

## CAPTURAS DEL EXAMEN

Script de creación de la base de datos, de la entidad resultados\_exámenes, auditoria y trigger:

```
DESKTOP-2NFKL52.B...de base de datos  BionetDatabase.sql...2.BioNetDB (sa (54)) -> X
id INT IDENTITY(1,1) PRIMARY KEY,
operacion NVARCHAR(10), -- 'INSERT' o 'UPDATE'
paciente_id INT,
tipo_examen NVARCHAR(100),
fecha DATETIME DEFAULT GETDATE()
);
GO

-- Trigger para registrar inserciones o actualizaciones
CREATE TRIGGER trg_log_resultados
ON resultados_exámenes
AFTER INSERT, UPDATE
AS
BEGIN
    INSERT INTO log_cambios_resultados (operacion, paciente_id, tipo_examen)
    SELECT
        CASE
            WHEN EXISTS (SELECT * FROM inserted EXCEPT SELECT * FROM deleted) THEN 'UPDATE'
            ELSE 'INSERT'
        END,
        paciente_id,
        tipo_examen
    FROM inserted;
END;
GO
```

100 %

Messages

Commands completed successfully.

Completion time: 2025-04-23T19:00:46.6171700-05:00

Tablas creadas:

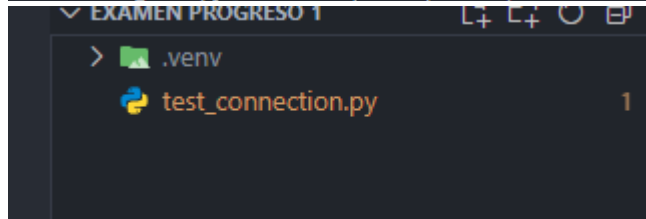
log_cambios_resultados	
id	
operacion	
paciente_id	
tipo_examen	
fecha	

resultados_exámenes	
id	
laboratorio_id	
paciente_id	
tipo_examen	
resultado	
fecha_examen	

## CAPTURAS DEL EXAMEN

Creación del proyecto con Python:

```
Kevito@DESKTOP-2NFKL52 MINGW64 /d/Universidad/Sistemas Integrados/Examen progreso 1
$ pip install pyodbc pandas
Defaulting to user installation because normal site-packages is not writeable
Collecting pyodbc
  Downloading pyodbc-5.2.0-cp313-cp313-win_amd64.whl.metadata (2.8 kB)
Collecting pandas
  Downloading pandas-2.2.3-cp313-cp313-win_amd64.whl.metadata (19 kB)
Collecting numpy>=1.26.0 (from pandas)
```



Instalación de Python-dotenv para manejar las variables de entorno y prueba de conexión a la base de datos:

```
Bienvenido | test_connection.py 2 x | .env
1 import os
2 from dotenv import load_dotenv
3 import pyodbc
4
5 # Cargar variables de entorno
6 load_dotenv()
7
8 # Parámetros desde el archivo .env
9 server = os.getenv('DB_SERVER')
10 database = os.getenv('DB_NAME')
11 username = os.getenv('DB_USER')
12 password = os.getenv('DB_PASSWORD')
13
14 try:
15     conn = pyodbc.connect(
16         f'DRIVER={{ODBC Driver 17 for SQL Server}};SERVER={server};DATABASE={database};UID={username};PWD={password}'
17     )
18     print("✅ Conexión exitosa a SQL Server (usando .env)")
19     conn.close()
20 except Exception as e:
21     print("❌ Error de conexión:", e)
22
```

PROBLEMAS 2 | SALIDA | CONSOLA DE DEPURACIÓN | TERMINAL | PUERTOS

```
Kevito@DESKTOP-2NFKL52 MINGW64 /d/Universidad/Sistemas Integrados/Examen progreso 1
$ python test_connection.py
✅ Conexión exitosa a SQL Server (usando .env)
```

## CAPTURAS DEL EXAMEN

Creación de script de transferencia de archivos:

```
import os
import shutil
import csv

# Directorios
INPUT_DIR = '../input-labs'
PROCESSED_DIR = '../processed'
ERROR_DIR = '../error'

# Encabezados esperados
EXPECTED_HEADERS = ['laboratorio_id', 'paciente_id', 'tipo_examen', 'resultado', 'fecha_examen']

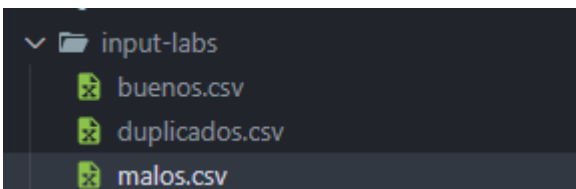
def validar_csv(file_path):
    try:
        with open(file_path, 'r', encoding='utf-8') as f:
            reader = csv.reader(f)
            headers = next(reader)
            return headers == EXPECTED_HEADERS
    except Exception as e:
        print(f"❌ Error al leer archivo {file_path}: {e}")
        return False

def procesar_archivos():
    for filename in os.listdir(INPUT_DIR):
        if filename.endswith('.csv'):
            file_path = os.path.join(INPUT_DIR, filename)
            print(f"📁 Revisando archivo: {filename}")

            if validar_csv(file_path):
                shutil.move(file_path, os.path.join(PROCESSED_DIR, filename))
                print(f"✅ {filename} es válido y fue movido a /processed")
            else:
                shutil.move(file_path, os.path.join(ERROR_DIR, filename))
                print(f"❌ {filename} inválido. Movido a /error")

if __name__ == '__main__':
    procesar_archivos()
```

