

Clase N° 8.1

Python y Bases de Datos

© Lic. Ricardo Thompson

Bases de Datos

- Existen muchos tipos de bases de datos.
- Uno de los más utilizados es el modelo *relacional*, en el que los datos se almacenan en forma de *tablas*.
- Una tabla es una organización de datos en filas y columnas, similar a una planilla Excel.

© Lic. Ricardo Thompson

Integración con Python

- Python está preparado para interactuar con distintos administradores de bases de datos.
- Existen módulos para conectarse a bases MySQL, PostgreSQL, SQLite, etc.
- En esta clase utilizaremos SQLite3, una base de datos de dominio público liviana y robusta que no requiere instalación.

© Lic. Ricardo Thompson

Importante

Los temas relacionados con bases de datos analizados en esta clase se brindan a modo de complemento al programa de estudios de la materia, y por lo tanto no serán evaluados en ningún examen.

© Lic. Ricardo Thompson

Ejemplo 1

**Creación de una base de datos de
códigos postales a partir de un
archivo CSV.**

© Lic. Ricardo Thompson

Archivo CSV

```
1426;11 DE SEPTIEMBRE;201;1800;CAPITAL FEDERAL
1428;11 DE SEPTIEMBRE;1801;2800;CAPITAL FEDERAL
1429;11 DE SEPTIEMBRE;2801;4900;CAPITAL FEDERAL
1416;12 DE OCTUBRE;1601;1900;CAPITAL FEDERAL
1427;14 DE JULIO;1;1500;CAPITAL FEDERAL
1430;14 DE JULIO;1501;1700;CAPITAL FEDERAL
1130;15 DE NOVIEMBRE DE 1889;1101;1800;CAPITAL FEDERAL
1261;15 DE NOVIEMBRE DE 1889;1801;2700;CAPITAL FEDERAL
1437;1RO. DE MAYO;1801;2000;CAPITAL FEDERAL
1439;2 DE ABRIL DE 1982;6401;7000;CAPITAL FEDERAL
1431;20 DE FEBRERO;5601;6100;CAPITAL FEDERAL
1761;20 DE JUNIO;;;BUENOS AIRES
[ . . . ]
```

© Lic. Ricardo Thompson

```
import sqlite3
```

```
try:
```

```
    # Creamos la conexión con la base de  
    # datos, o la base de datos si no existe.
```

```
    bd = sqlite3.connect("codpos.db")
```

```
    # bd = sqlite3.connect(":memory:")
```

```
    # Creamos un cursor para manejarla
```

```
    cursor = bd.cursor( )
```

```
    print("Base de datos abierta")
```

© Lic. Ricardo Thompson

```
# Comenzamos a enviar comandos a la base de datos
```

```
cursor.execute("DROP TABLE IF EXISTS codigos")
```

```
cursor.execute("""CREATE TABLE codigos  
                (id INT PRIMARY KEY NOT NULL,  
                 codigo INT NOT NULL,  
                 calle TEXT NOT NULL,  
                 desde TEXT,  
                 hasta TEXT,  
                 provincia TEXT)""")
```

```
print("Tabla creada")
```

```
print("Insertando registros desde archivo CSV...")
```

```
arch=open("codpos.csv","rt")
```

© Lic. Ricardo Thompson

```
k = 1
for linea in arch:
    cod, calle, desde, hasta, prov = linea.rstrip('\n').split(';')
    comando = "INSERT INTO codigos (id, codigo, calle, desde, hasta, provincia) \
        VALUES ('+str(k)+' , '"+cod+"', '"+calle+"', '"+desde+"', '"+hasta+"', '"+prov+"')\"
    cursor.execute(comando)
    k = k + 1
bd.commit( )
print(k, "registros insertados")
```

© Lic. Ricardo Thompson

```
except IOError:
    print("No se puede leer el archivo.")
except sqlite3.OperationalError as error:
    print("ERROR EN LA BASE DE DATOS:", error)
finally:
    try:
        bd.close( )
        arch.close( )
    except NameError:
        pass
```

© Lic. Ricardo Thompson

Ejemplo 2

Desarrollar una aplicación que permita consultar la base de datos de códigos postales.

