

# PROYECTO INTEGRADOR II

Diego Lopez Castan



# INDICE

- Objetivo Proyecto
- Preguntas de negocio
- KPIs
- Fuentes de Datos
- Modelado Dimensional
- Transformaciones y DBT
- Dashboard
- Repositorio

# OBJETIVO



Este proyecto nace con la necesidad ayudar a una empresa de comercio electrónico que se encuentra en pleno crecimiento a organizar de mejor manera los datos. Para ello nos han proporcionado una estructura de datos y unos script sql para cargar los datos.

Según indican el principal desafío que nos proponen son, diseñar e implementar un modelo de datos relacional y dimensional que permita consolidar la información, facilitar el análisis y empoderar a los analistas de negocio para generar reportes, KPIs e insights con autonomía.

Para lograrlo, vamos a aplicar principios de modelado dimensional, crear un modelo que refleje la realidad del negocio y construirlo con herramientas modernas como DBT. No solo se trata de transformar datos, sino de darles sentido, documentar cada paso y facilitar que otras personas puedan analizarlos sin depender del equipo técnico. La meta final es empoderar a quienes toman decisiones con información confiable, accesible y bien organizada.

# PREGUNTAS DE NEGOCIO



Uno de los temas más importantes que tenemos como Data Engineer es poder crear preguntas relevantes para describir el negocio y por otro lado para optimizarlos. Por ello pense en entender mejor a los clientes, optimizar las ventas, mejorar la logística y anticipar oportunidades o problemas antes de que se vuelvan críticos.

Las preguntas principales que podemos hacer a un e-commerce son por ejemplo: ¿Qué productos están funcionando mejor? ¿Dónde se generan las demoras en las entregas? ¿Qué comportamiento tienen los clientes más fieles? ¿Cómo varía la facturación a lo largo del tiempo? Son preguntas que cualquier negocio debería poder responder sin depender de procesos manuales o intuiciones.

Más adelante en el proyecto voy a definir y crear los principales KPIs que sean relevantes a este negocio. Obviamente teniendo en cuenta la posibilidad de poder controlarlos con los datos suministrados.

# PREGUNTAS DE NEGOCIO

Algunas preguntas que podría consultar por tema son:

## Ventas y órdenes

- ¿Cuántas órdenes se realizaron por mes?
- ¿Cuál fue el total facturado por mes o por año?
- ¿Qué órdenes quedaron en estado “pendiente” o “cancelado”?

## Clientes

- ¿Cuáles son los clientes con mayor volumen de compras?
- ¿Cuál es el ticket promedio por cliente?
- ¿De qué países o ciudades provienen los clientes que más compran?

## Productos

- ¿Cuáles son los productos más vendidos?
- ¿Qué productos tienen más stock y menor rotación?
- ¿Cuál es el ingreso generado por cada producto?



# PREGUNTAS DE NEGOCIO



## Detalle por categorías

- ¿Qué categorías de productos tienen mayores ventas?
- ¿Qué categoría tiene el mayor ticket promedio?

## Pagos

- ¿Qué métodos de pago son los más utilizados?
- ¿Hay correlación entre monto y tipo de pago?

## Logística

- ¿Cuál es el tiempo promedio entre la fecha de envío y la fecha de entrega?
- ¿Qué shippers tienen más entregas demoradas?
- ¿Cuál es el estado de los envíos (en tránsito, entregado, demorado)?

# KPIs



Los KPIs permiten medir lo que importa, detectar desvíos a tiempo y tomar decisiones basadas en datos reales y no en intuiciones. En este proyecto, los KPIs funcionan como el puente entre las preguntas del negocio y las respuestas que debe ofrecer el modelo.

Diseñar buenos indicadores no es solo una cuestión técnica. Implica entender qué está buscando el negocio, qué se puede medir con los datos disponibles y cómo presentar esa información de forma simple y accionable. Un buen KPI no genera más dudas, sino claridad.

