



# TRABAJO PRÁCTICO 3: Minería de Datos con Pandas

**Materia:** Modelizado de Minería de Datos

**Docente:** Fernández, David

**Alumno:** Oviedo, Lucas Nahuel

**Carrera:** Tecnicatura en Ciencia de Datos e Inteligencia Artificial

**Institución:** INSTITUTO SUPERIOR DE FORMACIÓN TÉCNICA N° 197

**Año:** 2025

## 1. Introducción

Este trabajo práctico se centra en el análisis de datos de ventas utilizando Python y la biblioteca Pandas. Se realiza un análisis exploratorio de datos (EDA) sobre un conjunto de datos de ventas, incluyendo clientes, productos, facturas y rubros. El objetivo es extraer insights valiosos sobre el comportamiento de ventas, ranking de clientes, productos más vendidos y análisis temporal.

## 2. Descripción del Problema

El problema consiste en analizar datos de ventas de una empresa ficticia. Los datos incluyen información sobre clientes, productos, facturas y ventas. Se busca responder preguntas como: ¿Quiénes son los clientes más importantes?, ¿Cuáles son los productos más vendidos?, ¿Cómo varían las ventas a lo largo del tiempo?, entre otros.

### 2.1 Datasets Utilizados

- **clientes.csv:** Información de los clientes (id\_cliente, nombre, etc.)
- **productos.csv:** Catálogo de productos (id\_producto, descripcion, precio, etc.)
- **rubros.csv:** Clasificación de productos por rubros

- **facturas\_encabezados\_simulados.csv:** Encabezados de facturas (id\_factura, fecha, id\_cliente, etc.)
- **facturas\_detalle\_simulados.csv:** Detalles de facturas (id\_factura, id\_producto, cantidad, etc.)
- **ventas\_simuladas.csv:** Datos de ventas consolidados

## 3. Implementación del Análisis

### 3.1 Carga de Datos

Se cargan los archivos CSV utilizando Pandas y se verifican las columnas de cada dataset.

```
productos = pd.read_csv('productos.csv')
rubros = pd.read_csv('rubros.csv')
clientes = pd.read_csv('clientes.csv')
facturas_encabezados = pd.read_csv('facturas_encabezados_simulados.csv')
facturas_detalle = pd.read_csv('facturas_detalle_simulados.csv')
ventas = pd.read_csv('ventas_simuladas.csv')
```

### 3.2 Unificación de Datos

Se unen los datasets para crear una vista consolidada de ventas con información de clientes.

```
ventas_clientes = pd.merge(ventas, facturas_encabezados, on='id_factura', how='left')
ventas_clientes = pd.merge(ventas_clientes, clientes, on='id_cliente', how='left')
```

### 3.3 Análisis de Clientes

Se calcula el ranking de clientes por total de ventas y el ticket promedio por cliente.

```
ranking_clientes = ventas_clientes.groupby('nombre_cliente')
['total_venta'].sum().reset_index()
ranking_clientes = ranking_clientes.sort_values(by='total_venta', ascending=False)

ticket_promedio = ventas_clientes.groupby('nombre_cliente')
['total_venta'].mean().reset_index()
```

### 3.4 Análisis Temporal

Se analiza la evolución de ventas mensuales y semanales (específicamente para septiembre).

```
ventas_clientes['fecha'] = pd.to_datetime(ventas_clientes['fecha'], format='%d/%m/%Y')
ventas_por_mes = ventas_clientes.groupby(ventas_clientes['fecha'].dt.to_period('M'))
['total_venta'].sum().reset_index(name='total_venta')

septiembre_ventas = ventas_clientes[ventas_clientes['fecha'].dt.month == 9]
ventas_semanales_septiembre =
septiembre_ventas.groupby(septiembre_ventas['fecha'].dt.isocalendar().week)
['total_venta'].sum().reset_index()
```

### 3.5 Análisis de Productos

Se identifica el producto más vendido en cantidad y el ranking por facturación total.

```
productos_vendidos = facturas_detalle.groupby('id_producto')['cantidad'].sum().reset_index()
productos_vendidos = pd.merge(productos_vendidos, productos, on='id_producto')
```

```
detalle_conPrecio = pd.merge(facturas_detalle, productos, on='id_producto')
detalle_conPrecio['importe'] = detalle_conPrecio['cantidad'] * detalle_conPrecio['precio']
ranking_productos = detalle_conPrecio.groupby('descripcion')['importe'].sum().reset_index()
```

