

1. ¿Qué hace este programa?

Este programa lee un archivo de texto con deudas de clientes, valida el formato de cada línea y carga los datos en una tabla MySQL llamada deudas. Al final, devuelve un código de resultado indicando si todo salió bien o hubo errores.

2. Requisitos

Antes de usarlo, el programador necesita:

1. MySQL instalado y funcionando.
2. Python instalado (3.x).
3. La librería de Python para conectar con MySQL:

Código fuente

```
#bash  
pip install mysql-connector-python
```

Código fuente

```
#MySQL  
CREATE DATABASE deudas_clientes;  
USE deudas_clientes;  
  
CREATE TABLE deudas (  
    id_deuda          VARCHAR(15) NOT NULL,  
    id_cliente        VARCHAR(15) NOT NULL,  
    nombre_cliente    VARCHAR(60) NOT NULL,  
    correo            VARCHAR(60) NOT NULL,  
    monto_deuda       DECIMAL(20,2) NOT NULL,  
    fecha_vencimiento DATE        NOT NULL,  
    PRIMARY KEY (id_deuda)  
);
```

3. Estructura de archivos del proyecto

En una carpeta (por ejemplo PruebaEstudiantes) debe tener:

- cargar_deudas.py → script principal.
- deuda_clientes.txt → archivo con las deudas.

Ejemplo de contenido de deuda_clientes.txt:

```
#txt  
1546587676-4;juan perez saez;juan.perez@gmail.com;19000;zbc1233;12-03-2021  
20456789-3;camila rojas muñoz;camila.rojas@example.com;37500;zxq7890;25-11-  
2025
```

Formato de cada línea:

```
#txt  
id_cliente;nombre;correo;monto;id_deuda;fecha(DD-MM-YYYY)
```

4. Configuración dentro de cargar_deudas.py

En el script, el programador debe revisar dos cosas:

1. **Ruta del archivo** (si está en la misma carpeta, basta con el nombre):

```
#Python  
RUTA_ARCHIVO = "deuda_clientes.txt"
```

2. Datos de conexión a MySQL:

```
#Python  
DB_CONFIG = {  
    "host": "localhost",  
    "user": "root",  
    "password": "1234",  
    "database": "deudas_clientes",  
}
```

Si su usuario o contraseña son distintos, los cambia ahí.

5. Cómo ejecutar el proceso de carga

1. Abrir la carpeta del proyecto en VS Code (o terminal).
2. Abrir una terminal en esa carpeta. Debe verse algo así:

```
#Python  
C:\Users\Usuario\Documents\PruebaEstudiantes>
```

3. Ejecutar el script:

```
#Python  
python cargar_deudas.py
```

4. Observar el mensaje en consola.

El programa puede devolver:

- OK: archivo cargado correctamente → **código 0 (éxito)**
 - ERROR: archivo no encontrado → **código 1**
 - ERROR de formato: ... → **código 2**
 - ERROR BD: ... → **código 3**
-

6. Cómo verificar que se cargaron las deudas

En MySQL Workbench (o consola):

```
#MySQL
```

```
USE deudas_clientes;  
SELECT * FROM deudas;
```

Ahí debería ver los mismos datos que están en deuda_clientes.txt, pero ya ordenados en columnas.

7. Cómo agregar más deudas

Para cargar nuevas deudas:

1. Abrir deuda_clientes.txt.
2. Agregar nuevas líneas al final con el mismo formato:

```
#txt
```

```
12345678-9;pedro diaz;pedro.diaz@example.com;65000;abc1234;10-01-2026
```

3. Guardar el archivo.

4. Volver a ejecutar:

```
#Bash
```

```
python cargar_deudas.py
```

Importante: id_deuda **no debe repetirse** porque es clave primaria.

8. Cómo “resetear” la tabla para probar de nuevo

Si quiere empezar desde cero sin datos:

#MySQL

```
USE deudas_clientes;  
TRUNCATE TABLE deudas;
```

Luego vuelve a ejecutar:

#Bash

```
python cargar_deudas.py
```