

API REST



Flask

web development,
one drop at a time

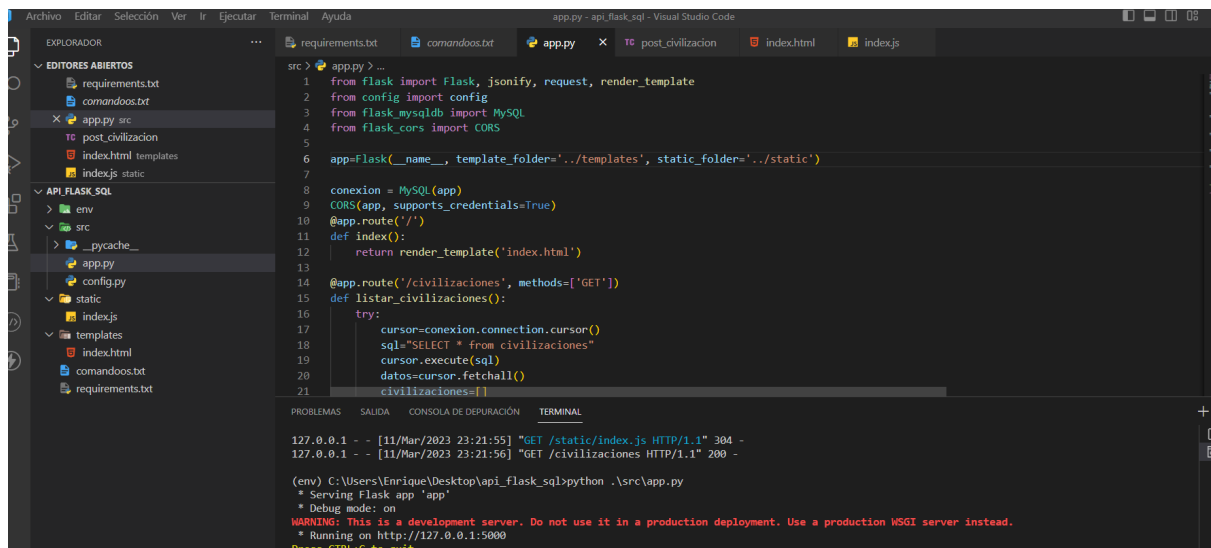
Enrique García Herrador

ÍNDICE

1. TECNOLOGÍAS Y RECURSOS
2. INTRODUCCIÓN
3. MÉTODOS
 - a. GET
 - b. POST
 - c. UPDATE
 - d. DELETE
4. TEMPLATE EJEMPLO USO
5. COMENTARIOS FINALES

1 TECNOLOGÍAS

Las tecnologías utilizadas han sido Python como lenguaje de programación de la aplicación a través de flask y su entorno virtual y en el caso de la base de datos se ha utilizado MySql a través de Xampp y PHPMyAdmin. Como IDE hemos utilizado VS y para realizar las diferentes peticiones HTTP hemos utilizado un plugin del mismo ThunderClient.



