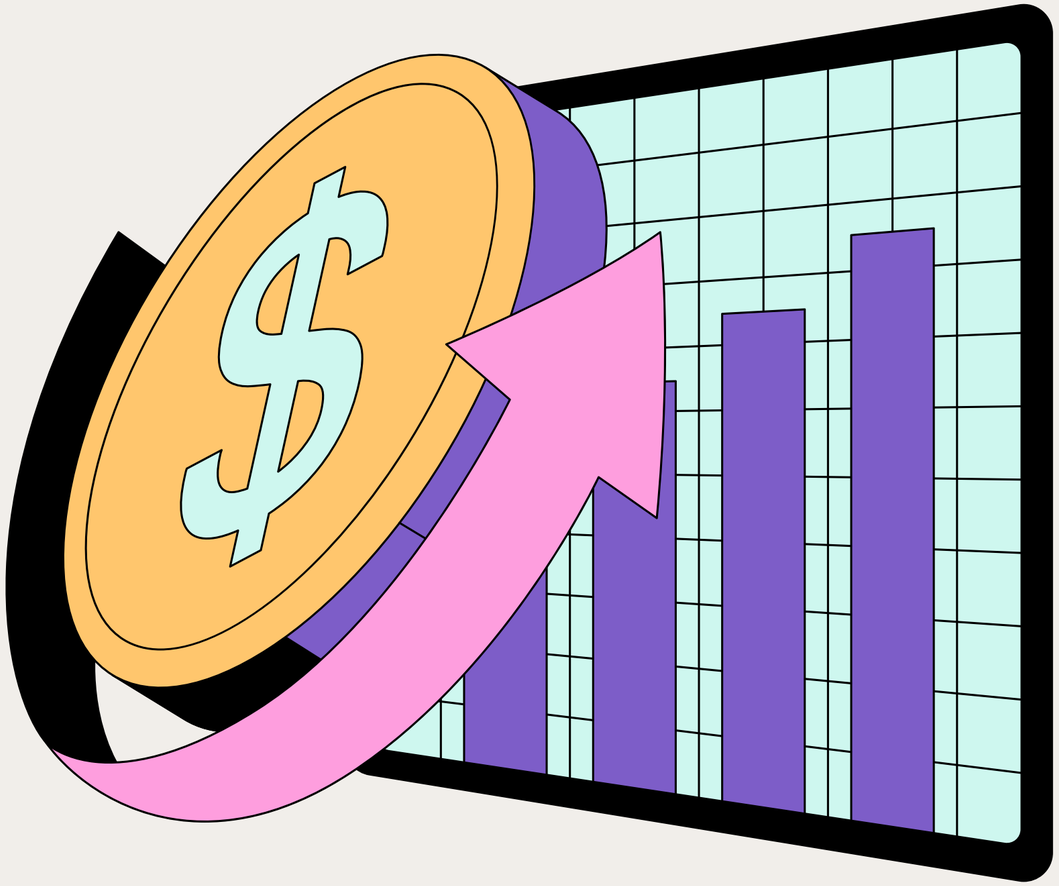
Proyecto final

Diseño e Implementación Base de datos: Banco Qapital



Grupo #1

Presentado por:

Ángela Valentina Sierra Rodríguez

+57 321 9888510

www.linkedin.com/in/angelavalentinasierra

Yirleyson Toro Ramírez

+57 314 6019212

https://www.linkedin.com/in/yirleyson-toro-ramirez/

María Fernanda Becerra

+57 316 2912054

www.linkedin.com/in/maríafernandabecerrav

Contexto

Descripción

Diseño e implementación de una base de datos integral para una entidad bancaria, enfocada en la gestión segura y eficiente de:

- Información de clientes
- Transacciones financieras
- Cartera de productos y control de mora

La solución permitirá:

- Almacenar datos personales, de autenticación y productos contratados.
- Registrar y consultar movimientos financieros (depósitos, retiros, transferencias, pagos, compras).
- Administrar cartera: saldos, límites de crédito, fechas de vencimiento y pagos realizados.
- Monitorear mora: clientes en incumplimiento, montos adeudados y procesos de recuperación.