### 3.6 Matriz BCG

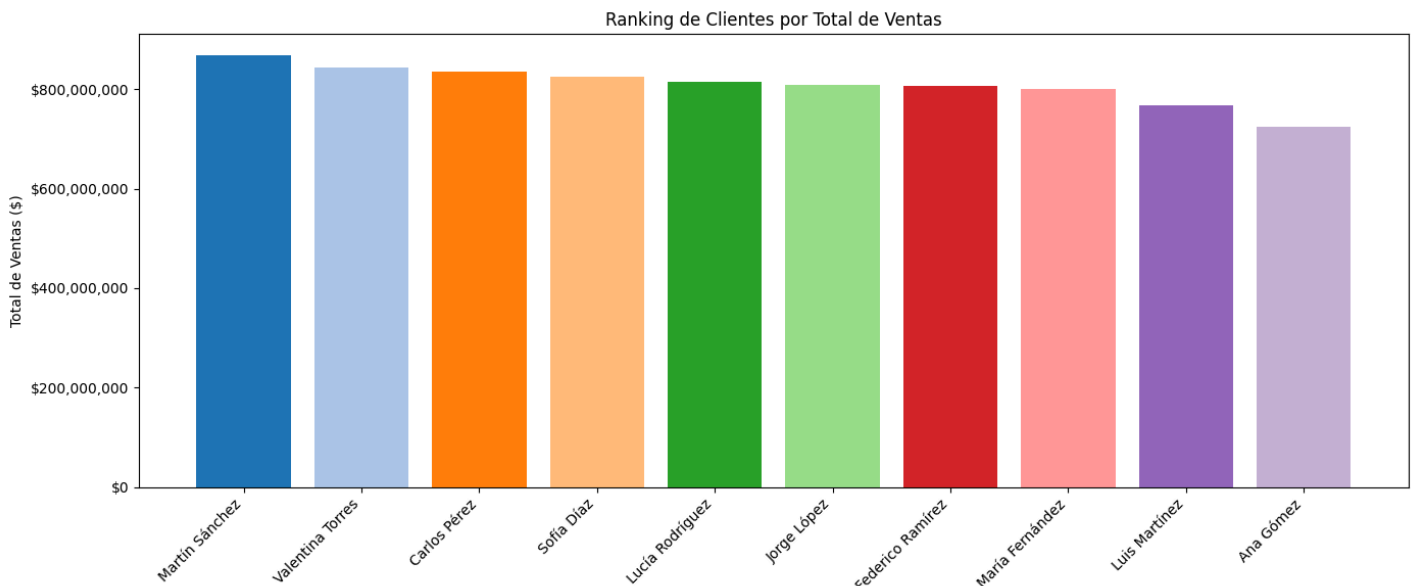
Se implementa la matriz BCG para clasificar productos según crecimiento e ingresos.

```
detalle = pd.merge(facturas_detalle, productos, on='id_producto')
detalle = pd.merge(detalle, facturas_encabezados[['id_factura', 'fecha']],
left_on='id_facturaENC', right_on='id_factura', how='left')
detalle['fecha'] = pd.to_datetime(detalle['fecha'], format='%d/%m/%Y')
detalle['mes'] = detalle['fecha'].dt.to_period('M')
detalle['importe'] = detalle['cantidad'] * detalle['precio_unitario']
ingresos_mes = detalle.groupby(['descripcion', 'mes'])['importe'].sum().reset_index()
pivot = ingresos_mes.pivot(index='descripcion', columns='mes', values='importe').fillna(0)
pivot['ingresos_totales'] = pivot.sum(axis=1)
# Cálculo de crecimiento y asignación de cuadrantes...
```

