



códigofacilito



# REST

Bootcamp Backend Avanzado: Microservicios

Carolina - Backend Developer





- **Senior Backend Developer Gorilla Logic**
- **Co organizadora Python Colombia y Django Girls Colombia**
- **Amante del mar y deportes acuáticos.**
-  **/carolinagomezt**





Código de la clase 🦆





>\_ ¿Qué es REST?

>\_ ¿Qué es RESTFul?

>\_ Ventajas y Desventajas

>\_ Demo

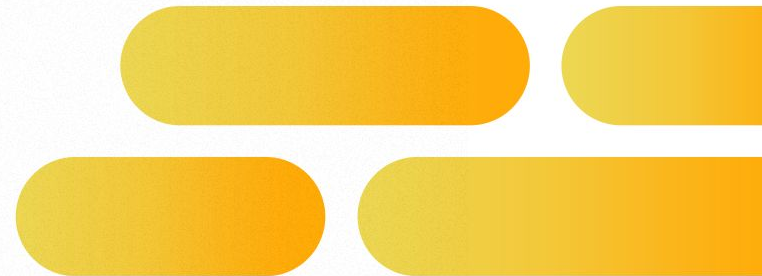
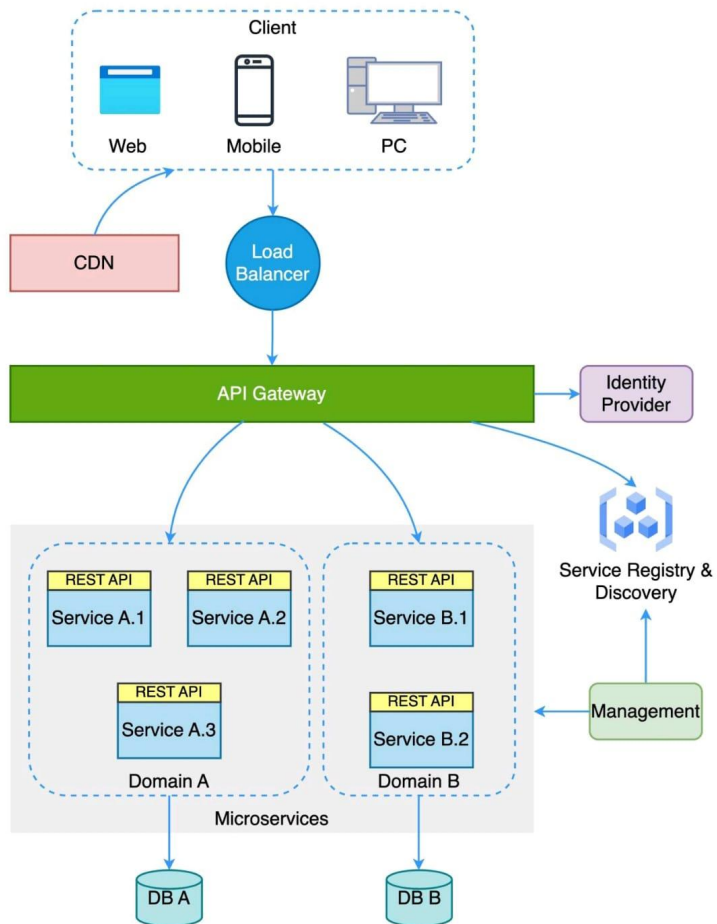






