

**TECNOLÓGICO DE ESTUDIOS SUPERIORES DE IXTAPALUCA
ORGANISMO PÚBLICO DESCENTRALIZADO DEL GOBIERNO DEL ESTADO DE
MEXICO**

PROYECTO DE INVESTIGACIÓN.

Investigación documental.

DIVISIÓN

INGENIERÍA EN SISTEMAS COMPUTACIONALES.

QUE PRESENTAN:

GERARDO MARTINEZ BARRIOS.

ORLANDO RAMIREZ CURIEL.

JUAN MANUEL SEGOBIA GARCIA.

Ing. Ebner Juárez Elías
Asesor Técnico

Ing. Ebner Juárez Elías
Asesor Metodológico

Licda. María Elena Orozco Álvarez
Subdirectora de Estudios
Profesionales

Noviembre 22 de 2021.

Objetivo.

General: Instalar sqlite para poder desarrollar la practica para poder retroalimentar los conocimientos a cerca de manejar tablas y datos en las bases de datos.

Introducción.

SQLite es una biblioteca de C que provee una base de datos ligera basada en disco que no requiere un proceso de servidor separado y permite acceder a la base de datos usando una variación no estándar del lenguaje de consulta SQL. Algunas aplicaciones pueden usar SQLite para almacenamiento interno. También es posible prototipar una aplicación usando SQLite y luego transferir el código a una base de datos más grande como PostgreSQL u Oracle también nos dice Siahaan 2019 (“La gran flexibilidad y movilidad de la base de datos SQLite3 la convierten en la primera opción para que cualquier desarrollador la use y la integre con cualquier producto con el que trabaje.”)...

Desarrollo

Primero se debe instalar DB Browser for SQL lite, descargamos el archivo desde la página sqlitebrowser.org.

Una vez teniendo el archivo lo ejecutamos y listo.



Para iniciar la practica debemos hacer una base de datos. Primero debemos importar el módulo SQL lite 3, después se debe crear una conexión con la base de datos.

```
In [1]: import sqlite3
```

Después con el siguiente comando creamos una conexión a la base de datos por que como dice Dominguez 2021 ("Un objeto de conexión se crea utilizando la función `connect ():`")...

```
In [2]: cnexion = sqlite3.connect("ejemplo.db")
```

Ahora creamos una tabla, para esto creamos una variable de tipo cursor.

```
In [4]: conexion = sqlite3.connect("ejemplo.db")  
  
#Creamos el cursor  
cursor = conexion.cursor()  
  
conexion.close()
```

Ahora ejecutamos el código y buscamos el archivo que se acaba de crear. En nuestro caso lo guardamos en el escritorio.



Lo abrimos y vemos que aún no tiene nada.



Ahora colocamos los siguientes comandos para que se pueda ingresar datos, hacer tablas y etc.

```
In [4]: conexion = sqlite3.connect("ejemplo.db")

#Creamos el cursor
cursor = conexion.cursor()

conexion.close()
```

Creamos una tabla para almacenar información llamando al método execute.

```
In [10]: conexion = sqlite3.connect("ejemplo.db")
cursor = conexion.cursor()

#creamos una tabla llamada estudiantes
cursor.execute("CREATE TABLE estudiantes (email VARCHAR(100), carrera VARCHAR(100), nombre VARCHAR(100), edad INTEGER)")

conexion.close()
```

Ahora ingresamos datos en nuestra tabla.

```
In [11]: conexion = sqlite3.connect("ejemplo.db")
cursor = conexion.cursor()

#Insertamos un registro en la tabla
cursor.execute("INSERT INTO estudiantes VALUES ('bluenote@google.com', 'Artes', 'Sharon', 27)")

#Guardamos los cambios haciendo commit
conexion.commit()

conexion.close()
```

Ahora ingresamos estos comandos para que se puedan leer los datos.

```
In [12]: conexion = sqlite3.connect('ejemplo.db')
        cursor = conexion.cursor()

        #Seleccionamos todos los registros en la tabla
        cursor.execute("SELECT * FROM estudiantes")

        usuarios = cursor.fetchone()
        print(usuarios)

('bluenote@google.com', 'Artes', 'Sharon', 27)
```

1.4 INSERTAR Y LEER DATOS EN LA BASE DE DATOS 1

Creamos una tabla mediante el comando CREATE TABLE

```
In [10]: conexion = sqlite3.connect("ejemplo.db")
        cursor = conexion.cursor()

        #creamos una tabla llamada estudiantes
        cursor.execute("CREATE TABLE estudiantes (email VARCHAR(100), carrera VARCHAR(100), nombre VARCHAR(100), edad INTEGER)")
        conexion.close()
```

INSERT INTO es el comando para ingresar datos

```
In [11]: conexion = sqlite3.connect("ejemplo.db")
        cursor = conexion.cursor()

        #Insertamos un registro en la tabla
        cursor.execute("INSERT INTO estudiantes VALUES ('bluenote@google.com','Artes','Sharon',27)")

        #Guardamos los cambios haciendo commit
        conexion.commit()

        conexion.close()
```

SELECT * FROM es el comando para leer el contenido de una tabla

```
In [12]: conexion = sqlite3.connect('ejemplo.db')
        cursor = conexion.cursor()

        #Seleccionamos todos los registros en la tabla
        cursor.execute("SELECT * FROM estudiantes")

        usuarios = cursor.fetchone()
        print(usuarios)

('bluenote@google.com', 'Artes', 'Sharon', 27)
```

