



TECNOLÓGICO DE ESTUDIOS SUPERIORES DE IXTAPALUCA

ORGANISMO PÚBLICO DESCENTRALIZADO DEL GOBIERNO DEL ESTADO DE MÉXICO

**PROYECTO DE INVESTIGACIÓN  
MANEJO DE BASE DE DATOS CON SQLITE 3**

**DIVISIÓN  
INGENIERÍA EN SISTEMAS COMPUTACIONALES**

**ASIGNATURA  
SISTEMAS BASADOS EN EL CONOCIMIENTO**

**MAESTRO  
ING. JUÁREZ ELÍAS EBNER**

**QUE PRESENTAN:  
CARRILLO CABAÑAS HIRAM  
MORALES JUAREZ ABIGAHÍ**

**GRUPO: 1952    TURNO: VESPERTINO**

**IXTAPALUCA, ESTADO DE MÉXICO, 22 NOVIEMBRE 2021**

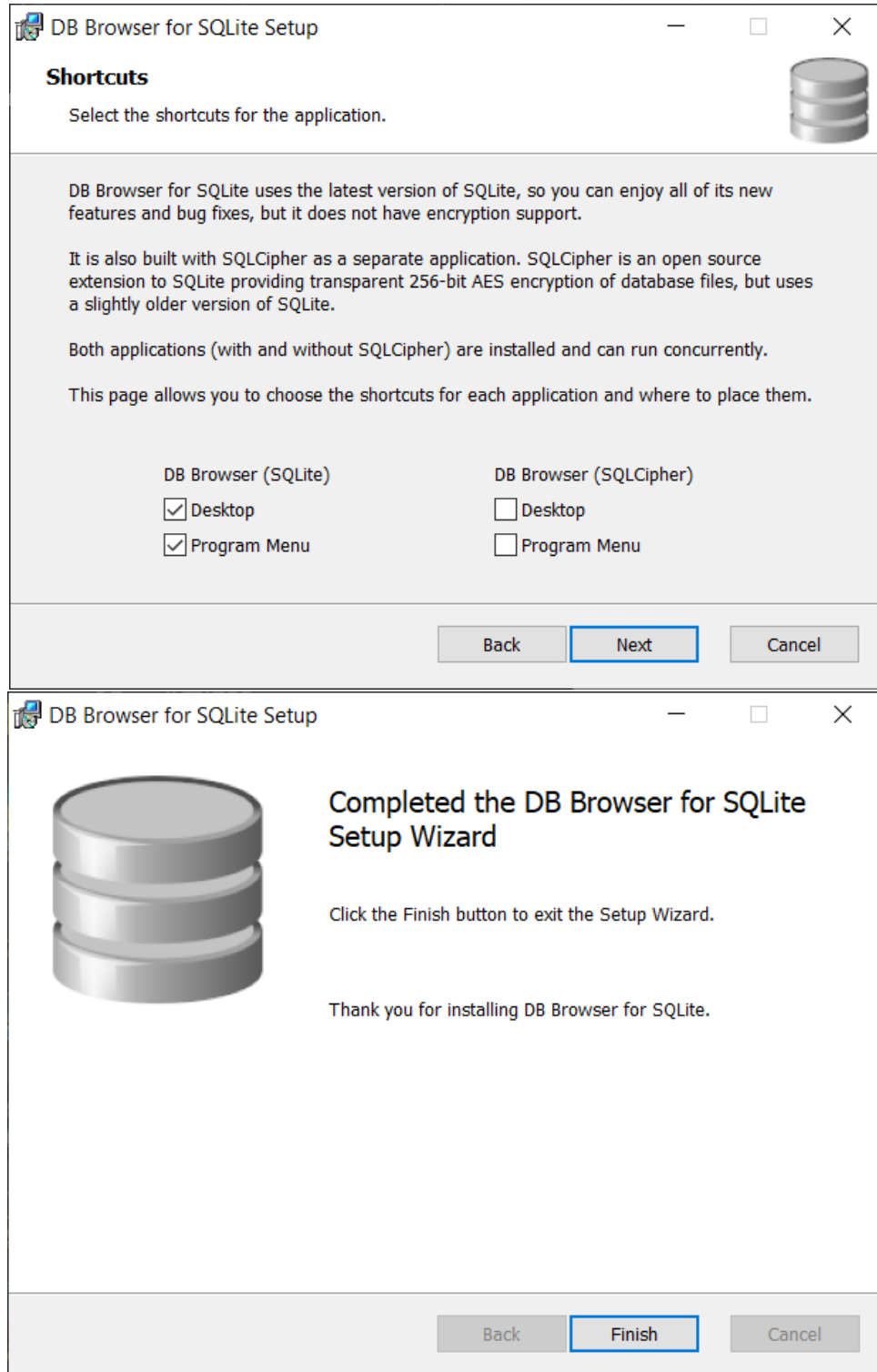
## **INTRODUCCIÓN**

SQLite es una herramienta de software libre, que permite almacenar información en dispositivos empujados de una forma sencilla, eficaz, potente, rápida y en equipos con pocas capacidades de hardware, como puede ser una PDA o un teléfono celular. SQLite implementa el estándar SQL92 y también agrega extensiones que facilitan su uso en cualquier ambiente de desarrollo.

