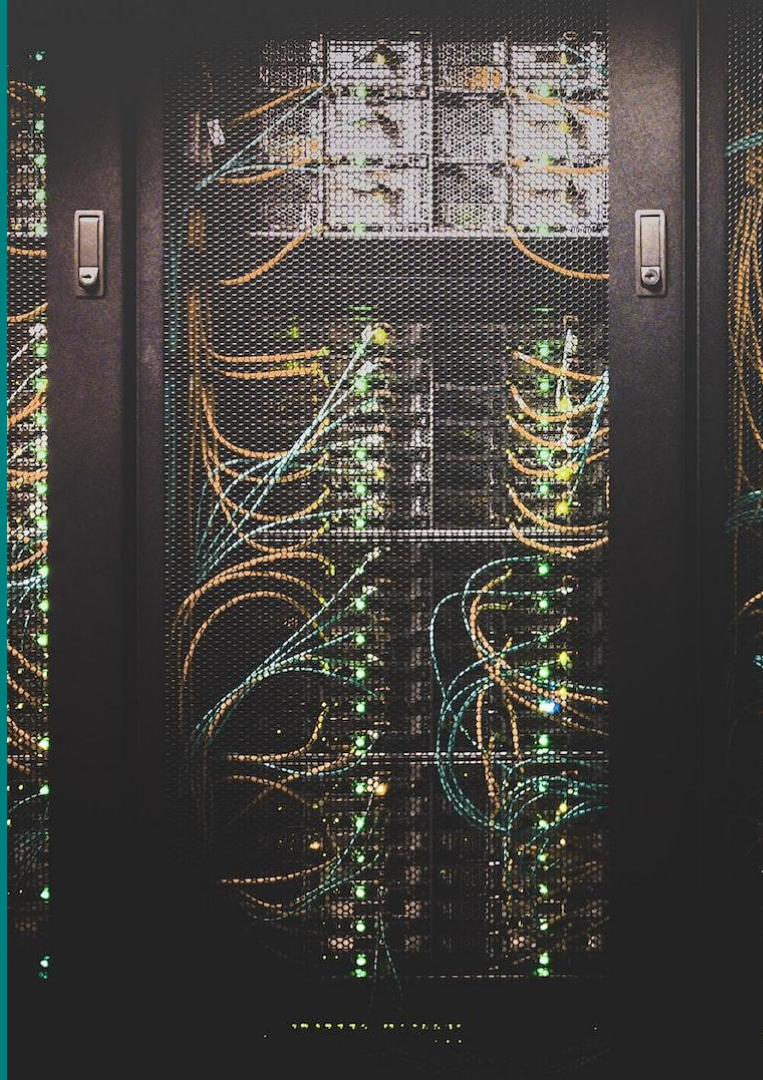
Fundamentos de backend



WOMEN WHO CODE®/medellín

Tabla de contenidos

HTTP

Cómo se direccionan los datos en la web

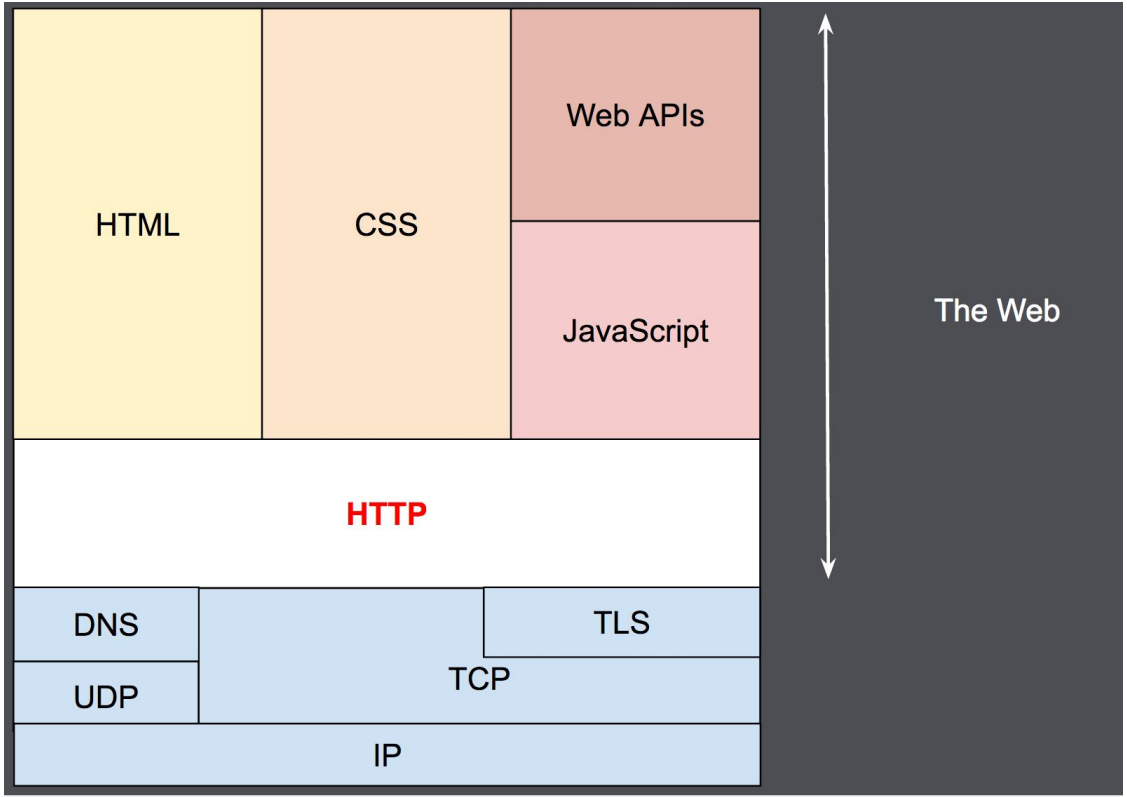
¿Cómo haces para llegar una carta a tu amigo por correo físico?