Requisitos Clave

- Seguridad y confidencialidad de los datos, cumpliendo estándares y normativas bancarias.
- Escalabilidad y alto rendimiento para soportar gran volumen de clientes y transacciones.
- Estructura robusta que facilite análisis, consultas avanzadas y toma de decisiones.

Objetivos

General

Diseñar e implementar una base de datos relacional eficiente que permita gestionar la información clave de una entidad bancaria, optimizando la administración de operaciones, el análisis de datos y la toma de decisiones.

Específicos

1. Analizar los requerimientos funcionales del banco
2. Diseñar el modelo entidad-relación que refleje con precisión los procesos financieros y operativos de la organización.
3. Implementar la base de datos utilizando un sistema de gestión adecuado, garantizando integridad, seguridad y escalabilidad.
4. Desarrollar consultas avanzadas y reportes analíticos que faciliten el monitoreo del negocio y la toma de decisiones.

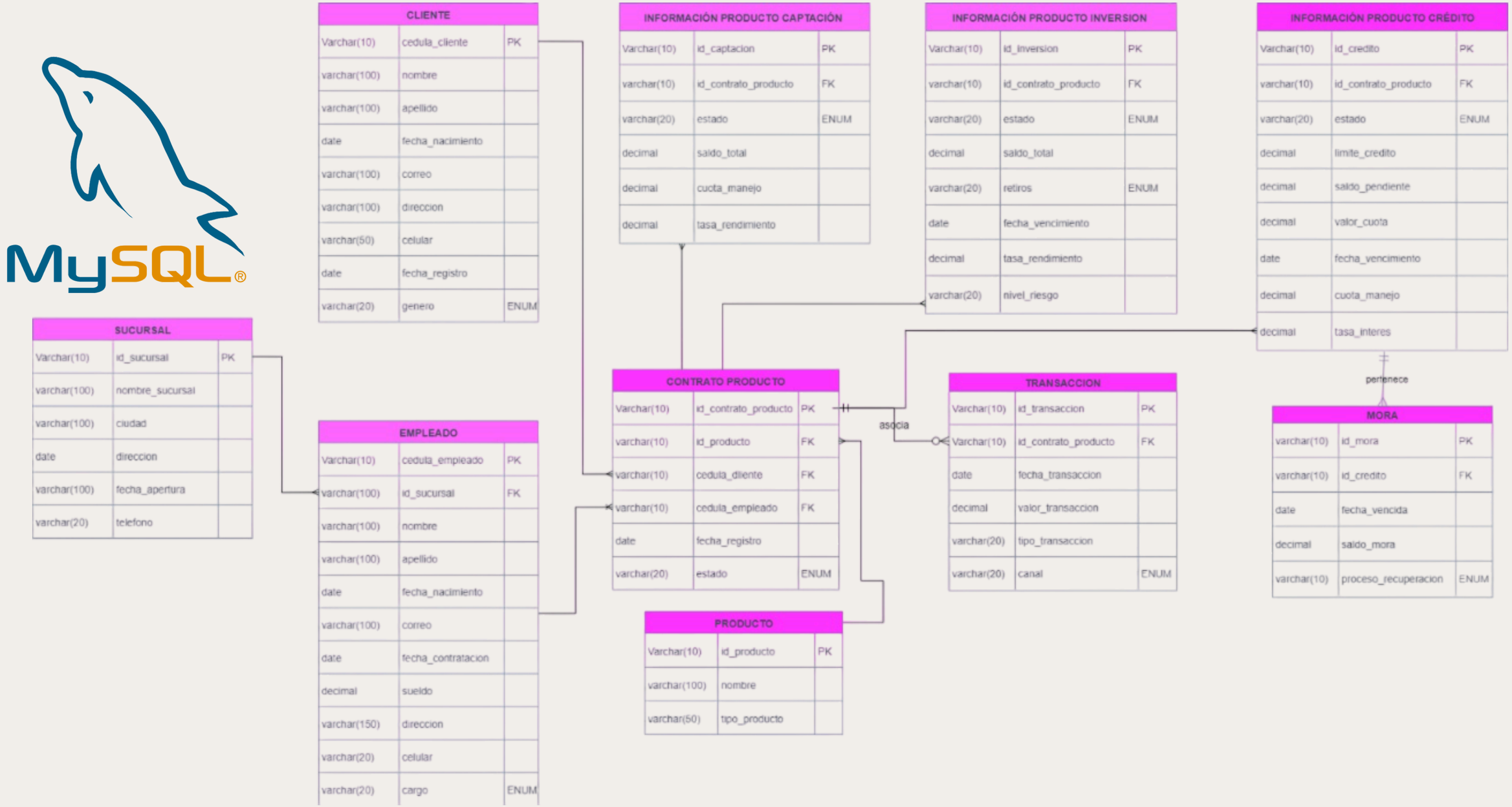
Metodología

El desarrollo del proyecto se llevó a cabo en varias etapas clave:

1. Análisis de requerimientos
2. Diseño e implementación: Modelo relacional en workbench Myqsl y Consultas
3. Población de datos desde Python + librería Faker
4. Se diseñaron paneles de visualización en Power BI

Esta metodología permitió establecer un flujo de trabajo coherente y completo, desde la definición del problema hasta la generación de información útil para el proceso de toma de decisiones en un contexto empresarial real.

