



## PDI - TP 3

Alumnos:

Alonso Facundo

Alvarez Poli Bautista

Kloster Agustin



Luego de haber instalado todo lo necesario para el tp (SQLite, Flask, sqlalchemy, DB Browser) creamos la base de datos, en vez de en el terminal, en el DB Browser con el nombre "datos\_sensores.db".

Luego ejecutamos el código en sensores\_r2 para crear la tabla

```
1 import time
2 import random
3 import sqlite3
4 from flask import Flask, render_template, jsonify
5 from datetime import datetime
6 from funciones import geo_latlon
7
8 app = Flask(__name__)
9
10 def create_table():
11     conn = sqlite3.connect('datos_sensores.db')
12     cursor = conn.cursor()
13     cursor.execute('CREATE TABLE IF NOT EXISTS lectura_sensores (
14                     id INTEGER PRIMARY KEY,
15                     co2 REAL,
16                     temp REAL,
17                     hum REAL,
18                     fecha TEXT,
19                     lugar TEXT,
20                     altura REAL,
21                     presion REAL,
22                     presion_nm REAL,
23                     temp_ext REAL
24                 )')
```

Verificamos con el gestor que haya sido creado correctamente.

Agregamos nuevos valores a través del código python

```
while cont < cant_capturas:
    cont += 1
    verdadero = 1
    if verdadero == 1:
        print("Datos Disponibles!")
        co2_medido = random.uniform(250, 1100)
        temp_sensor = random.uniform(temp_ext, temp_ext + 10)
        humedad_relativa = random.uniform(40, 80)
        print("CO2: %d PPM" % co2_medido)
        print("Temperatura: %0.2f degrees C" % temp_sensor)
        print("Humedad: %0.2f %% rH" % humedad_relativa)

        d = datetime.now()
        print("Fecha", d)
        timestampStr = d.strftime("%d-%b-%Y (%H:%M:%S.%f)")

        conn = sqlite3.connect('datos_sensores.db')
        cursor = conn.cursor()
        cursor.execute('INSERT INTO lectura_sensores (co2, temp, hum, fecha, lugar, altura, presion, presion_nm, temp_ext)
                        VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)',
                        (co2_medido, temp_sensor, humedad_relativa, timestampStr, lugar, altura, presion, presion_nm, temp_ext))
        conn.commit()
        conn.close()

        print("Registro insertado..., acumulados:", cont, "\n")
        time.sleep(delta_t_capturas)
        print("\nEsperando nuevo registro de datos ...\n")
```



Desde el programa en Python importo las dependencias.

Adaptar el código en github de acuerdo al criterio de diseño elegido.

@app.route('/') muestra una tabla sobre los datos (gracias a un archivo html)

### Sensores de prueba

Type a keyword...												
id	CO2	CO2 Norm	Temp	Hum	Fecha	Lugar	Altura	Presion	Presion nm	Temp ext	Temp ref	
1	430.07		23.48	63.31	10-Jun-2024 (20:01:59.533864)	abc	3.00	1014.00	1014.00	16.11		
2	1001.00		21.41	75.32	10-Jun-2024 (20:02:00.543867)	abc	3.00	1014.00	1014.00	16.11		
3	494.04		24.16	77.05	10-Jun-2024 (20:02:01.556176)	abc	3.00	1014.00	1014.00	16.11		
4	799.91		16.43	58.03	10-Jun-2024 (20:02:02.571512)	abc	3.00	1014.00	1014.00	16.11		
5	688.60		22.69	61.43	10-Jun-2024 (20:02:03.582586)	abc	3.00	1014.00	1014.00	16.11		
6	582.80		24.71	45.80	10-Jun-2024 (20:22:08.401282)	abc	4.00	1014.00	1014.00	16.87		
7	1016.14		19.90	57.93	10-Jun-2024 (20:22:09.410033)	abc	4.00	1014.00	1014.00	16.87		
8	862.42		18.43	70.00	10-Jun-2024 (20:22:10.423069)	abc	4.00	1014.00	1014.00	16.87		
9	251.21		18.93	68.46	10-Jun-2024 (20:22:11.437999)	abc	4.00	1014.00	1014.00	16.87		
10	604.79		19.67	60.39	10-Jun-2024 (20:22:12.445610)	abc	4.00	1014.00	1014.00	16.87		
11	670.07		19.23	44.61	10-Jun-2024 (20:23:27.369365)	bcd	4.00	1014.00	1014.00	16.87		
12	432.75		18.79	62.69	10-Jun-2024 (20:23:28.378715)	bcd	4.00	1014.00	1014.00	16.87		
13	417.44		18.66	62.52	10-Jun-2024 (20:23:29.393156)	bcd	4.00	1014.00	1014.00	16.87		
14	448.41		25.75	49.24	10-Jun-2024 (20:23:30.407784)	bcd	4.00	1014.00	1014.00	16.87		

@app.route('/api/prueba')

```
@app.route('/api/prueba')
def index2():
    return "Prueba de ruta API..."
```

Se devuelve un mensaje para probar que la ruta funciona

@app.route('/api/todos-los-datos')

```
@app.route('/api/todos-los-datos')
def mostrar_todos_los_datos():
    try:
        records = LecturaSensores.query.all()
        if not records:
            return jsonify({'mensaje': 'No se encontraron datos'}), 404
        return jsonify([record.to_dict() for record in records])
    except Exception as e:
        return jsonify({'error': 'Error al buscar datos', 'detalle': str(e)}), 500
```



Se realiza un try en el cual se buscan los datos

Si la lista records está vacía, devuelve un mensaje JSON indicando que no se encontraron con un error 404 (not found)

Si ocurre algún error, devuelve un mensaje de "Error al buscar datos" con un error 500

### @app.route('/api/primer-registro')

A diferencia de /todos-los-datos, este unicamente obtiene el primer registro de la tabla, o sea el primero almacenado

### @app.route('/api/directorio-db')

Muestra el directorio de la base de datos.

```
{
  "directorio": "/home/Poli/.vscode-server/Protocolos/",
  "path": "/home/Poli/.vscode-server/Protocolos/datos_sensores.db"
}
```

### @app.route('/api/insertar-dato')

Inserta un nuevo dato con

co2=400.00, temp=22.50, hum=45.00, fecha="2023-06-08", lugar="Oficina", altura=100.00, presion=1013.25, presion\_nm=1010.00, temp\_ext=18.00

11	670.07		19.23	44.61	10-Jun-2024 (20:23:27.369365)	bcd	4.00	1014.00	1014.00	16.87
12	432.75		18.79	62.69	10-Jun-2024 (20:23:28.378715)	bcd	4.00	1014.00	1014.00	16.87
13	417.44		18.66	62.52	10-Jun-2024 (20:23:29.393156)	bcd	4.00	1014.00	1014.00	16.87
14	448.41		25.75	49.24	10-Jun-2024 (20:23:30.407784)	bcd	4.00	1014.00	1014.00	16.87
15	400.00		22.50	45.00	2023-06-08	Oficina	100.00	1013.25	1010.00	18.00