HTTP

Hypertext Transfer Protocol (HTTP) (o Protocolo de Transferencia de Hipertexto en español) es un protocolo que nos ayuda a transmitir contenidos entre un cliente y un servidor web.

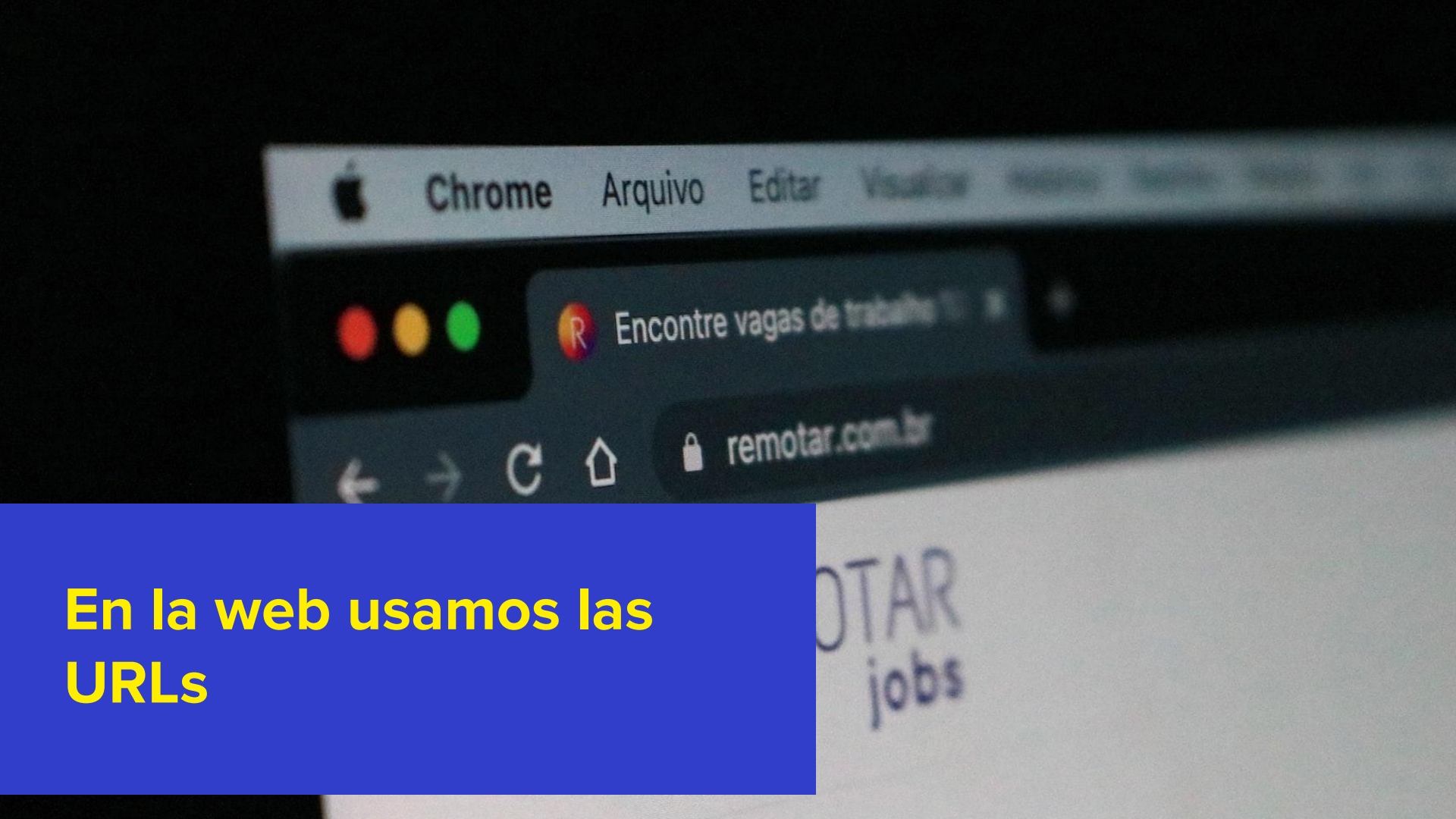
HTTP es él mensajero de la web

Existen otras capas dentro de internet



HTTP es él mensajero de la web, pero ¿cómo sabemos a quién le envían los datos? 🤔


