

In [4]:
Proyecto 3 - Análisis de Datos con Visualizaciones
VERSIÓN CORREGIDA - Imágenes se guardan correctamente

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import os
from datetime import datetime

# =====
# CONFIGURACIÓN
# =====
CARPETA_DATOS = 'datos'
carpeta_imagenes = "graficos"

if not os.path.exists(carpeta_imagenes):
    os.makedirs(carpeta_imagenes)

plt.style.use('seaborn-v0_8')
sns.set_palette("husl")

# =====
# CARGA DE DATOS
# =====
print(" Cargando archivos CSV...")

try:
    clientes = pd.read_csv(f'{CARPETA_DATOS}/clientes.csv')
    productos = pd.read_csv(f'{CARPETA_DATOS}/productos.csv')
    facturas_encabezado = pd.read_csv(f'{CARPETA_DATOS}/facturas_encabezado.csv')
    facturas_detalle = pd.read_csv(f'{CARPETA_DATOS}/facturas_detalle.csv')
    rubros = pd.read_csv(f'{CARPETA_DATOS}/rubros.csv')
    sucursales = pd.read_csv(f'{CARPETA_DATOS}/sucursales.csv')
    condicion_iva = pd.read_csv(f'{CARPETA_DATOS}/condicion_iva.csv')

    print(" Todos los archivos cargados correctamente")

except FileNotFoundError as e:
    print(f" Error: No se encontró {e.filename}")
    exit()

# =====
# PREPARACIÓN DE DATOS
# =====
print("\n Preparando datos...")

# Renombrar columnas para evitar conflictos
productos_renom = productos.rename(columns={'descripcion': 'nombre_producto'})
rubros_renom = rubros.rename(columns={'descripcion': 'nombre_rubro'})
condicion_iva_renom = condicion_iva.rename(columns={'descripcion': 'tipo_iva'})

# Merge seguro de detalles con productos y rubros
detalles_completos = (facturas_detalle
    .merge(productos_renom, on='id_producto', how='left')
    .merge(rubros_renom, on='id_rubro', how='left')
)
# Merge seguro de facturas con clientes y condición IVA
facturas_completas = (facturas_encabezado
    .merge(clientes, on='id_cliente', how='left')
    .merge(condicion_iva_renom, on='id_condicion_iva', how='left')
    .merge(sucursales, on='id_sucursal', how='left')
)
# Convertir fecha
facturas_encabezado['fecha'] = pd.to_datetime(facturas_encabezado['fecha'])

print(" Datos preparados correctamente")

# =====
# ANÁLISIS BÁSICO
```

```

# =====
print("\n" + "*50)
print("ANÁLISIS BÁSICO")
print("*50)

total_ventas = facturas_encabezado['total_venta'].sum()
total_facturas = len(facturas_encabezado)
total_clientes = len(clientes)

print(f" Ventas totales: ${total_ventas:.2f}")
print(f" Total facturas: {total_facturas}")
print(f" Total clientes: {total_clientes}")
print(f" Total productos: {len(productos)}")

# =====
# FUNCIÓN CORREGIDA PARA GRÁFICOS INDIVIDUALES
# =====

def crear_y_guardar_grafico_individual(nombre_archivo, funcion_grafico):
    """Crea y guarda un gráfico individual SIN usar plt.show()"""
    fig, ax = plt.subplots(figsize=(10, 6))
    funcion_grafico(ax) # Pasar el axes para que dibuje
    plt.tight_layout()

    # PRIMERO GUARDAR
    archivo = os.path.join(carpeta_imagenes, f"individual_{nombre_archivo}.png")
    fig.savefig(archivo, dpi=300, bbox_inches='tight')
    print(f" individual_{nombre_archivo}.png")

    # LUEGO MOSTRAR (opcional)
    plt.show()

    # CERRAR
    plt.close(fig)

# =====
# DEFINICIÓN DE GRÁFICOS INDIVIDUALES
# =====

def grafico_01_ventas_mensuales(ax):
    ventas_mensuales = facturas_encabezado.groupby(facturas_encabezado['fecha'].dt.month)['total_venta'].sum()
    meses = ['Ene', 'Feb', 'Mar']
    ventas_mensuales.index = [meses[i-1] for i in ventas_mensuales.index if i <= len(meses)]
    ax.bar(ventas_mensuales.index, ventas_mensuales.values, color='skyblue', alpha=0.7)
    ax.set_title('Ventas por Mes', fontsize=14, fontweight='bold')
    ax.set_xlabel('Mes')
    ax.set_ylabel('Ventas ($)')
    ax.grid(True, alpha=0.3)

def grafico_02_ventas_sucursal(ax):
    ventas_sucursal = facturas_completas.groupby('id_sucursal')['total_venta'].sum()
    ax.pie(ventas_sucursal.values, labels=[f'Sucursal {i}' for i in ventas_sucursal.index],
           autopct='%1.1f%%', startangle=90, colors=['#ff9999','#66b3ff','#99ff99'])
    ax.set_title('Ventas por Sucursal', fontsize=14, fontweight='bold')

def grafico_03_top_productos_ventas(ax):
    ventas_producto = detalles_completos.groupby('nombre_producto')['subtotal_linea'].sum().nlargest(5)
    y_pos = range(len(ventas_producto))
    ax.barch(y_pos, ventas_producto.values, color='lightgreen')
    ax.set_title('Top 5 Productos por Ventas', fontsize=14, fontweight='bold')
    ax.set_xlabel('Ventas Totales ($)')
    ax.set_yticks(y_pos)
    ax.set_yticklabels(ventas_producto.index)
    ax.grid(True, alpha=0.3, axis='x')

def grafico_04_ventas_tipo_iva(ax):
    ventas_iva = facturas_completas.groupby('tipo_iva')['total_venta'].sum()
    ax.bar(ventas_iva.index, ventas_iva.values, color='lightcoral')
    ax.set_title('Ventas por Tipo de IVA', fontsize=14, fontweight='bold')
    ax.set_xlabel('Tipo de IVA')
    ax.set_ylabel('Ventas ($)')
    ax.tick_params(axis='x', rotation=45)
    ax.grid(True, alpha=0.3, axis='y')

```

```

def grafico_05_distribucion_montos(ax):
    ax.hist(facturas_encabezado['total_venta'], bins=8, alpha=0.7, color='orange', edgecolor='black')
    ax.set_title('Distribución de Montos de Venta', fontsize=14, fontweight='bold')
    ax.set_xlabel('Monto de Venta ($)')
    ax.set_ylabel('Frecuencia')
    ax.grid(True, alpha=0.3)

def grafico_06_stock_rubro(ax):
    stock_rubro = productos_renom.merge(rubros_renom, on='id_rubro').groupby('nombre_rubro')['stock'].sum()
    ax.bar(stock_rubro.index, stock_rubro.values, color='gold')
    ax.set_title('Stock por Rubro', fontsize=14, fontweight='bold')
    ax.set_xlabel('Rubro')
    ax.set_ylabel('Stock Total')
    ax.tick_params(axis='x', rotation=45)
    ax.grid(True, alpha=0.3, axis='y')

def grafico_07_productos_mas_vendidos(ax):
    cantidad_producto = detalles_completos.groupby('nombre_producto')['cantidad'].sum().nlargest(8)
    y_pos = range(len(cantidad_producto))
    ax.barr(y_pos, cantidad_producto.values, color='lightblue')
    ax.set_title('Productos Más Vendidos (Cantidad)', fontsize=14, fontweight='bold')
    ax.set_xlabel('Unidades Vendidas')
    ax.set_yticks(y_pos)
    ax.set_yticklabels(cantidad_producto.index)
    ax.grid(True, alpha=0.3, axis='x')

def grafico_08_ventas_rubro(ax):
    ventas_rubro = detalles_completos.groupby('nombre_rubro')['subtotal_linea'].sum()
    ax.pie(ventas_rubro.values, labels=ventas_rubro.index, autopct='%1.1f%%', startangle=90)
    ax.set_title('Ventas por Rubro', fontsize=14, fontweight='bold')