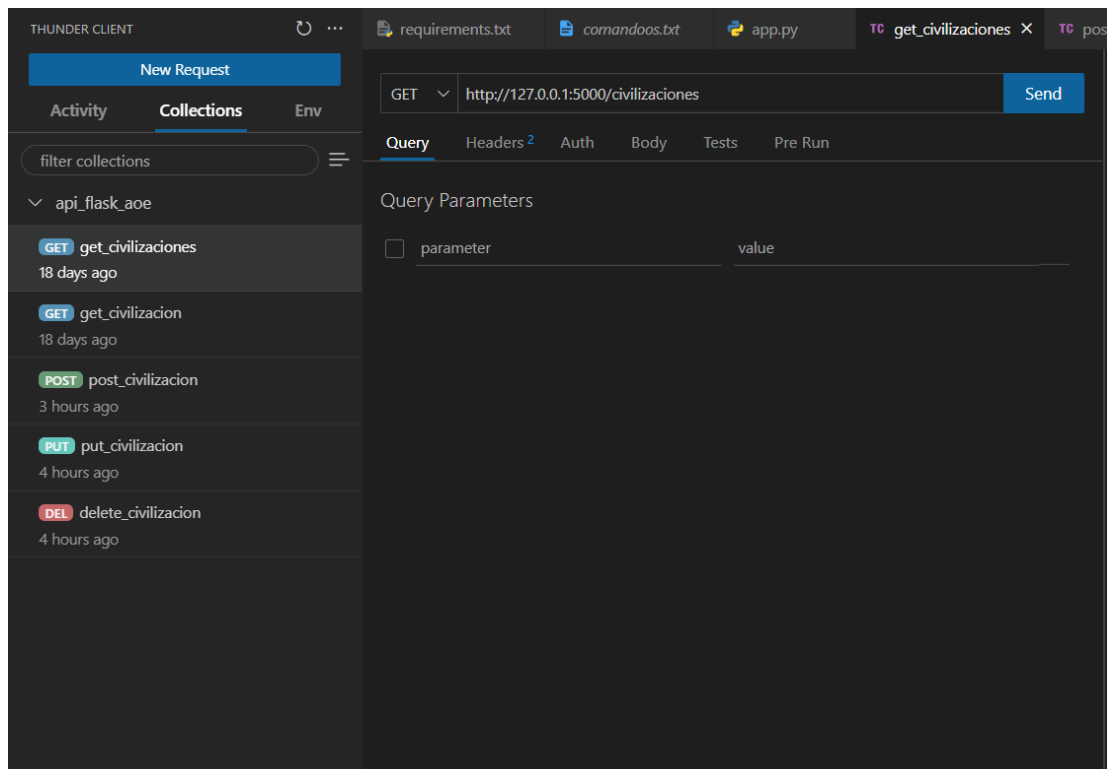
```
src > app.py -
1 from flask import Flask, jsonify, request, render_template
2 from config import config
3 from flask_mysqldb import MySQL
4 from flask_cors import CORS
5
6 app=flask(__name__, template_folder='../templates', static_folder='../static')
7
8 conexion = MySQL(app)
9 CORS(app, supports_credentials=True)
10 @app.route('/')
11 def index():
12     return render_template('index.html')
13
14 @app.route('/civilizaciones', methods=['GET'])
15 def listar_civilizaciones():
16     try:
17         cursor=conexion.connection.cursor()
18         sql="SELECT * from civilizaciones"
19         cursor.execute(sql)
20         datos=cursor.fetchall()
21         civilizaciones=[]
```

```
127.0.0.1 - - [11/Mar/2023 23:21:55] "GET /static/index.js HTTP/1.1" 304 -
127.0.0.1 - - [11/Mar/2023 23:21:56] "GET /civilizaciones HTTP/1.1" 200 -

(env) C:\Users\Enrique\Desktop\api_flask_sql>python .\src\app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on https://127.0.0.1:5000
Press CTRL+C to quit
```



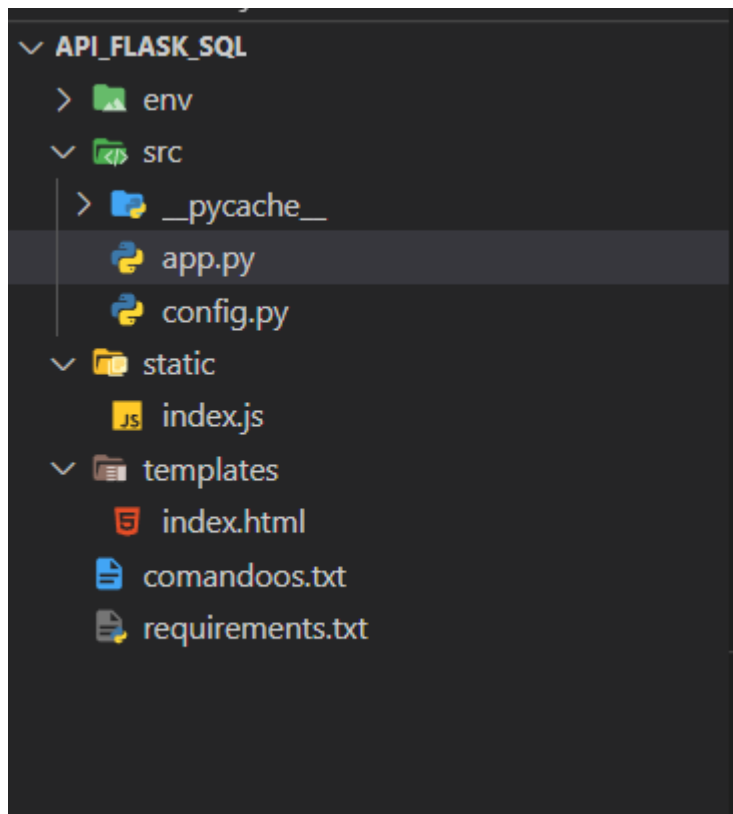
#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
<input type="checkbox"/>	1 NOMBRE	varchar(30)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	2 ESPECIALIDAD	varchar(30)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	3 UNIDAD_UNICA	varchar(30)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	4 TECNOLOGIA_UNICA	varchar(30)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	5 IMAGEN	varchar(1000)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más



2 INTRODUCCIÓN

Se trata de una API que es capaz de devolver, insertar, modificar y eliminar datos trabajando con el formato de datos JSON y en este caso se ha hecho temática sobre las civilizaciones del Age Of Empires 2 DE. A continuación iremos viendo los diferentes métodos y sus funcionalidades.