Unsplash
360 Saint-Jacques #G101
Montreal, QC H2Y 1P5
Canada

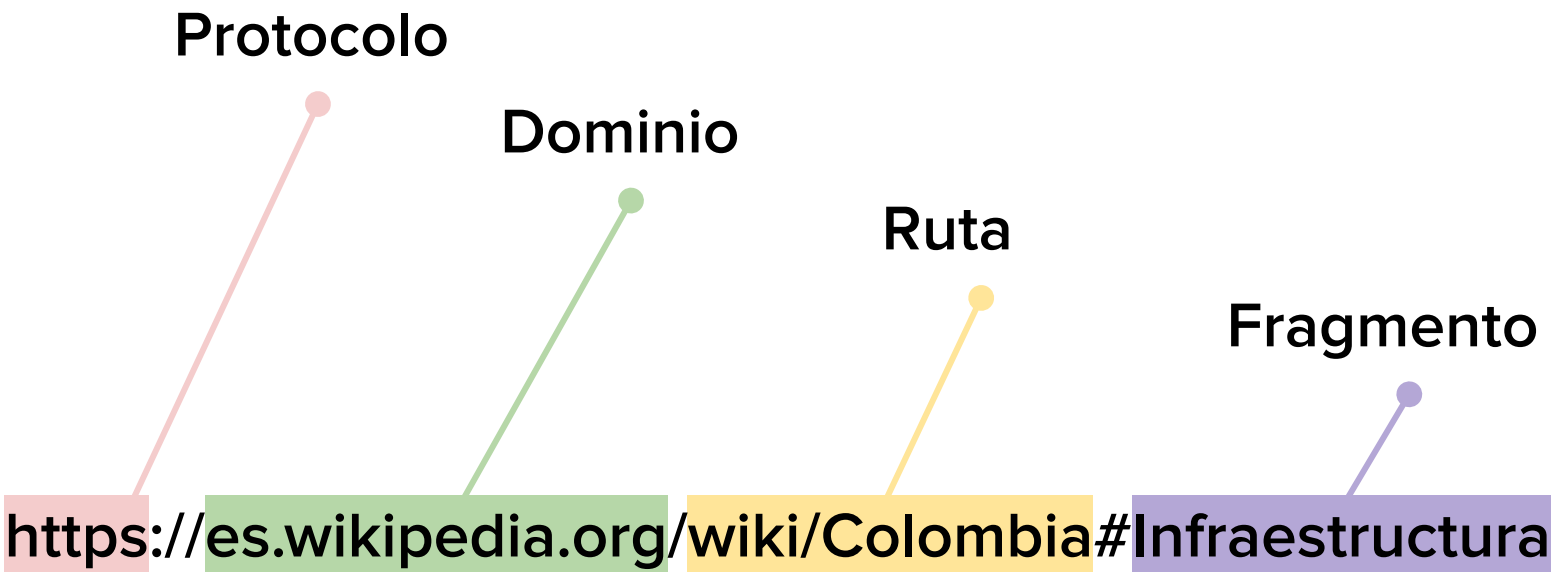


En la web usamos las URLs

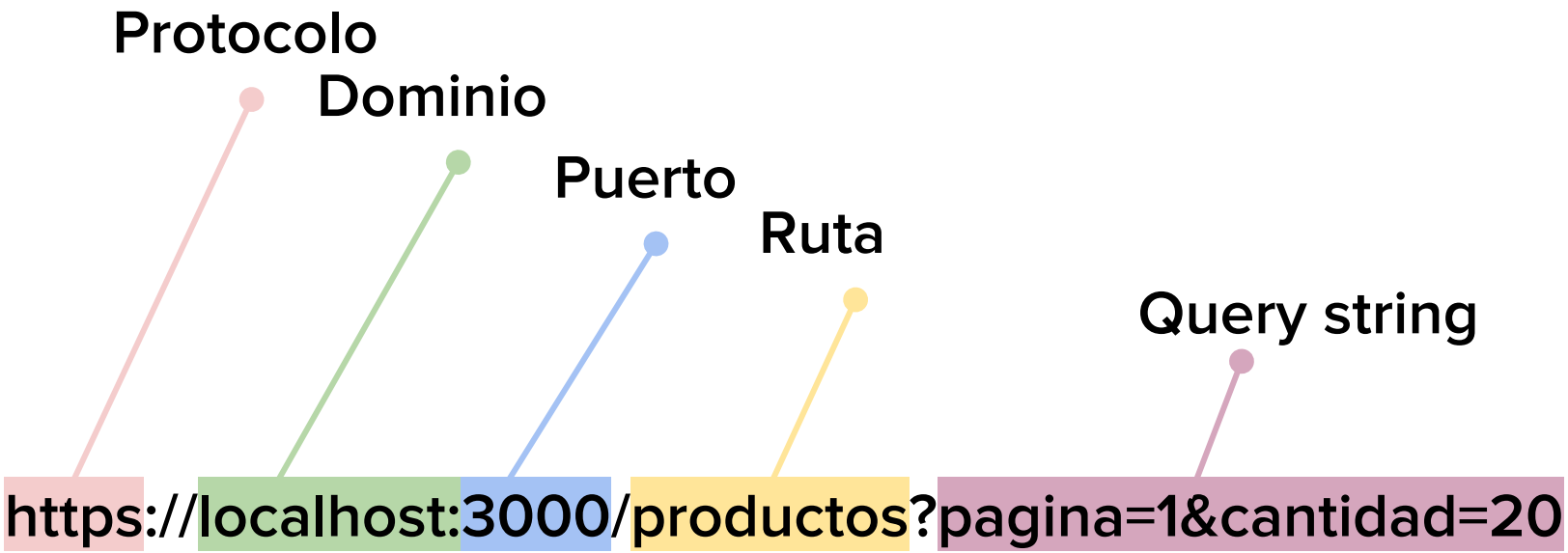
URL