def grafico_09_precio_promedio(ax):
    precio_promedio = detalles_completos.groupby('nombre_producto')['precio_unitario'].mean().nlargest(8)
    y_pos = range(len(precio_promedio))
    ax.barr(y_pos, precio_promedio.values, color='lightgreen')
    ax.set_title('Precio Promedio por Producto (Top 8)', fontsize=14, fontweight='bold')
    ax.set_xlabel('Precio Promedio ($)')
    ax.set_yticks(y_pos)
    ax.set_yticklabels(precio_promedio.index)
    ax.grid(True, alpha=0.3, axis='x')

def grafico_10_precio_vs_stock(ax):
    ax.scatter(productos_renom['precio'], productos_renom['stock'], alpha=0.6, color='purple', s=60)
    ax.set_xlabel('Precio ($)')
    ax.set_ylabel('Stock')
    ax.set_title('Relación Precio vs Stock', fontsize=14, fontweight='bold')
    ax.grid(True, alpha=0.3)

def grafico_11_productos_por_rubro(ax):
    productos_por_rubro = productos_renom.merge(rubros_renom, on='id_rubro')['nombre_rubro'].value_counts()
    ax.bar(productos_por_rubro.index, productos_por_rubro.values, color='coral')
    ax.set_title('Cantidad de Productos por Rubro', fontsize=14, fontweight='bold')
    ax.set_xlabel('Rubro')
    ax.set_ylabel('Cantidad de Productos')
    ax.tick_params(axis='x', rotation=45)
    ax.grid(True, alpha=0.3, axis='y')

def grafico_12_ticket_promedio(ax):
    ax.hist(facturas_encabezado['total_venta'], bins=10, alpha=0.7, color='teal', edgecolor='black')
    ax.set_title('Distribución del Ticket Promedio', fontsize=14, fontweight='bold')
    ax.set_xlabel('Ticket Promedio ($)')
    ax.set_ylabel('Frecuencia')
    ax.grid(True, alpha=0.3)

# =====
# GENERAR GRÁFICOS INDIVIDUALES
# =====
print("\n Generando gráficos individuales...")

graficos_individuales = [
    ("01_ventas_mensuales", grafico_01_ventas_mensuales),
    ("02_ventas_sucursal", grafico_02_ventas_sucursal),
]

```

```

("03_top_productos_ventas", grafico_03_top_productos_ventas),
("04_ventas_tipo_iva", grafico_04_ventas_tipo_iva),
("05_distribucion_montos", grafico_05_distribucion_montos),
("06_stock_rubro", grafico_06_stock_rubro),
("07_productos_mas_vendidos", grafico_07_productos_mas_vendidos),
("08_ventas_rubro", grafico_08_ventas_rubro),
("09_precio_promedio", grafico_09_precio_promedio),
("10_precio_vs_stock", grafico_10_precio_vs_stock),
("11_productos_por_rubro", grafico_11_productos_por_rubro),
("12_ticket_promedio", grafico_12_ticket_promedio)
]

for nombre, funcion in graficos_individuales:
    crear_y_guardar_grafico_individual(nombre, funcion)

# =====
# DASHBOARDS (también corregidos)
# =====

print("\n Generando Dashboard Principal...")
fig1, axes1 = plt.subplots(2, 3, figsize=(18, 12))
fig1.suptitle('Dashboard Principal - Análisis de Ventas', fontsize=20, fontweight='bold')

# Aplicar las mismas funciones a los subplots
grafico_01_ventas_mensuales(axes1[0, 0])
grafico_02_ventas_sucursal(axes1[0, 1])
grafico_03_top_productos_ventas(axes1[0, 2])
grafico_04_ventas_tipo_iva(axes1[1, 0])
grafico_05_distribucion_montos(axes1[1, 1])
grafico_06_stock_rubro(axes1[1, 2])

plt.tight_layout()

# PRIMERO GUARDAR
archivo_principal = os.path.join(carpeta_imagenes, "dashboard_principal.png")
fig1.savefig(archivo_principal, dpi=300, bbox_inches='tight')
print(f" Dashboard principal guardado: {archivo_principal}")

# LUEGO MOSTRAR
plt.show()
plt.close(fig1)

print("\n Generando Dashboard de Productos...")
fig2, axes2 = plt.subplots(2, 3, figsize=(18, 12))
fig2.suptitle('Dashboard de Análisis de Productos', fontsize=20, fontweight='bold')

# Aplicar funciones a los subplots
grafico_07_productos_mas_vendidos(axes2[0, 0])
grafico_08_ventas_rubro(axes2[0, 1])
grafico_09_precio_promedio(axes2[0, 2])
grafico_10_precio_vs_stock(axes2[1, 0])
grafico_11_productos_por_rubro(axes2[1, 1])
grafico_12_ticket_promedio(axes2[1, 2])

plt.tight_layout()

# PRIMERO GUARDAR
archivo_productos = os.path.join(carpeta_imagenes, "dashboard_productos.png")
fig2.savefig(archivo_productos, dpi=300, bbox_inches='tight')
print(f" Dashboard de productos guardado: {archivo_productos}")

# LUEGO MOSTRAR
plt.show()
plt.close(fig2)

# =====
# RESUMEN FINAL
# =====

print("\n" + "*50)
print(" ANÁLISIS COMPLETADO")
print("*50)