El proyecto se ha organizado en las carpetas “src”, “templates” y “static”. En el directorio “src” están situados los archivos app.py y config.py, este último alberga las configuraciones correspondientes a la conexión a base de datos e inicio de la app. A parte tenemos el fichero requirements.txt para realizar las instalaciones necesarias del proyecto.



3 MÉTODOS

Nuestra API cuenta con varios métodos para interactuar con la base de datos.

a. GET

La API tiene dos métodos, el primero devuelve todas las civilizaciones que existen en base de datos en formato JSON y el segundo devuelve la civilización que se recibe por parámetro.

1

```

@app.route('/civilizaciones', methods=['GET'])
def listar_civilizaciones():
    try:
        cursor=conexion.connection.cursor()
        sql="SELECT * from civilizaciones"
        cursor.execute(sql)
        datos=cursor.fetchall()
        civilizaciones=[]
        for fila in datos:
            civilizacion={'nombre':fila[0],'especialidad':fila[1],'unidad_unica':fila[2],'tecnologia_unica':fila[3],'imagen':fila[4]}
            civilizaciones.append(civilizacion)
        return jsonify({'civilizaciones':civilizaciones})

    except Exception as ex:
        return "Error"

```

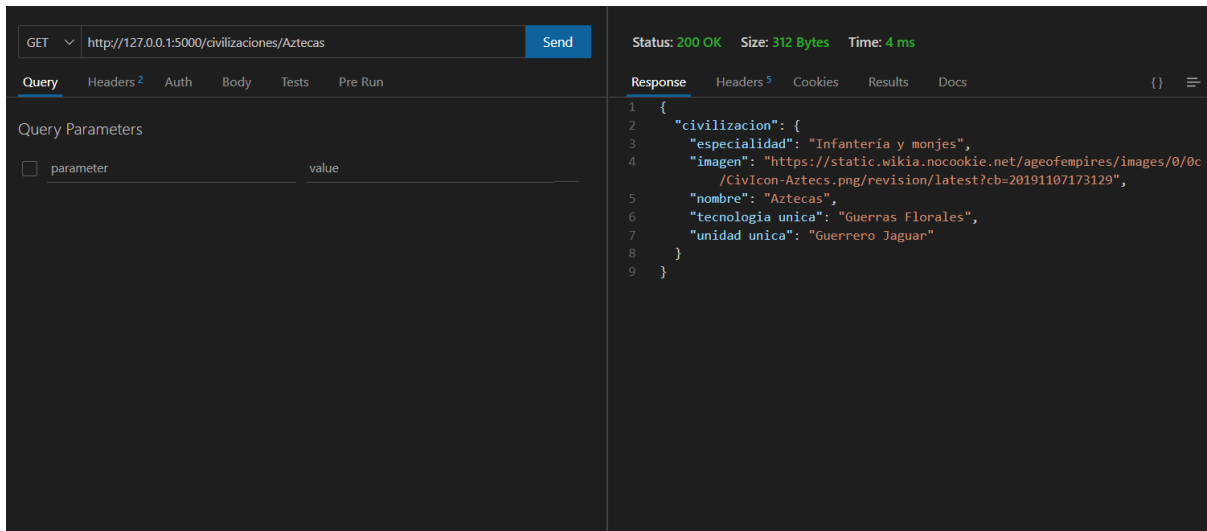
Query	Headers	Auth	Body	Tests	Pre Run	Status: 200 OK	Size: 1.22 KB	Time: 7 ms				
<p>GET <input type="text" value="http://127.0.0.1:5000/civilizaciones"/> <input type="button" value="Send"/></p> <p>Query Parameters</p> <table border="1"> <thead> <tr> <th>parameter</th> <th>value</th> </tr> </thead> <tbody> <tr> <td><input type="text"/></td> <td><input type="text"/></td> </tr> </tbody> </table>						parameter	value	<input type="text"/>	<input type="text"/>	<p>Response</p> <pre> 1 { 2 "civilizaciones": [3 { 4 "especialidad": "Infantería y monjes", 5 "imagen": "https://static.wikia.nocookie.net/ageofempires/images/ 6 /0c/CivIcon-Aztecs.png/revision/latest?cb=20191107173129", 7 "nombre": "Aztecas", 8 "tecnologia_unica": "Guerras Florales", 9 "unidad_unica": "Guerrero Jaguar" 10 }, 11 { 12 "especialidad": "Elefantes y Naval", 13 "imagen": "https://static.wikia.nocookie.net/ageofempires/images/ 14 /ce/Civ-Icon_Bengalies.png/revision/latest?cb 15 =20220429115713&path-prefix=es", 16 "nombre": "Bengalies", 17 "tecnologia_unica": "Mahayana", 18 "unidad_unica": "Ratha" 19 }, 20 { 21 "especialidad": "Camellos y harcos", 22 "imagen": "https://static.wikia.nocookie.net/ageofempires/images/ 23 /2c/CivIcon-Berberes.png/revision/latest?cb 24 =20191118225639&path-prefix=es", 25 "nombre": "Bereberes", 26 "tecnologia_unica": "Alcazabas", 27 "unidad_unica": "Arquero a camello" 28 }, 29 { 30 "especialidad": "Arqueros", 31 "imagen": "https://static.wikia.nocookie.net/ageofempires/images/ 32 /0e/CivIcon-Mayas.png/revision/latest?cb=20191118225430&path 33 -prefix=es", 34 "nombre": "Mayas", </pre>		
parameter	value											
<input type="text"/>	<input type="text"/>											