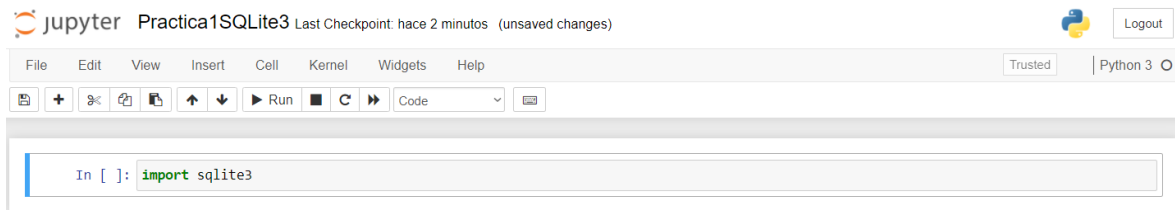
Esto permite que SQLite soporte desde las consultas más básicas hasta las más complejas del lenguaje SQL, y lo más importante es que se puede usar tanto en dispositivos móviles como en sistemas de escritorio, sin necesidad de realizar procesos complejos de importación y exportación de datos, ya que existe compatibilidad al 100% entre las diversas plataformas disponibles, haciendo que la portabilidad entre dispositivos y plataformas sea transparente.

## DESARROLLO

Iniciamos descargando el programa BD Browser SQLite y posteriormente se instala.



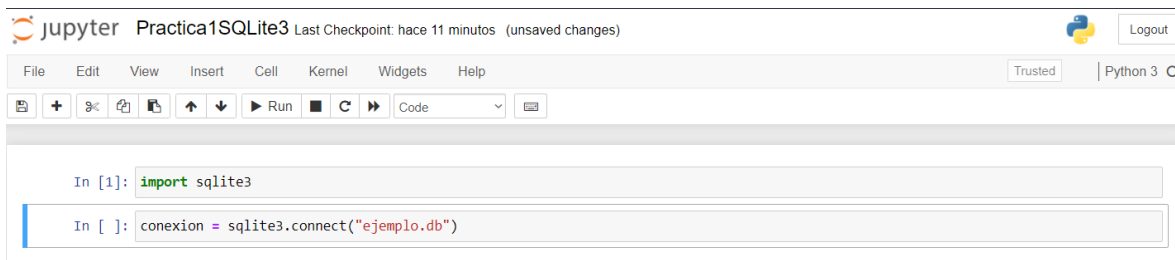
Comenzamos por crear una base de datos para ello es necesario importa el módulo SQLite3, ingresamos el siguiente comando.



A screenshot of a Jupyter Notebook interface. The title bar shows 'jupyter Practica1SQLite3' and 'Last Checkpoint: hace 2 minutos (unsaved changes)'. The menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. The toolbar has icons for file operations and execution. The code area contains a single cell with the text: `In [ ]: import sqlite3`.

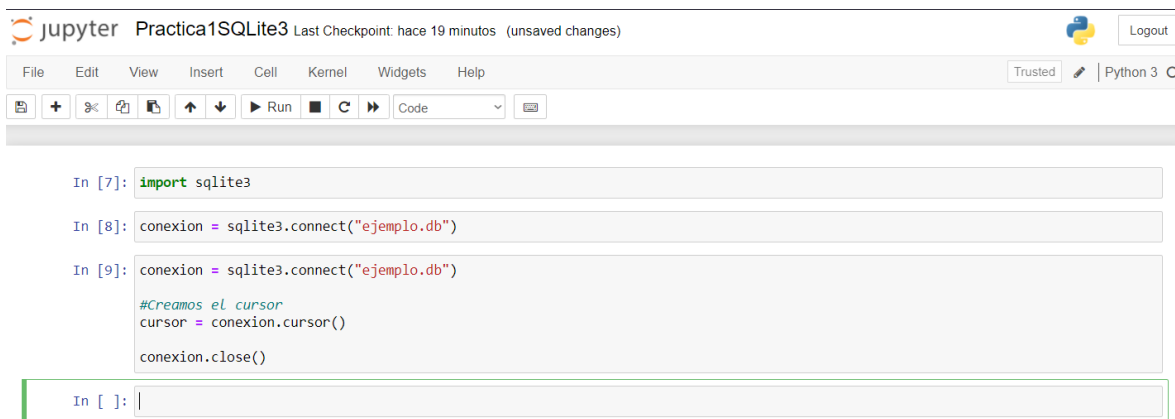
Una vez importado el módulo de SQLite3, con el siguiente comando crearemos una conexión a la base de datos.

