



**Docente:** Juan Pablo Ruiz Rosero [jpabloruiz@unicauca.edu.co](mailto:jpabloruiz@unicauca.edu.co)

### 3.Base de datos en Python

Una base de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. SQLite es un sistema de gestión de bases de datos relacional, contenida en una relativamente pequeña biblioteca escrita en C. SQLite se enlaza con el programa pasando a ser parte integral del mismo, por lo cual no es un servicio independiente.

1. Desde la consola Linux instale el explorador de base de datos SQLite, SQLite Browser

```
sudo apt-get install sqlitebrowser
```

Nota: Para Windows lo puede descargar desde: <http://sqlitebrowser.org/>

2. Cree el archivo `testDatabase.py` y copie el siguiente contenido:

```
import sqlite3
conn = sqlite3.connect('example.db3')

c = conn.cursor()

# Create table
c.execute('''CREATE TABLE students (fecha text, nombre text, apellido text, edad integer)''')

# Insert a row of data
c.execute("INSERT INTO students VALUES ('2017-09-04','Pedro','Dominguez',30)")

# Save (commit) the changes
conn.commit()

# Close connection
conn.close()
```

3. Ejecute el archivo `testDatabase.py` y abra con el programa SQLite Browser el archivo de base de datos `example.db3`. Seleccione la pestaña `Browse data` y encuentre los datos introducidos de Pedro Dominguez.
4. Cree una carpeta llamada `guia04` y copie el archivo adjunto `dbFunctions.py`
5. Cree un archivo llamado `globals.py` donde guardará las variables y constantes globales de este ejercicio.

```
LOCATION_X = 1
LOCATION_Y = 2
GATEWAY_ID = "User_Laptop"
```

Nota: Las constantes se escriben en mayúsculas y las variables en minúsculas. Reemplace la `LOCATION_X` y `LOCATION_Y` con la ubicación de su puesto respecto a la puerta.

6. Cree un archivo llamado `createTable.py` desde donde importe la librería `dbFunctions` y llame a la función `createTable()` así:

```
import dbFunctions
```

```
dbFunctions.createTable()
```

Ejecute este archivo y corrobore que se halla creado el archivo de base de datos `sensors.db3`

7. Cree el archivo `insertValue.py` mediante el cual inserte el el SensorID = 'sensor1' a la base de datos con el tiempo actual:

```
import time
```

```
import dbFunctions
```

```
sensor = {}
```

```
sensor["date"] = time.time()
```

```
sensor["sensorID"] = "sensor8"
```

```
dbFunctions.insertSensorEvent(sensor)
```

8. Cree el archivo `insertValues.py` el cual mediante la función `raw_input()` y un while loop inserte múltiples sensorID con el tiempo actual, hasta que se inserte el sensorID "fin".
9. Cree el archivo `getAllEvents.py` el cual usando la función `dbFunctions.getAllEvents()` imprima mediante un ciclo for todos los eventos de sensores registrados en la base de datos.
10. Cree el archivo `getAllEventsFrom.py` el cual usando la función `dbFunctions.getAllEventsFrom()` imprima mediante un ciclo for todos los eventos de sensores registrados en la base de datos, del "sensor1".
11. Cree el archivo `ConvertCSVtoDB.py` con el cual lea los datos del archivo `dataPopayan.csv` y los guarde en el archivo base de datos `popayan.db3`. Añada funciones a la librería `dbFunctions.db` si lo considera necesario.
12. Cree el archivo `MaxTemperatureDate.py` el cual abra la información de la base de datos `popayan.db3` e imprima la fecha y la temperatura correspondiente a la máxima registrada.