



LET'S GO!/>

# Análisis y visualización de datos de ventas del sector retail

PIM3 - Data Science



→ [soyhenry.com](https://soyhenry.com)



# Proyecto Integrador

## Contexto del proyecto y rol del estudiante:

TechCore es una cadena de tiendas minoristas especializada en la venta y distribución de computadores de todas las gamas, ha contratado a DataVision Analytics para centralizar y analizar la información de sus ventas a nivel nacional. El objetivo es transformar los datos crudos de facturación en información estructurada, crear un modelo relacional y desarrollar un dashboard interactivo en Power BI que permita extraer conclusiones estratégicas. Para ello, se ha diseñado un reto técnico dirigido a candidatos para el rol de Científico de Datos Junior, cuyo objetivo es demostrar su capacidad para integrar múltiples fuentes de datos, analizarlas y generar insights accionables.

Como parte del proceso de selección para el rol de Analista de Datos Junior, DataVision Analytics ha diseñado un reto técnico en el que el candidato deberá completar las 3 fases del proyecto.





# Objetivos del PI

- Utilizar Power Query para aplicar técnicas de limpieza y transformación de datos garantizando consistencia, integridad y calidad.
- Crear las tablas de un modelo relacional con Python a partir de la base de datos inicial, identificando las entidades principales y sus relaciones.
- Desarrollar un dashboard interactivo en Power BI para visualizar métricas clave como ventas totales, ventas por ciudad, productos más vendidos y comportamiento de clientes.
- Interpretar los resultados del análisis para generar recomendaciones orientadas a mejorar las ventas y optimizar la estrategia comercial.
- Documentar el proceso de forma clara, estructurada y reproducible.





<Proyecto Integrador>

# Entregable

# Final



# Proyecto Integrador



El entregable debe ser un link a una carpeta de drive con la siguiente identificación:

**ProyectoM3\_NombreEstudiante**

**Contenido:**

**Carpeta principal:** ProyectoM3\_NombreEstudiante/

## 1. Notebooks

- | — Avances/
  - | | — Avance\_1/
    - | | | — Avance\_1\_Limpieza\_Transformacion.pbix
    - | | | — ventas.csv
    - | | | — ventasTransformed.csv
  - | | — Avance\_2/
    - | | | — Avance\_2\_Modelo\_Relacional.ipynb
    - | | | — modeloVentas.xlsx
  - | | — Avance\_3/
    - | | | — Avance\_3\_Dashboard\_PowerBI.pbix



Recuerda habilitar los accesos a la carpeta





# Proyecto Integrador

## 3. Documentación

- | | — README.pdf o .docx: Pasos realizados, hallazgos principales
- | | — Conclusiones\_Recomendaciones.pdf: Insights y propuestas de mejora





<Proyecto Integrador>

# Detalle de **Avances**



# Proyecto Integrador



## Detalle de Avance#1:

- **Detalle de avance:**

Importar la base de datos cruda de facturas a Power BI.

Estandarizar nombres de columnas y tipos de datos.

Detectar y manejar valores nulos, duplicados y registros inconsistentes.

Normalizar campos como categorías de productos, nombres de ciudades y métodos de pago.

Exportar el dataset limpio para su uso en Python (Avance 2).

- **Conocimientos necesarios:**

Uso de Power Query para transformación y limpieza de datos.

Manejo de tipos de datos en Power BI.

Técnicas de normalización y estandarización de valores.

- **Tech Stack necesario:**

Power BI Desktop y Power Query





# Proyecto Integrador



## Detalle de Avance#1:

### → Notas extra:

#### Recomendaciones para la limpieza.

*Renombrar columnas para uniformidad:* cambiar Prod\_Name a ProductoNombre.

*Convertir tipos de datos:* pasar fechas de texto a formato date, precios a decimal number, etc.

*Eliminar duplicados:* registros repetidos de facturas exactamente iguales.

*Reemplazar valores nulos:* por ejemplo, si MetodoPago es nulo, poner "No especificado" (y para las demás variables categóricas), imputar valores numéricos nulos por cero, en caso de que sea adecuado, etc.

*Normalizar categorías:*

"elec-trónica", "Electronica" → "Electrónica".

"medellin", "Medellín" → "Medellín".

*Calcular columnas derivadas que puedan mejorar el análisis:* por ejemplo, sacar el año y mes de la FechaVenta.

