



Universidad Nacional de Rosario
Facultad de Ciencias Exactas, Ingeniería y Agrimensura
Escuela de Ingeniería Electrónica

Universidad Nacional de Rosario
Facultad de Ciencias Exactas, Ingeniería y Agrimensura
Escuela de Ingeniería Electrónica
Bases de datos

Trabajo Práctico N.º 2
IoT-SQLite

Autor:

Grupo N.º	
Nombre y Apellido	N.º de Legajo
Buono Luciano	B-5649/9

Corrigió	Calificación



1. Tareas desarrolladas

En el archivo `initsqlite.py` se crearon tres tablas: `table_Terrazatemperatura`, `table_Jardintemperatura`, `table_Patiotemperatura` donde cada una posee un campo real para el valor de la temperatura, un campo de texto para la unidad y un campo real para la fecha.

```
# SQLite DB Table Schema
table_schema="""
drop table if exists table_Terrazatemperatura ;
create table table_Terrazatemperatura (
    value real,
    unit text,
    timestamp real primary key
);
drop table if exists table_Patiotemperatura ;
create table table_Patiotemperatura (
    value real,
    unit text,
    timestamp real primary key
);
drop table if exists table_Jardintemperatura ;
create table table_Jardintemperatura (
    value real,
    unit text,
    timestamp real primary key
);"""
```

Dentro del archivo `subscribe.py` se realizaron las subscripciones a los tres tópicos y dentro de la función `on_message` se ejecuta la función **`write_to_db`**.

`On_message` se ejecuta cuando el Broker le pasa información al cliente MQTT.

Luego dentro de `sensors.py` se crearon los 3 sensores a utilizar en este trabajo además de otros como `table_PatioHumedad`, `table_TerrazaHumedad`, `table_TerrazaNoisemeter`.

Por último, dentro de `sqlitehandler.py` tenemos la función `write_to_database` que se encarga de realizar el query a ingresar en la base de datos.

```
def write_to_db(topic, jsonData):

    dbObj = DatabaseManager()
    data = json.loads(jsonData)
    query = '''INSERT INTO ''' + topic + '''(value, unit, timestamp)
VALUES(?,?,?)'''
```



```
print (query)
dbObj.add_del_update_db_record(query,(data['value'], data['unit'],
data['timestamp']))
del dbObj
```

2. Consultas.

- Obtener el valor promedio de algún sensor de temperatura.

```
cur.execute('select AVG(value) FROM(table_Terrazatemperatura)')
aux=cur.fetchone()
print ("1. Temperatura promedio de un sensor (Terraza):
"+str(aux[0])+"°C" )
```

- Obtener el valor promedio de todos los sensores de temperatura.

```
cur.execute('SELECT AVG(value) FROM (SELECT value FROM
table_Terrazatemperatura UNION SELECT value FROM table_Jardintemperatura
UNION SELECT value FROM table_Patiotemperatura)')
aux= cur.fetchone()
print("2. Temperatura promedio de todos los sensores: "+str(aux[0])+"°C"
)
```

- Obtener el valor promedio del último día de todos los sensores de temperatura. (HINT: el timestamp de las lecturas de los sensores está en formato UNIX-time).

```
# --Agarro el timestamp del ultimo elemento(ultimo del dia 3)
cur.execute('SELECT timestamp FROM table_Terrazatemperatura order by
timestamp desc limit 1')
aux1= cur.fetchone()
#Le resto al ultimo elemento 64800. de esta forma me aseguro de solo
agarrar los elementos del ultimo dia
cur.execute('SELECT AVG(value) FROM (SELECT value,timestamp FROM
table_Terrazatemperatura UNION SELECT value,timestamp FROM
table_Jardintemperatura UNION SELECT value,timestamp FROM
table_Patiotemperatura) WHERE timestamp > ('+str(aux1[0])+' -86400)')
aux= cur.fetchone()
print("3. El valor promedio de temperatura de todos los sensores en el
ultimo dia es: "+str(aux[0])+"°C")
```

- Obtener la fecha del momento más caluroso de la casa. (La máxima lectura de alguno de los sensores de temperatura)