Creación de csv para pruebas, buenos, duplicados, y mal formato:

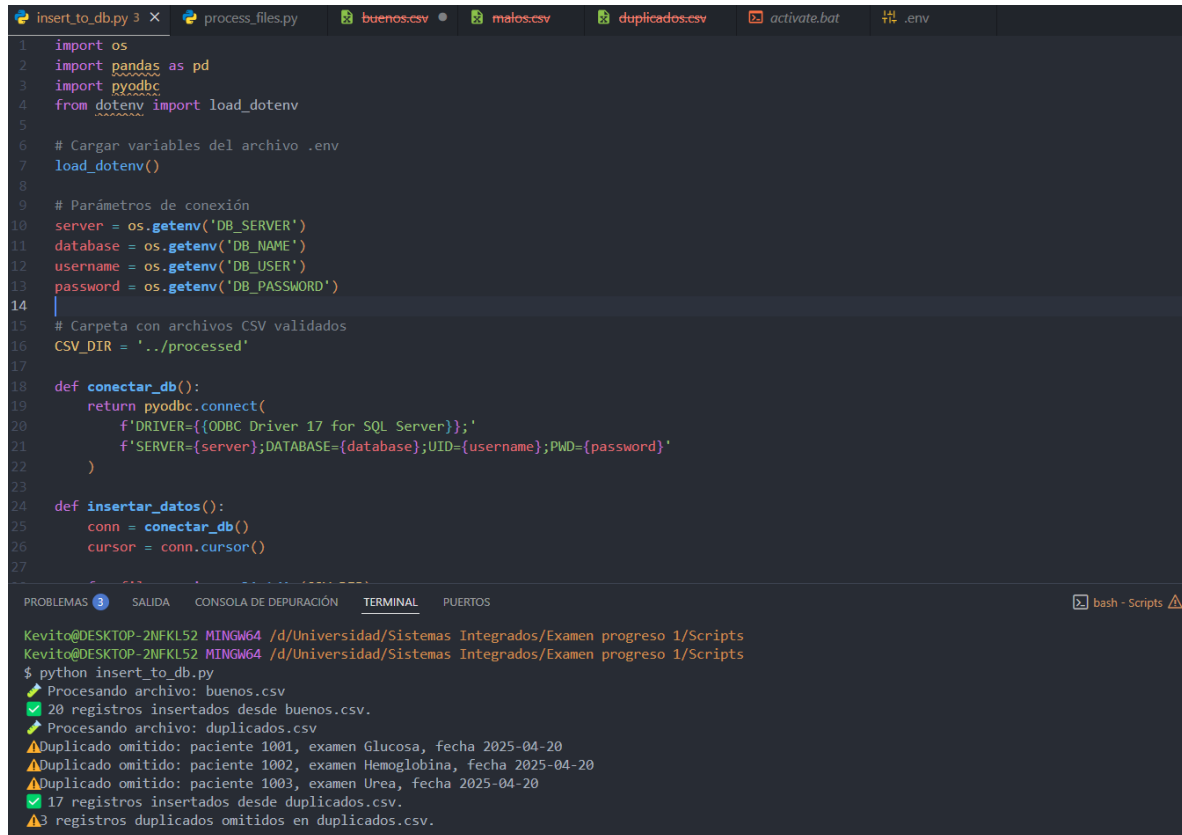


Corrida del script con los csv creados (El que tiene error no se mueve a la carpeta procesado y los duplicados se controlaran en el siguiente script):

```
Kevito@DESKTOP-2NFKL52 MINGW64 /d/Universidad/Sistemas Integrados/Examen progreso 1/Scripts
$ python process_files.py
📁 Revisando archivo: buenos.csv
✅ buenos.csv es válido y fue movido a /processed
📁 Revisando archivo: duplicados.csv
✅ duplicados.csv es válido y fue movido a /processed
📁 Revisando archivo: malos.csv
❌ malos.csv inválido. Movido a /error
```

## CAPTURAS DEL EXAMEN

Insertión de los datos a la base de datos (Validación de registros duplicados y omisión):



