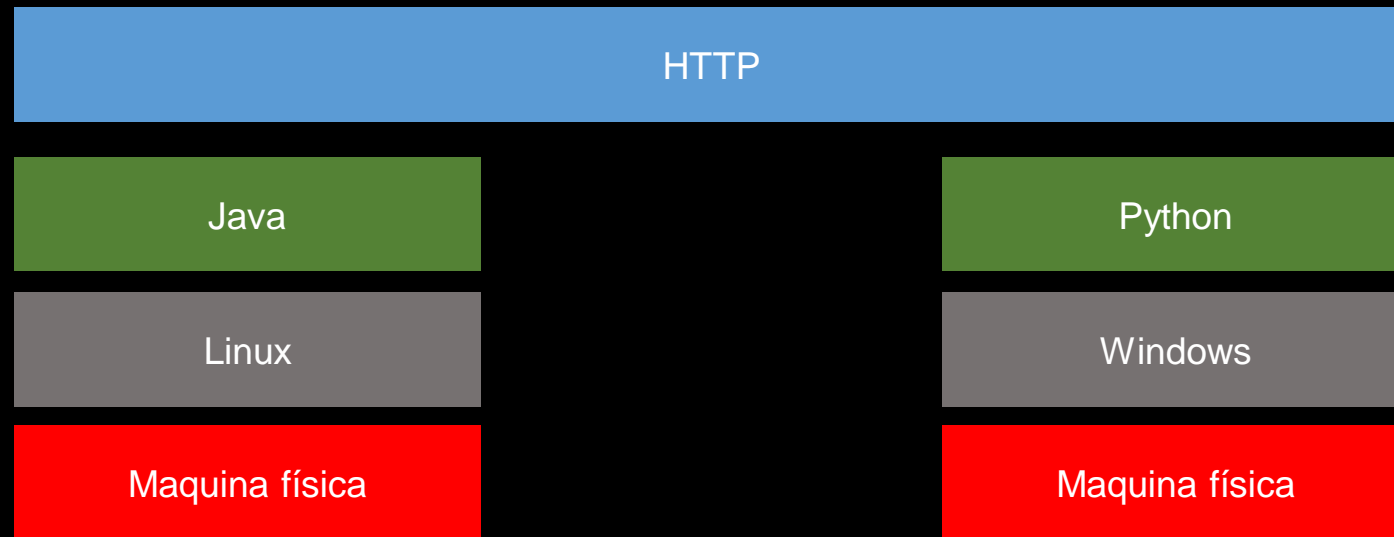


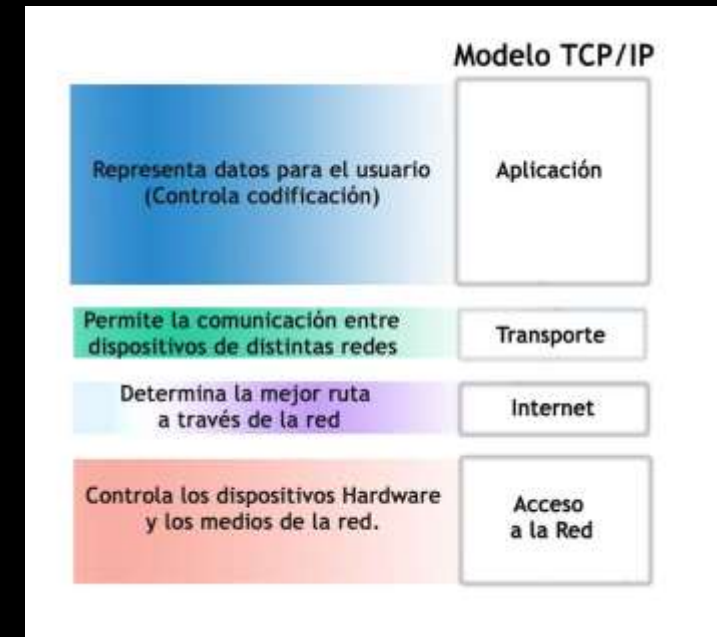
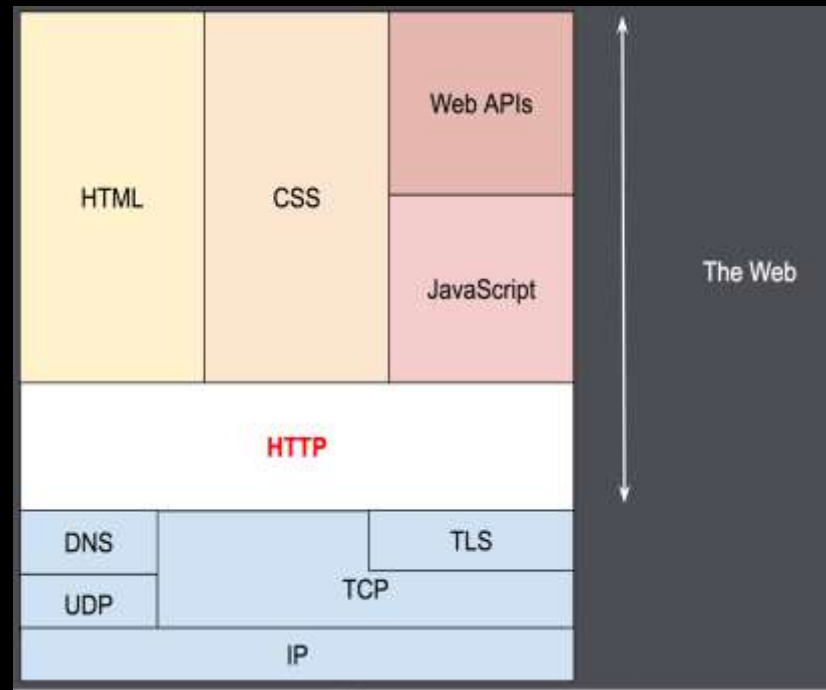
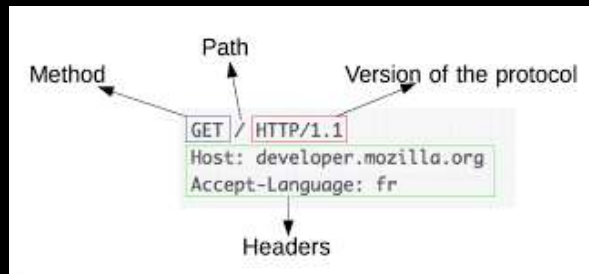