A lo largo del trabajo se construirán métricas clave como el total facturado por mes, el ticket promedio, los productos más vendidos, el porcentaje de entregas demoradas, o el nivel de recompra por cliente. Estos indicadores se desarrollarán con una mirada analítica, pero también con foco en el uso: deben poder ser consultados fácilmente por quienes toman decisiones, integrarse en dashboards o informes y actualizarse de forma automática.

# KPIS



## 1. Valor total de ventas

- Descripción: Monto total facturado en un período de tiempo determinado.
- Fórmula: Suma de Total en la tabla Ordenes donde Estado = 'Completada'.
- Objetivo: Medir el rendimiento global del negocio y monitorear ingresos a lo largo del tiempo.

## 2. Ticket promedio

- Descripción: Valor promedio gastado por los clientes en cada orden.
- Fórmula:  $SUM(Total) / COUNT(OrdenID)$  (tabla Ordenes con Estado = 'Completada').
- Objetivo: Evaluar el comportamiento de compra y detectar oportunidades de upselling.

## 3. Índice de satisfacción de productos

- Descripción: Promedio de calificaciones recibidas en las reseñas de productos.
- Fórmula:  $AVG(Calificacion)$  (tabla ReseñasProductos).
- Objetivo: Evaluar la calidad percibida de los productos y detectar productos con baja satisfacción.

## 4. Recompra de clientes (retención)

- Descripción: Porcentaje de clientes que realizaron más de una compra.
- Fórmula:  $(\text{Usuarios con más de una orden} / \text{Total de usuarios con al menos una orden}) * 100$
- Objetivo: Medir la fidelización y satisfacción del cliente con la experiencia general.

# FUENTES DE DATOS



Para llevar a cabo los diferentes análisis del proyecto, se nos proporcionaron 12 archivos .sql. Los mismo contenían las estructura y los datos de carga para ser ejecutados en una base de datos de SQL Server. Lo que hice fue cambiar los archivos para ser ejecutados por una base de datos en postgres.

El archivo 1.Create\_ddl\_postgres.sql define la estructura de todas las tablas de la base de datos, es decir, contiene las sentencias CREATE TABLE necesarias para generar el esquema completo.

Los archivos restantes (2.usuarios\_postgres.sql a 12.historial\_pagos\_postgres.sql) contienen las instrucciones para cargar cada tabla con los datos correspondientes.

# ESTRUCTURA BASE DE DATOS



## Tabla Usuarios

UserID: Identificador único del usuario. Se genera automáticamente.  
Nombre: Nombre del usuario.  
Apellido: Apellido del usuario.  
DNI: Documento de identidad del usuario. Debe ser único.  
Email: Dirección de correo electrónico del usuario. También debe ser única.  
Contraseña: Contraseña encriptada del usuario.  
FechaRegistro: Fecha y hora en que el usuario se registró en la plataforma. Por defecto es la actual.

## Tabla Categorías

CategorialID: Identificador único de la categoría.  
Nombre: Nombre de la categoría de productos.  
Descripción: Breve descripción de la categoría.

## Tabla Productos

ProductID: Identificador único del producto.  
Nombre: Nombre del producto.  
Descripción: Descripción detallada del producto.  
Precio: Precio unitario del producto.  
Stock: Cantidad disponible en inventario.  
CategorialID: Categoría a la que pertenece el producto. Referencia a Categorías.

# ESTRUCTURA BASE DE DATOS



## Tabla Ordenes

OrdenID: Identificador único de la orden de compra.  
UsuarioID: Usuario que realizó la orden. Referencia a Usuarios.  
FechaOrden: Fecha y hora en que se generó la orden.  
Total: Monto total de la orden.  
Estado: Estado de la orden (por ejemplo: Pendiente, Procesada, Cancelada).