print(f"\n ARCHIVOS GENERADOS EN: {carpeta_imagenes}/")

```

```
print(" 2 dashboards y 12 gráficos individuales")
print(" Ventas totales: ${total_ventas:.2f}")
```

Cargando archivos CSV...

Todos los archivos cargados correctamente

Preparando datos...

Datos preparados correctamente

ANÁLISIS BÁSICO

Ventas totales: \$1,911,316.00

Total facturas: 8

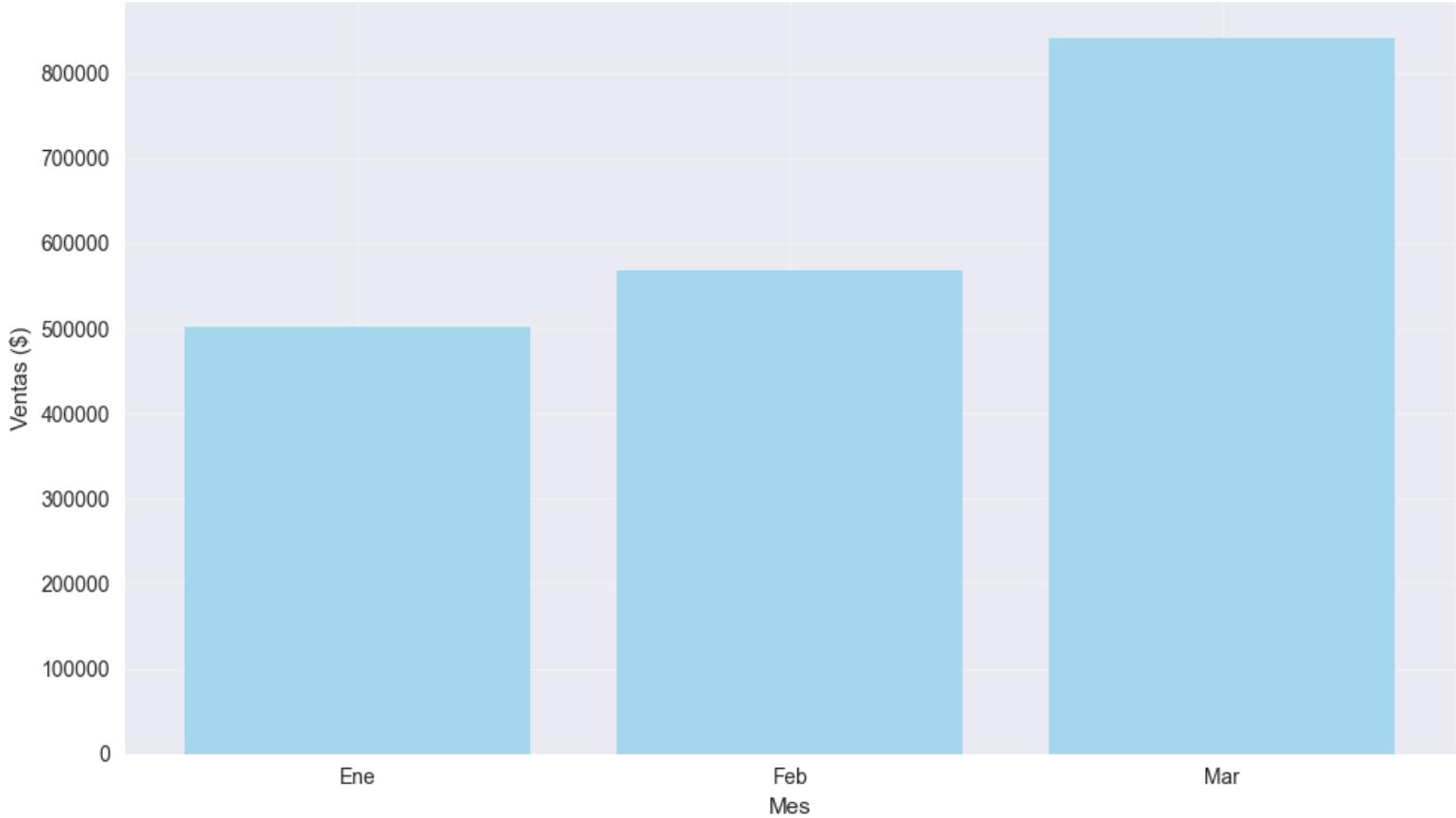
Total clientes: 10