# ¿Cómo se comunican dos servicios via Internet?

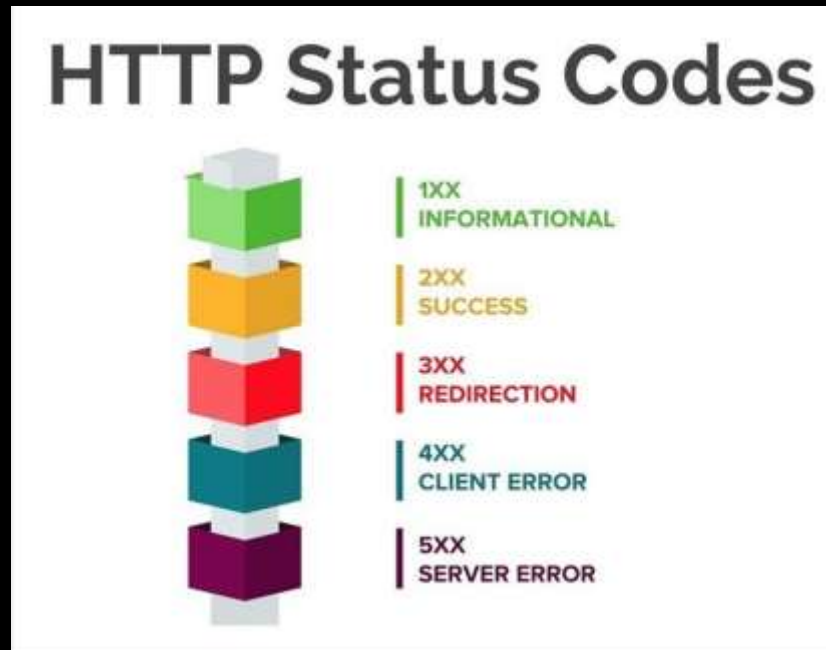


# Protocolo HTTP

Protocolo de aplicación diseñado en los 90s. Se usa para transmisión de datos, documentos, imágenes o vídeos



