

# Introducción al Marketing Online y los Negocios Digitales

Trabajo práctico

## Consigna

### Objetivo

Diseñar e implementar un mini–ecosistema de datos comercial (online + offline) y construir un dashboard que sirva como reporte para un área comercial, con KPIs clave: Ventas, Usuarios Activos, Ticket Promedio, NPS, Ventas por Provincia y Ranking Mensual por Producto.

Se trabajará con las tablas y campos descriptos en el documento (canales, provincias, categorías, clientes, direcciones, tiendas, productos, pedidos, ítems, pagos, envíos, sesiones web y NPS).

### Tareas

#### 1. Modelado & DDL

- Crear el esquema estrella según modelado de kimball con **PK**, **FK** y tipos adecuados para todas las tablas provistas según lo visto en clase.
- Documentar en un diccionario de datos, claves, dominios y supuestos.

#### 2. Generación de datos de muestra

- En el siguiente repositorio [https://github.com/AugustoCarmona/mkt\\_tp\\_final](https://github.com/AugustoCarmona/mkt_tp_final) se encuentra dentro de la carpeta RAW cada una de las tablas descritas a continuación en formato .CSV para trabajar de forma local luego de haber hecho el fork correspondiente al repositorio.

#### 3. Transformaciones (Python)

- Dentro del repositorio clonado se deben desarrollar los archivos .py correspondientes para calcular cada KPI (ver definiciones abajo).
- El objetivo es que cada script python convierte las tablas denormalizadas a un nuevo directorio DW, el cual actuará como data warehouse y contendrá los archivos .CSV con la información de las tablas denormalizadas.
- Se deben crear todas las dimensiones y hechos posibles más allá de las requeridas para configurar el dashboard solicitado.

#### 4. Dashboard

- Con la herramienta LookerStudio tal como veremos en la próxima clase se deberá desarrollar un Dashboard como el presente más adelante en este documento que tome las tablas de la carpeta DW como fuente.
- Filtros: **fecha**, **canal**, **provincia** y **producto**.
- Vistas mínimas:

- Ventas (serie temporal y tarjeta \$M).
- Usuarios Activos (serie temporal y tarjeta nK).
- Ticket Promedio (\$K).
- NPS (tarjeta y tendencia).
- Ventas por Provincia (mapa o barra).
- Ranking mensual por Producto (top N).

## 5. Entregables

- **Repositorio** con: scripts de carga, capturas/enlace del dashboard y README. Debe contener todas las buenas prácticas vistas en clase incluyendo la gestión de un entorno virtual con requirements.txt para la instalación local y conventional commits como estándar de commit. Toda la gestión del repositorio debe hacerse con la consola y el uso de la consola se evaluará durante la entrega del mismo.
- **README** con: instrucciones de ejecución, supuestos, diccionario de datos y consultas clave.
- **Tablero** terminado en LookerStudio.

## 6. Criterios de evaluación (orientativo)

- Correctitud del modelo y consultas (40%).
- Calidad del dashboard (40%).
- Claridad del README/documentación (15%).
- Profundidad de hallazgos y nuevas ideas al dashboard (5%).

# Caso de uso: EcoBottle

EcoBottle AR vende dos botellas reutilizables: Classic A y Sport B. Opera tienda online y 4 tiendas físicas (BA, CBA, SF, MZA). En ONLINE, los clientes compran con tarjeta o mercado pago; los pedidos se envían con Correo Argentino. En OFFLINE, las tiendas registran ventas en POS y stock local. Marketing corre campañas en redes (tráfico “ads”) y email (tráfico “direct/referral”).

Atención al cliente envía una encuesta NPS 7 días después del pedido entregado (en tienda, al día siguiente). El management monitorea diariamente: ventas totales, ticket promedio, usuarios activos, NPS y el ranking mensual por producto.

Desde la gerencia comercial solicitaron al área de Data hacer un tablero para poder compartir con el C-level y otros stakeholders interesados en conocer el crecimiento de las ventas por producto, provincia y canal de venta.

**Meta del trimestre:** crecer 15% ventas en Córdoba y reducir tiempos de entrega en Mendoza.

## Catálogos / Referencias

- **channel** catálogo de canales de venta (online/offline u otros). Se usa para clasificar pedidos, sesiones y respuestas NPS.
  - channel\_id INT
  - code VARCHAR(20) UNIQUE — ej. 'ONLINE', 'OFFLINE'
  - name VARCHAR(50)
- **province** catálogo de provincias/estados. Normaliza la localización de direcciones para análisis geográfico.
  - province\_id INT
  - name VARCHAR(50) — ej. Buenos Aires, Córdoba, Santa Fe, Mendoza
  - code VARCHAR(10) UNIQUE — opcional
- **product\_category** jerarquía de categorías de producto (opcionalmente con padre-hijo). Permite agrupar ventas por familia.
  - category\_id INT
  - name VARCHAR(80)
  - parent\_id INT NULL → product\_category.category\_id

## Maestros

- **customer** maestro de clientes con datos de identificación y alta/baja. Referenciado por pedidos, sesiones web y NPS.
  - customer\_id INT
  - email VARCHAR(120) UNIQUE
  - first\_name VARCHAR(80)
  - last\_name VARCHAR(80)
  - phone VARCHAR(30) NULL
  - status CHAR(1) DEFAULT 'A' CHECK (status IN ('A','I'))
  - created\_at TIMESTAMP NOT NULL
- **address** direcciones normalizadas (facturación, envío, tiendas). Relaciona con province para mapas y agregaciones.
  - address\_id INT
  - line1 VARCHAR(120)
  - line2 VARCHAR(120) NULL
  - city VARCHAR(80)
  - province\_id INT → province.province\_id
  - postal\_code VARCHAR(20)
  - country\_code CHAR(2) DEFAULT 'AR'
  - created\_at TIMESTAMP NOT NULL
- **store** (para canal offline) puntos de venta físicos. Enlaza a address; se usa en pedidos del canal offline e inventario.
  - store\_id INT