## Tabla DetallesOrdenes

DetalleID: Identificador único del ítem dentro de una orden.  
OrdenID: Orden a la que pertenece el detalle. Referencia a Ordenes.  
ProductoID: Producto incluido en el detalle. Referencia a Productos.  
Cantidad: Cantidad del producto en esa orden.  
PrecioUnitario: Precio unitario del producto al momento de la orden.

## Tabla DireccionesEnvio

DireccionID: Identificador único de la dirección.  
UsuarioID: Usuario al que pertenece la dirección. Referencia a Usuarios.  
Calle: Calle y número.  
Ciudad: Ciudad del domicilio.  
Departamento: Departamento o unidad funcional (opcional).  
Provincia: Provincia o estado.  
Distrito: Distrito (opcional).  
Estado: Estado (opcional, en países donde aplique).  
CodigoPostal: Código postal.  
Pais: País de residencia.

# ESTRUCTURA BASE DE DATOS



## Tabla Carrito

CarritoID: Identificador único del ítem en el carrito.  
UsuarioID: Usuario al que pertenece el carrito.  
ProductoID: Producto agregado al carrito.  
Cantidad: Cantidad de ese producto en el carrito.  
FechaAgregado: Fecha y hora en que se agregó el producto.

## Tabla MetodosPago

MetodoPagоЯD: Identificador único del método de pago.  
Nombre: Nombre del método (por ejemplo, Tarjeta, Transferencia).  
Descripcion: Detalles adicionales sobre el método.

## Tabla OrdenesMetodosPago

OrdenMetodoID: Identificador único del registro.  
OrdenID: Orden asociada al pago.  
MetodoPagоЯD: Método de pago utilizado.  
MontoPagado: Monto pagado con ese método.