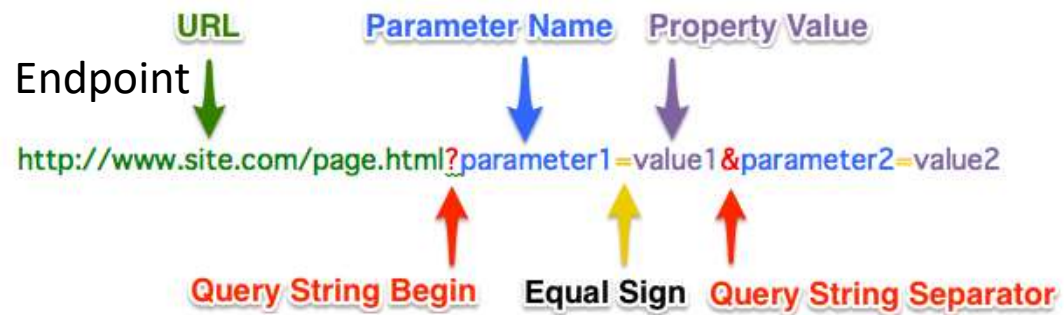
# Protocolo respuesta



HTTP Status Codes		
<b>Level 200 (Success)</b> 200 : OK 201 : Created 203 : Non-Authoritative Information 204 : No Content	<b>Level 400</b> 400 : Bad Request 401 : Unauthorized 403 : Forbidden 404 : Not Found 409 : Conflict	<b>Level 500</b> 500 : Internal Server Error 503 : Service Unavailable 501 : Not Implemented 504 : Gateway Timeout 599 : Network timeout 502 : Bad Gateway

Información sobre cada código

<https://developer.mozilla.org/es/docs/Web/HTTP/Status>



La URL no solo sirve para identificar el protocolo de comunicación con el servidor, y la dirección del servidor, sino que también permite establecer ciertos parámetros que usa el servidor para hacer consultas