(Ebrahim, 2019) “Para utilizar SQLite3 en Python, primero deberás importar el módulo `sqlite3` y luego crear un objeto de conexión para conectarnos a la base de datos. Este nos permitirá ejecutar las sentencias SQL. Un objeto de conexión se crea utilizando la función `connect ()`:”



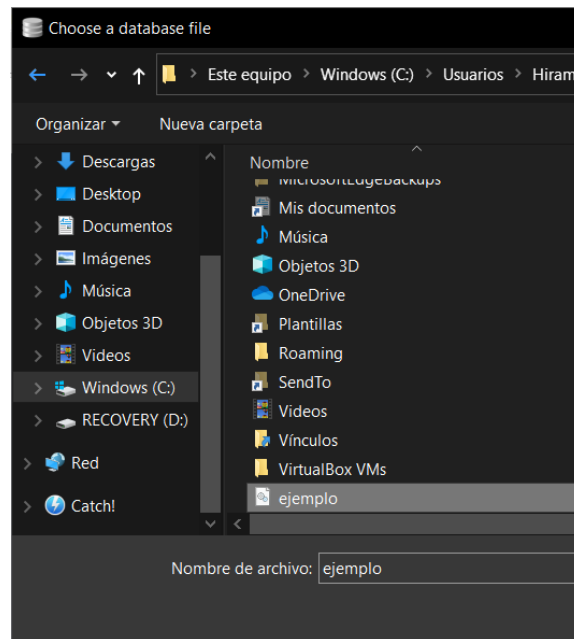
A screenshot of a Jupyter Notebook interface. The title bar shows 'jupyter Practica1SQLite3' and 'Last Checkpoint: hace 11 minutos (unsaved changes)'. The code area contains two cells. The first cell has the text: `In [1]: import sqlite3`. The second cell has the text: `In [ ]: conexion = sqlite3.connect("ejemplo.db")`.

Para crear una tabla es necesario crear una variable de tipo cursor.

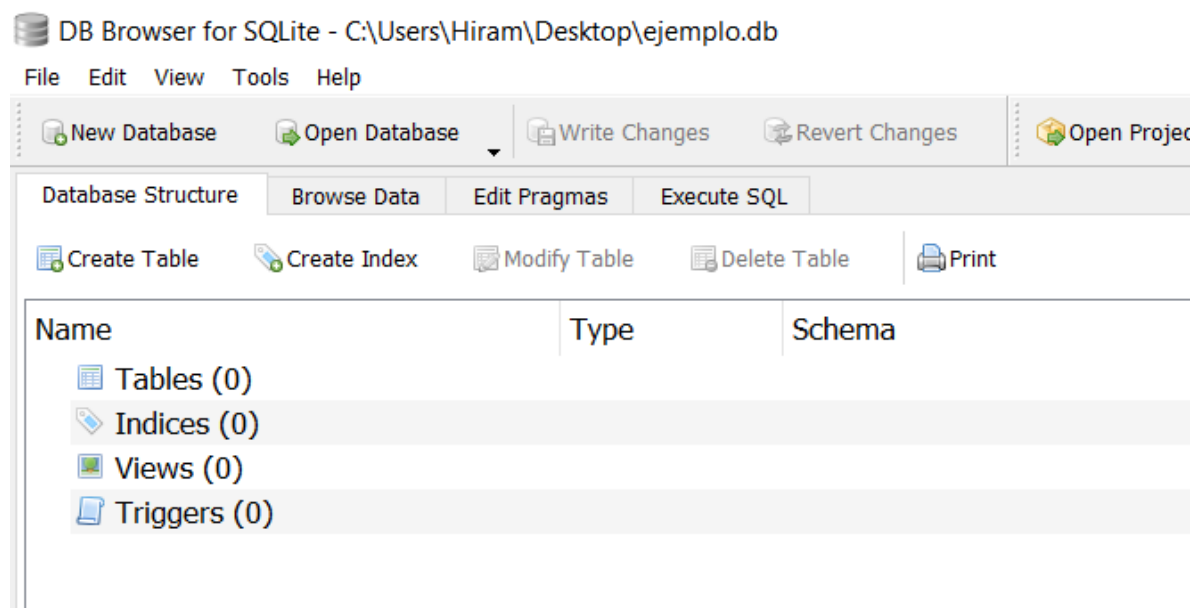


A screenshot of a Jupyter Notebook interface. The title bar shows 'jupyter Practica1SQLite3' and 'Last Checkpoint: hace 19 minutos (unsaved changes)'. The code area contains three cells. The first cell has the text: `In [7]: import sqlite3`. The second cell has the text: `In [8]: conexion = sqlite3.connect("ejemplo.db")`. The third cell has the text: `In [9]: conexion = sqlite3.connect("ejemplo.db")`, `#Creamos el cursor`, `cursor = conexion.cursor()`, and `conexion.close()`. Below these is an empty code cell with the text: `In [ ]: |`.

