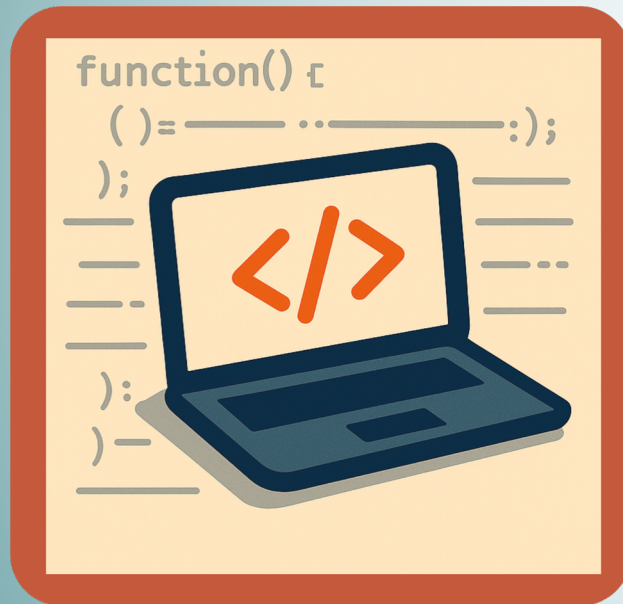


Programación

TEMA 10. BASES DE DATOS



Javier González Villa

David Lázaró Urrutia

DEPARTAMENTO DE MATEMÁTICA APLICADA
Y CIENCIAS DE LA COMPUTACIÓN

Este material se publica bajo la siguiente licencia:

[Creative Commons BY-NC-SA 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/)



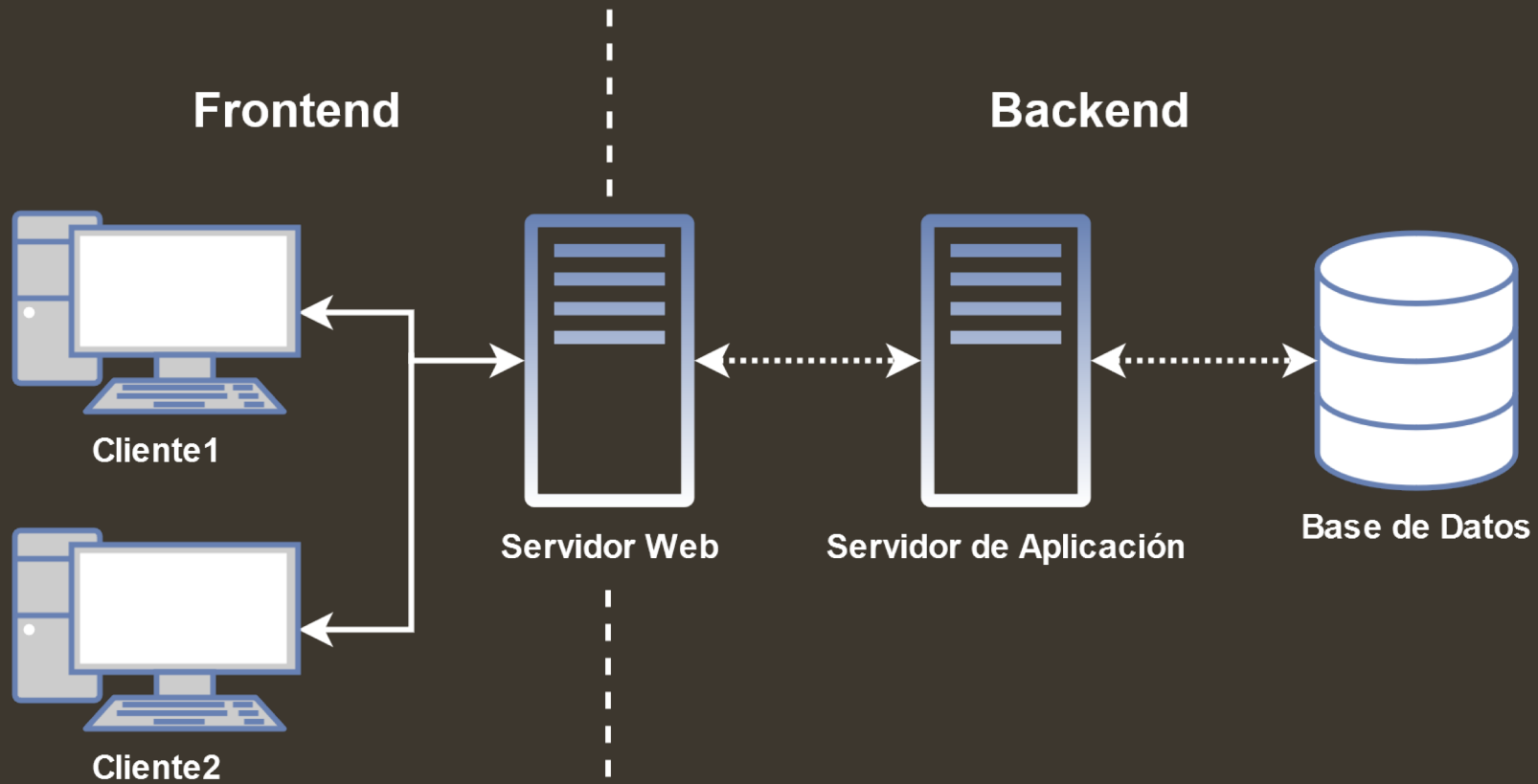
Contenidos

1. Bases de Datos
 1. Creación de bases de datos
 2. Ejecución de instrucciones SQL
 3. Repertorio de instrucciones SQL básico