## 4. Resultados y Visualizaciones

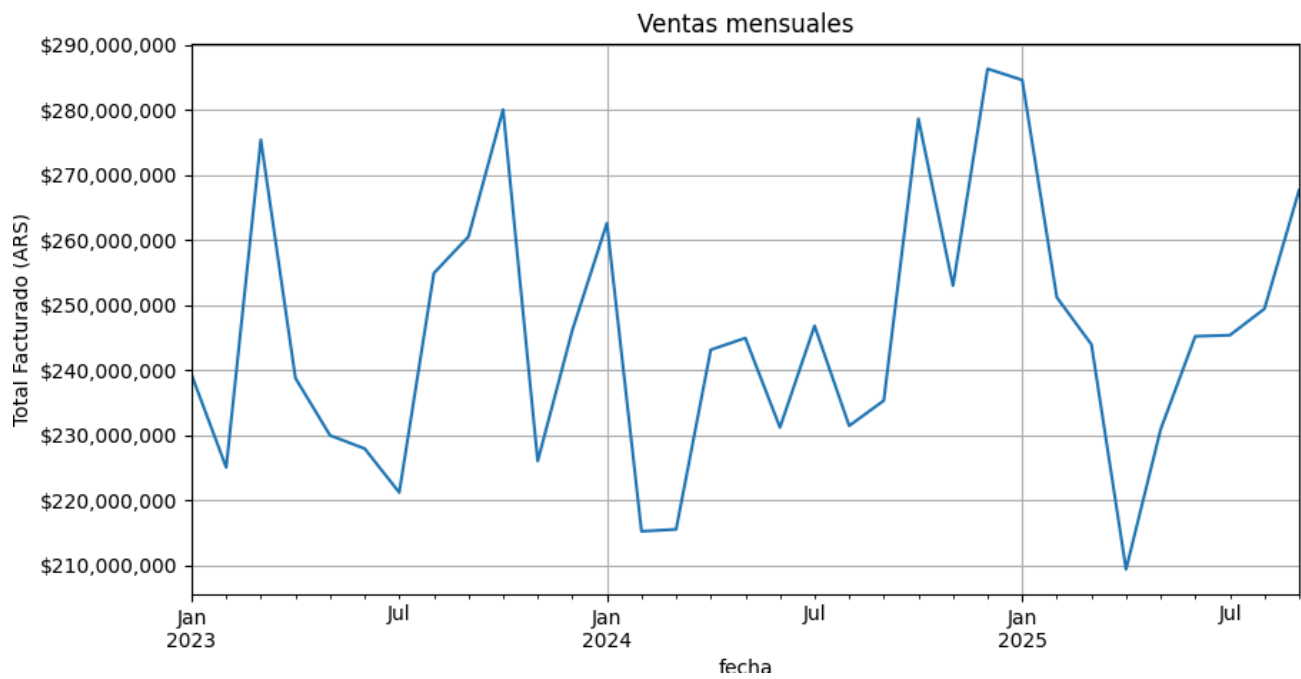
### 4.1 Ranking de Clientes

Se presenta un gráfico de barras mostrando el total de ventas por cliente, ordenado de mayor a menor.