Una vez ejecutado el código se busca el archivo creado en BD Browser SQLite.



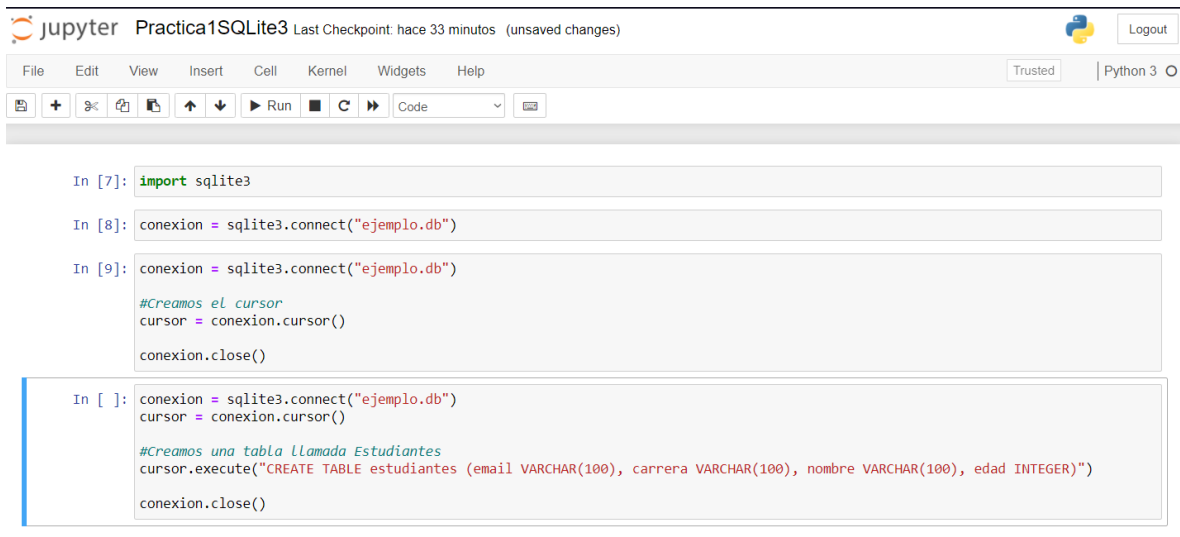
Al abrir el archivo se visualiza que no tiene contenido ya que fue recién creado.



Ejecutamos nuevamente el comando.

```
In [9]: conexion = sqlite3.connect("ejemplo.db")  
  
#Creamos el cursor  
cursor = conexion.cursor()  
  
conexion.close()
```

Para crear una tabla será por el método execute.



The image shows a Jupyter Notebook interface with the title 'Practica1SQLite3'. The top bar indicates 'Last Checkpoint: hace 33 minutos (unsaved changes)' and includes a 'Logout' button. The menu bar includes 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', 'Widgets', and 'Help'. The toolbar shows icons for file operations, running, and code execution. The notebook contains four code cells:

