Propuesta de generación de ventas en Voltaje S.R.L.

Alumno: Toledo, Alvaro Julian
Legajo: 52721
Cátedra: Práctica supervisada

Objetivo del proyecto

El proyecto **synthetic_sales_generator** tiene como objetivo generar un conjunto completo de **datos sintéticos de ventas** y cargarlos en una base de datos MySQL para su posterior consulta y análisis. El sistema simula el funcionamiento de un escenario real de ventas, incluyendo clientes, ventas y detalle de ventas, manteniendo coherencia entre los datos generados.

El propósito principal del proyecto es **disponer de un dataset de ventas realista sin utilizar datos reales**, que pueda usarse para prácticas académicas, análisis de datos y pruebas de consultas, sin depender de información sensible ni de sistemas productivos.

Arquitectura general del sistema

Por un lado, **PostgreSQL** se usa únicamente como fuente del catálogo de productos. En esta base existen las tablas *articulo*, *categoria* y *tipo*, que provienen del proyecto previo **classification_project**, donde los productos ya fueron clasificados y organizados.

Por otro lado, **MySQL** es la base de datos destino donde se almacenan los datos sintéticos generados. Allí se guardan los clientes, las ventas y el detalle de cada venta. Esta separación permite mantener el catálogo de productos independiente de los datos de ventas, imitando una arquitectura común en sistemas reales.

La infraestructura de MySQL se levanta mediante **Docker Compose**, lo que permite crear, reiniciar o eliminar la base de datos de forma simple y controlada.

Funcionamiento

1. Generación de clientes

La generación de clientes se realiza como una etapa independiente del proceso. Para ello, el proyecto cuenta con un agente que utiliza la API de OpenAI para generar **clientes con datos completos y coherentes**, como nombre, apellido, DNI, teléfono, correo electrónico y domicilio.

Se indica explícitamente que los nombres y apellidos deben ser típicos de Argentina. El agente trabaja por lotes y valida que cada cliente tenga la estructura correcta.

El resultado de esta etapa es un archivo **JSON** (*data/clientes.json*), que funciona como entrada para la carga posterior en la base MySQL.

2. Extracción del catálogo de productos

Antes de generar las ventas, el sistema se conecta a la base PostgreSQL para **leer el catálogo de productos**. En esta etapa se obtiene, para cada artículo, su identificador, nombre, categoría y tipo. Además, el sistema intenta detectar si el artículo tiene algún precio definido en la base; si no lo tiene, ese valor se generará luego dentro de un rango configurado.

Esta etapa define el conjunto de productos que pueden aparecer en las ventas y garantiza que todos los artículos utilizados existan realmente en el catálogo clasificado.

3. Generación de ventas y detalles

La generación de ventas se realiza simulando un año completo de actividad comercial. Para cada día se crea una cantidad variable de ventas, y cada venta se asocia a un cliente y contiene varias líneas de detalle. En esta etapa se generan las fechas de venta, los clientes involucrados, las cantidades de artículos, los precios unitarios y los totales, manteniendo coherencia entre todos estos valores.

Un aspecto clave de esta etapa es la forma en que se seleccionan los artículos dentro de cada venta. En lugar de elegirlos de manera uniforme, el sistema aplica un sesgo de popularidad, implementado en el módulo generate_sales.py. Al inicio del proceso se asigna a cada artículo un peso de popularidad, de modo que algunos productos tengan muchas más probabilidades de aparecer que otros. Este enfoque refleja un comportamiento común en escenarios reales, donde unos pocos artículos concentran gran parte de las ventas y la mayoría se vende con menor frecuencia. Estos pesos se mantienen constantes durante todo el año, lo que hace que los mismos productos tiendan a dominar las ventas a lo largo del tiempo.

Para evitar que la popularidad dependa del orden original de los artículos, los rangos de popularidad se asignan de forma aleatoria. Luego, en cada venta, los artículos se seleccionan mediante un muestreo ponderado sin reemplazo. Esto significa que dentro de una misma venta no se repite el mismo artículo, pero que los artículos más populares tienen mucha más probabilidad de ser elegidos. Como en cada venta se seleccionan varios artículos usando esta misma lógica, los productos más vendidos tienden a aparecer juntos de forma natural, sin necesidad de reglas

explícitas. Como resultado, el dataset final presenta pocos artículos con muchas apariciones, muchos artículos con pocas ventas y combinaciones de productos que se repiten de manera consistente durante el año, lo que lo hace más adecuado para análisis posteriores.

4. Carga de datos en MySQL

Una vez generados los datos, el sistema crea las tablas destino en MySQL: **cliente**, **venta** y **detalle_venta**. Primero se insertan los clientes, luego las ventas y finalmente los detalles de cada venta.

