

Ejecutamos estas instrucciones por medio del método ***commit()*** y cerramos la conexión con la base de datos mediante el método ***close()***. Debemos asegurarnos de ejecutar ***commit()*** o los comandos no se ejecutarán y los datos se perderán.

```
16     con.commit()
17     con.close()
```

Estos datos son persistentes y se pueden volver a cargar posteriormente.

ATENCIÓN!!!!: No debes llamar a tus archivos .py con el mismo nombre de las librerías que estás probando. Python intentará encontrar los métodos invocados en ese archivo y generará un error. **Esto es válido para cualquier import con el que estemos trabajando.**

..2) Ahorrando líneas

Si nos encontramos en la situación en la que ya existen datos para nuestra base de datos, tal vez necesitemos hacer una carga inicial.

Podemos agregar varios registros en la misma instrucción, cargándolos previamente a una lista e invocar el método **executemany()**, enviando una consulta paramétrica a la base de datos.

```
13     lista = [
14         ("2154489", "Sebastián", "Álvarez", 7.75),
15         ("2154563", "Analía", "Rovere", 8.75),
16         ("2154896", "Jaime", "Felice", 8),
17         ("2154492", "Rosa", "Medina", 9)
18     ]
19     cur.executemany('INSERT INTO alumnos VALUES (?, ?, ?, ?)', lista)
```

Debemos recordar comentar las líneas en las que agregamos datos una vez ejecutados, para que el programa no intente volver a agregarlos en la siguiente ejecución del programa.

Podrás conseguir el código completo de este ejemplo en el archivo ejemplo_sqlite3.py

..3) Herramientas adicionales (opcional)

DB Browser for SQLite: <https://sqlitebrowser.org/dl/>

Programa de interfaz gráfica sencillo y liviano para visualizar rápidamente una base de datos en SQLite.