# URL

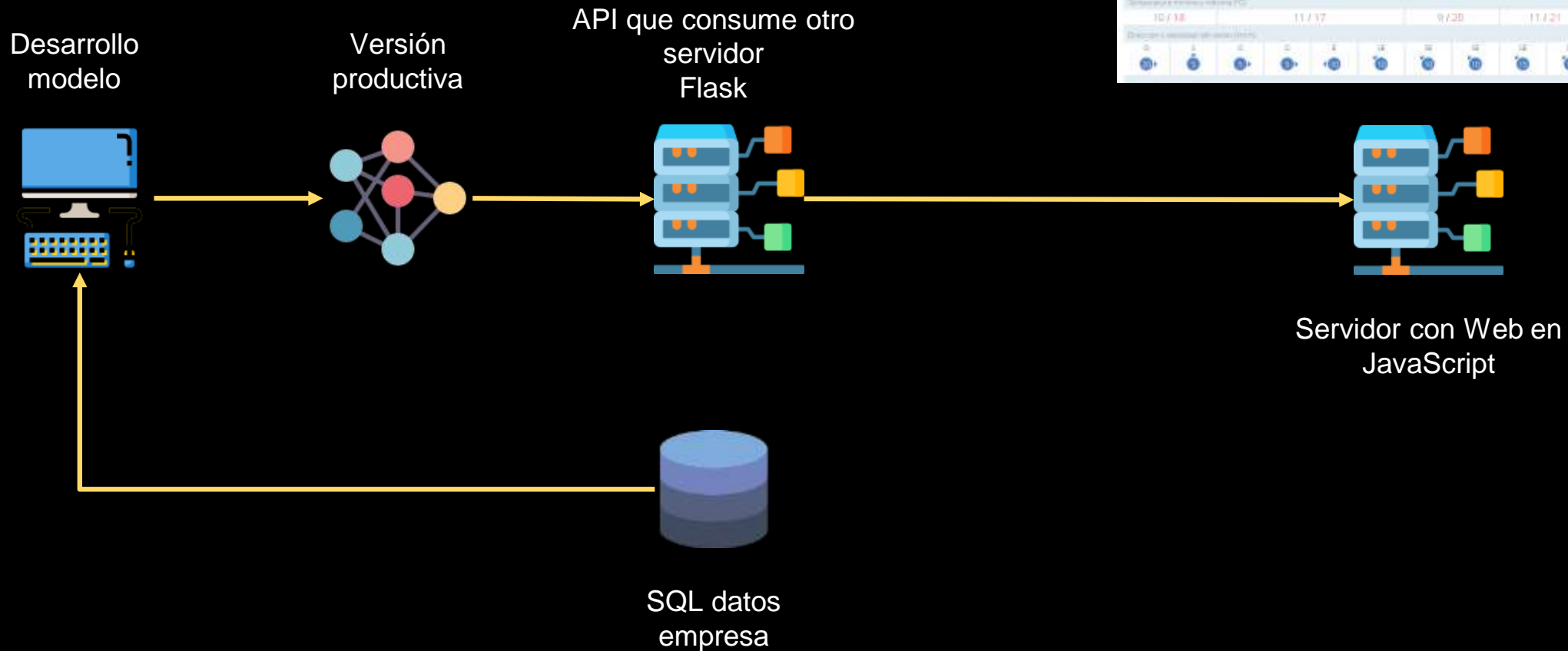
## /books

GET	/books	Lists all the books in the database
DELETE	/books/{bookId}	Deletes a book based on their id
POST	/books	Creates a Book
PUT	/books/{bookId}	Method to update a book
GET	/books/{bookId}	Retrieves a book based on their id

# Peticiones HTTP

Dependiendo de la acción que se quiera realizar sobre el servidor, habrá un tipo de petición diferente

# Arquitectura

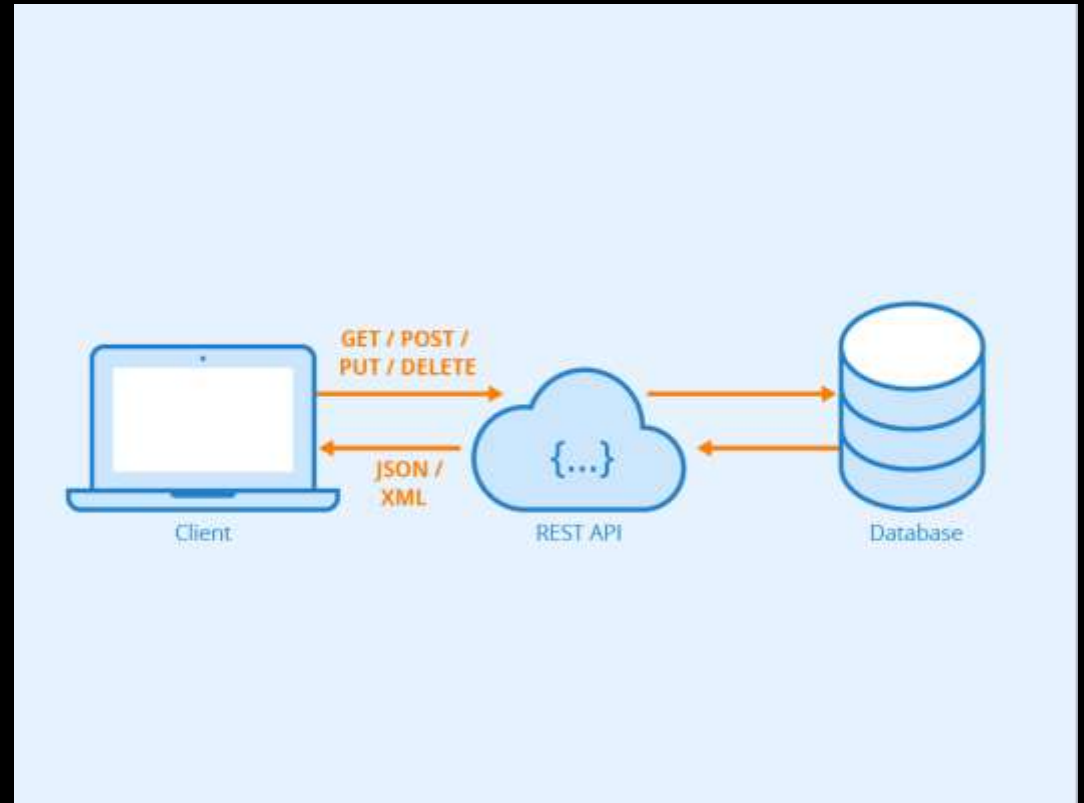


# ¿Qué es una API?

## *Application Programming Interfaces*

Se trata de:

1. Punto de acceso
2. Interfaz de comunicación entre dos piezas de software completamente diferentes
3. Esto es posible ya que se desarrolla sobre HTTP, que es el protocolo de comunicación utilizado en Internet