Modelo relacional banco Qapital



Script de creación: MySQL

```sql
create Database banco;
Use banco;

CREATE TABLE clientes (
    cedula_cliente BIGINT PRIMARY KEY,
    nombre VARCHAR(50) NOT NULL,
    apellido VARCHAR(50) NOT NULL,
    fecha_nacimiento DATE NOT NULL,
    correo VARCHAR(100) UNIQUE,
    direccion VARCHAR(150),
    celular VARCHAR(20),
    fecha_registro DATE DEFAULT (CURRENT_DATE),
    genero ENUM('Masculino','Femenino')
);
CREATE TABLE sucursal (
    id_sucursal INT AUTO_INCREMENT PRIMARY KEY,
    nombre_sucursal VARCHAR(100) NOT NULL,
    direccion VARCHAR(150) NOT NULL,
    ciudad VARCHAR(50) NOT NULL,
    telefono VARCHAR(20),
    fecha_apertura DATE
);
```

```sql
CREATE VIEW clientes_con_mas_productos_activos AS
SELECT c.cedula_cliente, c.nombre, c.apellido,
       COUNT(cp.id_contrato_producto) AS total_productos
FROM clientes c
JOIN contrato_producto cp ON c.cedula_cliente = cp.cedula_cliente
JOIN producto p ON cp.id_producto = p.id_producto
WHERE p.tipo_producto IN ('credito','debito','virtual')
GROUP BY c.cedula_cliente, c.nombre, c.apellido
ORDER BY total_productos DESC;

CREATE VIEW clientes_en_mora AS
SELECT c.cedula_cliente, c.nombre, c.apellido,
       m.saldo_mora,
       DATEDIFF(CURDATE(), m.fecha_vencida) AS dias_atraso
FROM clientes c
JOIN contrato_producto cp ON c.cedula_cliente = cp.cedula_cliente
JOIN producto_credito pc ON cp.id_contrato_producto = pc.id_contrato_producto
JOIN mora m ON pc.id_credito = m.id_credito
WHERE m.saldo_mora > 0;


CREATE VIEW productos_mas_usados AS
SELECT p.nombre_producto,
       COUNT(t.id_transaccion) AS total_transacciones
FROM producto p
JOIN contrato_producto cp ON p.id_producto = cp.id_producto
JOIN transaccion t ON cp.id_contrato_producto = t.id_contrato_producto
GROUP BY p.nombre_producto
ORDER BY total_transacciones DESC;
```

```python
import mysql.connector
from faker import Faker
import random

# Conexión a MySQL
conexion = mysql.connector.connect(
    host="localhost",
    user="root",
    password="root",
    database="banco"
)
cursor = conexion.cursor()
faker = Faker("es_CO")  # Datos realistas en español de Colombia

# ==========================
# 1. Insertar sucursales
# ==========================
def insertar_sucursales():
    sucursales_fijas = [
        ("Sucursal Medellín", "Calle 10 #20-30", "Medellín", "6041234567"),
        ("Sucursal Bogotá", "Cra 15 #45-67", "Bogotá", "6017654321"),
        ("Sucursal Cali", "Av. Roosevelt #25-50", "Cali", "6021112233"),
        ("Sucursal Barranquilla", "Calle 72 #53-45", "Barranquilla", "6053334444"),
        ("Sucursal Cartagena", "Cra 2 #31-45", "Cartagena", "6057778888")
    ]

    sql = """INSERT INTO sucursal (nombre_sucursal, direccion, ciudad, telefono, fecha_apertura)
