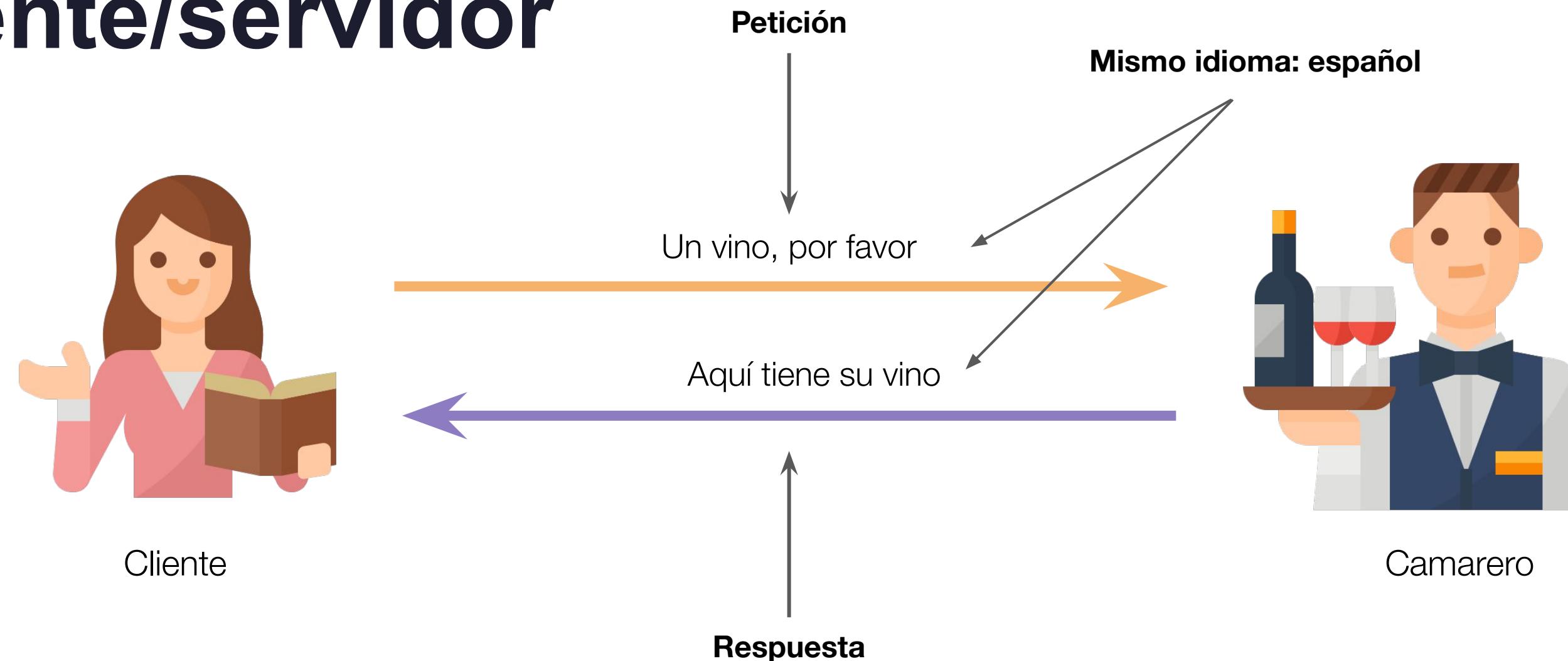


keep coding

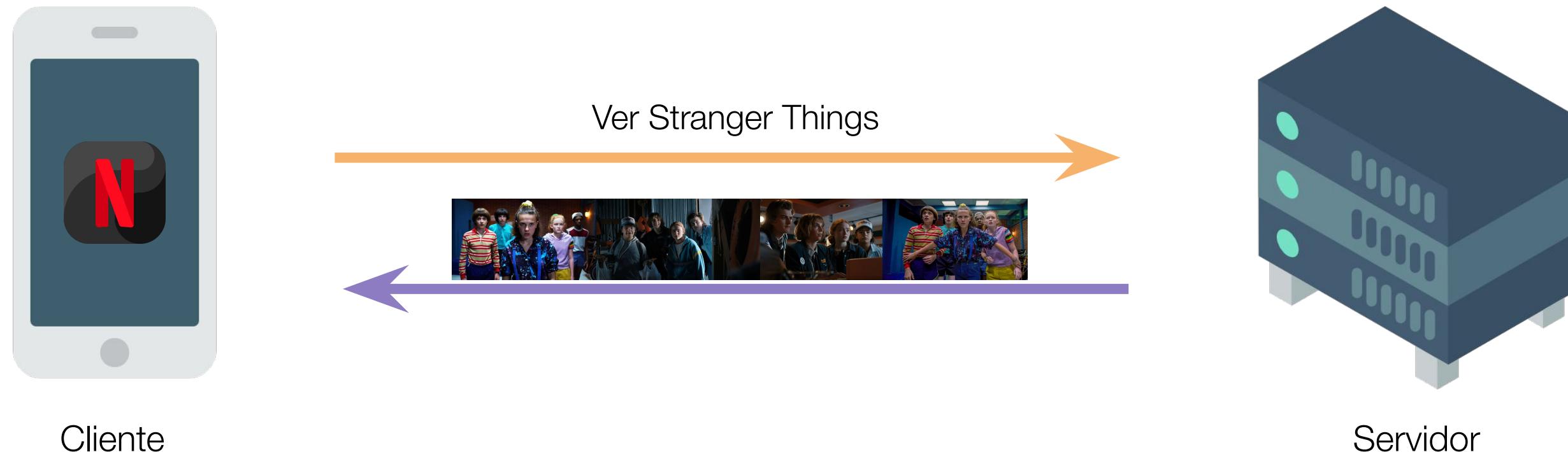
Web 101

¿Cómo funciona la web?

