

# Ejercicios SQLITE.

## Desarrollo de aplicaciones sobre sistemas operativos de proposito general - CESE. FIUBA.

### Ejercicio 1

Escribir un programa que reciba por línea de comandos el nombre de un archivo de base de datos con el cual operar. El programa deberá permitir al usuario escribir consultas SQL de cualquier tipo. En el caso de ser una consulta del tipo SELECT, deberá imprimir el resultado de la consulta. Al ingresar el comando “exit” el programa terminará.

### Ejercicio 2

Desarrollar un servicio web con flask y SQLite para un sistema de domótica. El mismo cuenta con una lista de lámparas las cuales tienen un ID, IP, un nombre y un estado (On/Off). El servicio web contará con dos direcciones:

- /devices
  - Método: GET
  - Devolverá en formato JSON la lista de lámparas con todos sus datos.
- /devices/ID
  - Método: PUT
  - ID: Id de la lámpara
  - Body: “state”:X
    - donde X: Estado, 0: OFF. 1: ON
  - Respuesta: {“state”:”OK”} o {“state”:”ERROR”}

NOTAS: Usar el programa cliente\_python provisto probar.

### Ejercicio 3

Tomar el sistema del ejercicio 2, y agregar una tabla de LOG, que indique cada vez que se modifica el estado de una lámpara, el ID de la lámpara y el estado que se modificó, junto con un timestamp. Luego agregar el servicio web que permita consultar dicho log:

- /log?page=X&size=Y
  - Método: GET
  - X: Número de página (desde cero)
  - Y: Cantidad de entradas de cada página.

### Ejercicio 4

Crear un programa que consulte la lista de lámparas al servicio web cada medio segundo y cuando detecte un cambio en alguna de ellas, envíe un paquete UDP a la IP de dicha lámpara en el puerto 4096 con el siguiente formato ascii: “>ST:ON\n” o “>ST:OFF\n”.

### Ejercicio 5

Para probar el sistema, Crear un programa uxiliar que sea un server UDP para probar que lleguen las tramas, simulando ser las lámparas. Utilizar el servicio web /device/ID, para hacer UPDATES en los estados de las lámparas, prendiendo y apagándolas para probar el funcionamiento del sistema completo.