© Lic. Ricardo Thompson

```
import sqlite3

def menu( ):
    while True:
        try:
            print( )
            print("=" * 55)
            print("Consulta de Códigos Postales\n".center(55))
            print("1. Listado de Códigos Postales")
            print("2. Búsqueda por calle o localidad")
            print("3. Búsqueda por provincia")
            print("4. Salir")
            print("=" * 55)
            opc = int(input("Ingrese opción: "))
            assert 1<=opc<=4
            break
        except (ValueError, AssertionError):
            print("*** Debe ingresar un número del 1 al 4 ***")
    return opc
```

© Lic. Ricardo Thompson

A whiteboard with a black frame is mounted on a wooden desk. The whiteboard contains Python code. To the left of the whiteboard are a red book, a blue book, and a brown geometric object. To the right is a glass jar with blue liquid and a wooden stick. A pencil and an eraser are on the desk in front of the whiteboard. The background is a light gray wall.

```
try:
```

```
# Creamos la conexión con la base de datos
```

```
bd = sqlite3.connect("codpos.db")
```

```
# Creamos un cursor para manejarla
```

```
cursor = bd.cursor( )
```

```
print("Base de datos abierta")
```

```
opcion = 0
```

© Lic. Ricardo Thompson

```
while opcion!=4:
```

```
    opcion = menu( )
```

```
    comando = "SELECT * FROM codigos "
```

```
    if opcion==2: # 2: Búsqueda por calle o localidad
```

```
        dato = input("Calle o localidad a buscar? ")
```

```
        comando = comando + "WHERE calle LIKE '%" + dato + "%'"
```

```
    elif opcion==3: # 3: Búsqueda por provincia
```

```
        dato = input("Provincia a buscar? ")
```

```
        comando = comando + "WHERE provincia LIKE '%" + dato + "%'"
```

```
    if opcion<4 : # 4: Salir
```

```
        cursor.execute(comando)
```

```
        k = 0
```

```
        for registro in cursor:
```

```
            print(registro[1], registro[2], registro[3], registro[4], registro[5])
```

```
            k = k + 1
```

```
        print("\nTotal:", k, "registros")
```

© Lic. Ricardo Thompson

```
except sqlite3.OperationalError as error:
    print("ERROR EN LA BASE DE DATOS:", error)
finally:
    try:
        bd.close( )
    except NameError:
        pass
```

© Lic. Ricardo Thompson

Importante

Los dos ejemplos y el archivo CSV para crear la base de datos están disponibles en Webcampus para su descarga.

© Lic. Ricardo Thompson

Ejercitación

Agregar opciones de menú para:

- 1. Listado de Códigos Postales (Sólo CABA)**
- 2. Listado de Códigos Postales (Sólo Interior y GBA)**

(Buscar información sobre la cláusula WHERE)

© Lic. Ricardo Thompson

Ejercitación

- 3. Exportar toda la base de datos a un archivo de texto con campos de longitud fija:**

- **Código Postal: 4**
- **Calle o Localidad: 60**
- **Altura inicial: 5**
- **Altura final: 5**
- **Provincia: 20**

(Agregar un espacio extra entre cada campo)

© Lic. Ricardo Thompson

Ejercitación

Modelo del archivo con campos de longitud fija

1426 11 DE SEPTIEMBRE
1428 11 DE SEPTIEMBRE
1429 11 DE SEPTIEMBRE
1416 12 DE OCTUBRE
1427 14 DE JULIO
1430 14 DE JULIO
1130 15 DE NOVIEMBRE DE 1889
1261 15 DE NOVIEMBRE DE 1889
1437 1RO. DE MAYO
1439 2 DE ABRIL DE 1982
1431 20 DE FEBRERO
1761 20 DE JUNIO
3158 20 DE SEPTIEMBRE
1155 20 DE SEPTIEMBRE
4504 23 DE AGOSTO
1170 24 DE NOVIEMBRE

201 1800 CAPITAL FEDERAL
1801 2800 CAPITAL FEDERAL
2801 4900 CAPITAL FEDERAL
1601 1900 CAPITAL FEDERAL
1 1500 CAPITAL FEDERAL
1501 1700 CAPITAL FEDERAL
1101 1800 CAPITAL FEDERAL
1801 2700 CAPITAL FEDERAL
1801 2000 CAPITAL FEDERAL
6401 7000 CAPITAL FEDERAL
5601 6100 CAPITAL FEDERAL
BUENOS AIRES
ENTRE RIOS
1 400 CAPITAL FEDERAL
JUJUY
1 500 CAPITAL FEDERAL

© Lic. Ricardo Thompson

Ejercitación

4. Campos calculados: Añadir una columna LONGITUD en los listados, que muestre la cantidad de cuabras que tiene cada calle de CABA. Se calcula por diferencia de alturas.

© Lic. Ricardo Thompson

Ejercitación

**Tú que estás
leyendo ésto.
Ponte a trabajar.**

© Lic. Ricardo Thompson