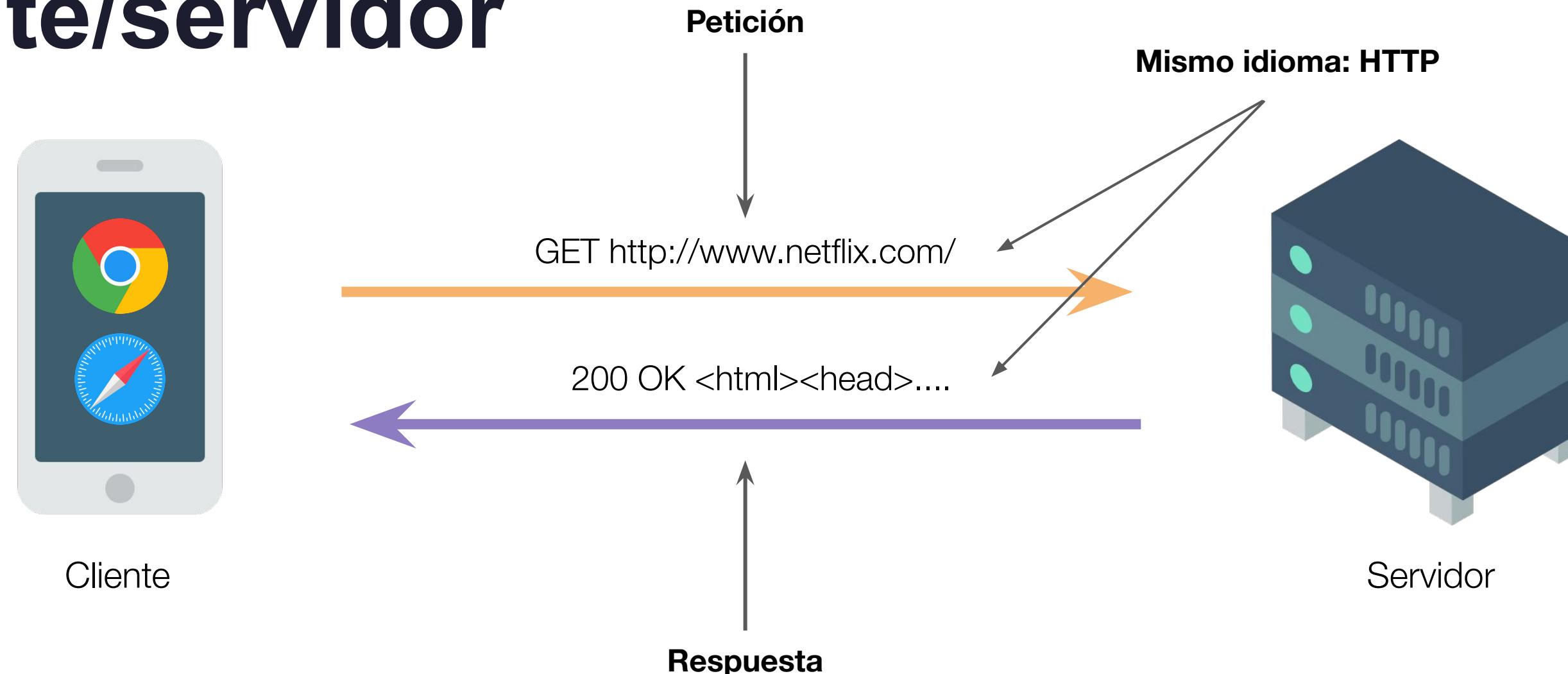
La arquitectura cliente/servidor



La arquitectura cliente/servidor