URL significa Uniform Resource Locator (Localizador de Recursos Uniforme) es una dirección que ubica de forma única a cada recurso que está en internet

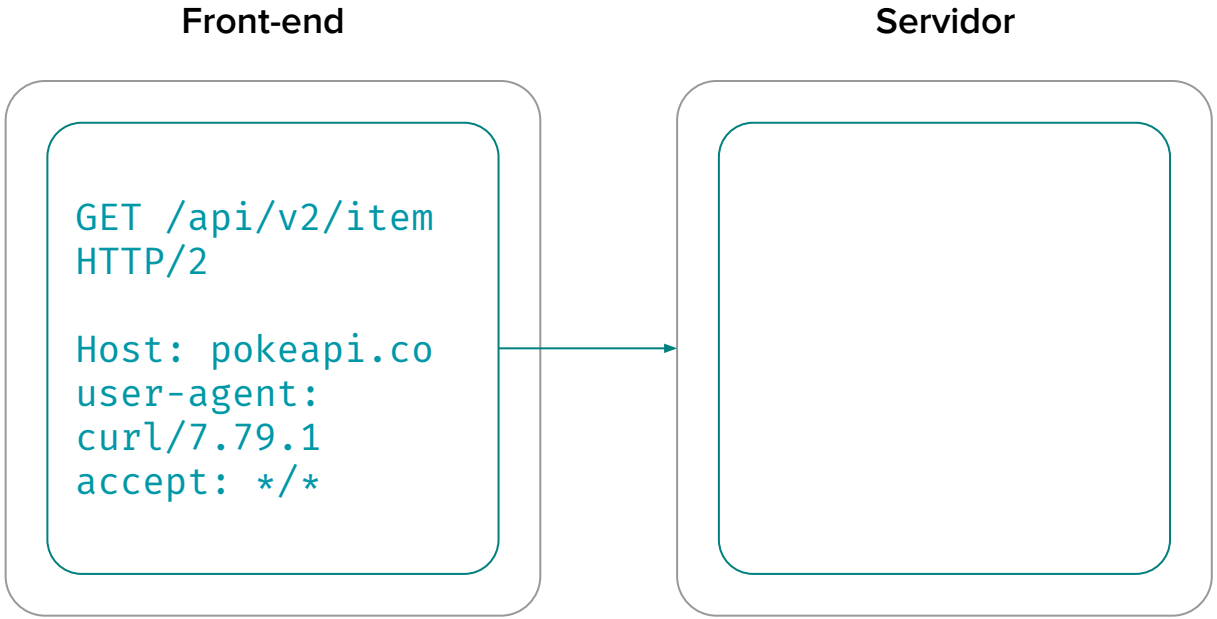
¿Cómo se ve una URL?