# ESTRUCTURA BASE DE DATOS

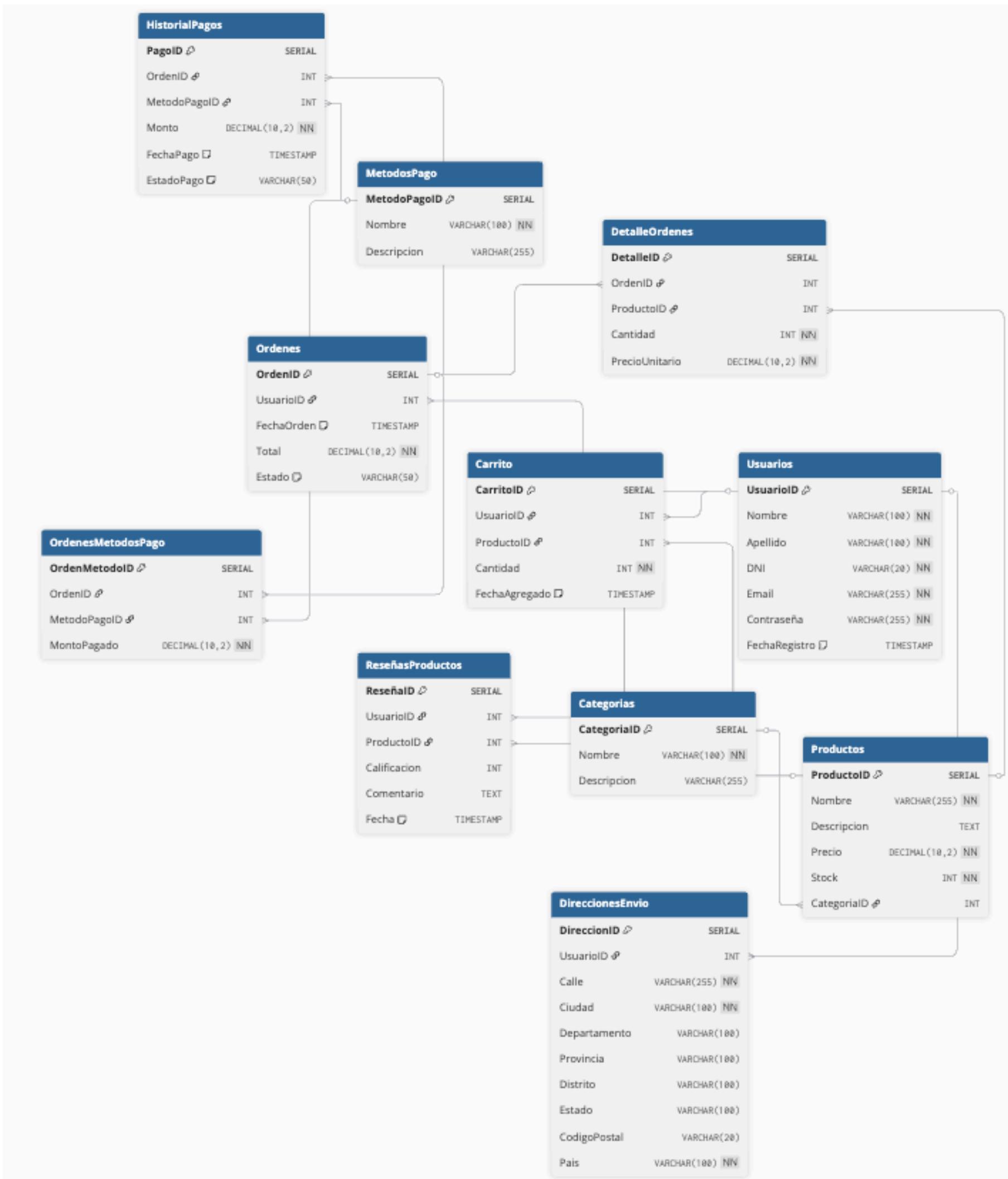


## Tabla ReseñasProductos

ReseñaID: Identificador único de la reseña.  
UsuarioID: Usuario que dejó la reseña.  
ProductoID: Producto reseñado.  
Calificacion: Puntaje entre 1 y 5.  
Comentario: Texto con la opinión del usuario.  
Fecha: Fecha de publicación de la reseña.

## Tabla HistorialPagos

PagoID: Identificador único del pago.  
OrdenID: Orden asociada al pago.  
MetodoPagoID: Método con el que se realizó el pago.  
Monto: Monto pagado.  
FechaPago: Fecha y hora del pago.  
EstadoPago: Estado del pago (por ejemplo: Procesando, Aprobado, Fallido).



# ANÁLISIS DE PREGUNTAS DE NEGOCIO Y DEFINICIÓN DE HECHOS Y DIMENSIONES (KIMBALL)



Para abordar las preguntas clave del negocio —como conocer qué productos se venden más, quiénes son los clientes más fieles o cuál es el ticket promedio— se identificaron los eventos transaccionales relevantes (hechos) y las entidades descriptivas relacionadas (dimensiones), siguiendo la metodología de modelado dimensional de Kimball.

## ► Tablas de hechos

- Ordenes: transacción principal que representa una compra realizada.
- DetalleOrdenes: descompone cada orden en sus productos y cantidades.
- OrdenesMetodosPago: permite analizar montos pagados por tipo de método.
- HistorialPagos: registra pagos concretos con su estado y fecha.
- Carrito: representa la intención de compra de los usuarios.
- ReseñasProductos: comportamiento post-compra que mide satisfacción.

## ► Tablas de dimensiones

- Usuarios: quién compra (nombre, email, país...).
- Productos: qué se vende.
- Categorías: clasificación de productos.
- MetodosPago: cómo se paga.
- DireccionesEnvio: a dónde se envía.

# DISEÑO DEL MODELO CONCEPTUAL Y LÓGICO (INCLUYENDO SCDS)

## ► Modelo conceptual

El modelo se estructura en torno a la tabla de hechos **Ordenes**, que representa una venta, y se conecta con otras tablas de hechos y dimensiones. Este enfoque permite responder múltiples preguntas de negocio con alta granularidad.