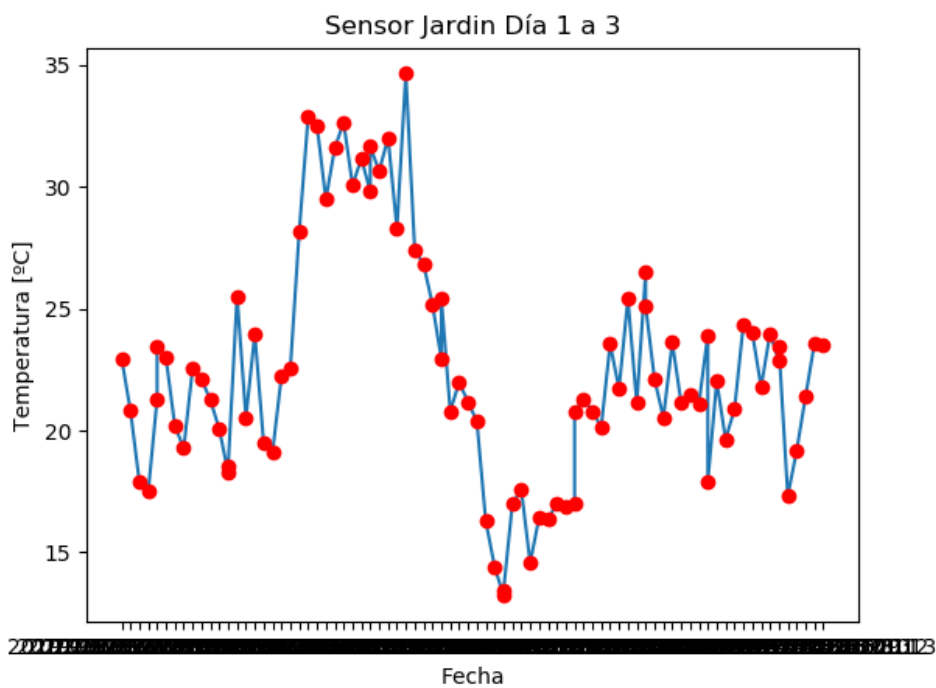
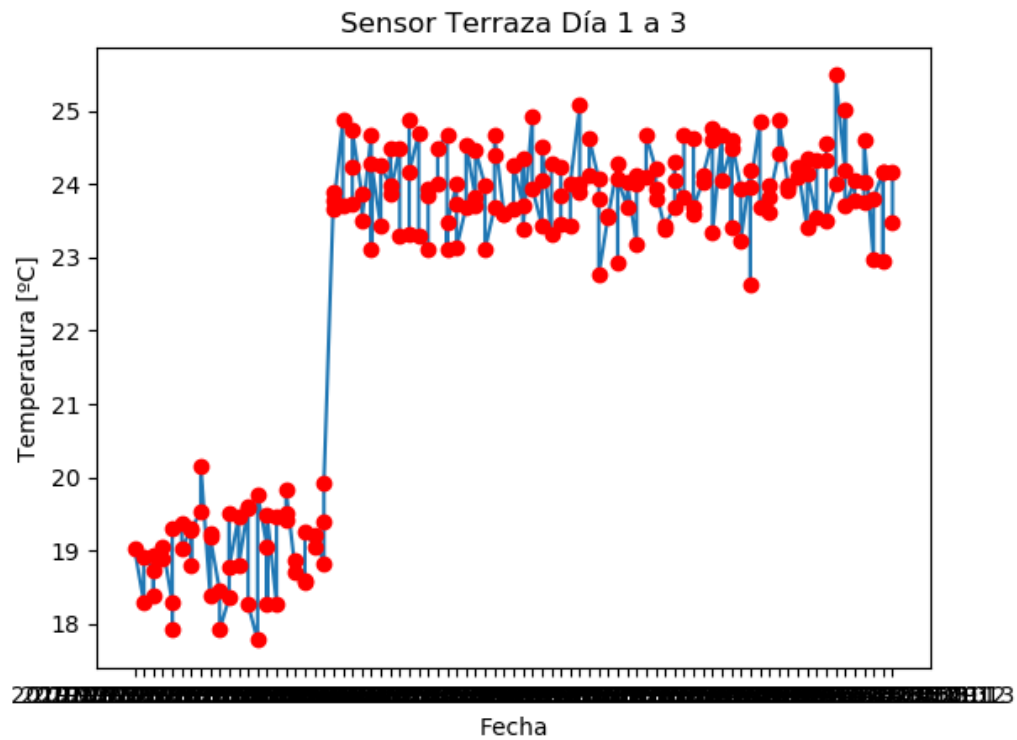
```
cur.execute('SELECT MAX(value), timestamp FROM (SELECT value,timestamp  
FROM table_Terrazatemperatura UNION SELECT value,timestamp FROM  
table_Jardintemperatura UNION SELECT value,timestamp FROM  
table_Patiotemperatura) ')  
aux= cur.fetchone()  
#Epoch a tiempo actual  
maxTempTime = time.asctime( time.localtime(aux[1] ) )  
print("4. El valor maximo de temp de todos los sensores fue:  
"+str(aux[0]) +"°C y sucedio: "+maxTempTime)
```