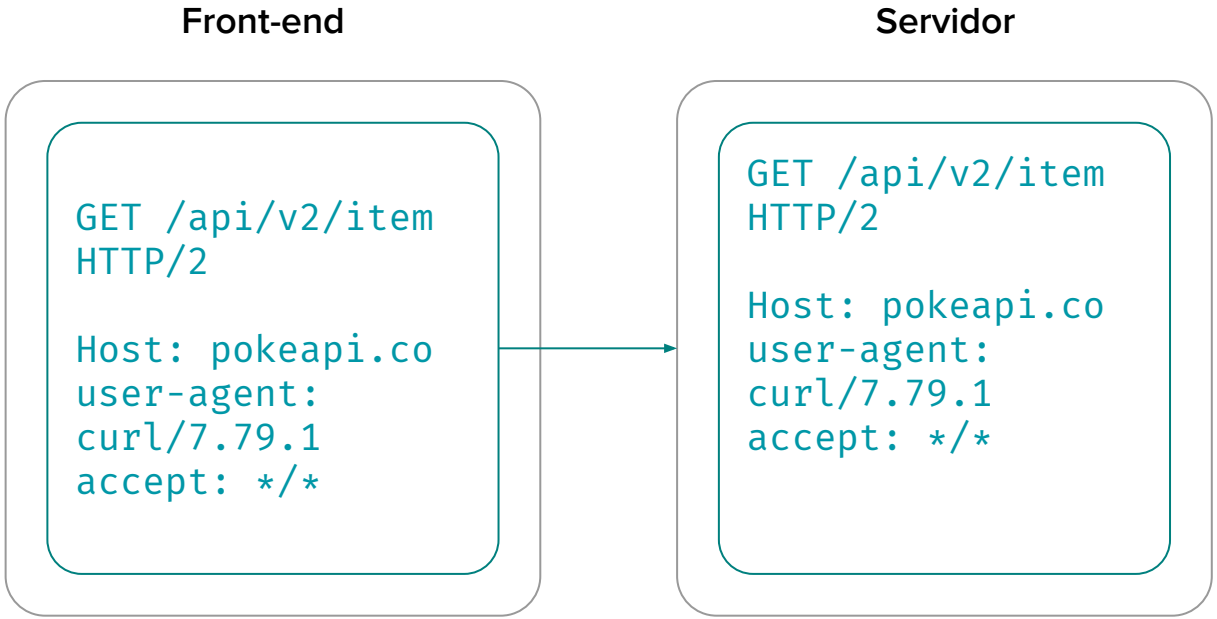
¿Cómo se ve una URL?



Peticiones HTTP



Peticiones HTTP



Peticiones HTTP

Front-end

```
GET /api/v2/item
HTTP/2

Host: pokeapi.co
user-agent:
curl/7.79.1
accept: */*
```

Servidor

```
GET /api/v2/item
HTTP/2

Host: pokeapi.co
user-agent:
curl/7.79.1
accept: */*
```

Peticiones HTTP

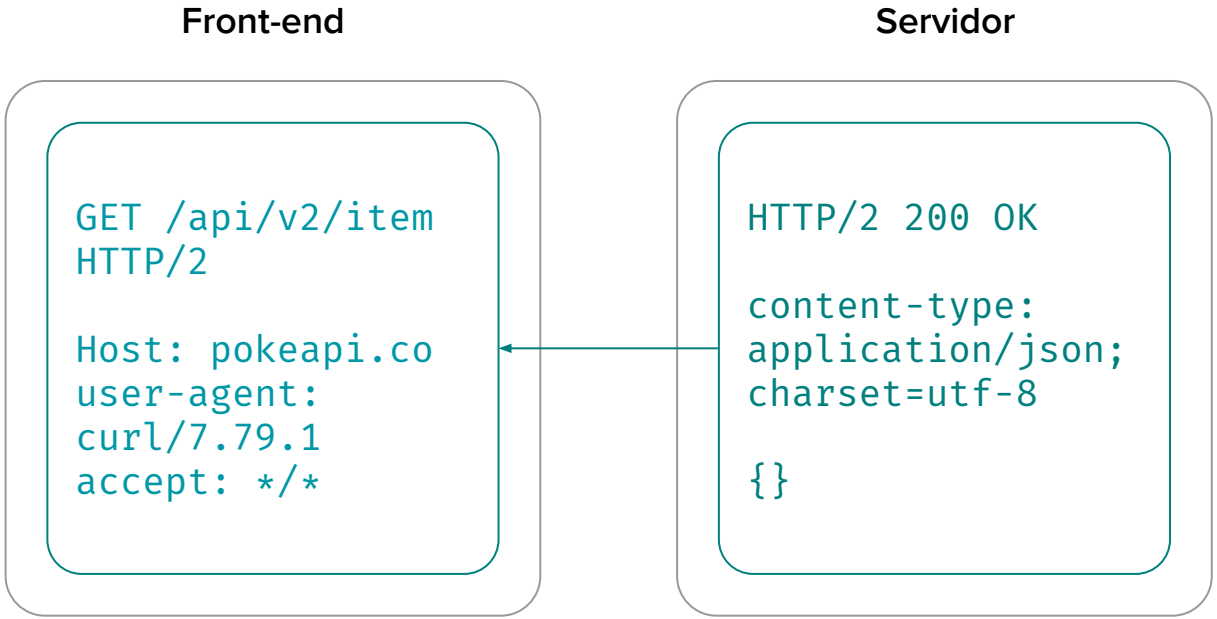
Front-end

```
GET /api/v2/item
HTTP/2

Host: pokeapi.co
user-agent:
curl/7.79.1
accept: */*
```