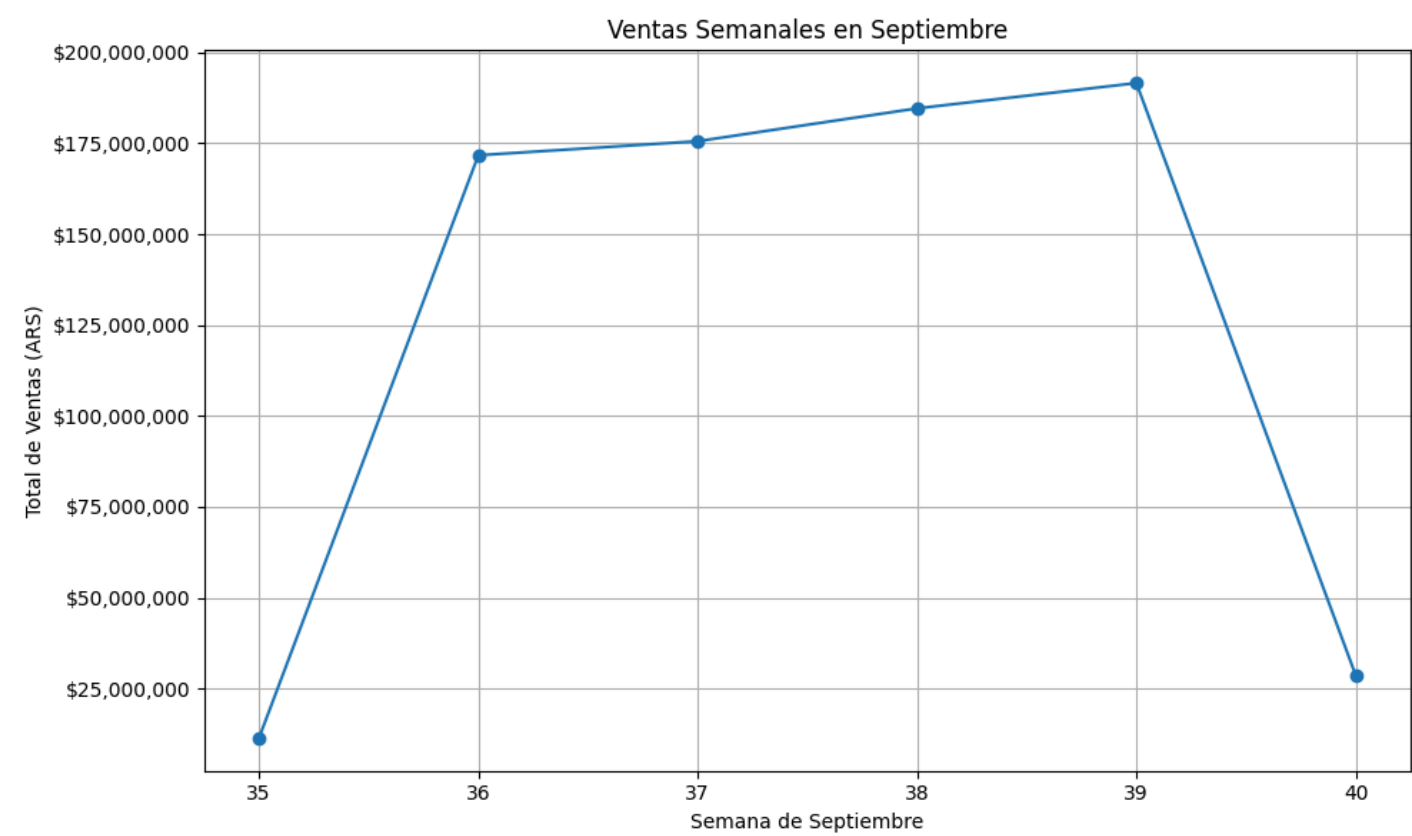
### 4.2 Ventas Mensuales

Gráfico de línea mostrando la evolución mensual de las ventas totales.