## ► Modelo lógico

El modelo sigue el esquema estrella (star schema), donde:

- **Ordenes** es el hecho central.
- Se conecta con las dimensiones: **Usuarios**, **Productos**, **MetodosPago**, **DireccionesEnvio**.
- **DetalleOrdenes** es una extensión que permite análisis a nivel de producto dentro de cada orden.

## ► SCDs (Slowly Changing Dimensions)

Se considera aplicar SCD Tipo 2 para:

- **Usuarios**: para mantener el historial si cambian de dirección o email.
- **Productos**: para conservar la historia si cambia el precio o la descripción.

Esto permitirá hacer análisis históricos precisos sin sobrescribir los datos anteriores.



# TRANSFORMACIONES Y DBT



La arquitectura medallón es un paradigma de lakehouse que organiza los datos en tres capas progresivas (Bronze, Silver y Gold), donde cada nivel agrega valor y refinamiento a los datos. Esta aproximación permite un balance óptimo entre flexibilidad, calidad y performance, facilitando tanto el análisis exploratorio como los casos de uso empresariales críticos.

Beneficios de la Arquitectura Medallón

Escalabilidad y Flexibilidad:

- Separación clara de responsabilidades permite equipos especializados
- Cada capa puede escalar independientemente según la demanda
- Facilita la incorporación de nuevas fuentes de datos

Calidad y Gobernanza:

- Control progresivo de calidad en cada capa
- Trazabilidad completa del linaje de datos
- Implementación consistente de políticas de datos

Performance y Eficiencia:

- Optimización específica por caso de uso
- Reutilización de transformaciones entre equipos
- Reducción de carga computacional en consultas finales

# BRONZE LAYER (STAGING)



La capa Bronze actúa como el punto de entrada de todos los datos al data lakehouse, preservando la fidelidad de la información original.

Características principales:

- Réplica exacta de datos fuente OLTP: Los datos se almacenan en su formato original sin transformaciones estructurales significativas
- Cambios mínimos: Se aplican únicamente transformaciones de tipado de datos básicas y limpieza mínima para garantizar la ingestión
- Preservación completa del historial: Mantiene un registro completo de todos los cambios y versiones de los datos
- Cargas incrementales con CDC: Implementa Change Data Capture para capturar únicamente los deltas, optimizando el rendimiento y reduciendo la latencia

Casos de uso:

- Auditoría y compliance regulatorio
- Análisis forense de datos
- Recuperación ante errores en capas superiores
- Exploración de datos no estructurados

# BRONZE LAYER (STAGING)



stg\_carrito.sql: Prepara los datos del carrito de compras, probablemente para analizar la conversión o el comportamiento del cliente durante el proceso de compra.

stg\_categorias.sql: Normaliza o categoriza los productos en la base de datos, facilitando análisis posteriores por tipo de producto o jerarquía.

stg\_detalle\_ordenes.sql: Desglosa cada orden en sus productos individuales, lo que permite hacer análisis detallados de ventas, combinaciones de productos y frecuencia de compra.

stg\_direcciones\_envio.sql: Procesa las direcciones de envío asociadas a las órdenes o a los usuarios, útil para análisis de distribución geográfica o eficiencia logística.

stg\_historial\_pagos.sql: Contiene los datos del historial de pagos realizados por los usuarios, fundamental para el análisis financiero, comportamiento de pago y tasas de conversión.

# BRONZE LAYER (STAGING)



stg\_metodos\_pago.sql: Relaciona los métodos de pago disponibles o utilizados por los usuarios, lo que permite entender preferencias y disponibilidad de medios de pago.

stg\_ordenes.sql: Representa las órdenes realizadas por los usuarios. Es una vista base clave para análisis de ventas, recurrencia de compra y rendimiento general.

stg\_productos.sql: Contiene información de los productos disponibles o vendidos. Sirve como fuente para análisis de catálogo, disponibilidad y rendimiento por producto.

stg\_resenas\_productos.sql: Vista que normaliza o transforma los datos de reseñas de productos, permitiendo análisis de satisfacción del cliente, calificaciones y feedback.

stg\_usuarios.sql: Contiene información sobre los usuarios registrados, como datos demográficos o de comportamiento. Fundamental para segmentaciones y análisis de clientes.