Total productos: 12

Generando gráficos individuales...

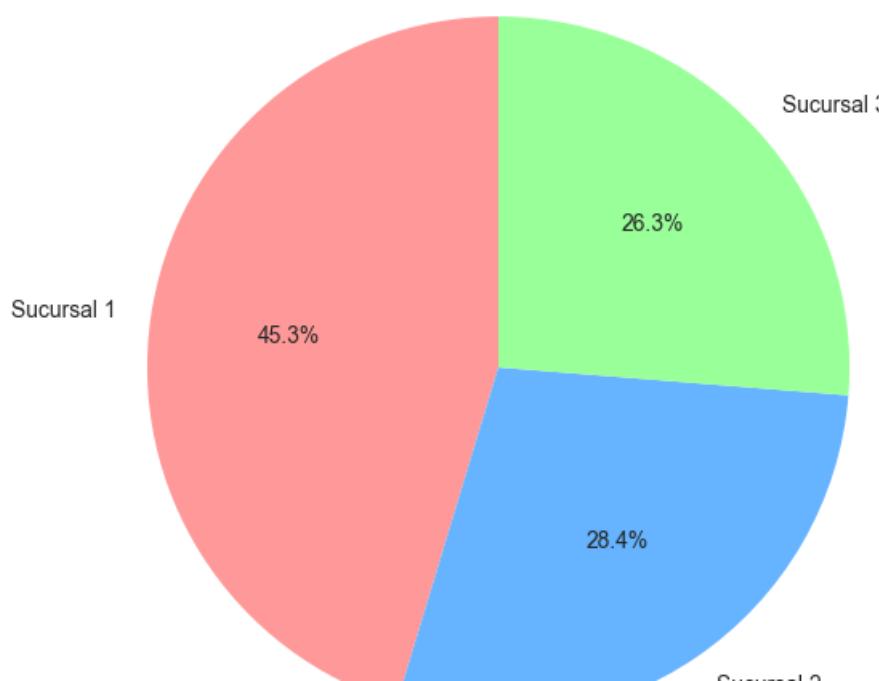
individual_01_ventas_mensuales.png

Ventas por Mes



individual_02_ventas_sucursal.png

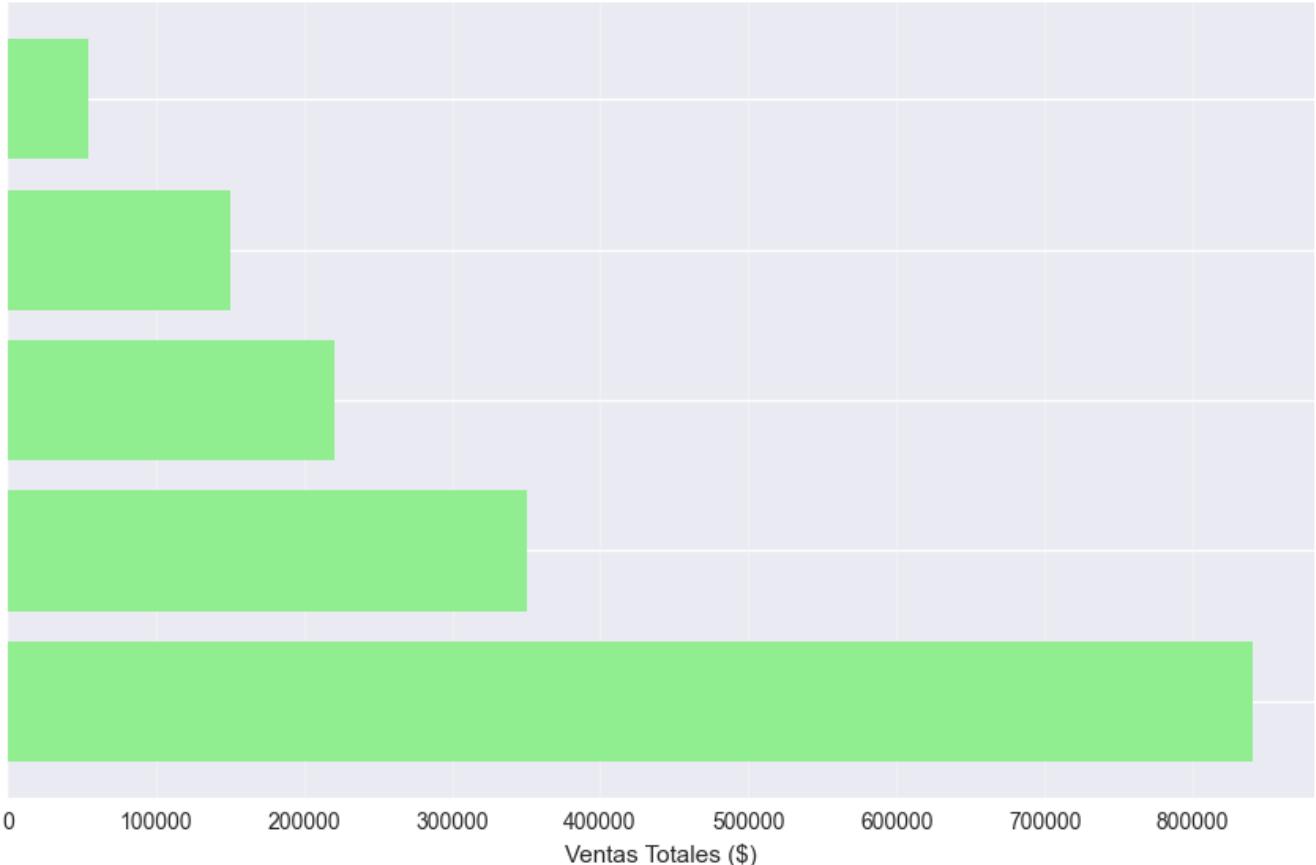
Ventas por Sucursal



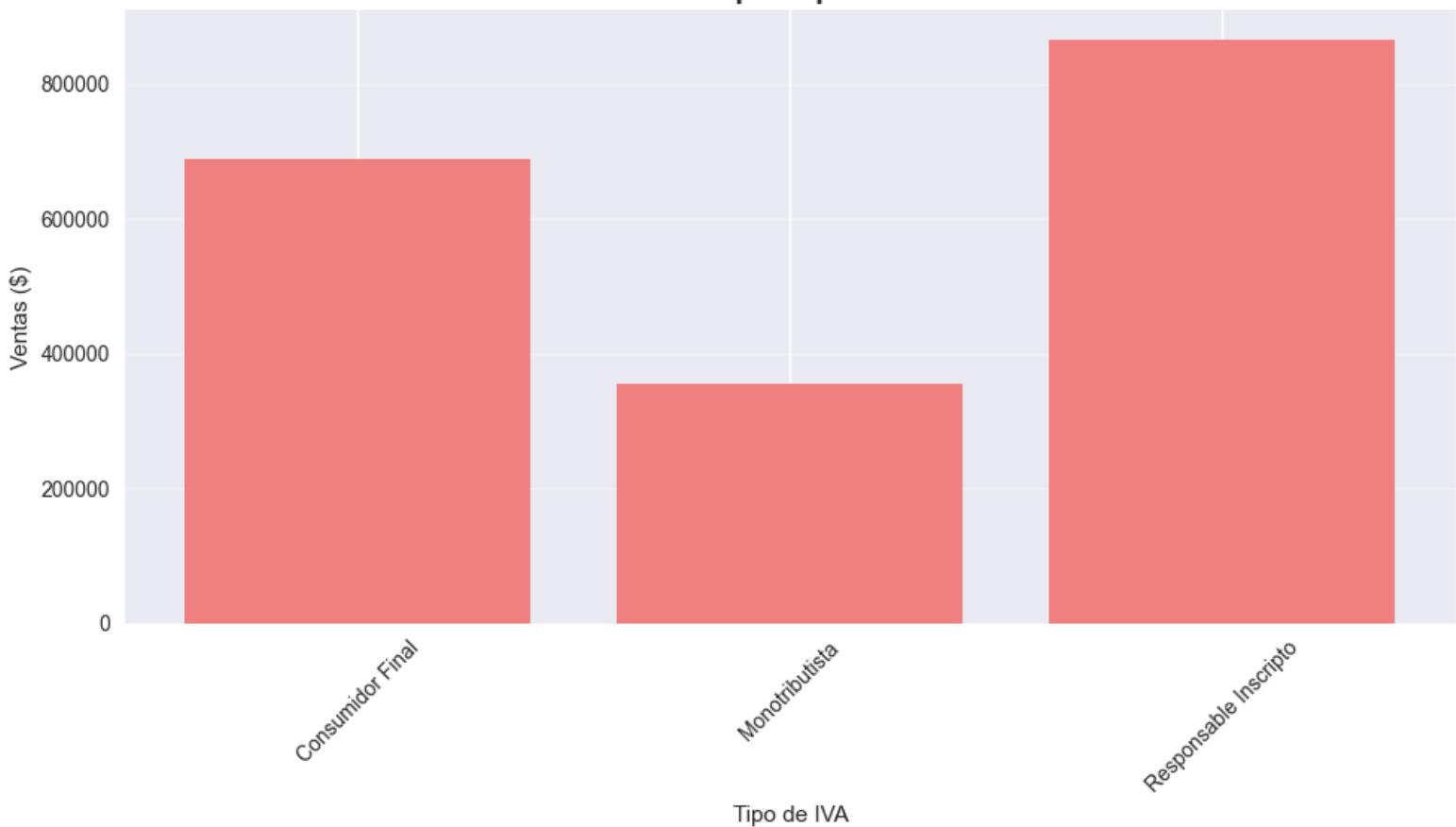
individual_03_top_productos_ventas.png

Top 5 Productos por Ventas

Auriculares Bluetooth