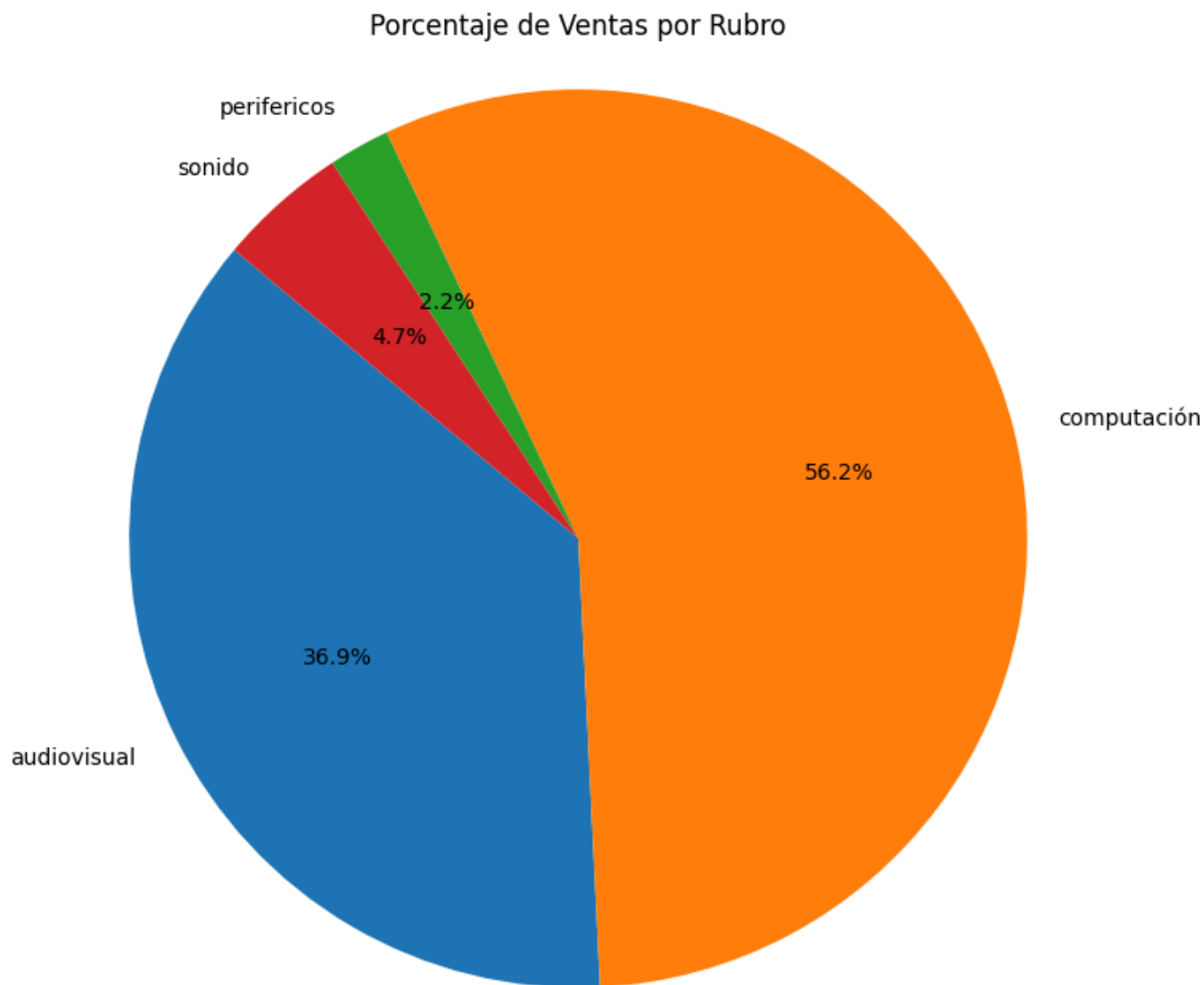
### 4.3 Ventas Semanales en Septiembre

Análisis detallado de las ventas por semana durante el mes de septiembre.



### 4.4 Ventas por Rubro

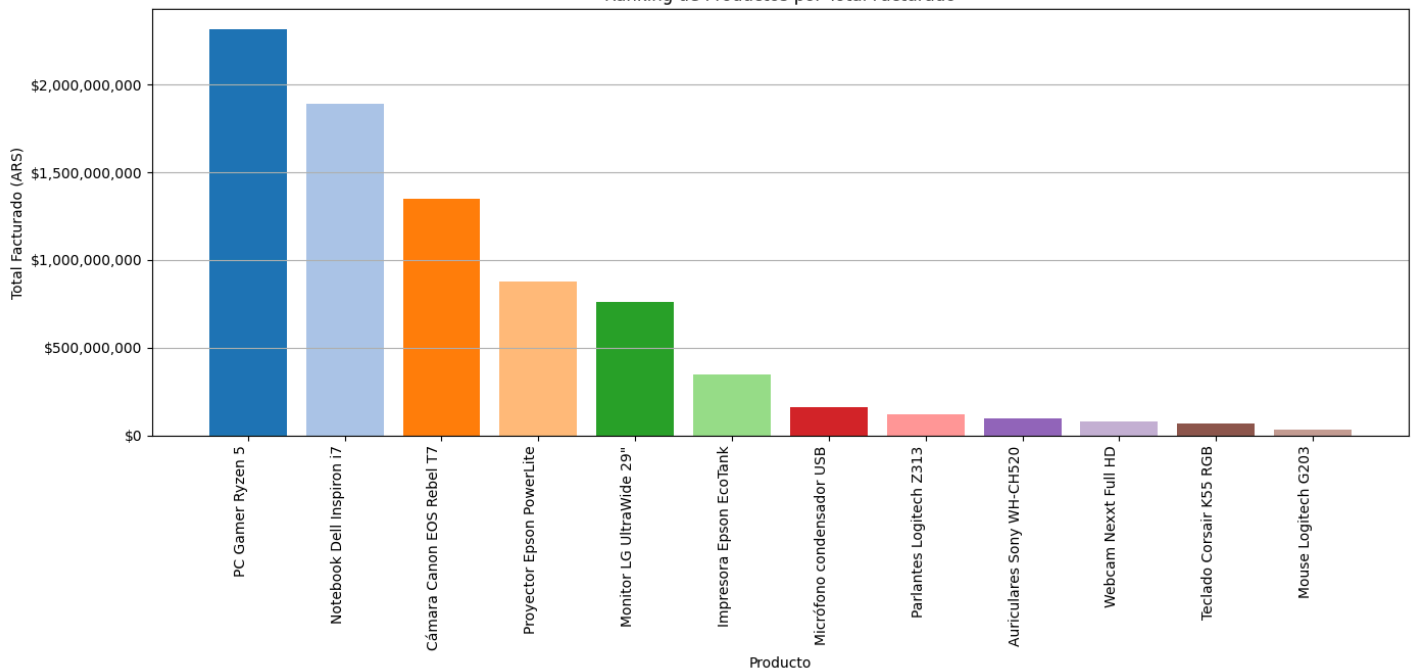
Distribución porcentual de ventas por rubro en formato de gráfico circular.