             VALUES (%s, %s, %s, %s, %s)"""

    for suc in sucursales_fijas:
        cursor.execute(sql, (suc[0], suc[1], suc[2], suc[3], faker.date_between(start_date="-20y", end_date="today")))

    conexion.commit()
    print("✅ Sucursales insertadas")
```

```python
def insertar_empleados(n):
    sql = """INSERT INTO empleados
             (cedula_empleado, nombre, apellido, fecha_nacimiento, correo,
              direccion, celular, fecha_contratacion, cargo, salario, id_sucursal)
             VALUES (%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)"""

    cargos = ["vendedor", "gerente", "asesor"]

    for _ in range(n):
        cedula = faker.unique.random_number(digits=10)
        nombre = faker.first_name()
        apellido = faker.last_name()
        fecha_nac = faker.date_of_birth(minimum_age=25, maximum_age=60)
        correo = faker.unique.email()
        direccion = faker.address()
        celular = faker.msisdn()[:10]
        fecha_contratacion = faker.date_between(start_date="-10y", end_date="today")
        cargo = random.choice(cargos)
        salario = round(random.uniform(1500000, 8000000), 2)
        id_sucursal = random.randint(1, 5)  # asegura que siempre apunte a sucursal existente

        valores = (cedula, nombre, apellido, fecha_nac, correo,
                   direccion, celular, fecha_contratacion, cargo,
                   salario, id_sucursal)

        cursor.execute(sql, valores)

    conexion.commit()
    print(f"✅ {n} empleados insertados")
```

```python
def insertar_clientes(n):
    sql = """INSERT INTO clientes
             (cedula_cliente, nombre, apellido, fecha_nacimiento, correo,
              direccion, celular, fecha_registro, genero)
             VALUES (%s,%s,%s,%s,%s,%s,%s,%s,%s)"""

    generos = ["Masculino", "Femenino"]

    for _ in range(n):
        cedula = faker.unique.random_number(digits=10)
        nombre = faker.first_name()
        apellido = faker.last_name()
        fecha_nac = faker.date_of_birth(minimum_age=18, maximum_age=80)
        correo = faker.unique.email()
        direccion = faker.address()
        celular = faker.msisdn()[:10]
        fecha_registro = faker.date_between(start_date="-5y", end_date="today")
        genero = random.choice(generos)

        valores = (cedula, nombre, apellido, fecha_nac, correo,
                   direccion, celular, fecha_registro, genero)

        cursor.execute(sql, valores)

    conexion.commit()
    print(f"✅ {n} clientes insertados")


# ==========================
# Ejecutar
# ==========================
insertar_sucursales()    # 5 sucursales
insertar_empleados(50)   # 50 empleados
insertar_clientes(1000)  # 1000 clientes

cursor.close()
conexion.close()
```