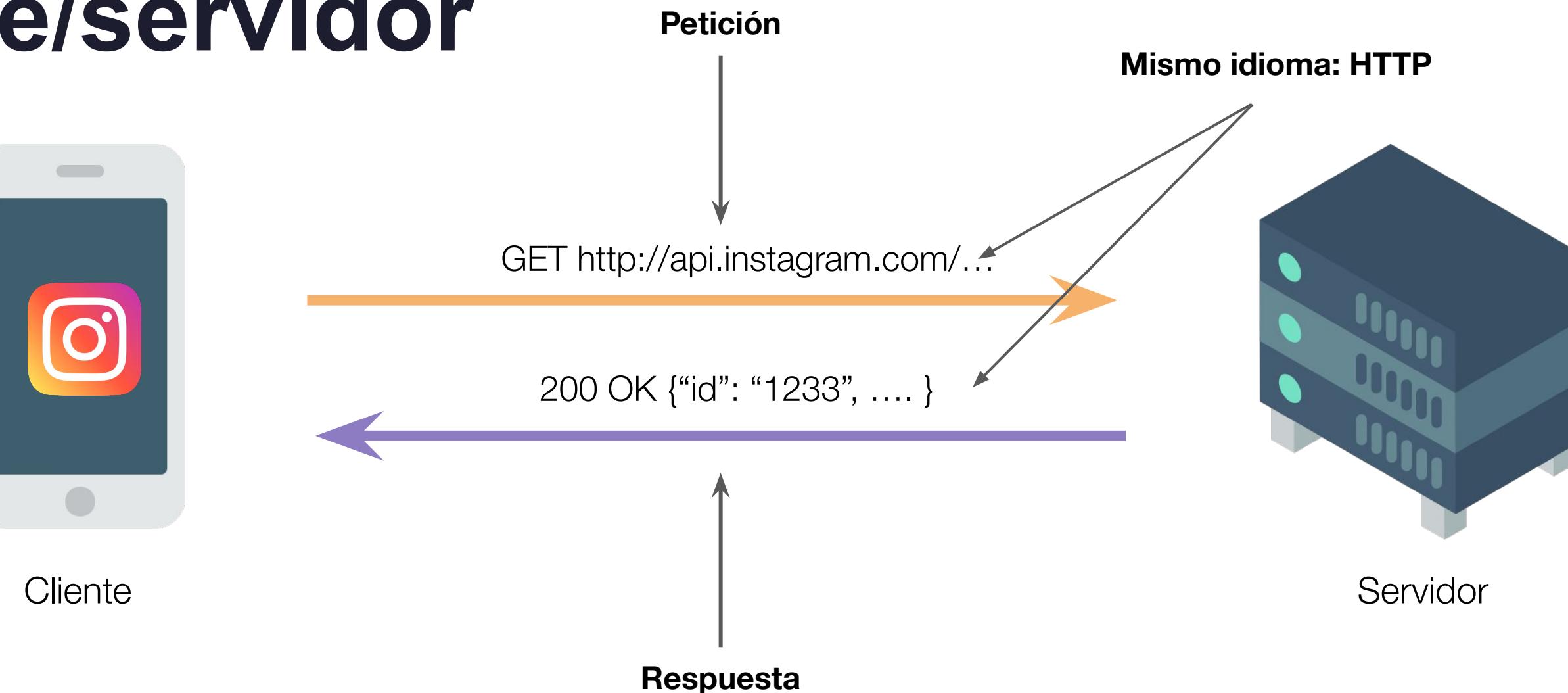
La arquitectura cliente/servidor



La arquitectura cliente/servidor



Cliente



¿Qué es HTTP?