The image shows a code editor with a Python script named `insert_to_db.py` and a terminal window below it. The script uses `os`, `pandas`, and `pyodbc` to connect to a database and insert data from CSV files. The terminal output shows the execution of the script, including the processing of `buenos.csv` and `duplicados.csv`, with messages indicating successful insertions and the omission of duplicate records.

```
1 import os
2 import pandas as pd
3 import pyodbc
4 from dotenv import load_dotenv
5
6 # Cargar variables del archivo .env
7 load_dotenv()
8
9 # Parámetros de conexión
10 server = os.getenv('DB_SERVER')
11 database = os.getenv('DB_NAME')
12 username = os.getenv('DB_USER')
13 password = os.getenv('DB_PASSWORD')
14
15 # Carpeta con archivos CSV validados
16 CSV_DIR = '../processed'
17
18 def conectar_db():
19     return pyodbc.connect(
20         f'DRIVER={{ODBC Driver 17 for SQL Server}};'
21         f'SERVER={server};DATABASE={database};UID={username};PWD={password}'
22     )
23
24 def insertar_datos():
25     conn = conectar_db()
26     cursor = conn.cursor()
27
28     # Procesar archivos CSV
29     for file in os.listdir(CSV_DIR):
30         df = pd.read_csv(os.path.join(CSV_DIR, file))
31         # Validar y omitir duplicados
32         # ... (código omitido) ...
33         # Insertar datos
34         cursor.executemany(
35             f'INSERT INTO {database} ({columns}) VALUES ({placeholders})',
36             df.to_records(index=False)
37         )
38     conn.commit()
39     cursor.close()
40     conn.close()
41
42 if __name__ == '__main__':
43     insertar_datos()
```

PROBLEMAS 3 SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS bash - Scripts

```
Kevito@DESKTOP-2NFKL52 MINGW64 /d/Universidad/Sistemas Integrados/Examen progreso 1/Scripts
Kevito@DESKTOP-2NFKL52 MINGW64 /d/Universidad/Sistemas Integrados/Examen progreso 1/Scripts
$ python insert_to_db.py
Procesando archivo: buenos.csv
✓ 20 registros insertados desde buenos.csv.
Procesando archivo: duplicados.csv
⚠ Duplicado omitido: paciente 1001, examen Glucosa, fecha 2025-04-20
⚠ Duplicado omitido: paciente 1002, examen Hemoglobina, fecha 2025-04-20
⚠ Duplicado omitido: paciente 1003, examen Urea, fecha 2025-04-20
✓ 17 registros insertados desde duplicados.csv.
⚠ 3 registros duplicados omitidos en duplicados.csv.
```

## CAPTURAS DEL EXAMEN

Verificación de registros en la base de datos:

SELECT \* FROM resultados\_examenes;

SELECT \* FROM log\_cambios\_resultados;

100 %

Results

Messages

	id	laboratorio_id	paciente_id	tipo_examen	resultado	fecha_examen
1	1	1	1001	Glucosa	85	2025-04-20
2	2	1	1002	Hemoglobina	13.5	2025-04-20
3	3	2	1003	Urea	30	2025-04-20
4	4	2	1004	Creatinina	0.9	2025-04-20
5	5	3	1005	Colesterol	190	2025-04-20
6	6	3	1006	TSH	1.5	2025-04-20
7	7	1	1007	T4 Libre	1.2	2025-04-20
8	8	2	1008	PCR	0.5	2025-04-20
9	9	3	1009	Femina	300	2025-04-20
10	10	1	1010	Triglicéridos	150	2025-04-20
11	11	1	1011	Hemoglobina	14.1	2025-04-20

	id	operacion	paciente_id	tipo_examen	fecha
1	1	UPDATE	1001	Glucosa	2025-04-23 19:26:12.893
2	2	UPDATE	1002	Hemoglobina	2025-04-23 19:26:12.897
3	3	UPDATE	1003	Urea	2025-04-23 19:26:12.897
4	4	UPDATE	1004	Creatinina	2025-04-23 19:26:12.897
5	5	UPDATE	1005	Colesterol	2025-04-23 19:26:12.900
6	6	UPDATE	1006	TSH	2025-04-23 19:26:12.900
7	7	UPDATE	1007	T4 Libre	2025-04-23 19:26:12.900
8	8	UPDATE	1008	PCR	2025-04-23 19:26:12.900
9	9	UPDATE	1009	Femina	2025-04-23 19:26:12.900
10	1...	UPDATE	1010	Triglicéridos	2025-04-23 19:26:12.903
11	1...	UPDATE	1011	Hemoglobina	2025-04-23 19:26:12.903
12	1...	UPDATE	1012	Glucosa	2025-04-23 19:26:12.903
13	1...	UPDATE	1013	Urea	2025-04-23 19:26:12.903
14	1...	UPDATE	1014	TSH	2025-04-23 19:26:12.903
15	1...	UPDATE	1015	Colesterol	2025-04-23 19:26:12.903
16	1...	UPDATE	1016	Creatinina	2025-04-23 19:26:12.903
17	1...	UPDATE	1017	T4 Libre	2025-04-23 19:26:12.903
18	1...	UPDATE	1018	PCR	2025-04-23 19:26:12.907
19	1...	UPDATE	1019	Triglicéridos	2025-04-23 19:26:12.907
20	2...	UPDATE	1020	Femina	2025-04-23 19:26:12.907
21	2...	UPDATE	2001	Examen_1	2025-04-23 19:26:12.907
22	2...	UPDATE	2002	Examen_2	2025-04-23 19:26:12.910
23	2...	UPDATE	2003	Examen_3	2025-04-23 19:26:12.910
24	2...	UPDATE	2004	Examen_4	2025-04-23 19:26:12.910