Guarda el .pbix con los pasos realizados en PowerQuery como **Avance\_1\_Limpieza\_Transformacion.pbix** y un **nuevo** .csv con el resultado de las transformaciones como: ventasTransformed.csv



# Proyecto Integrador

## Detalle de Avance#2:

- **Detalle de avance:**

- Cargar el dataset limpio exportado desde Power Query.
- Identificar entidades y relaciones necesarias para el modelo relacional.
- Definir claves primarias y foráneas para garantizar integridad referencial.
- Generar tablas base: Facturas, DetalleFacturas, Productos, Clientes, Sucursales y Vendedores. Realizar un dibujo detallado del diagrama entidad/relaciones identificando claramente las relaciones, su cardinalidad, llave primaria y llave foránea.
- Implementar el modelo relacional usando pandas y opcionalmente SQLite.
- Validar consistencia de datos y relaciones.
- Una vez creadas las tablas por medio de Python, generar el modelo relacional en Power BI con dichas tablas.

- **Conocimientos necesarios:**

- Lectura y manipulación de datasets con pandas.
- Diseño de modelos relacionales y normalización de datos.
- Uso de claves primarias y foráneas

### Tech Stack necesario:

Python, pandas, numpy, Power BI.

# Proyecto Integrador



## Detalle de Avance#2:

### Notas extra:

#### Recomendaciones para el modelo relacional.

- Crear tabla Facturas con: FacturaID, Fecha, SucursalID, ClienteID, VendedorID.
- Crear tabla DetalleFacturas con: DetalleID, FacturaID, ProductoID, Cantidad, Descuento.
- Crear tabla Productos con: ProductoID, Nombre, Marca, Precio.
- Crear tabla Clientes con: ClienteID, Nombre, Género, Edad, Teléfono, Email, Dirección.
- Crear tabla Ciudades con: CiudadID, NombreCiudad.

#### Validar integridad referencial. Algunos ejemplos:

- Si FacturaID en DetalleFacturas no existe en Facturas → error.
- Si ProductoID no está en Productos → error.

#### Generar reportes rápidos en Python para verificar datos:

- Total de ventas por marca.
- Top 10 productos más vendidos

Guarda el .ipynb que genera las tablas para alimentar el modelo relacional en Power BI como Avance\_2\_Modelo\_Relacional.ipynb. El resultado de este notebook en de Python debe ser un archivo de Excel modeloVentas.xlsx que finalmente será el que alimente el modelo para las visualizaciones del avance 3



# Proyecto Integrador



## Detalle de Avance#3:



### 1. Detalle de avance:

Importar el modelo relacional desde Python a Power BI.

Crear medidas DAX para indicadores clave: ventas totales, ticket promedio, margen, participación por ciudad/categoría.

Diseñar segmentaciones dinámicas por ciudad, marca de producto y periodo de tiempo.

Implementar visualizaciones interactivas con filtros y jerarquías.

Presentar el dashboard final con conclusiones y KPIs destacados.

### 1. Conocimientos necesarios:

Creación de medidas DAX en Power BI.

Diseño de dashboards interactivos.

Uso de segmentadores, jerarquías y filtros en Power BI.

### 1. Tech Stack necesario: Power BI Desktop, Lenguaje DAX



# Proyecto Integrador



## Detalle de Avance#3:

### → Notas extra:

#### Recomendaciones para la visualización

##### Filtros (segmentadores):

- Ciudad.
- marca de producto (Lenovo, Hacer, Asus...).
- Rango de fechas.
- Método de pago (Tarjeta, Efectivo, Transferencia...).
- Rango de edad del cliente.

##### Medidas DAX:

- VentasTotales
- TicketPromedio
- %ParticipacionCiudad

##### Visualizaciones interactivas:

- Mapa geográfico con ventas por ciudad.
- Gráfico de barras de ventas por marca.
- Línea de tiempo con evolución de ventas.
- Tarjetas con KPIs destacados (ventas, ticket promedio, Productos vendidos).

##### Jerarquías:

- Fecha → Año → Mes → Día.
- Ciudad → Sucursal.

## Detalle de Avance#4:

### EXTRA CREDIT

Como parte del reto técnico, se solicita al candidato implementar un esquema de seguridad a nivel de fila (Row-Level Security, RLS) en el modelo de datos de Power BI, con el fin de restringir el acceso a la información según el rol del usuario que visualiza el dashboard.

### Requisitos

- Definir dos roles de usuario en Power BI Desktop, tales como:
  - Gerente Nacional: tiene acceso completo a todas las ciudades.
  - Gerente Regional: puede visualizar únicamente los datos de ventas correspondientes a su ciudad.
  - Gerente Sucursal: visualiza solo los datos de su sucursal específica.
- Crear una tabla de usuarios (puede ser simulada) que contenga:
  - Nombre del usuario
  - Correo electrónico (simulado)
  - Rol asignado
  - Ciudad o tienda asociada
- Establecer relaciones entre la tabla de usuarios y las tablas de ventas, regiones y tiendas, para aplicar filtros dinámicos.
- Configurar las reglas de RLS en Power BI Desktop utilizando expresiones DAX que filtren los datos según el rol y la ubicación del usuario.