individual_04_ventas_tipo_iva.png

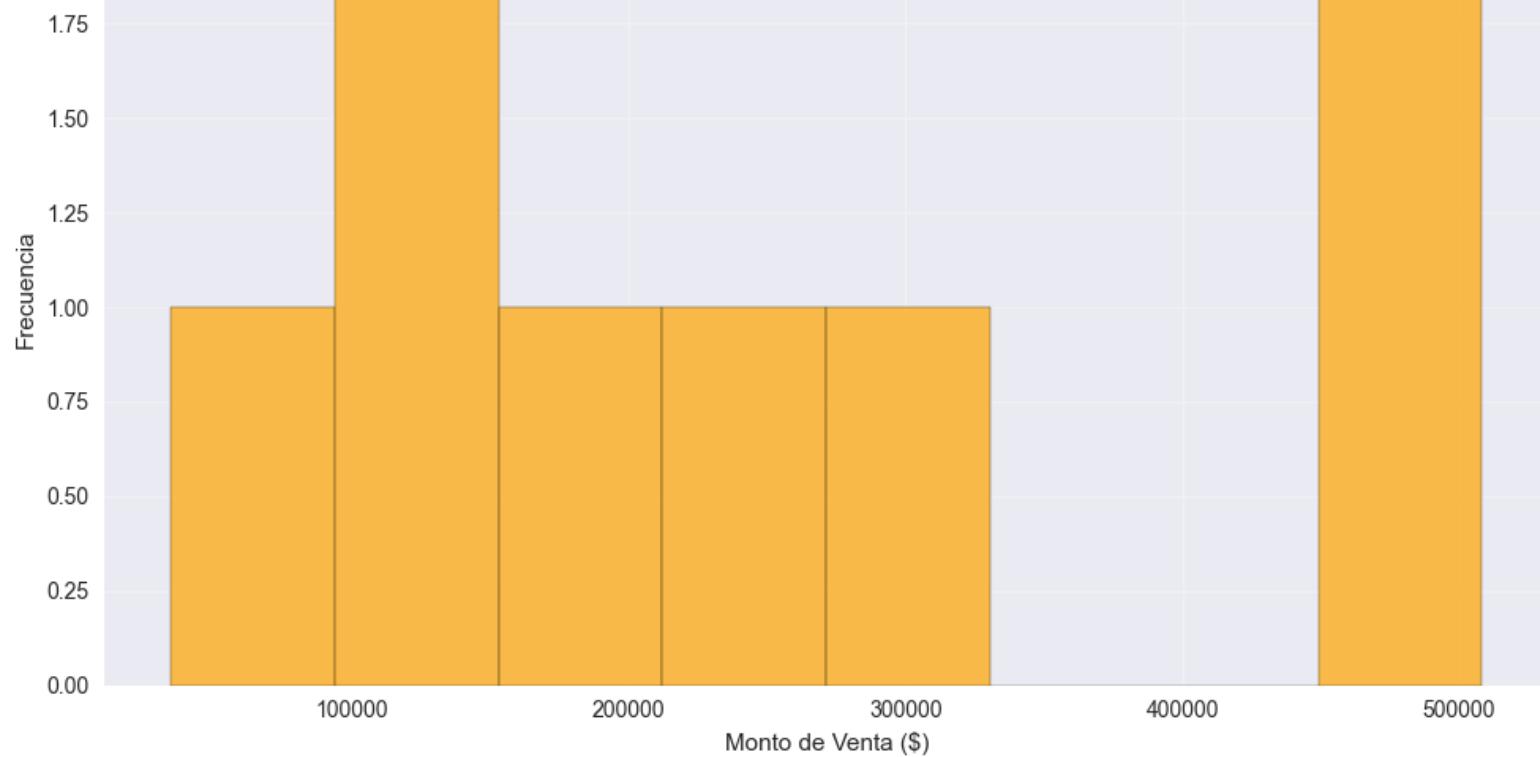
Ventas por Tipo de IVA

individual_05_distribucion_montos.png

Distribución de Montos de Venta

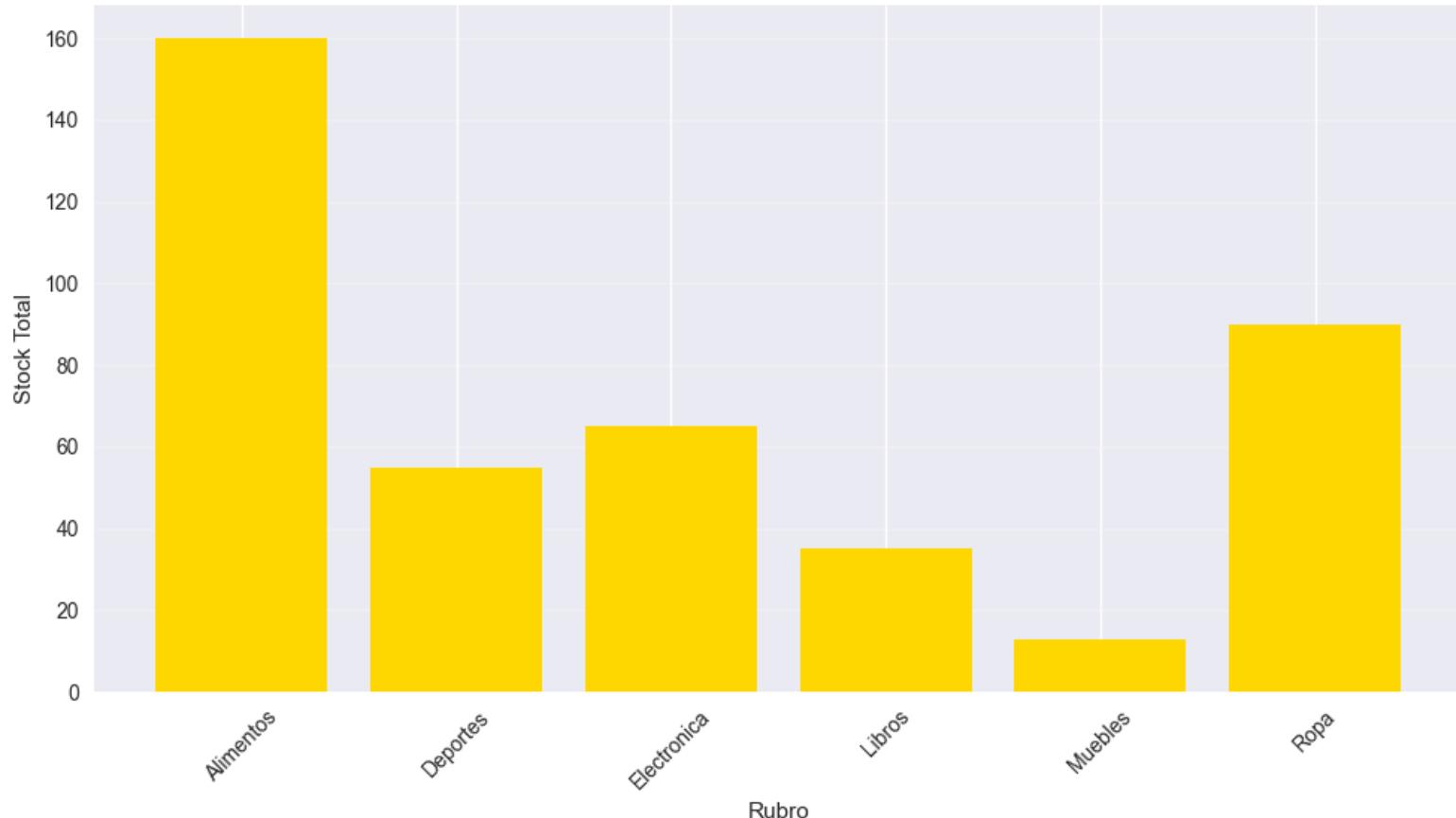
2.00





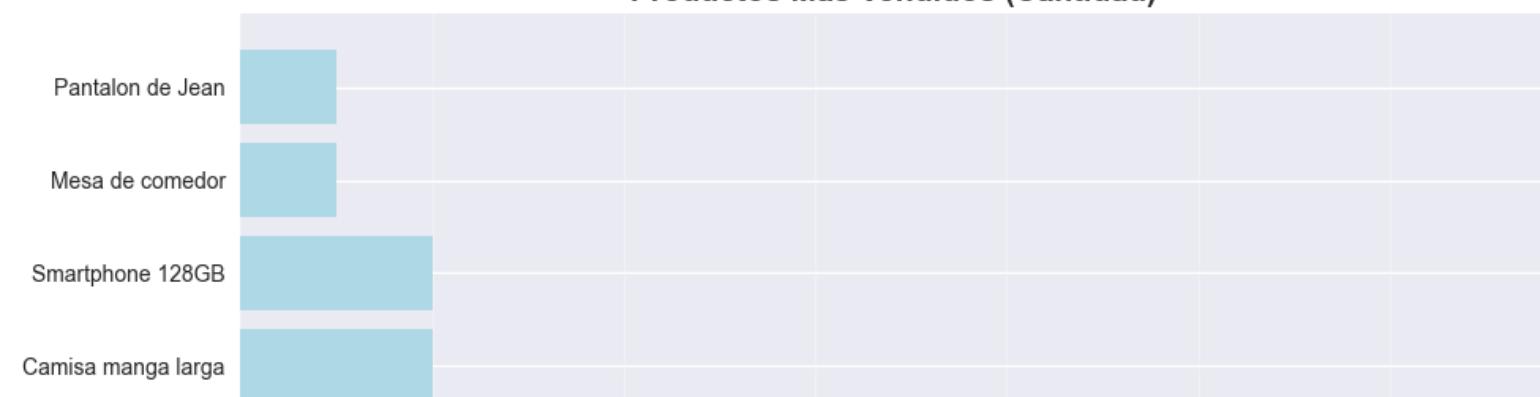
individual_06_stock_rubro.png

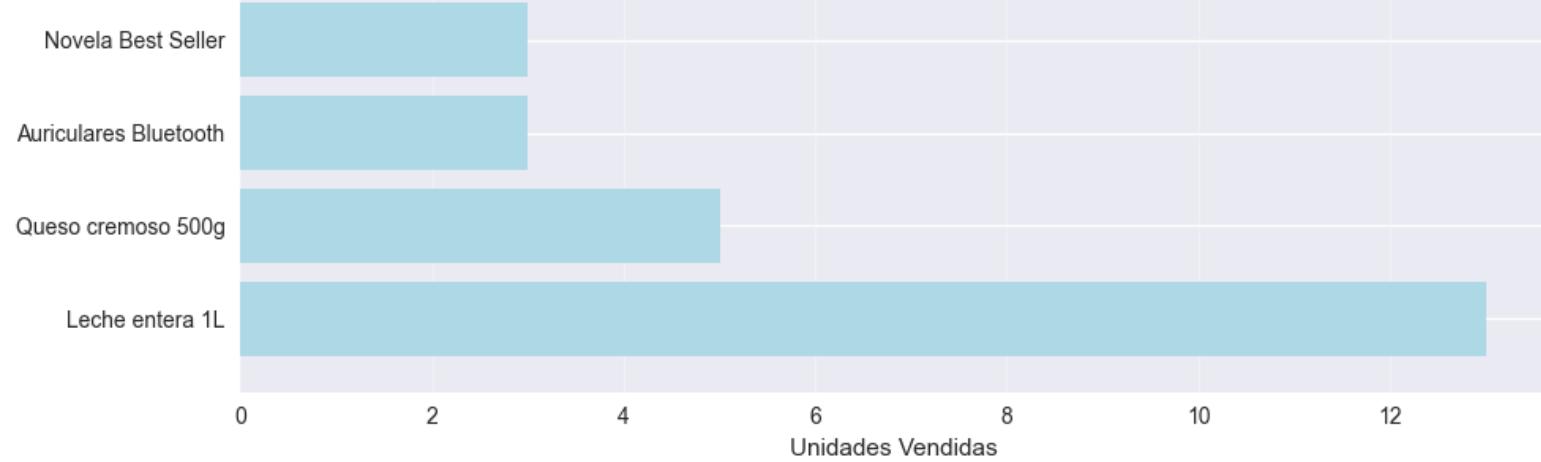
Stock por Rubro



individual_07_productos_mas_vendidos.png

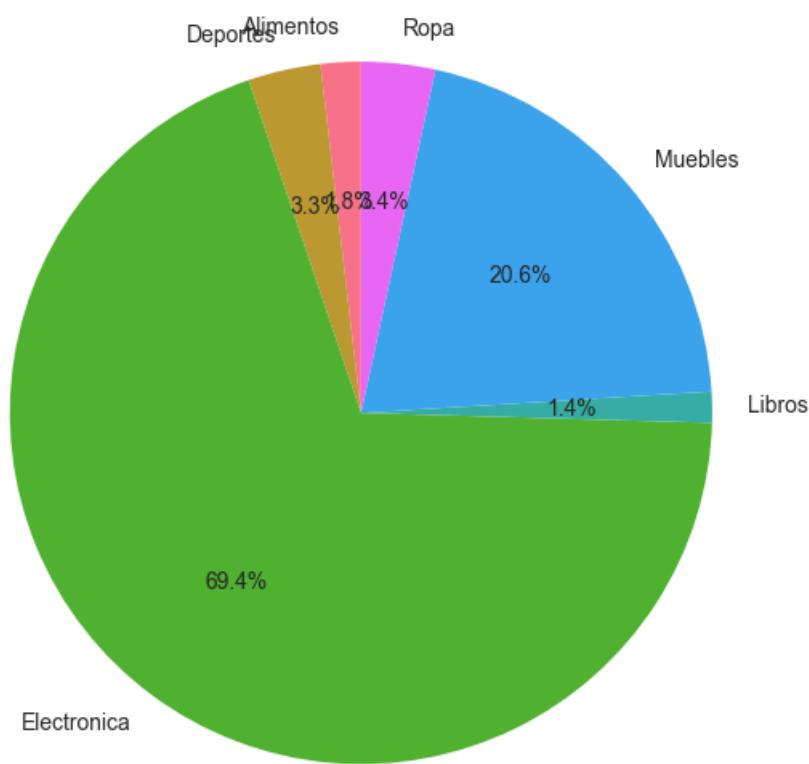