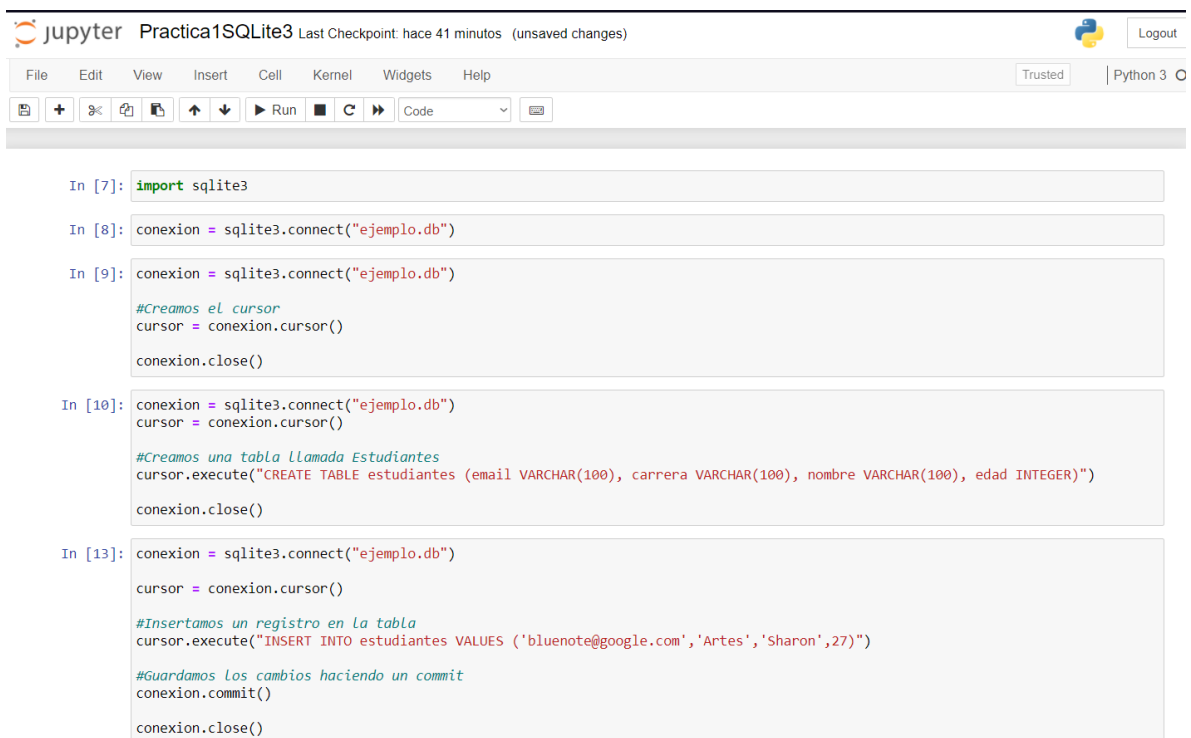
```
In [7]: import sqlite3

In [8]: conexion = sqlite3.connect("ejemplo.db")

In [9]: conexion = sqlite3.connect("ejemplo.db")
#Creamos el cursor
cursor = conexion.cursor()
conexion.close()

In [ ]: 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()
```

Para ingresar los datos a la tabla con INSERT INTO.