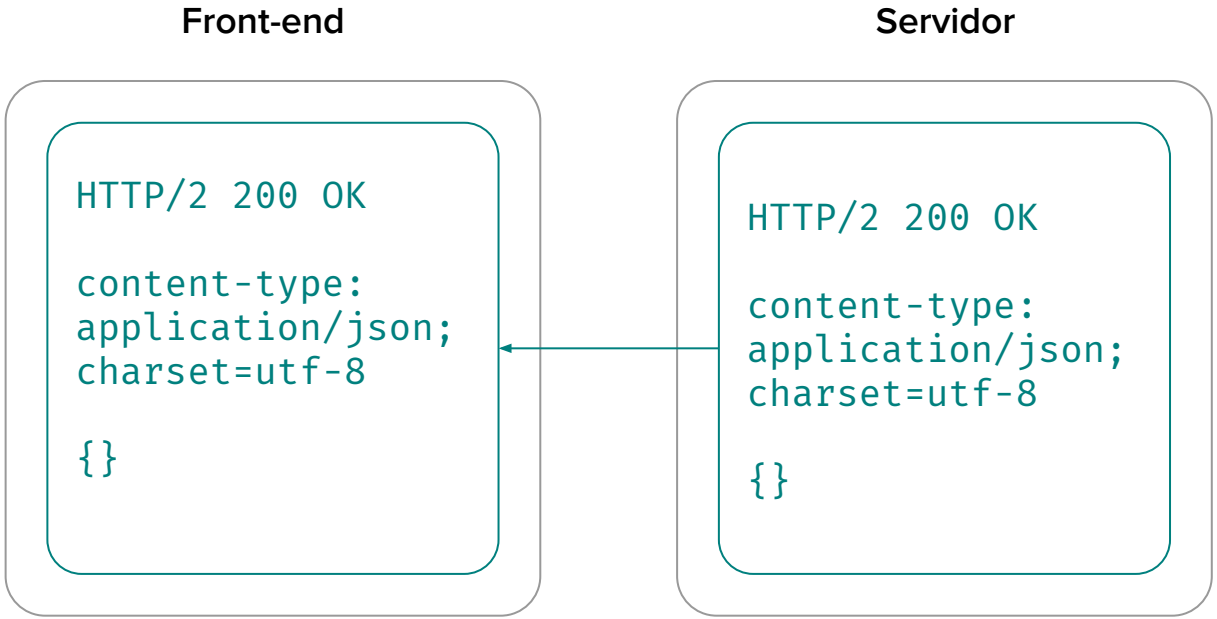
Servidor

```
HTTP/2 200 OK

content-type:
application/json;
charset=utf-8

{}
```

Peticiones HTTP



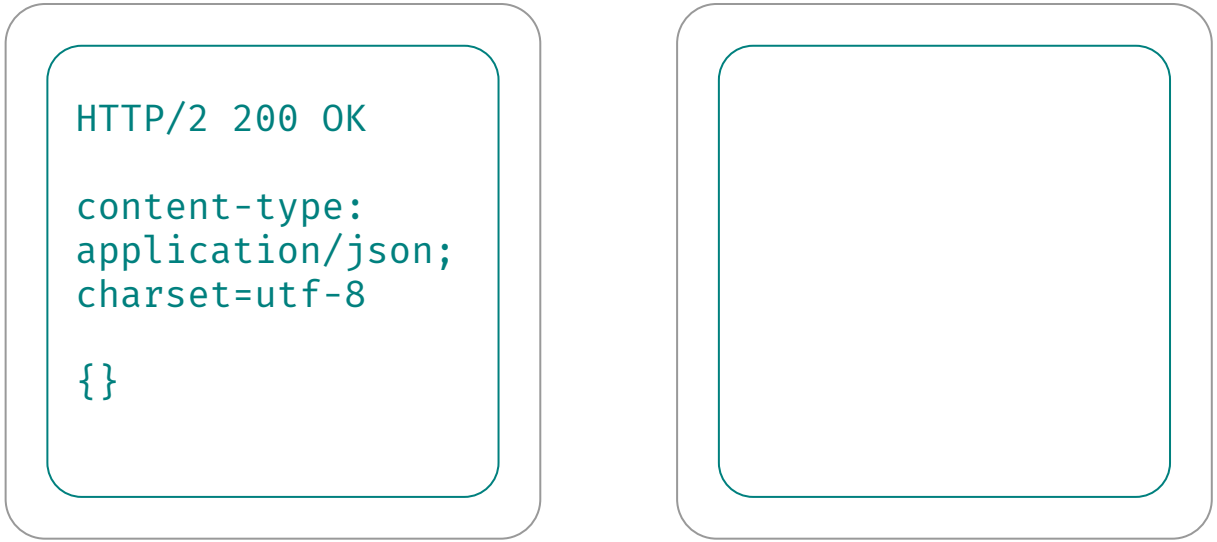
Peticiones HTTP



Peticiones HTTP

Front-end

Servidor



Códigos de status

Existen 5 grupos principales para los códigos de estatus de respuesta

Códigos de status

100 - 199	Respuestas informativas
200 - 299	Respuestas exitosas
300 - 399	Re-direcciones
400 - 499	Códigos de error del cliente
500 - 599	Códigos de error del servidor