Productos Más Vendidos (Cantidad)





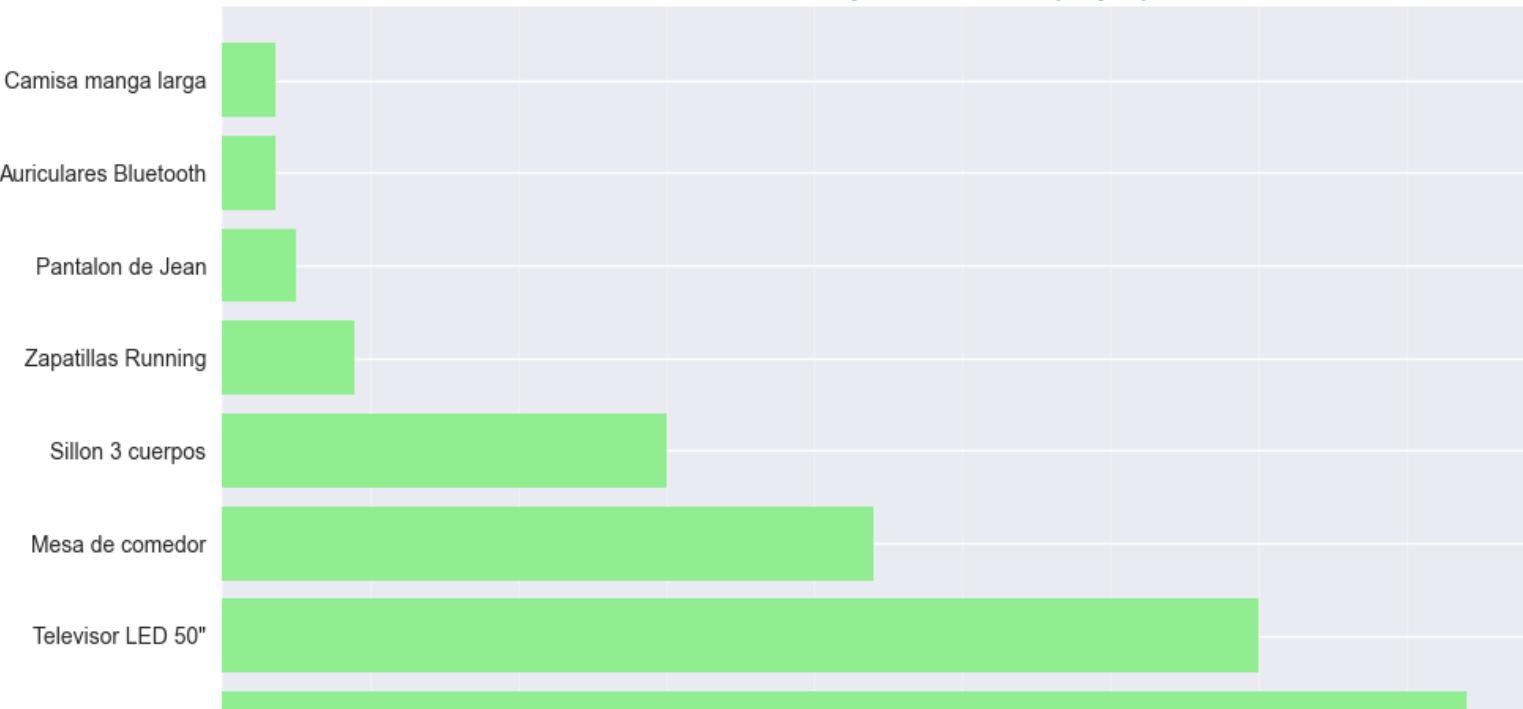
individual_08_ventas_rubro.png

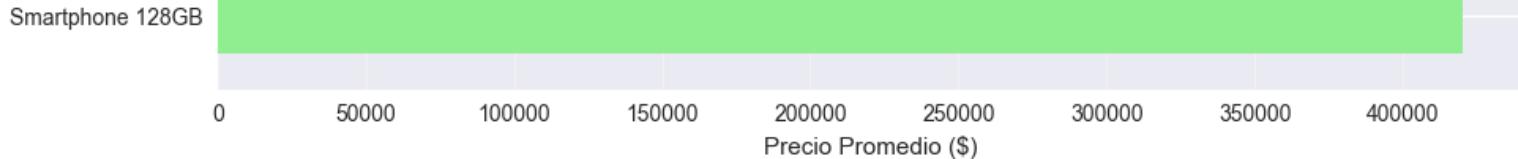
Ventas por Rubro



individual_09_precio_promedio.png

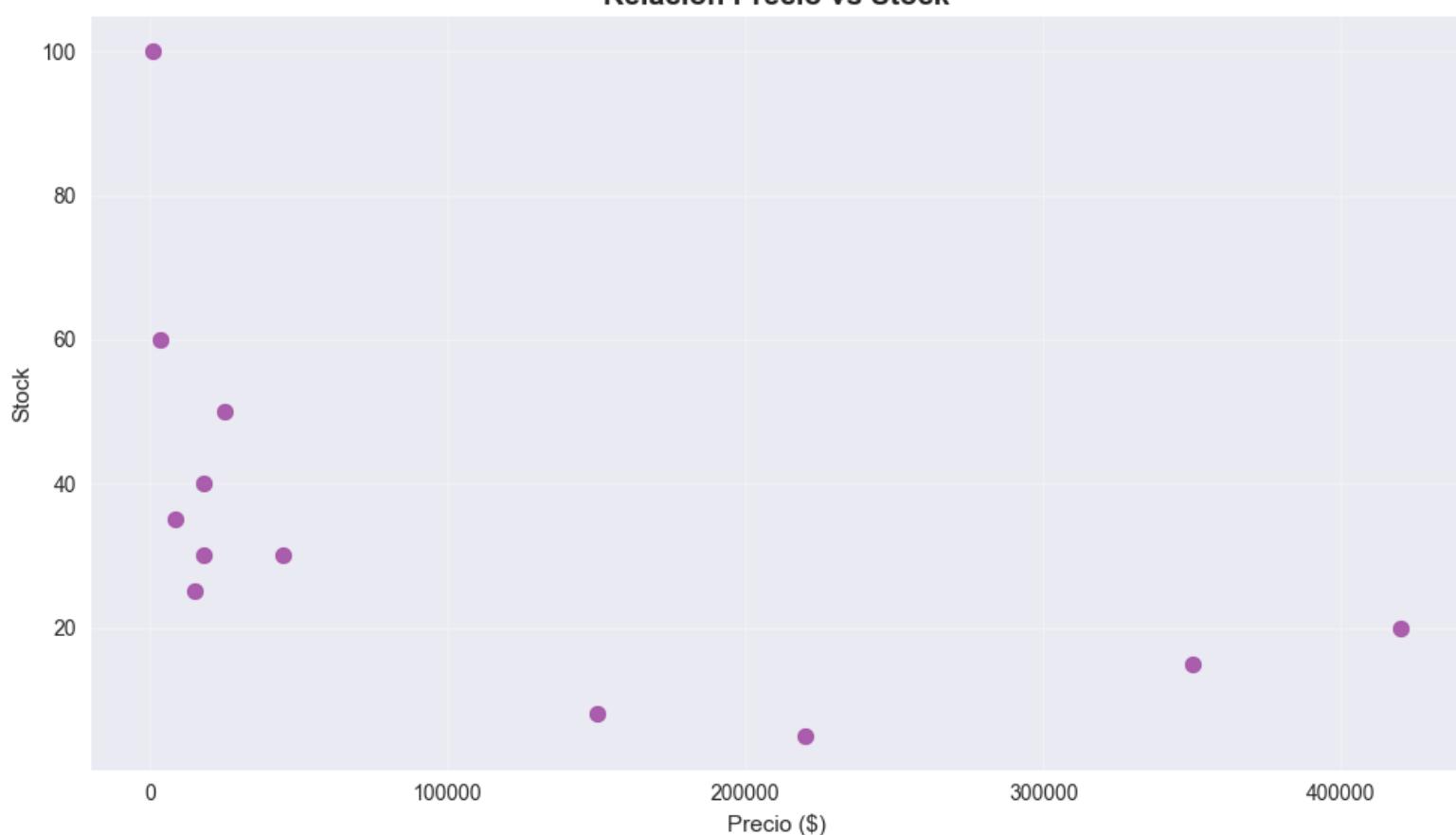
Precio Promedio por Producto (Top 8)





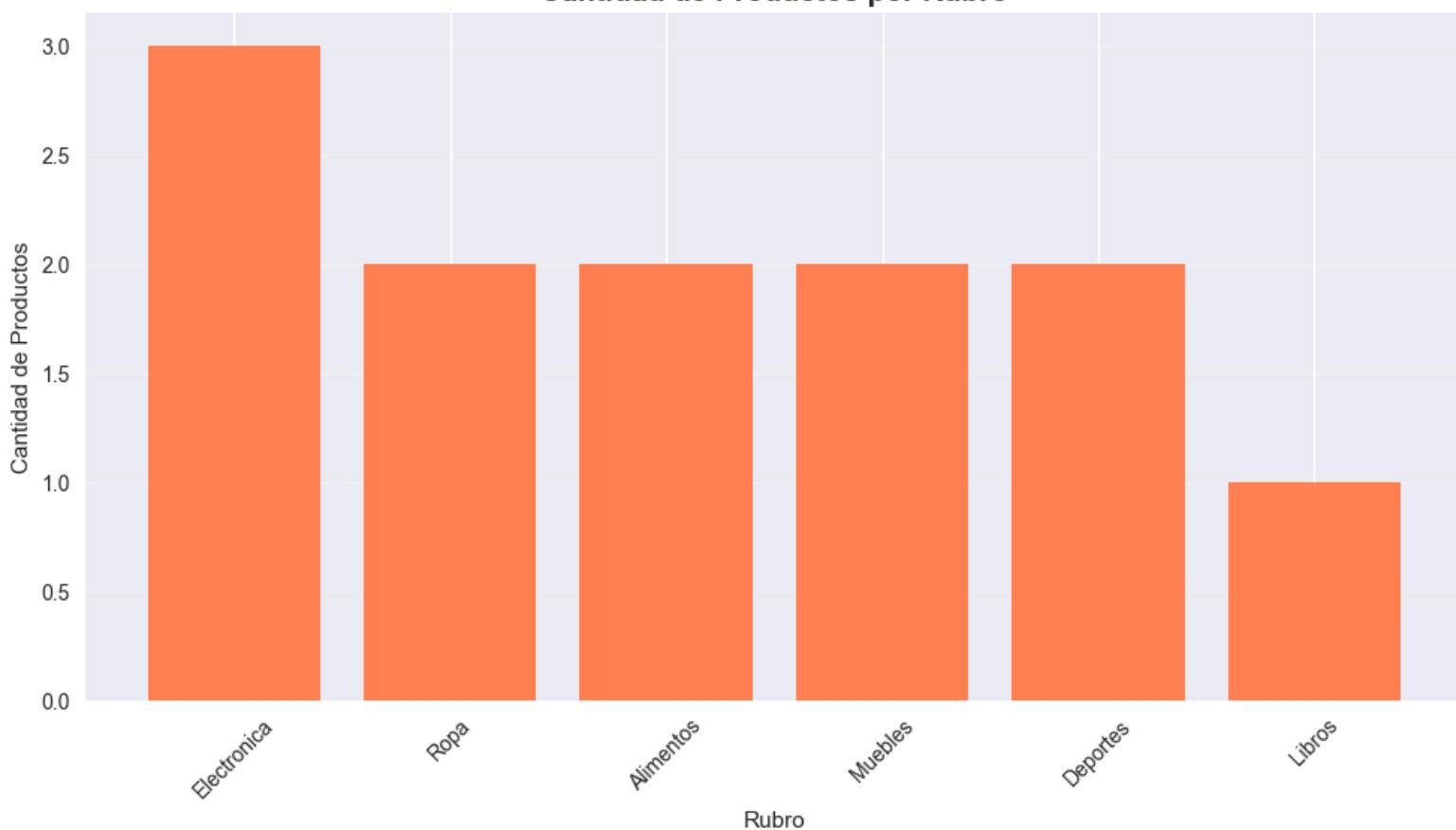
individual_10_precio_vs_stock.png

Relación Precio vs Stock