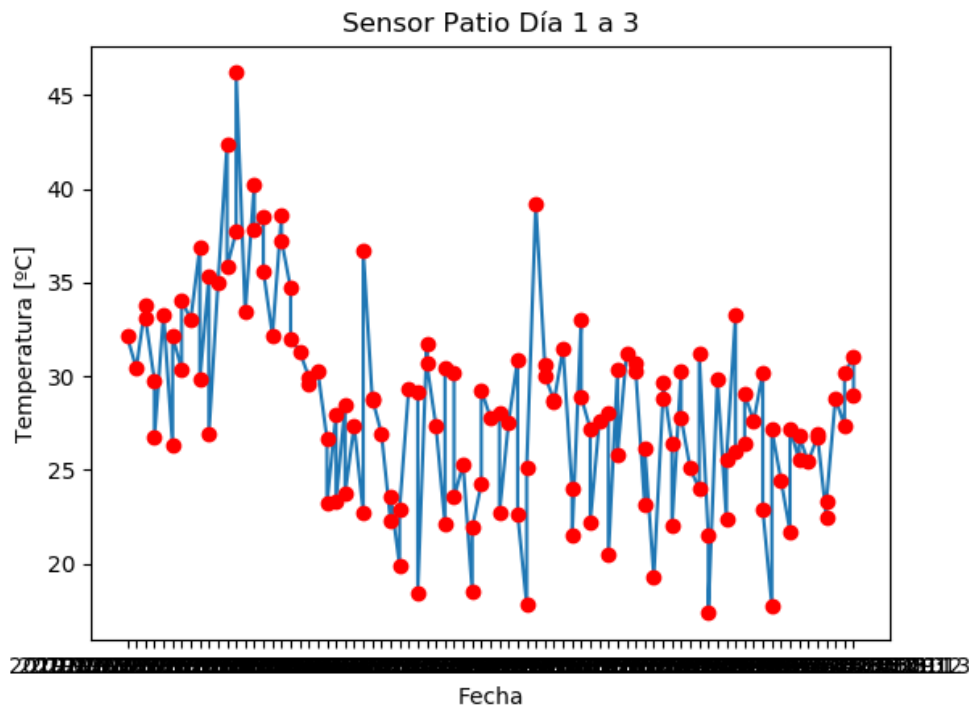
Resultados:

1. Temperatura promedio de un sensor (Terraza): 22.70430256781739°C
2. Temperatura promedio de todos los sensores: 24.467048525494114°C
3. El valor promedio de temperatura de todos los sensores en el ultimo dia es:
1564010413.7183506°C
4. El valor maximo de temp de todos los sensores fue: 46.21078704726974°C y sucedio: Sun
Jul 21 20:21:05 2019



3. Gráficos





4. Enlaces

El repositorio se puede encontrar en https://github.com/luciano-buono/sensors_async_sim/tree/TP2-LucianoBuono en branch: [TP2-LucianoBuono](#).