2

```

@app.route('/civilizaciones/<civilizacion>')
def listar_civilizacion(civilizacion):
    try:
        cursor=conexion.connection.cursor()
        sql="SELECT * FROM civilizaciones WHERE nombre = '{0}'".format(civilizacion)
        cursor.execute(sql)
        datos=cursor.fetchone()
        if datos != None:
            civilizacion={'nombre':datos[0],'especialidad':datos[1],'unidad_unica':datos[2],'tecnologia_unica':datos[3],'imagen':datos[4]}
            return jsonify({'civilizacion':civilizacion})
    except Exception as ex:
        return "Error"

```



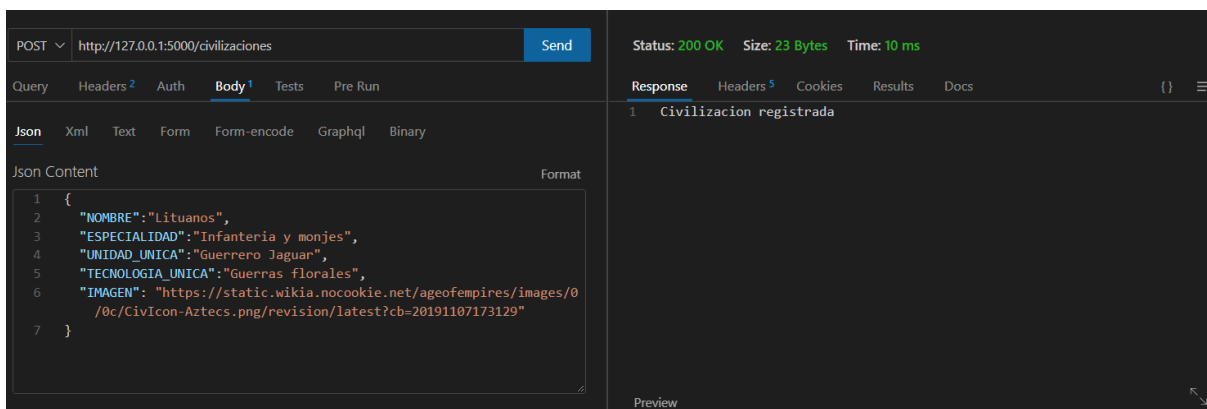
b. POST

Recibe por parámetros los datos en formato JSON y hace una inserción en base de datos.

```

@app.route('/civilizaciones', methods=['POST'])
def agregar_civilizacion():
    try:
        cursor=conexion.connection.cursor()
        sql="""INSERT INTO civilizaciones (NOMBRE, ESPECIALIDAD, UNIDAD_UNICA, TECNOLOGIA_UNICA, IMAGEN)
        VALUES ('{0}','{1}','{2}','{3}','{4}')""".format(request.json['NOMBRE'],request.json['ESPECIALIDAD'],request.json['UNIDAD_UNICA'],r
        cursor.execute(sql)
        conexion.connection.commit()
        return "Civilizacion registrada"
    except Exception as ex:
        return "Error"