1.4 INSERTAR Y LEER DATOS EN LA BASE DE DATOS 2

Podemos ingresar y leer varios registros al mismo tiempo

```
In [13]: conexion = sqlite3.connect("ejemplo.db")
        cursor = conexion.cursor()

        usuarios = [
            ('paella@vip.com', 'Medicina', 'Giuliana', 27),
            ('crimewave@vip.com', 'Derecho', 'Alicia', 20),
            ('prince@orange.com', 'Ingenieria', 'William', 24),
            ('elrisortdewelini@vip.com', 'Nutrición', 'Chefsito', 25),
            ('planb@vip.com', 'Artes', 'Chencho', 30),
            ('channel@boyz.com', 'Letras', 'Frank', 32),
            ('vinos@aouju.com', 'Nutrición', 'Eugenia', 23)
        ]

        cursor.executemany("INSERT INTO estudiantes VALUES (?, ?, ?, ?)", usuarios)

        conexion.commit()

        conexion.close()
```

1.5 LEER UN CSV E INSERTARLO EN UNA BASE DE DATOS

```
In [15]: import csv

        conexion = sqlite3.connect(r"C:\Users\HUTCH\ejemplo.db")
        cursor = conexion.cursor()

        archivo = open(r"C:\Users\HUTCH\datos_db.txt")

        filas = csv.reader(archivo)

        cursor.executemany("INSERT INTO estudiantes VALUES (?, ?, ?, ?)", filas)

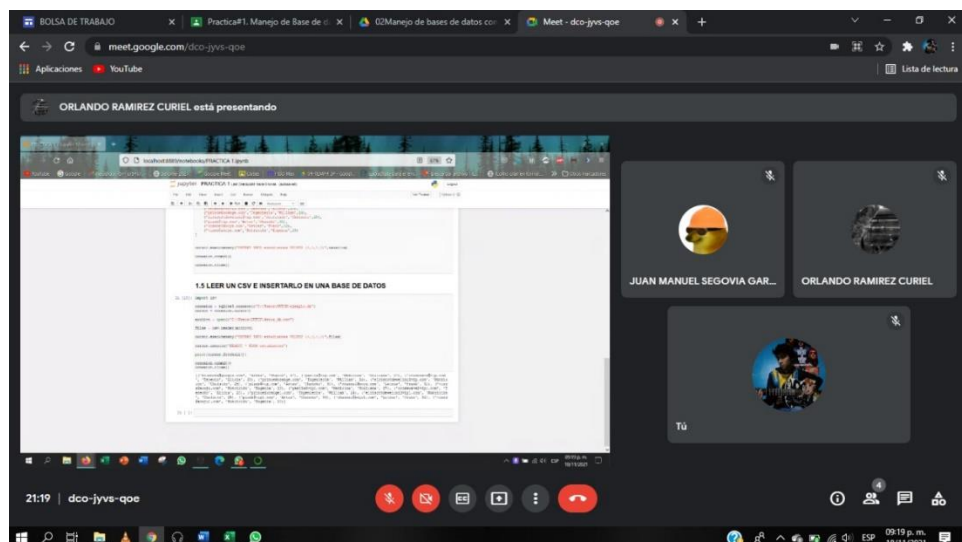
        cursor.execute("SELECT * FROM estudiantes")

        print(cursor.fetchall())

        conexion.commit()
        conexion.close()

[('bluenote@google.com', 'Artes', 'Sharon', 27), ('paella@vip.com', 'Medicina', 'Giuliana', 27), ('crimewave@vip.com', 'Derecho', 'Alicia', 20), ('prince@orange.com', 'Ingenieria', 'William', 24), ('elrisortdewelini@vip.com', 'Nutrición', 'Chefsito', 25), ('planb@vip.com', 'Artes', 'Chencho', 30), ('channel@boyz.com', 'Letras', 'Frank', 32), ('vinos@aouju.com', 'Nutrición', 'Eugenia', 23), ('paella@vip1.com', 'Medicina', 'Giuliana', 27), ('crimewave@vip1.com', 'Derecho', 'Alicia', 20), ('prince@orange1.com', 'Ingenieria', 'William', 24), ('elrisortdewelini@vip1.com', 'Nutrición', 'Chefsito', 25), ('planb@vip1.com', 'Artes', 'Chencho', 30), ('channel@boyz1.com', 'Letras', 'Frank', 32), ('vinos@aouju1.com', 'Nutrición', 'Eugenia', 23)]
```

Evidencia de equipo.



Conclusión

SQLite en general, es una base de datos server-less que se puede utilizar en casi todos los lenguajes de programación, incluido Python. Server-less significa que no hay necesidad de instalar un servidor separado para trabajar con SQLite para que pueda conectarse directamente con la base de datos.

Recordando la competencia conocer los conceptos fundamentales de los sistemas basados en conocimientos, así como el estado del arte de las áreas relacionadas al mismo lo cual consideramos comprendida y la práctica de sqlite y phyton la consideramos completada al 100%.

Github:

Juan: <https://github.com/SegoviaJuan/Analisis-y-Visualizacion-de-Datos->

Orlando: <https://github.com/OelandoRamirezCuriel-1/Analisis-y-Visualizacion-de-datos.git>

Gerardo: <https://github.com/Gerardomt515/analisisyvisualizaciondedatos>

Bibliografías:

Academy, P. L. (2020). *Python para Principiantes: 2 Libros en 1: Programación de Python para principiantes + Libro de trabajo de Python*. PROGRAMMING LANGUAGES ACADEMY.

Bailey, J., Djermanovic, D., & Dominguez, O. A. (2021). *Saving Data on Android (Second Edition): Learn Jetpack DataStore, Room, Firebase & SQLite with Kotlin*. Razeware LLC.

Siahaan, V., & Sianipar, R. H. (2019). *Learn SQLite with Python: Building Database-Driven Desktop Projects*. Independently published.