

# **Guía de Implementación - Sistema de Consulta de Clientes**

**Fecha creación**

11 Diciembre 2025

**Responsable de desarrollo**

Jefferson Pérez Bedoya

**Responsable de documentación**

Jefferson Pérez Bedoya

**Rios del Desierto SAS**

# Requisitos Previos

## Software Necesario

- **Python 3.8 o superior** (Recomendado: Python 3.11+)
- **pip** (gestor de paquetes de Python)
- **Navegador web moderno** (Chrome, Firefox, Edge)
- **Git** (opcional, para clonar el repositorio)

## Verificar Instalaciones

```
python --version pip --version
```

## Instalación Paso a Paso

### 1. Descargar el Proyecto

#### Opción A: Desde Git

```
git clone  
https://github.com/Jefer526/Prueba\_Tecnica/tree/main/Prueba\_Falabella
```

#### Opción B: Desde ZIP

1. Descargar el archivo ZIP
2. Extraer en una carpeta
3. Abrir terminal en esa carpeta

### 2. Instalar Dependencias del Backend

```
cd backend pip install -r requirements.txt
```

#### Si hay errores con pandas en Windows:

```
pip install --upgrade pip pip install Flask Flask-SQLAlchemy  
Flask-CORS openpyxl python-dateutil pip install pandas
```

#### Verificar instalación:

```
pip list
```

Deben aparecer:

- Flask
- Flask-SQLAlchemy
- Flask-CORS
- pandas
- openpyxl

### **3. Inicializar la Base de Datos**

#### **Opción A: Usando el Backend (Crea DB vacía)**

```
python app.py
```

Se creará automáticamente database.db con las tablas y tipos de documento.

**Detener el servidor:** Ctrl + C

#### **Opción B: Poblar con Datos de Prueba (Recomendado)**

```
# En otra terminal cd data python populate_db.py
```

Cuando pregunte, escribir s y presionar Enter.

Esto creará:

- 7 clientes de prueba
- 37 compras
- 3 clientes que califican para fidelización

### **4. Iniciar el Servidor Backend**

```
cd backend python app.py
```

**Salida esperada:**

```
 Base de datos inicializada correctamente * Serving Flask app  
'app' * Debug mode: on * Running on http://127.0.0.1:5000
```

**⚠ IMPORTANTE:** Dejar esta terminal abierta mientras uses la aplicación.

### **5. Abrir el Frontend**

#### **Opción A: Doble Click**

1. Navegar a la carpeta frontend/
2. Doble click en index.html

#### **Opción B: Desde el Navegador**

1. Abrir navegador
2. Presionar Ctrl + O
3. Navegar a: ruta/del/proyecto/frontend/index.html

# Configuración para Producción

## 1. Cambiar Modo Debug

En **backend/app.py**, cambiar la última línea:

```
# Desarrollo app.run(debug=True, host='0.0.0.0', port=5000)  #
Producción app.run(debug=False, host='0.0.0.0', port=5000)
```

## 2. Usar un Servidor WSGI (Producción)

### Instalar Gunicorn (Linux/Mac)

```
pip install gunicorn
```

### Ejecutar con Gunicorn

```
cd backend gunicorn -w 4 -b 0.0.0.0:5000 app:app
```

### Instalar Waitress (Windows)

```
pip install waitress
```

### Ejecutar con Waitress

```
cd backend waitress-serve --host=0.0.0.0 --port=5000 app:app
```

## 3. Configurar Variables de Entorno

Crear archivo **backend/.env**:

```
SECRET_KEY=tu-clave-secreta-muy-segura-aqui FLASK_ENV=production
DATABASE_URI=sqlite:///production.db
```

## 4. Servir Frontend con un Servidor Web

### Opción A: Python Simple HTTP Server

```
cd frontend python -m http.server 8080
```

Acceder en: <http://localhost:8080>

### Opción B: Nginx (Producción real)

Configurar Nginx para servir los archivos estáticos de frontend/.

# Verificación de Instalación

## 1. Verificar Backend

Abrir en navegador: <http://127.0.0.1:5000/api/tipos-documento>

Debe mostrar:

```
{ "tipos_documento": [ {"codigo": "CC", "descripcion": "Cédula de Ciudadanía"}, {"codigo": "NIT", "descripcion": "NIT"}, {"codigo": "PA", "descripcion": "Pasaporte"} ] }
```

## 2. Verificar Frontend

1. Abrir frontend/index.html
2. Debe mostrar la lista de clientes disponibles
3. Probar búsqueda con documento: 1234567890

## 3. Verificar Exportaciones

1. Buscar un cliente
2. Click en "Exportar CSV"
3. Verificar que se descargue el archivo

## 4. Verificar Reporte de Fidelización

1. Click en "Descargar Reporte de Fidelización"
2. Debe descargar un Excel con 3 clientes

## Datos de Prueba

### Clientes que CALIFICAN para fidelización (>5M COP):

Tipo	Número	Nombre
CC	1234567890	Juan Pérez García
CC	9876543210	Maria González López
NIT	900123456-1	Distribuidora El Éxito SAS

### Clientes que NO califican:

Tipo	Número	Nombre
CC	1122334455	Carlos Rodríguez Méndez
PA	AB123456	John Smith
CC	5566778899	Ana Martínez Silva
CC	4455667788	Luis Ramírez Torres

# Solución de Problemas

## Error: "Module not found"

```
pip install -r requirements.txt
```

## Error: CORS (Frontend no se conecta al Backend)

Verificar que Flask-CORS esté instalado:

```
pip install Flask-CORS
```

## Base de datos corrupta

Eliminar y recrear:

```
rm backend/database.db python backend/app.py python  
data/populate_db.py
```

# Estructura de Archivos

```
PRUBA_TECNICA_FALABELLA/  
|   └── backend/  
|       |   └── app.py           # Aplicación Flask principal  
|       |   └── models.py        # Modelos de base de datos  
|       |   └── config.py        # Configuración  
|       |   └── requirements.txt  # Dependencias |  
|       └── database.db        # Base de datos SQLite  
|           └── exports/        # Archivos exportados  
|   └── frontend/  
|       |   └── index.html      # Interfaz principal  
|       |   └── style.css        # Estilos  
|       |   └── script.js        # Lógica del frontend  
|   └── data/  
|       └── populate_db.py    # Script de datos de prueba  
└── docs/  
    |   └── INSTALACION.md     # Esta guía  
    |   └── DOCUMENTACION.md   # Documentación técnica
```