# SILVER LAYER (INTERMEDIATE)

La capa Silver transforma los datos brutos en información confiable y estructurada, aplicando reglas de negocio y estándares de calidad.

Características principales:

- Datos limpiados y enriquecidos: Eliminación de duplicados, corrección de inconsistencias, y enriquecimiento con datos de referencia
- Aplicación de reglas de negocio: Implementación de lógica empresarial compleja, cálculos derivados y validaciones específicas del dominio
- Slowly Changing Dimensions (SCD): Manejo de cambios históricos en dimensiones, preservando tanto el estado actual como los valores históricos
- Validaciones de calidad de datos: Implementación de controles automáticos para detectar anomalías, valores atípicos y problemas de integridad
- Seguimiento histórico: Mantenimiento de metadatos sobre la evolución y linaje de los datos

Casos de uso:

- Alimentación de modelos de machine learning
- Análisis de tendencias históricas
- Reportes operacionales complejos
- Integración de múltiples fuentes de datos





# SILVER LAYER (INTERMEDIATE)

`int_usuarios_enriquecidos.sql`: Vista intermedia que enriquece los datos de los usuarios con información adicional como comportamiento, historial de compras o ubicación.

`int_resenas_agregadas.sql`: Agrega o consolida las reseñas por producto o usuario. Es útil para calcular promedios de puntuación, cantidad de reseñas y clasificaciones generales.

`int_productos_enriquecidos.sql`: Vista que une información básica de productos con datos adicionales como categoría, métricas de rendimiento y reseñas.

`int_metricas_productos.sql`: Calcula métricas clave por producto como el promedio de calificación, número de ventas, cantidad de reseñas u otros KPIs relevantes.

`int_detalle_ordenes_enriquecido.sql`: Vista detallada de las órdenes que incluye información adicional como detalles del producto, usuario asociado y posiblemente reseñas o pagos.

# GOLD LAYER (DATA MARTS)



La capa Gold representa el nivel más refinado de datos, optimizada específicamente para el consumo analítico y la toma de decisiones empresariales.

Características principales:

- Datos agregados listos para el negocio: Información pre-procesada y estructurada según los requerimientos específicos de cada área de negocio
- Esquemas en estrella optimizados para KPI: Diseño dimensional que facilita consultas analíticas rápidas y intuitivas
- Métricas pre-calculadas: Indicadores clave de rendimiento computados previamente para garantizar respuestas instantáneas
- Optimización de performance para analytics: Particionado, indexación y materialización de vistas para maximizar la velocidad de consulta

Casos de uso:

- Dashboards ejecutivos en tiempo real
- Reportes regulatorios y de compliance
- Self-service analytics para usuarios de negocio
- Alimentación de herramientas de BI y visualización

# GOLD LAYER (DATA MARTS)



ventas\_mensuales.sql: Modelo que calcula las ventas mensuales agregadas, útil para análisis de crecimiento, comportamiento estacional y reportes periódicos.

productos\_mas\_vendidos.sql: Identifica los productos con mayor volumen de ventas. Este modelo permite detectar los artículos más populares, optimizar el inventario y definir estrategias de promoción.

mart\_ordenes\_completadas.sql: Contiene todas las órdenes que fueron completadas con éxito. Consolidación fundamental para análisis comerciales, tasas de conversión y métricas de rendimiento.

kpi\_valor\_total\_ventas.sql: Calcula el valor total de las ventas realizadas, una métrica financiera clave para evaluar el desempeño general del negocio.

kpi\_ticket\_promedio.sql: Métrica que estima el ticket promedio por orden, clave para entender el poder adquisitivo promedio de los clientes y evaluar campañas de upselling.

kpi\_satisfaccion\_productos.sql: KPI que evalúa la satisfacción de los productos a partir de reseñas y calificaciones, útil para monitorear la calidad percibida y la experiencia del cliente.

