

# PYTHON Y MYSQL

MAX RODRIGO DURÁN CANTEO  
TALLER #1



# ¿Qué es MySQL?

MySQL es un sistema de gestión de bases de datos relacional (RDBMS) de código abierto.



# Características

- 1 MySQL es conocido por su alta disponibilidad
- 2 Permite gestionar grandes volúmenes de datos
- 3 Ofrece un rendimiento rápido y eficiente
- 4 Su sintaxis es fácil de aprender y utilizar
- 5 Proporciona mecanismos robustos para proteger los datos almacenados



# ¿Qué es Python?

Python es un lenguaje de programación de alto nivel, interpretado y de propósito general.

Es conocido por su sintaxis clara y legible, lo que facilita el desarrollo de aplicaciones.

# Integración de MySQL y Python

Para utilizar MySQL con Python, vamos a trabajar con las siguientes librerías:

- flask
- Flask-MySQLdb

Para instalar las librerías lo haremos de la siguiente manera en la consola de nuestro proyecto:

```
pip install flask Flask-MySQLdb
```

# Integración de MySQL y Python

Crearemos una aplicación web con el framework Flask en python para trabajar con MySQL:

- Importamos las librerías que vamos a usar:
  - from flask import Flask, jsonify
  - from flask\_mysqlldb import MySQL
- Creamos la conexión entre MySQL y Flask:
  - Definimos credenciales:

```
class Configuracion():  
    DEBUG = True  
    MYSQL_HOST = 'localhost'  
    MYSQL_USER = 'root'  
    MYSQL_PASSWORD = 'root'  
    MYSQL_DB = 'voluntariado'  
  
configuracion = {  
    'configuracion': Configuracion  
}
```

# Integración de MySQL y Python

- Instanciamos la aplicación y la conexión:

```
app = Flask(__name__)  
conexion = MySQL(app)
```

- Definimos como empezara la aplicación y otorgamos las credenciales de acceso para la conexión:

```
if __name__ == '__main__':  
    app.config.from_object(configuracion['configuracion'])  
    app.run()
```

# Integración de MySQL y Python

- Creamos una ruta para hacer peticiones a nuestra aplicación:

```
@app.route('/<string:peticion>')
def consulta(peticion):
    try:
        cursor = conexion.connection.cursor()
        cursor.execute(diccionario[peticion])
        datos = cursor.fetchall()
        cursor.close()
        conexion.connection.commit()
        return jsonify({'Respuesta': datos})
    except Exception as e:
        return jsonify({'Respuesta': f'Error -> {e}'})
```

- Donde 'cursor' será la conexión con MySQL
- cursor.execute() servirá para ejecutar la consulta SQL
- datos servirá para traer los resultados de la consulta
- conexion.connection.commit() será para confirmar la petición y ejecutarla



# Integración de MySQL y Python

- La variable diccionario es un diccionario donde guardaremos las consultas o scripts de SQL :

```
class Diccionario():  
  
    consulta3 = '''  
        delete from clientes;  
        delete from paises;  
        ...  
    '''  
  
    consulta4 = '''  
        drop table clientes;  
        drop table paises;  
        ...  
    '''  
  
    consulta5 = '''  
        select * from clientes;  
        ...  
    '''  
  
    consulta6 = '''  
        select * from paises;  
        ...  
    '''  
  
    consulta7 = '''  
        select nombre from paises;  
        ...  
    '''  
  
    consulta8 = '''  
        select nombre, apellido, edad from clientes;  
        ...  
    '''  
  
    diccionario = {  
        'consulta3' : Diccionario.consulta3,  
        'consulta4' : Diccionario.consulta4,  
        'consulta5' : Diccionario.consulta5,  
        'consulta6' : Diccionario.consulta6,  
        'consulta7' : Diccionario.consulta7,  
        'consulta8' : Diccionario.consulta8,  
    }
```