- name VARCHAR(80)
  - address\_id INT → address.address\_id
- **product** maestro de productos (SKU, nombre, precio de lista, categoría, estado). Referenciado por ítems de pedido e inventario.
  - product\_id INT
  - sku VARCHAR(40) UNIQUE
  - name VARCHAR(120)
  - category\_id INT NULL → product\_category.category\_id
  - list\_price DECIMAL(12,2) NOT NULL
  - status CHAR(1) DEFAULT 'A' CHECK (status IN ('A','I'))
  - created\_at TIMESTAMP NOT NULL

## Transaccionales

- **sales\_order** cabecera del pedido/venta (cliente, canal, fechas, importes y estados). Origen de KPIs de ventas y ticket.
  - order\_id BIGINT
  - customer\_id INT → customer.customer\_id
  - channel\_id INT → channel.channel\_id
  - store\_id INT NULL → store.store\_id — NULL si ONLINE
  - order\_date TIMESTAMP NOT NULL
  - billing\_address\_id INT NULL → address.address\_id
  - shipping\_address\_id INT NULL → address.address\_id
  - status VARCHAR(20) CHECK (status IN ('CREATED','PAID','CANCELLED','FULFILLED','REFUNDED'))
  - currency\_code CHAR(3) DEFAULT 'ARS'
  - subtotal DECIMAL(12,2) NOT NULL
  - tax\_amount DECIMAL(12,2) NOT NULL
  - shipping\_fee DECIMAL(12,2) NOT NULL
  - total\_amount DECIMAL(12,2) NOT NULL
- **sales\_order\_item** detalle de líneas del pedido (producto, cantidad, precios y descuentos). Base para ranking por producto.
  - order\_item\_id BIGINT
  - order\_id BIGINT → sales\_order.order\_id
  - product\_id INT → product.product\_id
  - quantity INT NOT NULL CHECK (quantity > 0)
  - unit\_price DECIMAL(12,2) NOT NULL
  - discount\_amount DECIMAL(12,2) NOT NULL DEFAULT 0
  - line\_total DECIMAL(12,2) NOT NULL — (quantity\*unit\_price - discount\_amount)
- **payment** registros de cobros asociados a pedidos (método, estado, montos, timestamps). Permite conciliar ventas vs. pagos.
  - payment\_id BIGINT
  - order\_id BIGINT → sales\_order.order\_id

- method VARCHAR(20) CHECK (method IN ('CASH','CARD','TRANSFER','GATEWAY'))
  - status VARCHAR(20) CHECK (status IN ('PENDING','PAID','FAILED','REFUNDED'))
  - amount DECIMAL(12,2) NOT NULL
  - paid\_at TIMESTAMP NULL
  - transaction\_ref VARCHAR(80) NULL
- shipment** (para entregas ONLINE o envíos desde tienda) envíos asociados a pedidos (carrier, tracking, estado y fechas). Traza la logística y tiempos de entrega.
  - shipment\_id BIGINT
  - order\_id BIGINT → sales\_order.order\_id
  - carrier VARCHAR(40) NULL
  - tracking\_number VARCHAR(60) NULL
  - status VARCHAR(20) CHECK (status IN ('READY','SHIPPED','DELIVERED','CANCELLED'))
  - shipped\_at TIMESTAMP NULL
  - delivered\_at TIMESTAMP NULL

### Actividad Digital (para “Usuarios activos”)

- web\_session** sesiones de navegación/app (cliente opcional, timestamps, fuente y dispositivo). Fuente para “usuarios activos”.
  - session\_id BIGINT
  - customer\_id INT NULL → customer.customer\_id — permite anónimo
  - started\_at TIMESTAMP NOT NULL
  - ended\_at TIMESTAMP NULL
  - source VARCHAR(50) NULL — ej. ‘direct’, ‘ads’, etc.
  - device VARCHAR(30) NULL

### Encuestas (NPS)

- nps\_response** respuestas de encuesta NPS (score, comentario, canal, fecha y cliente opcional). Base para KPI de NPS.
  - nps\_id BIGINT
  - customer\_id INT NULL → customer.customer\_id
  - channel\_id INT → channel.channel\_id
  - score SMALLINT NOT NULL CHECK (score BETWEEN 0 AND 10)
  - comment TEXT NULL
  - responded\_at TIMESTAMP NOT NULL

### Cómo se cubre el tablero

- \$M Total Ventas:** SUM(total\_amount) de *sales\_order* con status IN ('PAID','FULFILLED'), filtrable por channel\_id y período en order\_date.
- nK Usuarios Activos:** COUNT(DISTINCT customer\_id) (o session\_id si anónimos) en *web\_session* por período.

- **\$K Ticket Promedio:**  $SUM(total\_amount) / COUNT(*)$  de *sales\_order* (mismo filtro que Ventas).
- **NPS:**  $((\%9-10) - (\%0-6)) * 100$  sobre *nps\_response* por período y canal.
- **Ventas por provincia:** agrupar *sales\_order.total\_amount* por *shipping\_address\_id* → *address.province\_id*.
- **Ranking ventas por producto (mensual):** agrupar *sales\_order\_item.line\_total* por *product\_id* y  $DATE\_TRUNC('month', order\_date)$ .

Modelo del tablero solicitado (imagen ilustrativa)