The image shows a Jupyter Notebook interface with the title 'Practica1SQLite3'. The top bar indicates 'Last Checkpoint: hace 41 minutos (unsaved changes)' and includes a 'Logout' button. The menu bar includes 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', 'Widgets', and 'Help'. The toolbar shows icons for file operations, running, and code execution. The notebook contains five code cells:

```
In [7]: import sqlite3

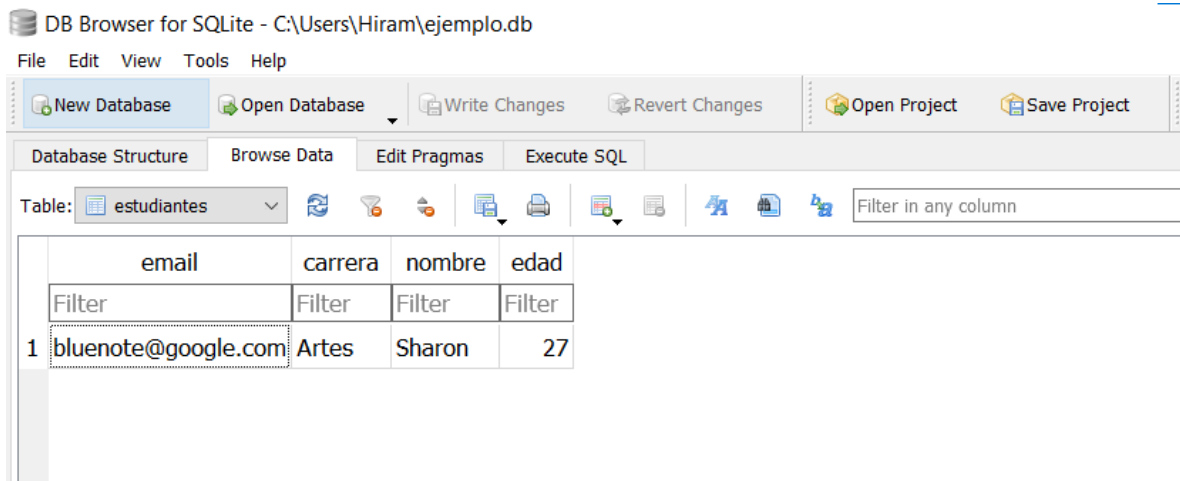
In [8]: conexion = sqlite3.connect("ejemplo.db")

In [9]: conexion = sqlite3.connect("ejemplo.db")
#Creamos el cursor
cursor = conexion.cursor()
conexion.close()

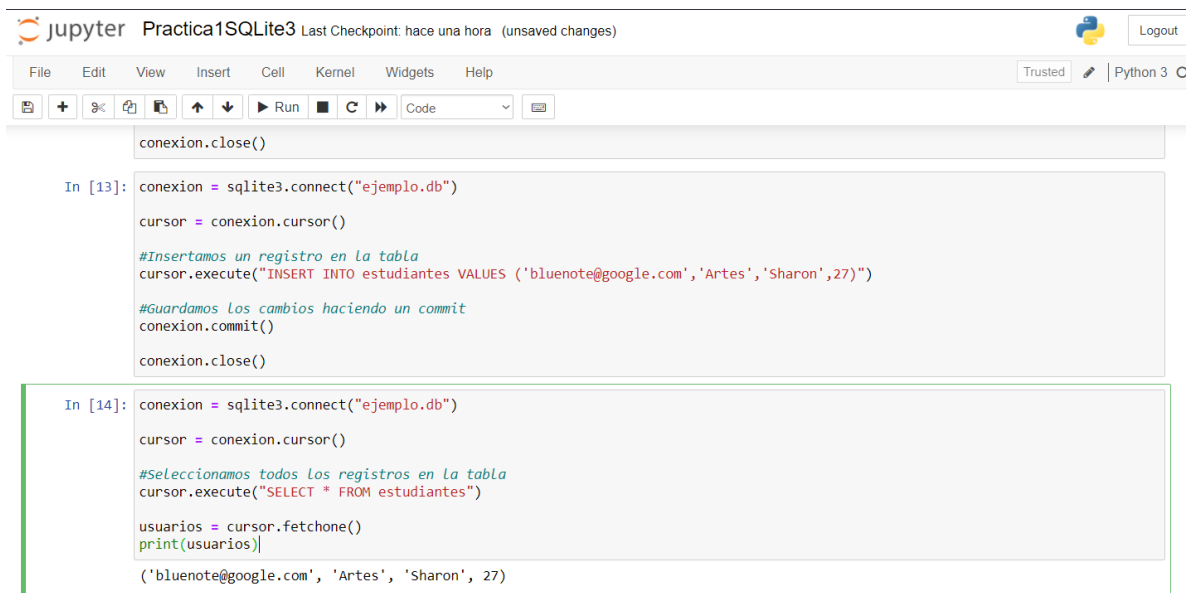
In [10]: conexion = sqlite3.connect("ejemplo.db")
cursor = conexion.cursor()
#Creamos una tabla llamada Estudiantes
cursor.execute("CREATE TABLE estudiantes (email VARCHAR(100), carrera VARCHAR(100), nombre VARCHAR(100), edad INTEGER)")
conexion.close()

In [13]: 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 un commit
conexion.commit()
conexion.close()
```

Al actualizar ya se visualizan datos en la base de datos.



Para leer el contenido de la tabla con SELECT \* FROM.



Ingresar varios datos y leerlos al mismo tiempo.

```
jupyter Practica1SQLite3 Last Checkpoint: hace una hora (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

cursor = conexion.cursor()

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

usuarios = cursor.fetchall()
print(usuarios)

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

In [15]: conexion = sqlite3.connect("ejemplo.db")
cursor = conexion.cursor()

usuarios = [
    ('abigahimorales@gmail.com', 'Ingenieria', 'Abigahi', 22),
    ('hirammetal14658@gmail.com', 'Ingenieria', 'Hiram', 24)
]

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

conexion.commit()

conexion.close()
```

Al ejecutar el comando.

DB Browser for SQLite - C:\Users\Hiram\ejemplo.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project