python™

Dashboard clientes: Banco Qpital

Power BI

Dashboard transacciones: Banco Qpital

Dashboard transacciones: Banco Qpital

Dashboard cartera/mora: Banco Qpital

Dashboard empleados: Banco Qpital

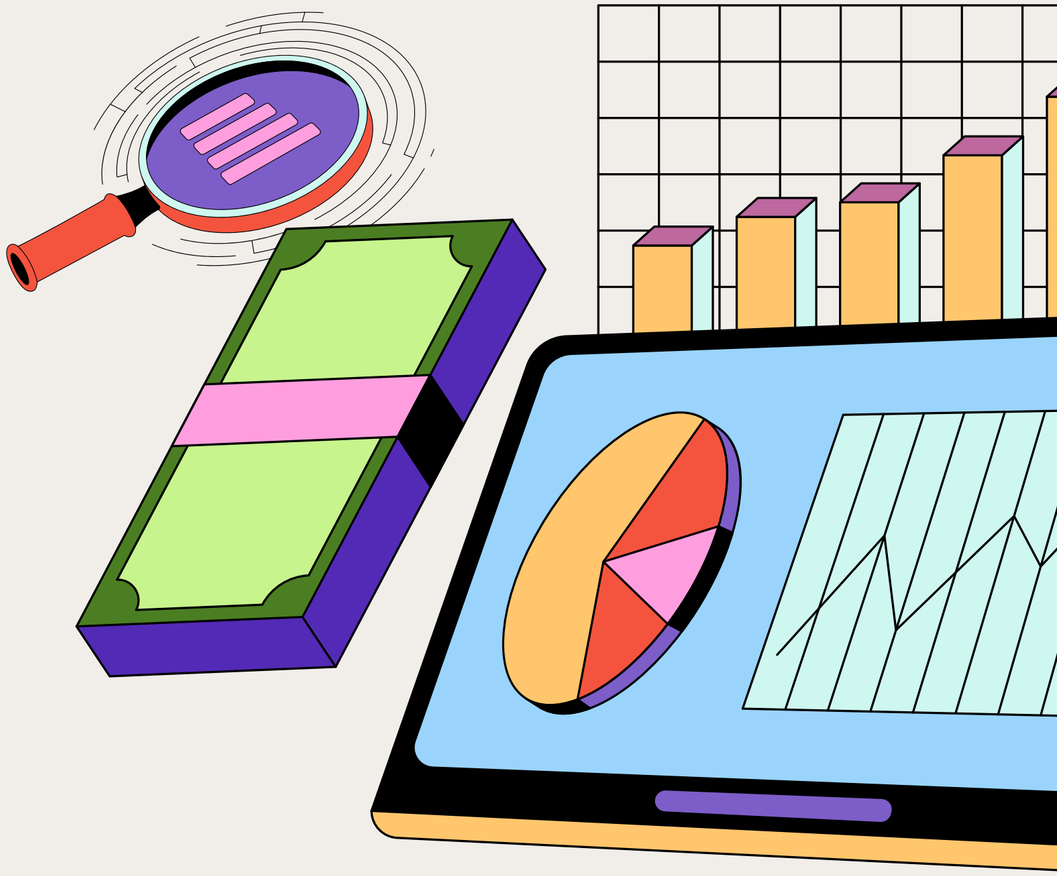
Conclusiones

1.Perfil de clientes: El banco cuenta con 1000 clientes: edad promedio de 49 años, lo que refleja un público adulto con alto potencial financiero.

- Recomendación: Ofrecer productos de inversión y ahorro a largo plazo enfocados en este segmento.

2. Cartera en mora: Se registran 179 clientes y 194 productos en mora, lo que indica un riesgo financiero considerable.

- Recomendación: Implementar planes de refinanciación y descuentos por pronto pago para recuperar cartera.