Métodos HTTP

GET Solicita un recurso de una entidad

POST Agrega un dato a la entidad

PUT Cambia todos los campos de una entidad

DELETE Borra un dato de la entidad

PATCH Modifica algunos campos de una entidad

Ahora si manos al código 💻

Vamos a crear nuestro primer servidor en node

RESTful API

La arquitectura más usada para él backend en JS

API

Application **P**rogramming **I**nterface

Interfaz de programación de aplicaciones

API

En términos simples: es un intermediario que permite que dos piezas de software se puedan comunicar

RestFul API

01 Interfaz uniforme

Cada recurso debe referirse a una entidad de la aplicación: libros, alumnos, películas, etc.

02 Sin estado

Las respuestas no dependen de los datos que tiene él cliente en ese momento en la aplicación

03 Operaciones definidas

Cada recurso expuesto debe tener una funcionalidad clara, no puede ser multifuncional

04 Sintaxis estandarizada

Cada recurso se puede acceder mediante URL

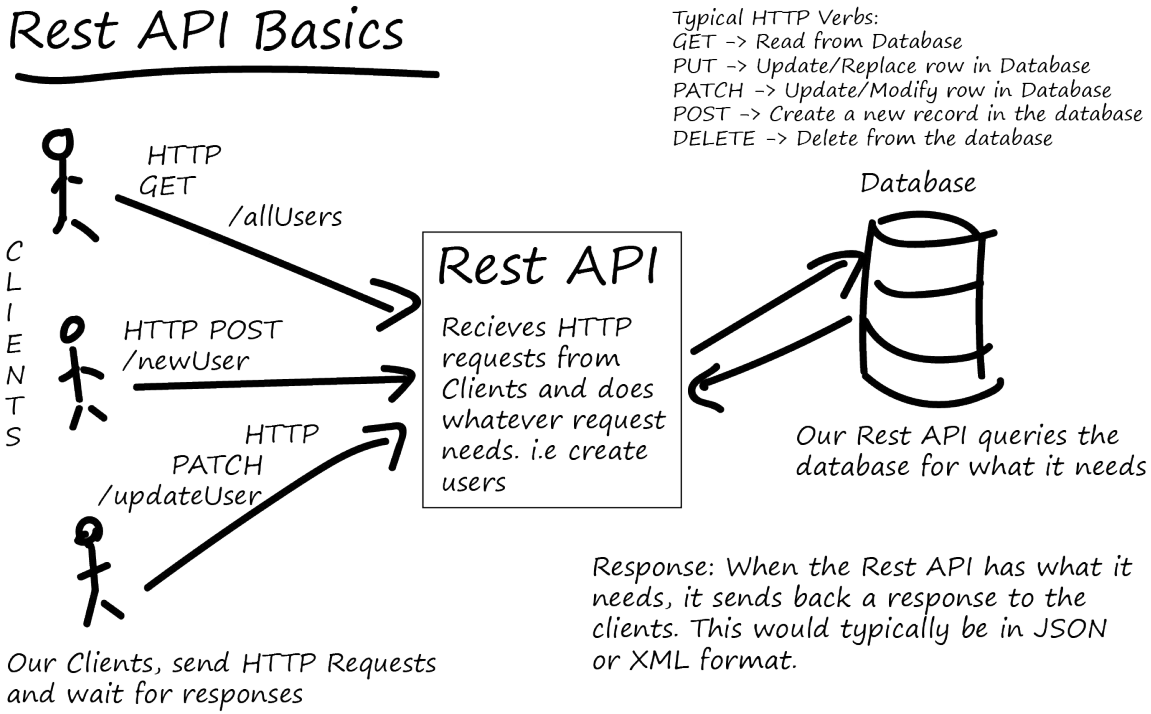
RestFul API

05

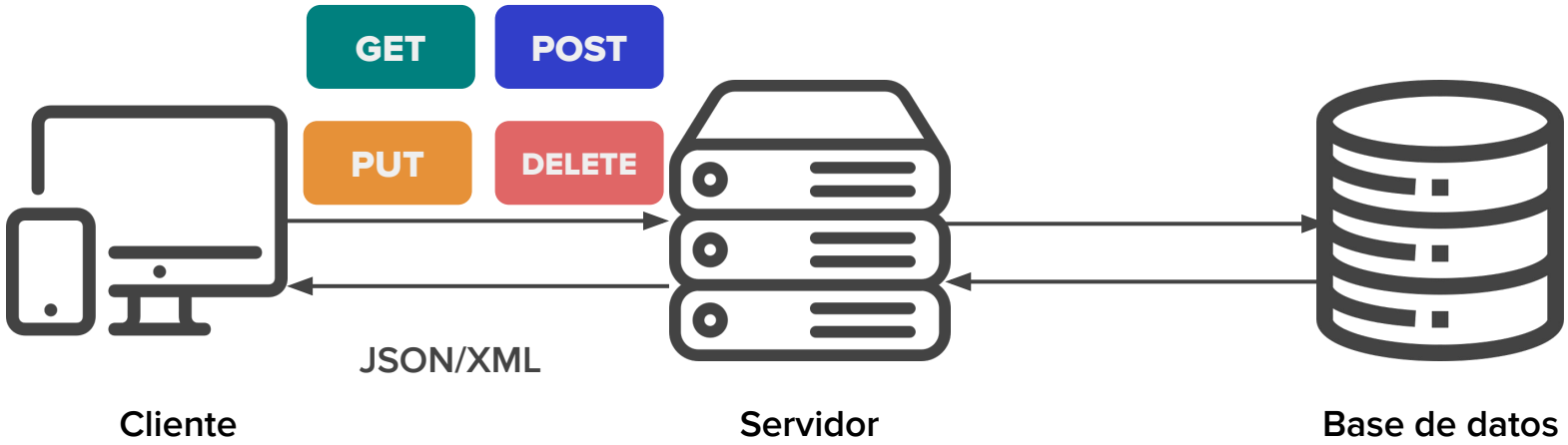
Uso de JSON

Generalmente se usan JSONs para comunicar los datos de la aplicación