## Microservice Architecture

ByteByteGo



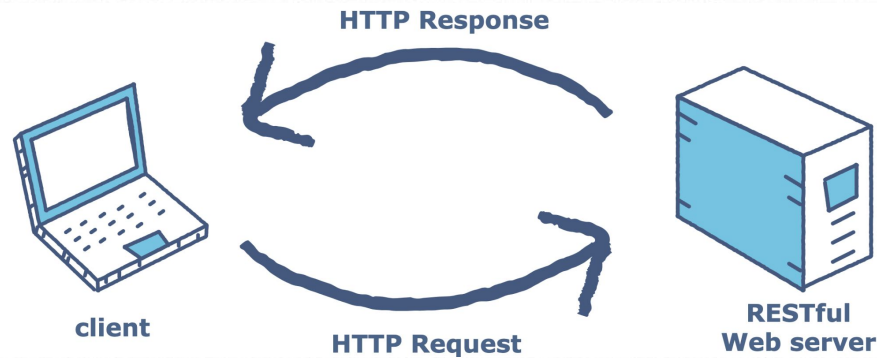


# ¿Qué es REST?





**Representational State Transfer o REST por sus siglas en inglés, es un estilo de arquitectura de software para realizar una comunicación cliente-servidor. Los clientes y los servidores intercambian datos mediante HTTP.**



Recurso recomendado: <https://aws.amazon.com/es/what-is/api/>





Recurso recomendado: <https://restfulapi.net/>

<https://restfulapi.net>

## Six Guiding Principles of REST

01

Uniform Interface

04

Cacheable

02

Client-Server

05

Layered System

03

Stateless

06

Code on Demand (Optional)



READ MORE





01

**Interfaz Uniforme**

- Identificación de recursos.
- Manipulación de recursos a través de representaciones.
- Mensajes descriptivos.
- Uso de Hipermedia como estado de la aplicación.

02

**Cliente - Servidor**

- El cliente solicita recursos.
- El servidor provee los recursos.
- El cliente y el servidor evolucionan independientemente.

03

**Sin Estado**

- Cada petición contiene la información necesaria para ejecutar la acción.

04

**Cacheable**

- Cada respuesta tiene información de si esta puede ser cacheada o no y por cuánto tiempo.

05

**Sistema de Capas**

- En un sistema de capas, cada una es responsable sólo por conocer información relacionada con ella. Ejemplo: MVC.

05

**Código a Demanda (Opcional)**

- El servidor puede proveer código ejecutable al cliente. Ejemplo: Javascript scripts.





# ¿Qué es RESTFul?



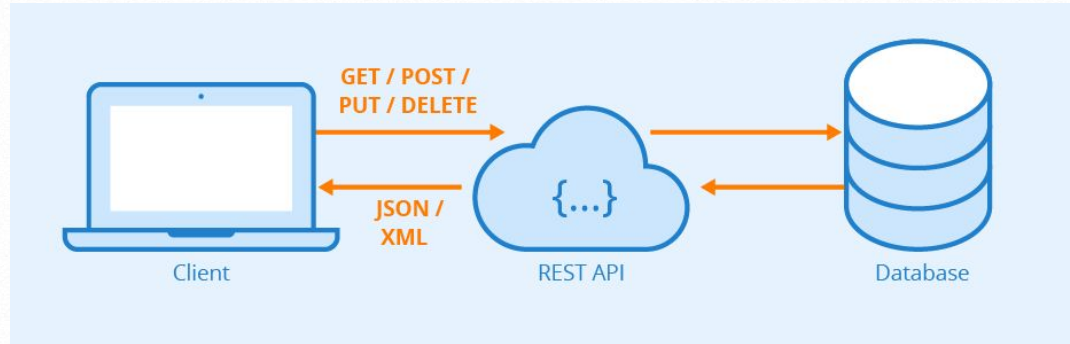


**Así como REST es el estilo de Arquitectura, RESTful es la implementación de dicha arquitectura. Define un conjunto de funciones como GET, PUT, DELETE, etc. que los clientes pueden utilizar para acceder a los datos del servidor. La principal característica de la API de REST es que no tiene estado.**