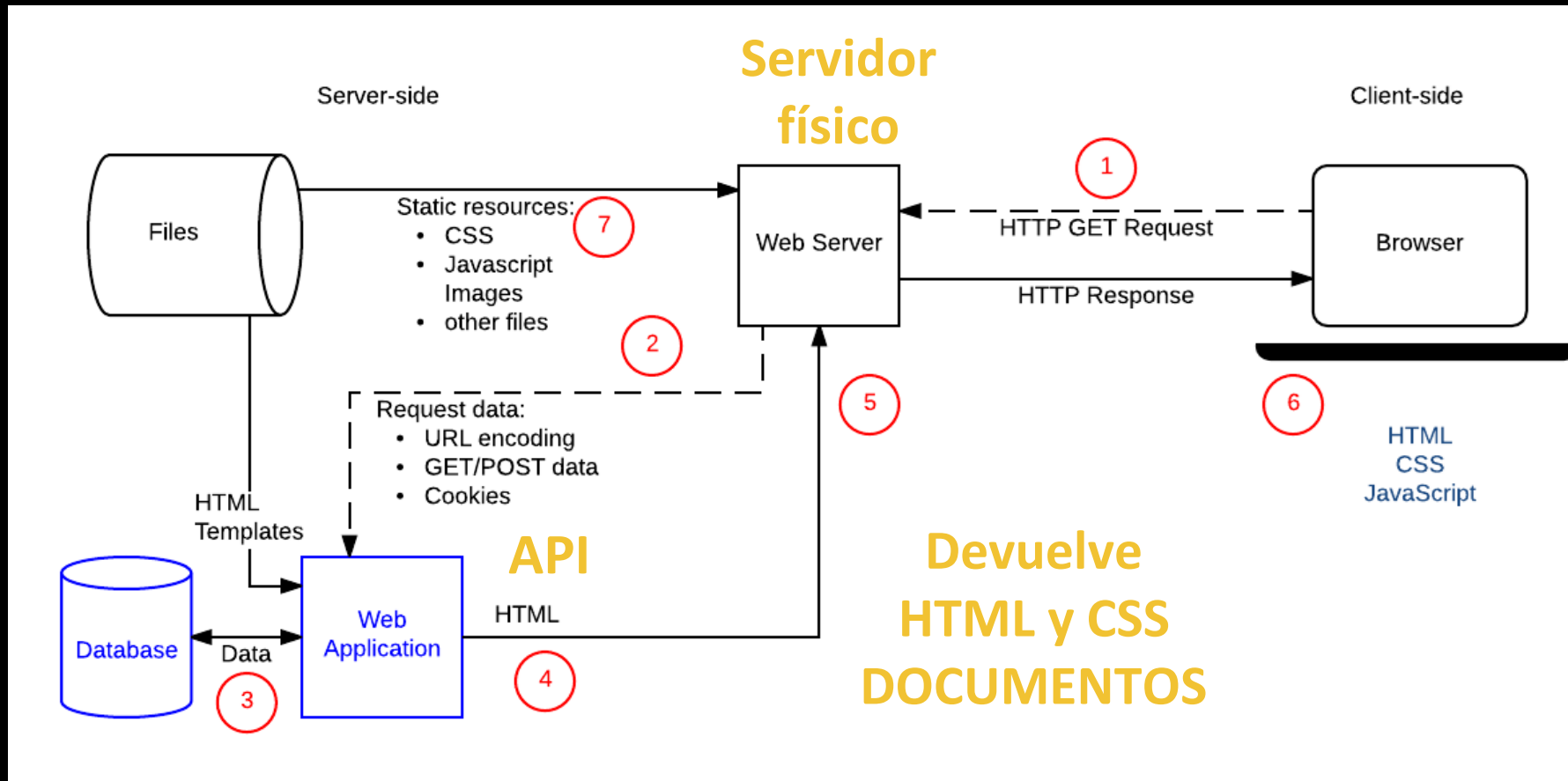




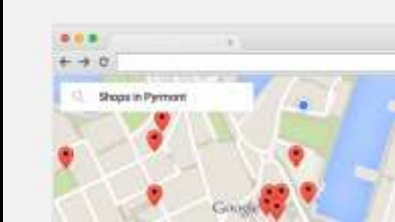
# Ejemplo de API

<https://www.thebridge.tech/>

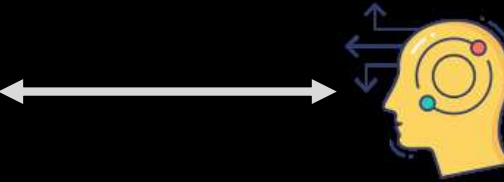
El Navegador(Cliente) pide recursos (la web a renderizar)