individual_11_productos_por_rubro.png

Cantidad de Productos por Rubro

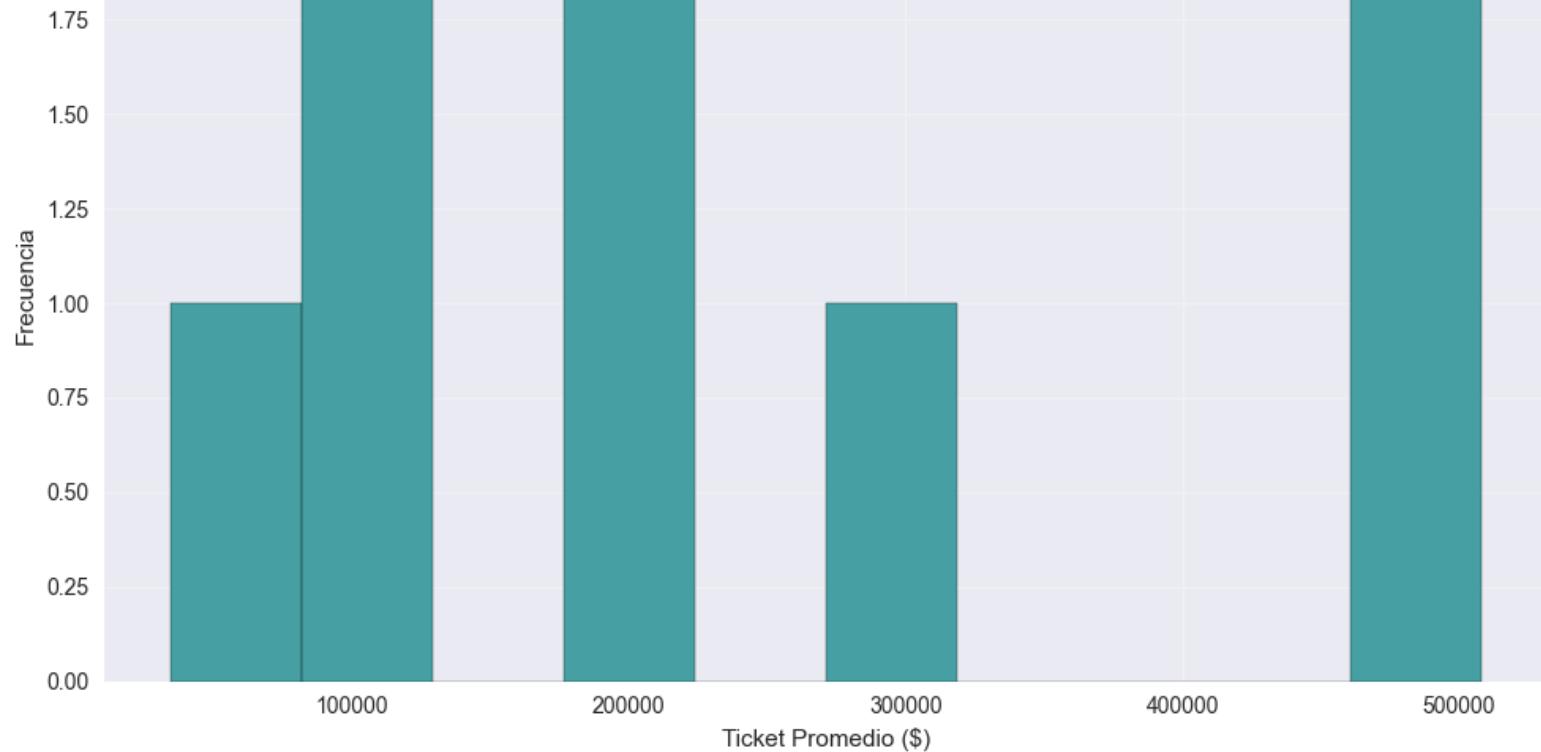


individual_12_ticket_promedio.png

Distribución del Ticket Promedio

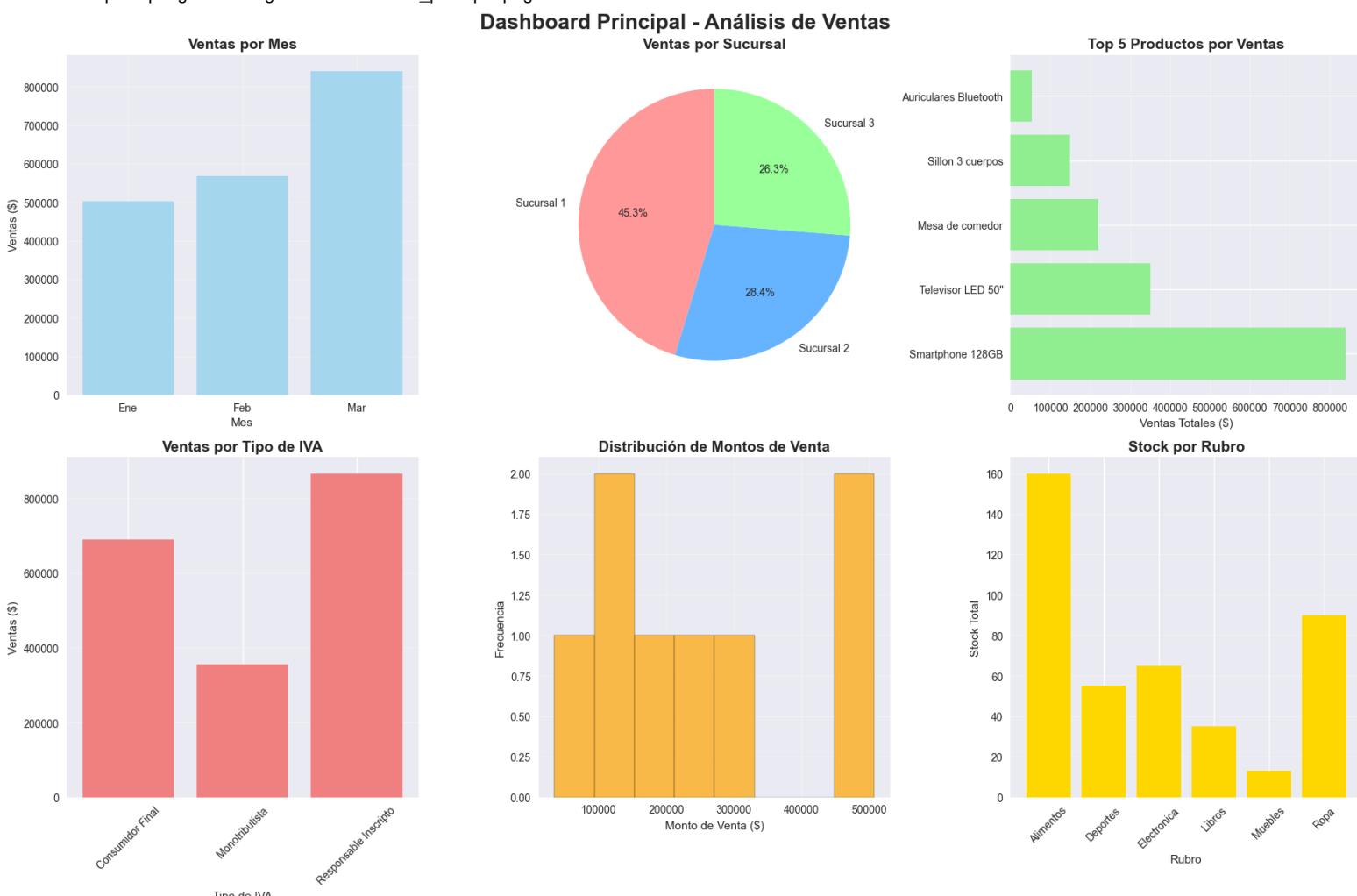


2.00



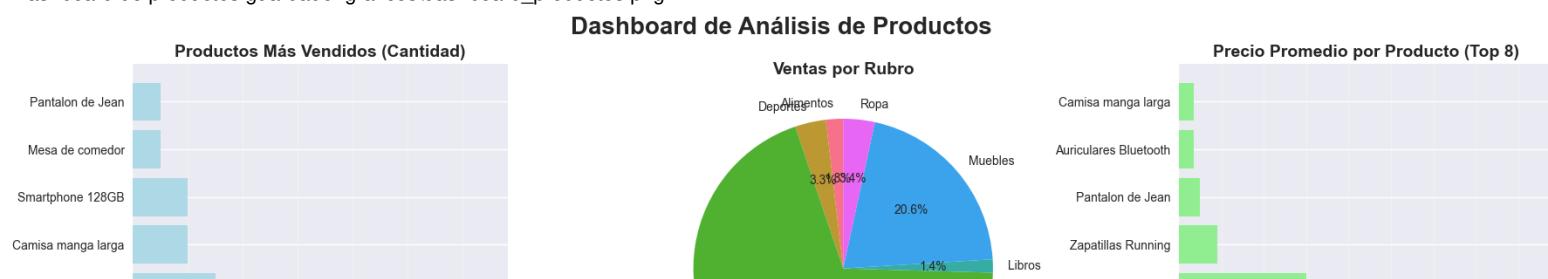
Generando Dashboard Principal...

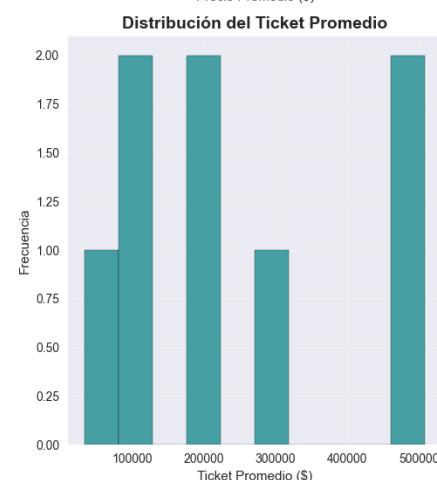
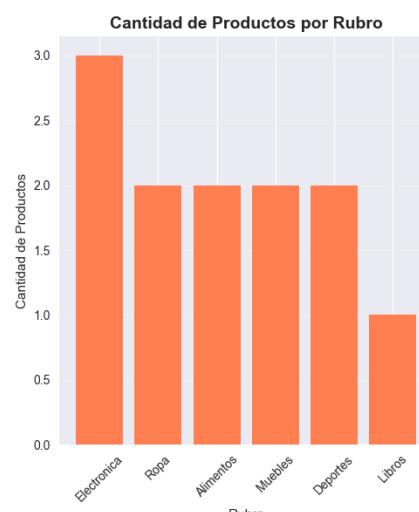