1. Bases de Datos

- Método de **almacenamiento** que se encarga de **conectar datos entre sí** al pertenecer éstos a un **mismo contexto** y estructurarlos de diversas formas para **facilitar su posterior uso**.
- Permite almacenar una **gran cantidad de información** y **acceder a ella de manera rápida, fácil y estructurada** al disponer de sistemas de indexación.
- Existen multitud de bases de datos de diversos tipos y muchas de ellas son **abiertas** permitiendo consultar, modificar, actualizar o borrar datos. Otras sin embargo pueden estar **protegidas** al contener datos históricos o restringidos.
- Existen **muchos tipos y estructuras de bases de datos** (jerárquicas, en red, transaccionales, relacionales, orientadas a objetos, documentales, etc.).

1. Bases de Datos



1. Bases de Datos

- Una de las librerías para la creación y manejo de bases de datos en Python es **SQLITE3** que permite realizar operaciones propias de manejo de datos mediante instrucciones **SQL en Python**.

<https://docs.python.org/3/library/sqlite3.html>

<https://www.w3schools.com/sql/>

- **SQL (Structured Query Language)** es un tipo de lenguaje específico para la manipulación de bases de datos.
- Por lo tanto en nuestro manejo de bases de datos se **intercalarán instrucciones propias de la librería junto con instrucciones SQL** para de manera combinada realizar operaciones sobre un motor de bases de datos.

1.1. Creación de bases de datos

- Para crear una base de datos se requiere realizar los siguientes pasos:

1. **Importar** la librería **sqlite3**:

```
import sqlite3
```

2. **Conectar** con la base de datos (**si existe se conecta sino la crea**):

```
conexion = sqlite3.connect("NombreBaseDeDatos.db")
```

3. **Confirmar** la conexión:

```
conexion.commit()
```

4. **Cerrar** la conexión:

```
conexion.close()
```

```
[1]: import sqlite3
      conexion=sqlite3.connect("BaseDeDatos.db")
      conexion.commit()
      conexion.close()
```

1.2. Ejecución de instrucciones SQL

- Para insertar datos en una base de datos se requiere realizar los siguientes pasos:

1. **Conectar** con la base de datos (si existe se conecta sino la crea):

```
conexion = sqlite3.connect( "NombreBaseDeDatos.db" )
```

2. **Ejecutar** la **instrucción SQL** que se desee:

```
conexion.execute( "Instrucción SQL" )
```

3. **Confirmar** la conexión:

```
conexion.commit()
```

4. **Cerrar** la conexión:

```
conexion.close()
```

```
[2]: import sqlite3
      conexion=sqlite3.connect("BaseDeDatos.db")
      try:
          conexion.execute("""CREATE TABLE Tabla1 (
                                Id integer primary key autoincrement,
                                Campo1 text
                                )""")
          print("Se creó la table Terminal correctamente.")
      except sqlite3.OperationalError:
          print("La tabla Tabla1 ya existe")
      conexion.commit()
      conexion.close()
```

Se creó la table Terminal correctamente.

1.3. Repertorio de instrucciones SQL básico

- Crear tabla:

```
"CREATE TABLE Nombre_Tabla (Id integer primary key autoincrement, Campo1 text, ...)"
```

```
"""CREATE TABLE Nombre_Tabla (  
    Id integer primary key autoincrement,  
    Campo1 text,  
    Campo2 integer,  
    Campo3 integer,  
    FOREIGN KEY (Campo2) REFERENCES Tabla2(Campo1),  
    FOREIGN KEY (Campo3) REFERENCES Tabla3(Campo5)"""
```

- Borrar tabla:

```
"DROP TABLE Nombre_Tabla"
```

- Insertar valor en tabla:

```
conexion.execute("INSERT INTO Tabla(Campo1,Campo2,Campo3) values (?, ?, ?)", (valor1, valor2, valor3))
```

1.3. Repertorio de instrucciones SQL básico

- Consultar valores insertados previamente:

- Todos los valores de una tabla:

```
cursor=conexion.execute("SELECT * FROM Tabla")
for fila in cursor:
    print(fila)
```

- Todos los valores de una tabla que cumplan una condición:

```
cursor=conexion.execute("SELECT * FROM Tabla WHERE Campo is not null")
for fila in cursor:
    print(fila)
```

```
cursor=conexion.execute("SELECT * FROM Tabla WHERE Campo = 3")
for fila in cursor:
    print(fila)
```

- Algunos campos de una tabla:

```
cursor=conexion.execute("SELECT Campo1, Campo3, Campo8 FROM Tabla")
for fila in cursor:
    print(fila)
```

- Algunos campos de una tabla que cumplan una condición de una relación de otra tabla:

```
cursor=conexion.execute("""SELECT Tabla1.Campo1, Tabla2.Campo5, Tabla1.Campo3 FROM Tabla1
                           INNER JOIN Tabla2 ON Tabla1.Campo2=Tabla2.Campo3""")
for fila in cursor:
    print(fila)
```