## 4.5 Ranking de Productos

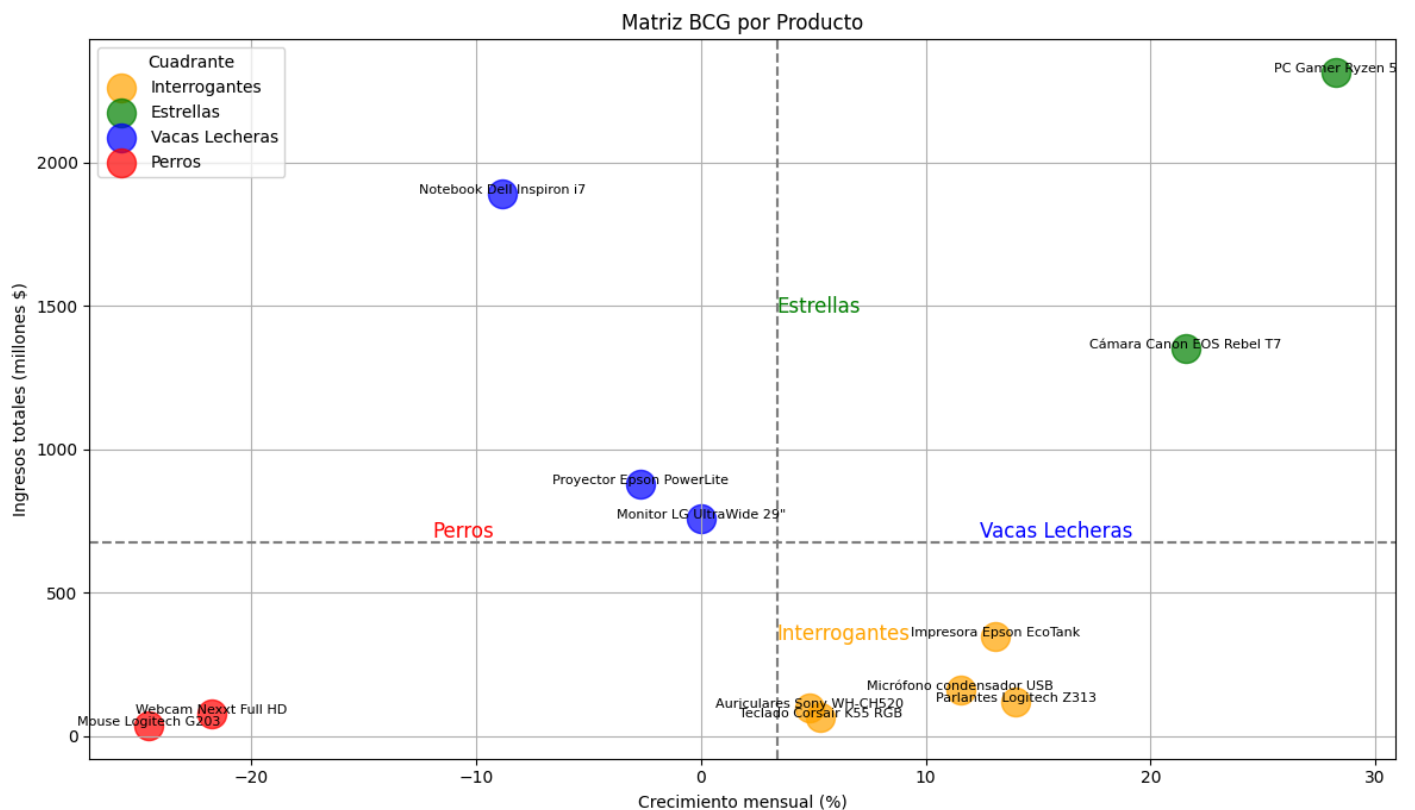
Gráfico de barras con el ranking de productos por total facturado.

Ranking de Productos por Total Facturado



## 4.6 Matriz BCG

Diagrama de dispersión clasificando productos en cuatro cuadrantes: Estrellas, Vacas Lecheras, Interrogantes y Perros.



## 5. Conclusión

Este análisis proporciona una visión completa del comportamiento de ventas de la empresa. Los insights obtenidos permiten identificar clientes clave, productos estrella y tendencias temporales. La implementación de técnicas de análisis de datos con Pandas demuestra su utilidad para la toma de

decisiones empresariales. Los resultados pueden ser utilizados para optimizar estrategias de marketing, gestión de inventario y planificación financiera.