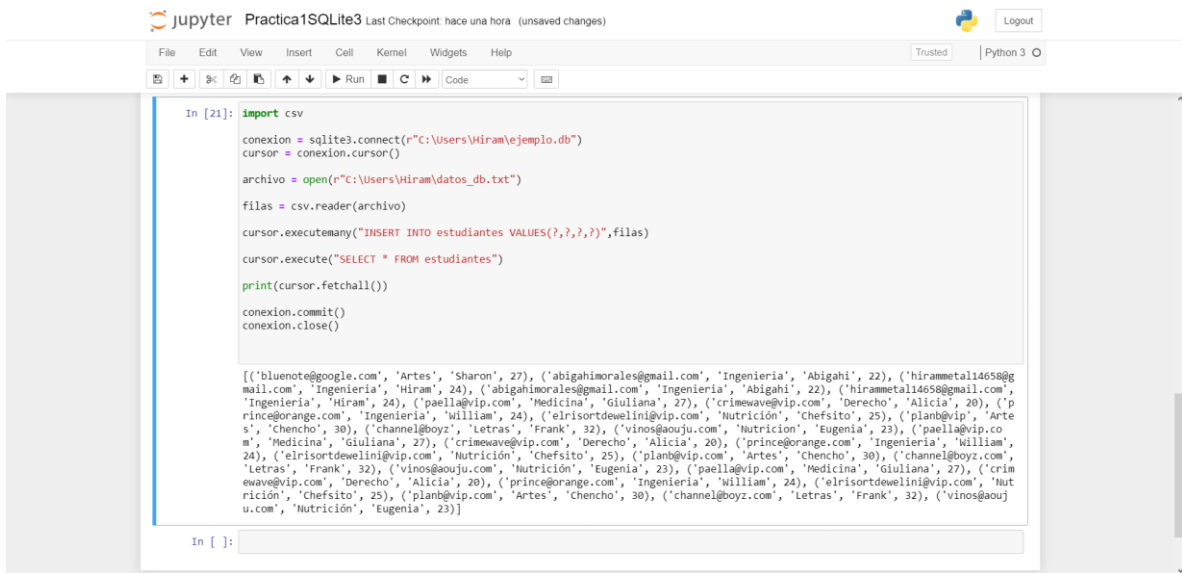
Database Structure Browse Data Edit Pragmas Execute SQL

Table: estudiantes

	email	carrera	nombre	edad
	Filter	Filter	Filter	Filter
1	bluenote@google.com	Artes	Sharon	27
2	abigahimorales@g...	Ingeni...	Abigahi	22
3	hirammetal14658@...	Ingeni...	Hiram	24



Para leer un CSV e insertarlo en una base de datos.



The image shows a Jupyter Notebook window titled "Practica1SQLite3 Last Checkpoint: hace una hora (unsaved changes)". The code in the cell is as follows:

```
In [21]: import csv

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

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

filas = csv.reader(archivo)

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

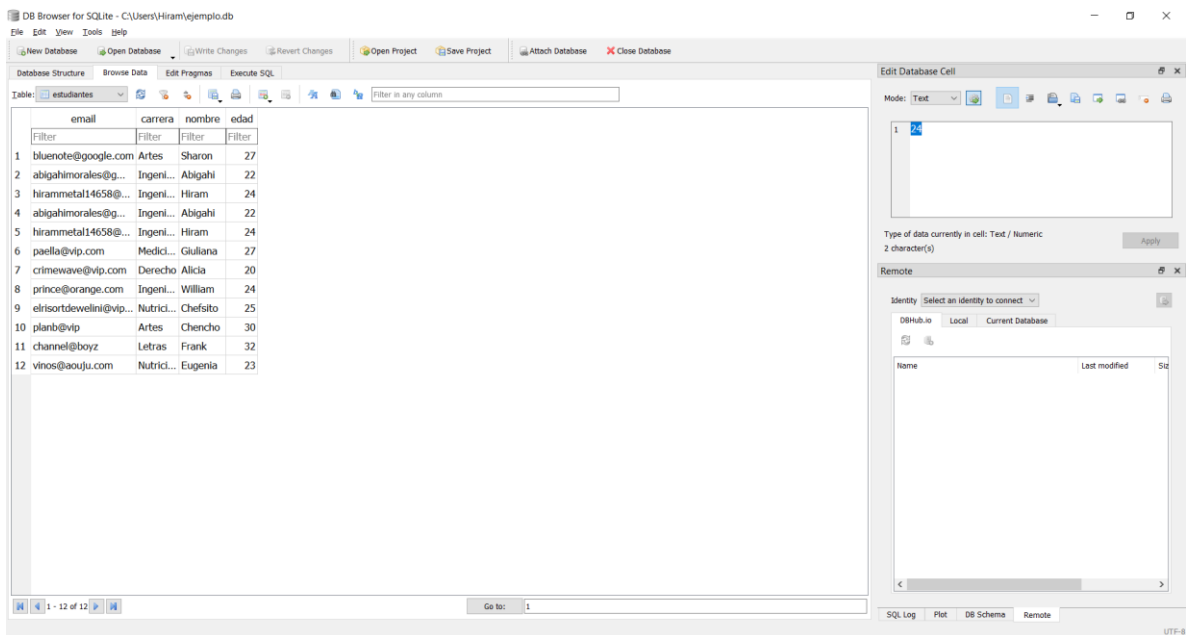
cursor.execute("SELECT * FROM estudiantes")

print(cursor.fetchall())

conexion.commit()
conexion.close()
```

The output of the code is a list of 12 tuples, each representing a student record with email, career, name, and age:

