos incompletos.

3. Estandarización del Formato de Fecha

Propósito: Convertir la columna de fechas a un tipo de dato estándar (datetime), esencial para cualquier análisis temporal y para asegurar la compatibilidad con el tipo de dato DATE de MariaDB.

- **Detalle:** pd.to_datetime(merged_df['invoice_date'], ...) transforma la columna invoice_date al formato de tiempo reconocido por Pandas, lo que permite realizar filtros y agrupaciones por año, mes o día.

4. Cálculo de la Columna Derivada: total_price

Propósito: Generar la métrica de ingresos clave para el análisis financiero del proyecto.

- **Detalle:** Se creó una nueva columna, total_price, mediante la multiplicación de los valores de quantity y price (merged_df['total_price'] = merged_df['quantity'] * merged_df['price']). Esta columna representa el ingreso total de cada transacción registrada.

5. Eliminación de Registros Duplicados

Propósito: Asegurar que cada registro en el *dataset* final represente una transacción única, evitando la doble contabilización y la distorsión de las métricas de frecuencia y volumen.

- **Detalle:** clean_df.drop_duplicates() remueve filas que son copias exactas de otras en el DataFrame, consolidando la unicidad del conjunto de datos antes de la carga a la base de datos.

6. Verificación Final de la Calidad del Data Set

La imagen adjunta muestra el resultado de la función de resumen del DataFrame (.info()) después de la fase de Transformación. Este paso comprueba la efectividad de la limpieza:
Conteo de Registros, Validación de Nulos y Tipos de Datos (Dtype)

```
Index: 99338 entries, 0 to 99456
Data columns (total 11 columns):
 #   Column          Non-Null Count  Dtype         
---  ------          --------------  -----         
 0   invoice_no      99338 non-null  object        
 1   customer_id     99338 non-null  object        
 2   category        99338 non-null  object        
 3   quantity        99338 non-null  int64         
 4   price           99338 non-null  float64       
 5   invoice_date    99338 non-null  datetime64[ns]
 6   shopping_mall   99338 non-null  object        
 7   gender          99338 non-null  object        
 8   age             99338 non-null  float64       
 9   payment_method  99338 non-null  object        
 10  total_price     99338 non-null  float64       
dtypes: datetime64[ns](1), float64(3), int64(1), object(6)
memory usage: 9.1+ MB
```

6. Análisis y Hallazgos

Análisis Exploratorio de Datos (EDA)

El propósito de este análisis es **identificar los patrones clave de ventas, precios y demografía de los clientes** con el fin de informar la toma de decisiones estratégicas en las áreas de marketing, inventario y gestión de centros comerciales.

== > Grafico Ilustrado ⇒

Al finalizar, podremos responder a las siguientes preguntas clave:

- **¿Quiénes son nuestros clientes más valiosos?** (Por género y grupo de edad).
- **¿Qué categorías de productos generan la mayor ganancia?** (Relación Precio vs. Frecuencia).
- **¿Cuáles son nuestros puntos de venta más estratégicos?** (Ranking por centro comercial).
- **¿Cómo ha evolucionado la tendencia de ventas?** (Análisis temporal y estacional).

7. Carga en MariaDB/MySQL

La fase de Carga se enfoca en transferir el DataFrame de Pandas, ya limpio y transformado (clean_df), a la tabla destino definida en MariaDB. Para esto, se utilizó la librería **SQLAlchemy**, que provee una interfaz de alto nivel para interactuar con bases de datos, y **PyMySQL** como driver de conexión.

Proceso de Carga y Configuración

1. **Configuración de Conexión:** Se definen las credenciales (DB_HOST='localhost', DB_USER='root', etc.) para construir la cadena de conexión URL. Se implementa un bloque try...except para manejar fallas comunes de conectividad.
2. **Inicialización del Motor:** Se utiliza create_engine() de SQLAlchemy para establecer la conexión al servidor de MariaDB.
3. **Carga de Datos:** El método clean_df.to_sql() de Pandas realiza la transferencia masiva de datos. El DataFrame completo se inserta en la tabla ventas_clientes_clean, lista para el análisis.

== > Proceso de Conexión ⇒

El DataFrame final (clean_df) fue migrado e insertado en la tabla ventas_clientes_clean en MariaDB, utilizando la librería SQLAlchemy. Las restricciones de PRIMARY KEY en invoice_id y NOT NULL en customer_id fueron aplicadas para mantener la consistencia de los datos en el destino.

- **Instrucción de Código Clave:** clean_df.to_sql('ventas_clientes_clean', con=engine, if_exists='replace', index=False)
    -

8. Conclusiones y Recomendaciones

8.1. Conclusiones

La culminación del proyecto valida el cumplimiento del objetivo general de implementar un proceso ETL estructurado y reproducible, asegurando la integración y el saneamiento de datos de fuentes diversas. La **gestión de valores nulos** y la **creación de la variable age_group** fueron determinantes para la calidad analítica del *dataset*. Los hallazgos confirman la predominancia del método de pago en efectivo y perfilan al cliente de mayor valor estratégico en el rango de 26 a 50 años.

8.2. Recomendaciones

1. **Estrategia de Segmentación de Mercado:** Se sugiere focalizar los esfuerzos de mercadeo hacia el grupo demográfico de **26-50 años**, dado su rol como principal motor de la facturación.

2. **Optimización de Procesos de Cobranza:** En virtud de la alta prevalencia del pago en **Efectivo (Cash)**, se recomienda evaluar e implementar mejoras logísticas en los puntos de venta con mayor concentración de transacciones en efectivo.
3. **Análisis de Rentabilidad en la Categoría *Cosmetics*:** Dada la detección del precio promedio más alto en **"Cosmetics"**, se requiere un análisis subsecuente para determinar si esta métrica se traduce en un margen de beneficio superior o si es indicativa de una estructura de costos elevada, lo cual podría requerir una reevaluación de la estrategia de precios.

== > Repositorio de Codigo Git ⇒