3. Preferencias de productos: El crédito de libre inversión es el más contratado y el fondo de inversión el más utilizado.

- Recomendación: Promocionar créditos con tasas preferenciales y crear productos complementarios a los fondos existentes.

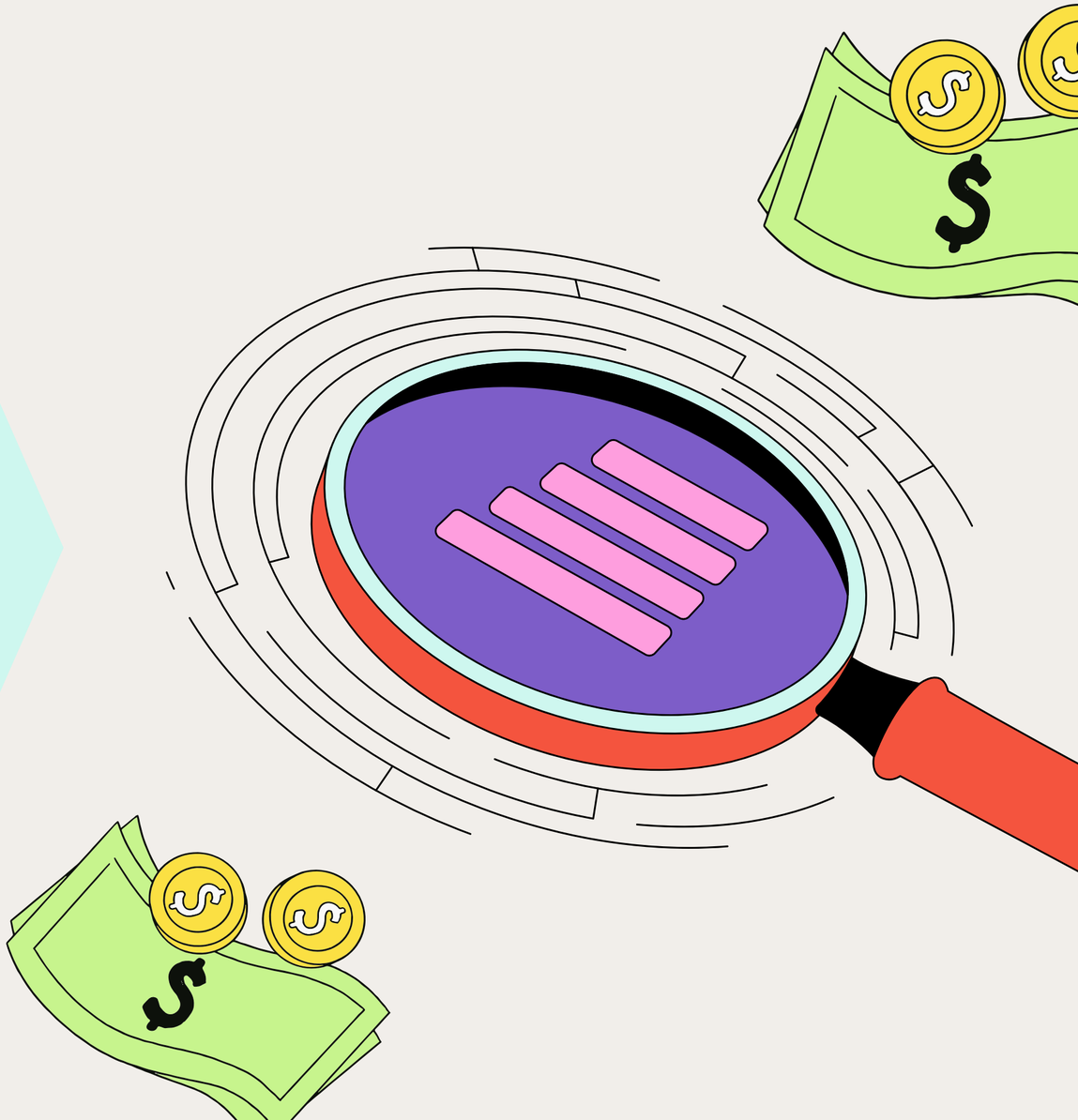
4. Productos inactivos: Existen 504 productos bloqueados y 498 vencidos, lo que representa oportunidades desaprovechadas.