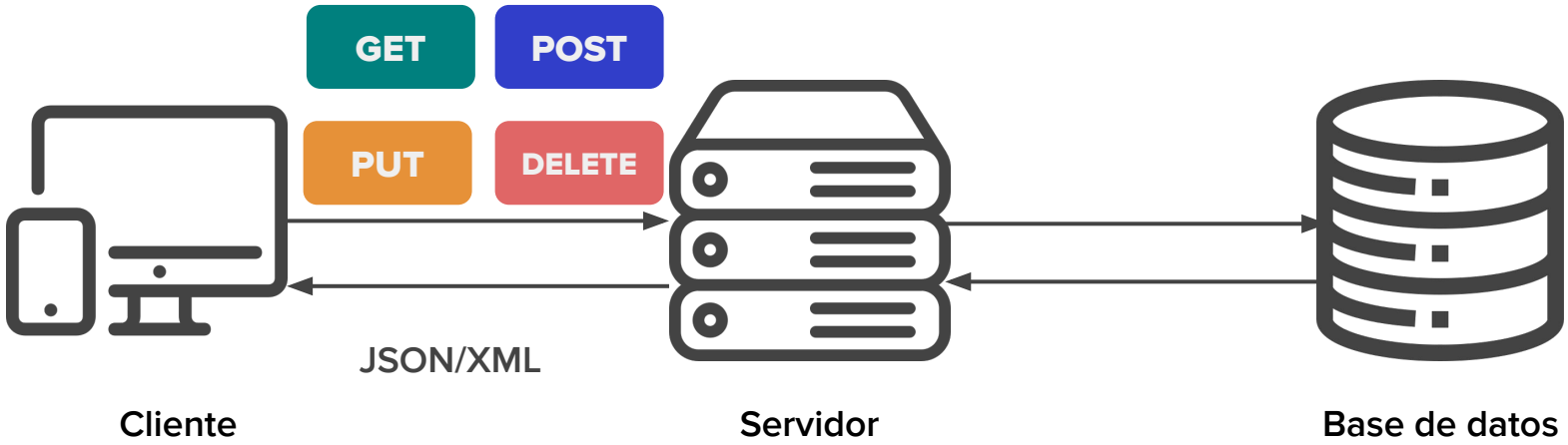
RestFul API


Rest API Basics
Typical HTTP Verbs:
GET -> Read from Database
PUT -> Update/Replace row in Database
PATCH -> Update/Modify row in Database
POST -> Create a new record in the database
DELETE -> Delete from the database
HTTP GET
/allUsers
CLIENTS
HTTP POST
/newUser
HTTP PATCH
/updateUser
Rest API
Recieves HTTP requests from Clients and does whatever request needs. i.e create users
Database
Our Rest API queries the database for what it needs
Response: When the Rest API has what it needs, it sends back a response to the clients. This would typically be in JSON or XML format.
Our Clients, send HTTP Requests and wait for responses

RestFul API



RestFul API



RestFul API Convenciones

Método	**/products**	**/products/{id}**
GET	Obtener lista de productos	Obtener producto
PUT	-	Reemplazar producto
PATCH	-	Actualizar producto
POST	Crear producto	-
DELETE	-	Eliminar productos

Express

Él framework definitivo de Node para crear él backend de tu aplicación en JavaScript

Ahora si manos al código 💻

Vamos a crear nuestro primer servidor en express