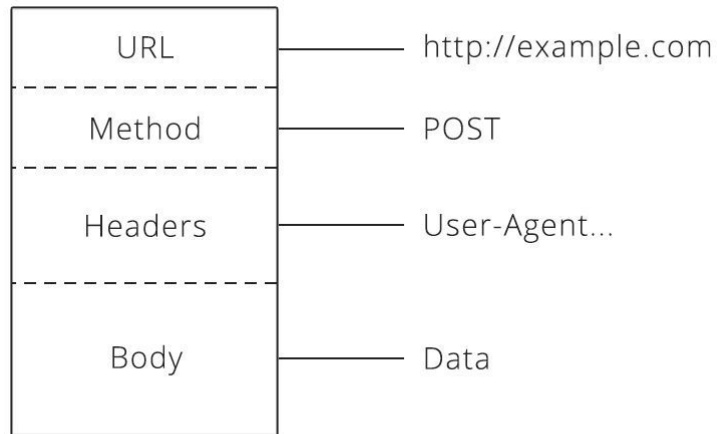


Recurso recomendado: <https://aws.amazon.com/es/what-is/api/>





Tomado de: <https://mechomotive.com/rest-api-in-restful-web-service/>



*Request*

### **Request Header:**

```
GET / HTTP/1.1
Host: www.msaleh.co.cc
User-Agent: Mozilla/5.0 (Windows; U; Windows
NT 5.1; en-US; rv:1.9.0.10)
Gecko/2009042316 Firefox/3.0.10
Accept: */*
Connection: close
```

### **Response Header:**

```
HTTP/1.1 200 OK
Date: Sat, 09 May 2009 12:27:54 GMT
Server: Apache/2.2.11 (Unix)
Last-Modified: Thu, 12 Feb 2009 15:29:42
GMT
Etag: "c3b-462ba63a46580"-gzip
Cache-Control: max-age=1200, private,
proxy-revalidate, must-revalidate
Expires: Sat, 09 May 2009 12:47:54 GMT
Accept-Ranges: bytes
Content-Length: 976
Content-Type: text/html
```



Recurso recomendado: <https://developer.mozilla.org/es/docs/Web/HTTP/Headers>



>\_



Tomado de: <https://codestreet.io/p/understanding-http-verbs-a-guide>





# Ventajas y Desventajas





Ventajas	Desventajas
<ul style="list-style-type: none"><li>→ Independencia de tecnologías y lenguajes.</li><li>→ Escalabilidad.</li><li>→ Flexibilidad.</li></ul>	<ul style="list-style-type: none"><li>→ Incremento en la complejidad del diseño.</li><li>→ Consideraciones adicionales de seguridad.</li><li>→ Manejo de errores.</li></ul>







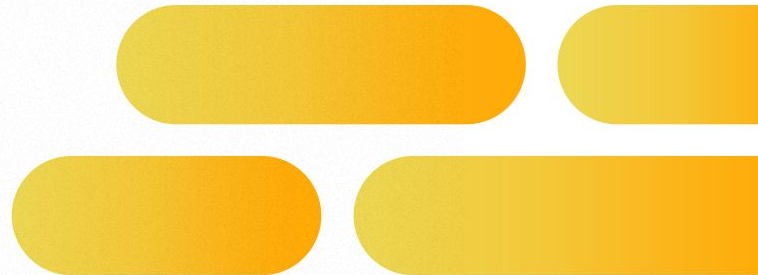
# Demo







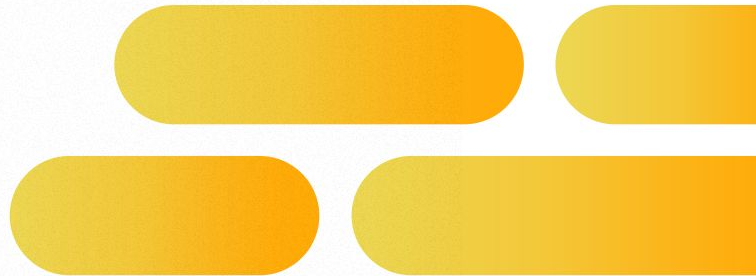
# ¿Preguntas?





# Tarea:

Buscar que es GraphQL y JWT.





# ¡Gracias!