# GOLD LAYER (DATA MARTS)



kpi\_recompra\_clientes.sql: Métrica que indica la tasa de recompra de los clientes, útil para medir la fidelización, retención y eficacia de programas de lealtad.

conversion\_por\_categoria.sql: Presenta la tasa de conversión segmentada por categoría de producto, lo que permite identificar qué categorías convierten mejor y dónde hay oportunidades de mejora.

# STORYTELLING CON DATOS



El negocio está experimentando un buen rendimiento en términos de ventas totales, pero se identifican áreas clave donde se pueden realizar mejoras para aumentar la satisfacción del cliente y mejorar la eficiencia operativa.

Ventas totales y comportamiento de compra

**Valor total de ventas:** ARS \$1,260,743.34

Esto indica una operación activa con un flujo considerable de ingresos.

**Ticket promedio por orden:** ARS \$504.70

Aunque hay un volumen interesante de ventas, el gasto por orden es relativamente bajo, lo que podría sugerir compras frecuentes de productos económicos.

Retención de clientes

**Tasa de recompra:** 78.1%

Un excelente indicador de fidelización. Significa que más de tres cuartas partes de los clientes realizan más de una compra, lo que habla muy bien del servicio postventa o la variedad de oferta.

# STORYTELLING CON DATOS



## Puntaje promedio de satisfacción: 3.0 de 5

Este es el punto más débil detectado. El nivel de satisfacción está justo en el umbral de lo aceptable. La mayoría de los productos tienen calificaciones entre 2.8 y 3.1, lo cual es bajo para productos con alta rotación.

## Productos y stock

Todos los productos analizados están en stock normal o alto.

Hay una fuerte presencia de productos de tecnología, moda y cuidado personal, todos con altas ventas, pero no todos con buenas calificaciones.

# INSIGHTS



Analizando más los datos se puede identificar varios puntos positivos como:

- Alta tasa de recompra: lo que indica una fidelización sólida.
- Alto nivel de ventas: buen volumen de ventas y stock bien abastecido.

Por otro lado existen algunas oportunidades de mejora como:

- Satisfacción del cliente baja: el promedio es muy bajo debemos poner foco en mejora de experiencia.
- Ticket promedio bajo: sería genial poder aumentar el valor del ticket por ejemplo sumando cross selling de otros productos.

# DASHBOARD

The screenshot shows a Streamlit dashboard with a yellow header bar containing the word "DASHBOARD". Below the header, there are three tabs: "KPIs" (selected), "Compras", and "Productos". The "KPIs" tab displays four performance indicators:

Indicador	Valor
valor_total_ventas	\$1,260,743.34
ticket_promedio	\$504.70
satisfaccion_productos	3.00
recompra_clientes	78.1%

Below the KPIs section, there is descriptive text explaining the dashboard's functionality and the data source.

Para ver los diferentes resultados cree un dashboard con la herramienta de Streamlit en cual cuenta con tres pestañas. Todos los datos son obtenidos desde las distintas vistas creadas en el modelado y las transformaciones DBT.

La primera como muestra la imagen se ven los cuatro kpis seleccionado.

En el segundo tab se pueden ver algunos gráficos de las compras por ejemplo:

- Ventas por categorías
- Cantidad vendida por producto
- Cantidad de stock por producto
- Total de órdenes por producto
- Cantidad vendida vs. Precio del producto

Por último en el tercera tab vamos a poder ver los diferentes gráficos relacionados con los productos. Entre ellos podemos ver:

- Top productos más vendidos.
- Distribución de calificaciones promedio.
- Precio vs calificación.



# GRACIAS!!

DIEGO LOPEZ CASTAN