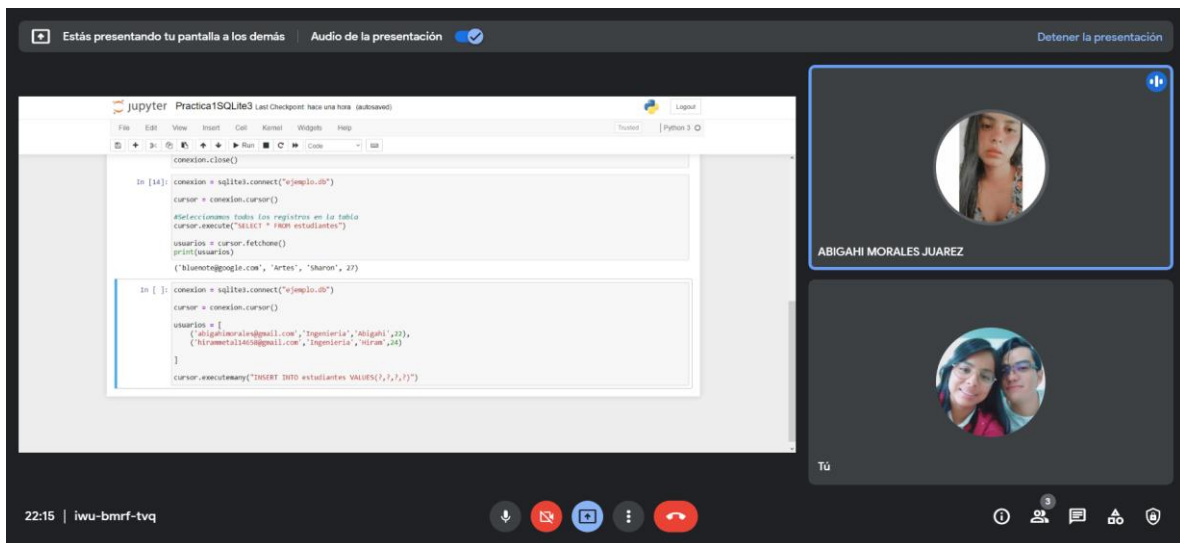
```
[('bluenote@google.com', 'Artes', 'Sharon', 27), ('abigahimorales@gmail.com', 'Ingenieria', 'Abigahi', 22), ('hirammetal14658@gmail.com', 'Ingenieria', 'Hiram', 24), ('paella@vip.com', 'Medicina', 'Giuliana', 27), ('crimewave@vip.com', 'Derecho', 'Alicia', 20), ('prince@orange.com', 'Ingenieria', 'William', 24), ('elrisortdewelini@vip.com', 'Nutrición', 'Chefsito', 25), ('planb@vip.com', 'Artes', 'Chencho', 30), ('channelboyz', 'Letras', 'Frank', 32), ('vinos@ouju.com', 'Nutrición', 'Eugenia', 23), ('crimewave@vip.com', 'Derecho', 'Alicia', 20), ('prince@orange.com', 'Ingenieria', 'William', 24), ('elrisortdewelini@vip.com', 'Nutrición', 'Chefsito', 25), ('planb@vip.com', 'Artes', 'Chencho', 30), ('channelboyz.com', 'Letras', 'Frank', 32), ('vinos@ouju.com', 'Nutrición', 'Eugenia', 23)]
```



The image shows the DB Browser for SQLite application. The main window displays the 'estudiantes' table with the following data:

email	carrera	nombre	edad
bluenote@google.com	Artes	Sharon	27
abigahimorales@gmail.com	Ingenieria	Abigahi	22
hirammetal14658@gmail.com	Ingenieria	Hiram	24
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
channelboyz	Letras	Frank	32
vinos@ouju.com	Nutrición	Eugenia	23

## EVIDENCIA DE EQUIPO



## CONCLUSIÓN

Gracias a que SQLite es software libre, es posible encontrar una gran cantidad de componentes, librerías y drivers para interactuar con SQLite desde una gran diversidad de lenguajes y plataformas de programación. Ya sea que estemos utilizando lenguajes modernos como Java, Perl, Python, PHP, Ruby, C#, lenguajes más antiguos como Pascal, SmallTalk, Clipper, o lenguajes poco conocidos como Suneido, REXX, S-Lang, para todos podemos encontrar librerías y ejemplos de código para SQLite.

## BIBLIOGRAFÍA

- Ebrahim, M. (24 de 01 de 2019). <https://likegeeks.com>. Recuperado el 22 de 11 de 2021, de <https://likegeeks.com/es/tutorial-de-python-sqlite3/>
- Rómmel, F. (s.f.). [sg.com.mx](https://sg.com.mx). Recuperado el 22 de 11 de 2021, de <https://sg.com.mx/revista/17/sqlite-la-base-datos-embebida>