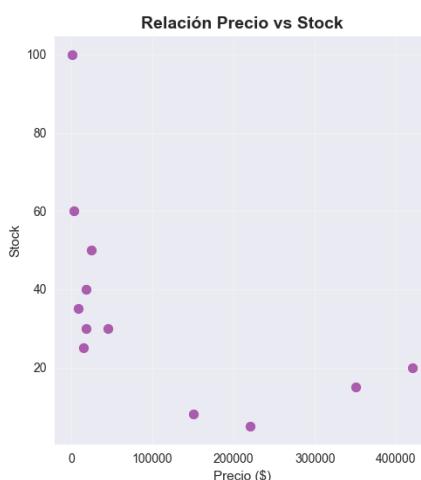
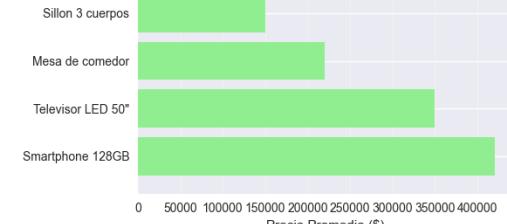
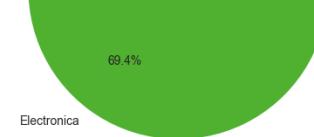
Dashboard principal guardado: graficos\dashboard_principal.png



Generando Dashboard de Productos...

Dashboard de productos guardado: graficos\dashboard_productos.png





ANÁLISIS COMPLETADO

ARCHIVOS GENERADOS EN: graficos/
2 dashboards y 12 gráficos individuales
Ventas totales: \$1,911,316.00

In []: