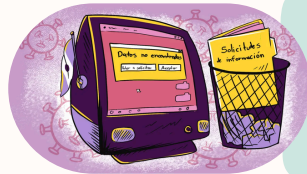


Cliente  
Envía una solicitud  
Servidor  
Recibe esa solicitud y te da una respuesta.



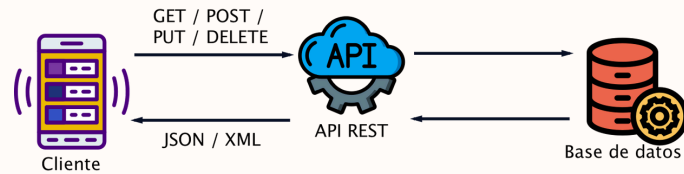
Sin estado  
El servidor es independiente ya que no recuerda la petición anteriores tiene que volver a solicitar la información

Recursos y URLs  
Acceder a lo que contiene.  
Ejemplo:  
`https://miapp.com/usuarios/123`  
Este recurso es el usuario con ID 123.



## ¿Cómo funciona?

## Arquitectura REST API



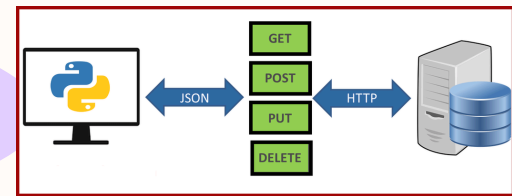
## Métodos HTTP

GET: **obtener** información

POST: **Crear** un nuevo recurso

DELETE: **Eliminar** un recurso

PUT: Actualizar  
Reemplazar  
Modificar  
Sobrescribir  
de los **recurso** existente



## Formato de datos (JSON)

Que es como un idioma que entienden las computadoras. Son las **listas** y **diccionarios** de python.

## API

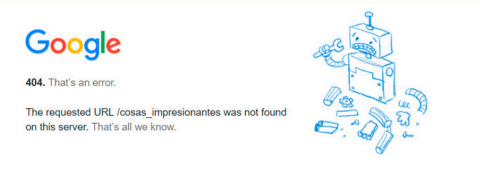
Marta Lugo López  
GIR0641

## Códigos de respuesta

Los códigos de respuesta son mensajes cortos que ayudan a entender lo que pasó con tu solicitud.

### Códigos de error del cliente

- 400 = No puedo entenderla
- 401 = No autorizado
- 404 = No se ha encontrado



### Códigos de redirección

- 301 = Movido permanentemente
- 302 = Movido temporalmente



### Códigos de éxito

- 200 = OK
- 201 = Creado



### Códigos de error del servidor

- 500 = Error interno del servidor
- 503 = Servicio no disponible

500  
Unexpected Error :(  
An error occurred and your request couldn't be com

## Flask REST API Python

```
from flask import Flask, jsonify, request
import game_controller
from db import create_tables

app = Flask(__name__)

@app.route('/games', methods=['GET'])
def get_games():
    games = game_controller.get_games()
    return jsonify(games)

if __name__ == '__main__':
    create_tables()
    """
    Here you can change debug and port
```

## ¿Cómo funciona?

```
@app.route('/usuarios', methods=['GET'])
def obtener_usuarios():
    return jsonify(usuarios)
```

@app.route('/'): Define la ruta principal.  
app.run(debug=True): Inicia el servidor y activa el modo debug para detectar errores.

```
if __name__ == '__main__':
    app.run(debug=True)
```

pip install flask

```
from flask import Flask, jsonify

app = Flask(__name__)
```

Definir rutas  
Ejemplo: /usuarios o /productos.  
Métodos HTTP  
Trabajar con JSON

```
{
  "id": 1,
  "nombre": "Juan",
  "edad": 25
}
```