La tabla *detalle_venta* no solo almacena el identificador del artículo, sino también su nombre, categoría y tipo. Esta decisión permite realizar consultas y análisis directamente sobre MySQL sin necesidad de conectarse nuevamente a PostgreSQL, facilitando el trabajo posterior con los datos.

Todo el proceso de creación de tablas, carga de clientes, generación de ventas e inserción de datos se ejecuta desde un único script principal, lo que simplifica la ejecución completa del proyecto.

Resultado final

Como resultado del proyecto, se obtiene una **base de datos MySQL completamente cargada** con clientes, ventas y detalles de ventas, basada en un catálogo real previamente clasificado. El dataset generado es consistente, fácil de regenerar y adecuado para análisis de ventas, consultas SQL y prácticas académicas.

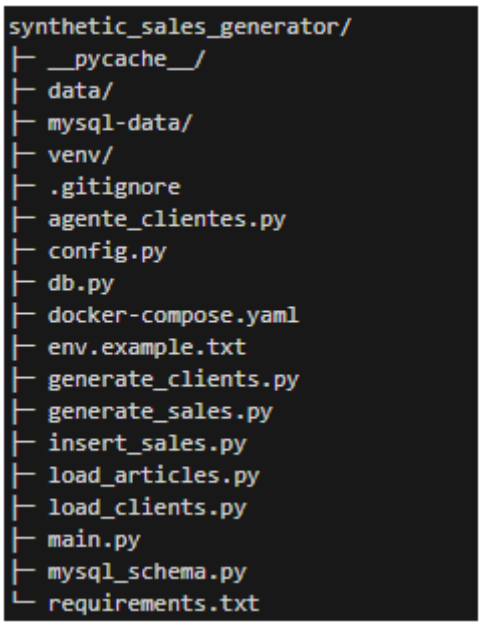

Imagen 1: Estructuración del proyecto

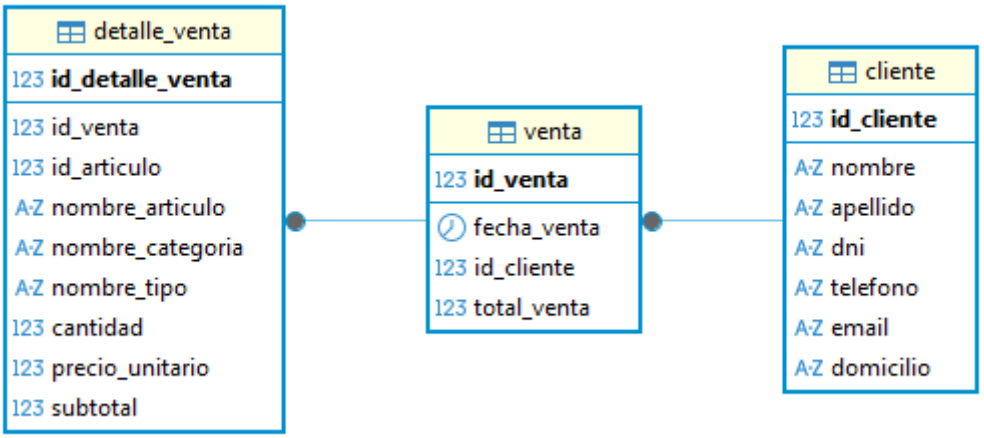

Imagen 2: Modelo de datos generado

Conclusión

El proyecto synthetic_sales_generator logra generar un conjunto completo de datos sintéticos de ventas, manteniendo coherencia entre clientes, ventas y detalles, y permitiendo que el resultado final se comporte como un sistema de ventas real desde el punto de vista del análisis. A lo largo del desarrollo se construyó un flujo simple y ordenado, donde cada etapa cumple una función clara: generación de clientes, lectura del catálogo de productos, simulación de ventas y carga en la base de datos. La separación entre PostgreSQL como fuente del catálogo y MySQL como base destino permite mantener una estructura limpia y organizada, evitando mezclar datos de referencia con datos generados y facilitando la reutilización del catálogo clasificado.

Además, el proyecto incorpora decisiones que mejoran la calidad del dataset generado, como la variabilidad diaria de ventas, la generación de múltiples artículos por operación y el uso de un sesgo de popularidad que hace que algunos productos se vendan con mayor frecuencia que otros. Estas decisiones permiten obtener datos menos uniformes y más cercanos a escenarios reales. El uso de archivos JSON como artefactos intermedios y de Docker para la infraestructura refuerza la reproducibilidad del proceso. En conjunto, el proyecto brinda una base sólida para realizar análisis de ventas, prácticas con bases de datos relacionales y futuros trabajos académicos sobre datos y sistemas de información.