```



☐ Lituanos Infanteria y monjes Guerrero Jaguar Guerras florales https://static.wikia.nocookie.net/ageofempires/ima...

c. PUT

Recibe los datos en formato JSON y en nombre qué es la PM en la tabla lo recibe por parámetro en la URL que es el parámetro que se utiliza para decidir la fila a modificar.

```
@app.route('/civilizaciones/<civilizacion>', methods=['PUT'])
def modificar_civilizacion(civilizacion):
    try:
        cursor=conexion.connection.cursor()
        sql="""UPDATE civilizaciones SET ESPECIALIDAD = '{0}', UNIDAD_UNICA = '{1}', TECNOLOGIA_UNICA = '{2}'
        WHERE nombre = '{3}'""".format(request.json['ESPECIALIDAD'], request.json['UNIDAD_UNICA'], request.json['TECNOLOGIA_UNICA'], civiliz
        cursor.execute(sql)
        conexion.connection.commit()
        return "Civilizacion Actualizada"

    except Exception as ex:
        return "Error"
```

PUT Send

Status: 200 OK Size: 24 Bytes Time: 12 ms

Response Headers Cookies Results Docs {}

1 Civilizacion Actualizada

Json Content Format

```
1 {
2   "ESPECIALIDAD": "Elefantes y Naval",
3   "UNIDAD_UNICA": "Ratha",
4   "TECNOLOGIA_UNICA": "Mahayana",
5   "IMAGEN": "https://static.wikia.nocookie.net/ageofempires/images/0
6   /0c/CivIcon-Aztecs.png/revision/latest?cb=20191107173129"
7 }
8
```

Preview

d. DELETE

El método delete elimina una fila recibiendo por parámetro el nombre de la fila a eliminar.

```
@app.route('/civilizaciones/<civilizacion>', methods=['DELETE'])
def eliminar_civilizacion(civilizacion):
    try:
        cursor=conexion.connection.cursor()
        sql="DELETE FROM civilizaciones WHERE nombre = '{0}'".format(civilizacion)
        cursor.execute(sql)
        conexion.connection.commit()
        return "Civilizacion eliminada"

    except Exception as ex:
        return "Error"
```

DELETE Send

Status: 200 OK Size: 22 Bytes Time: 8 ms

Response Headers Cookies Results Docs {}

1 Civilizacion eliminada

Query Parameters

parameter	value
-----------	-------

Preview









4 TEMPLATE

Se ha creado un template en el proyecto para que se pueda ver un ejemplo de uso de aplicación, en este se hace una llamada al método GET que devuelve todas las civilizaciones y se generan unas cartas con sus datos.



BIENVENIDO ESTOS ES UN EJEMPLO DE UTILIDAD DE LA API

MOSTRAR CARTAS

 <p>Aztecas Infantería y monjes, Guerrero Jaguar, Guerras Florales</p> <p>Go somewhere</p>	 <p>Bengalies Elefantes y Naval, Ratha, Mahayana</p> <p>Go somewhere</p>	 <p>Bereberes Camellos y barcos, Arquero a camello, Alcazabas</p> <p>Go somewhere</p>	 <p>Lituanos Infantería y monjes, Guerrero Jaguar, Guerras florales</p> <p>Go somewhere</p>
 <p>Mayas Arqueros, Arquero emplumado, El Dorado</p> <p>Go somewhere</p>	 <p>Aztecas Infantería y monjes, Guerrero Jaguar, Guerras Florales</p> <p>Go somewhere</p>	 <p>Bengalies Elefantes y Naval, Ratha, Mahayana</p> <p>Go somewhere</p>	 <p>Bereberes Camellos y barcos, Arquero a camello, Alcazabas</p> <p>Go somewhere</p>

5 COMENTARIOS FINALES

Se ha implementado también la captura del error 404 página no encontrada para mostrar un mensaje al usuario de la API.

```
def pagina_no_encontrada(error):  
    return """<h1>La página que has introducido no existe</h1>  
           <h1>:</h1>"""  
  
if __name__ == '__main__':  
    app.config.from_object(config['development'])  
    app.register_error_handler(404, pagina_no_encontrada)  
    app.run()
```

Se tenía previsto capturar más errores pero por motivos de tiempo no ha sido posible.