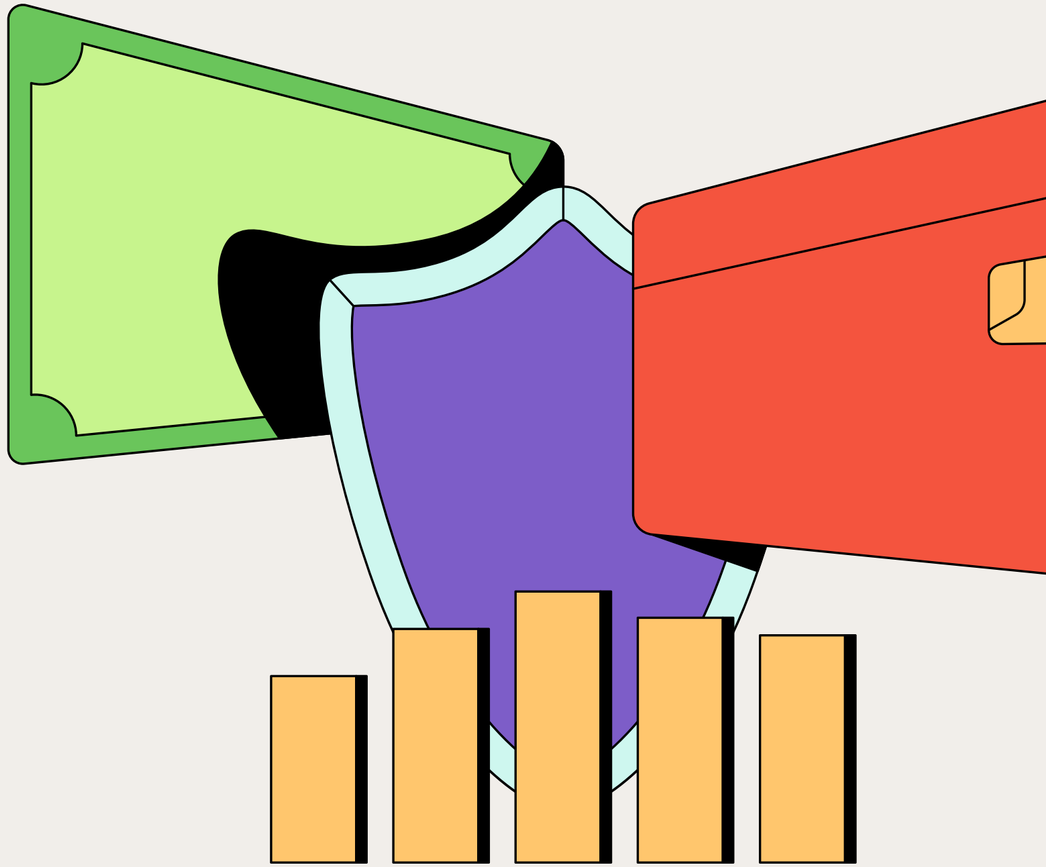
- Recomendación: Lanzar campañas de reactivación con beneficios especiales y simplificar procesos de renovación.

Conclusiones

Conclusión

Este proyecto demostró que el trabajo conjunto entre bases de datos relacionales, análisis automatizado y visualización dinámica es clave para optimizar la gestión financiera. Gracias a MySQL, Python y Power BI fue posible estructurar datos complejos, obtener indicadores relevantes y presentarlos de manera accesible para apoyar decisiones de alto impacto.



www.bancoQapital.com

Muchas gracias