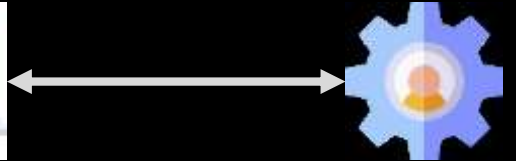
# Casos de uso



**Servicio de datos**  
*Google Maps, Idealista, Tripadvisor...*

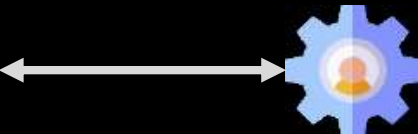


**Algoritmo de ML**  
*Despliegue en producción para nuevas predicciones*

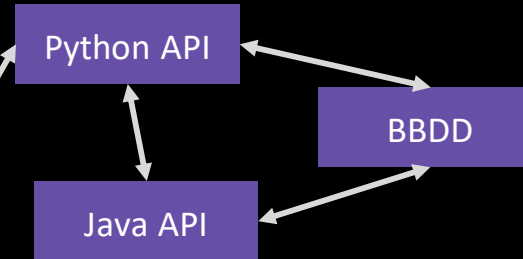


**Algoritmo**  
*Servicio de normalización de direcciones*

# Tipo de arquitectura



**Autoservicio de datos**



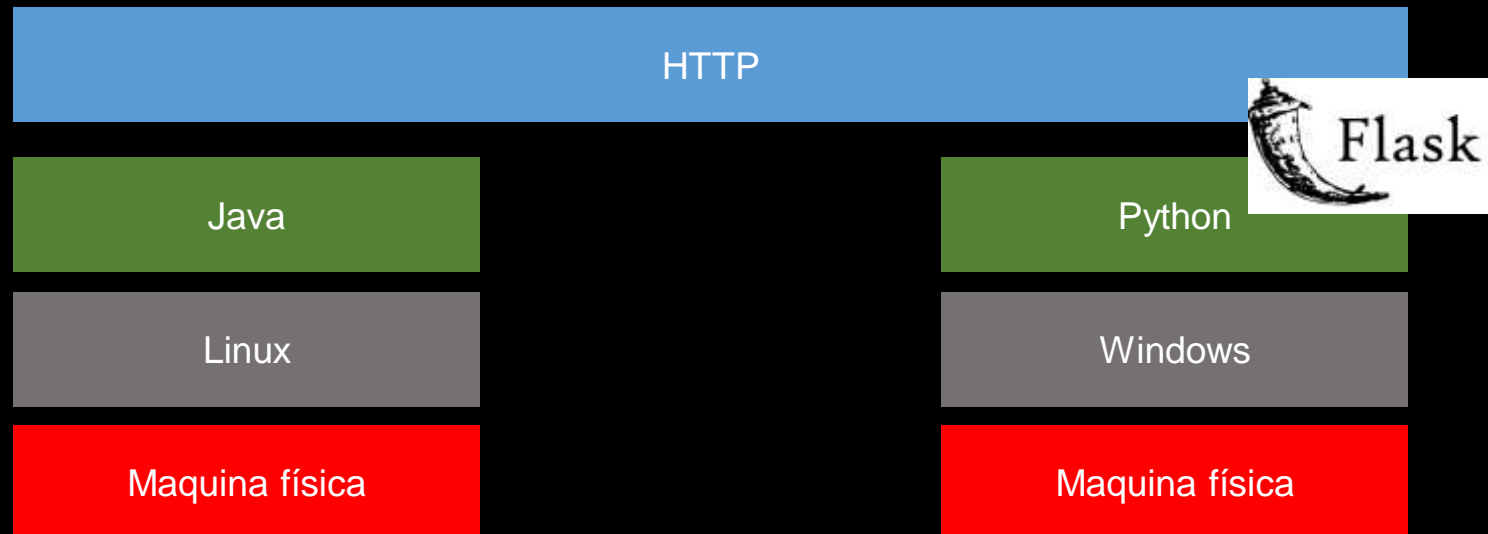
**Arquitectura de microservicios**

# Creación de APIs con Flask

# Flask

Framework web para Python que nos provee de las herramientas y funcionalidades necesarias para montar un servicio web, ya sea una página web, o un servicio tipo API. Este framework incluye el manejo de peticiones HTTP, por lo que no será necesario montarlas.

`pip install flask`  
Viene instalado en Anaconda



Vamos al código