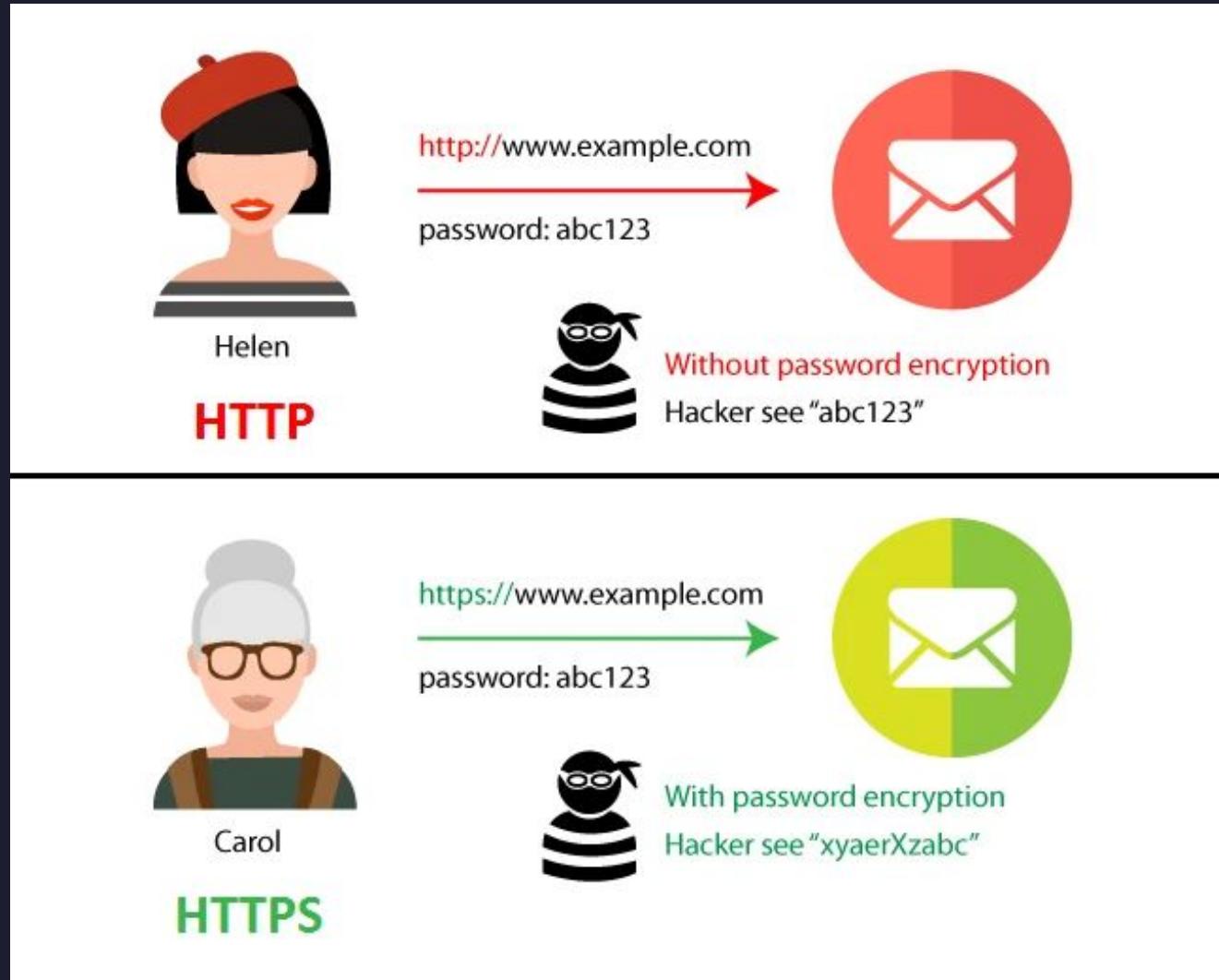
keep coding

- . Es el principal protocolo de comunicación que se utiliza para transferir datos usando tecnologías web.
- . Utiliza la arquitectura cliente servidor.
- . Permite varios tipos de peticiones: GET, POST, PUT, DELETE
- . Todas las respuestas comienzan por un código, los nos dicen si la petición ha ido bien, mal y por qué.

¿Y entonces, qué es HTTPS?

keep coding

- Es HTTP, pero seguro (la S es de Secure).
- Es HTTP, pero encriptado.

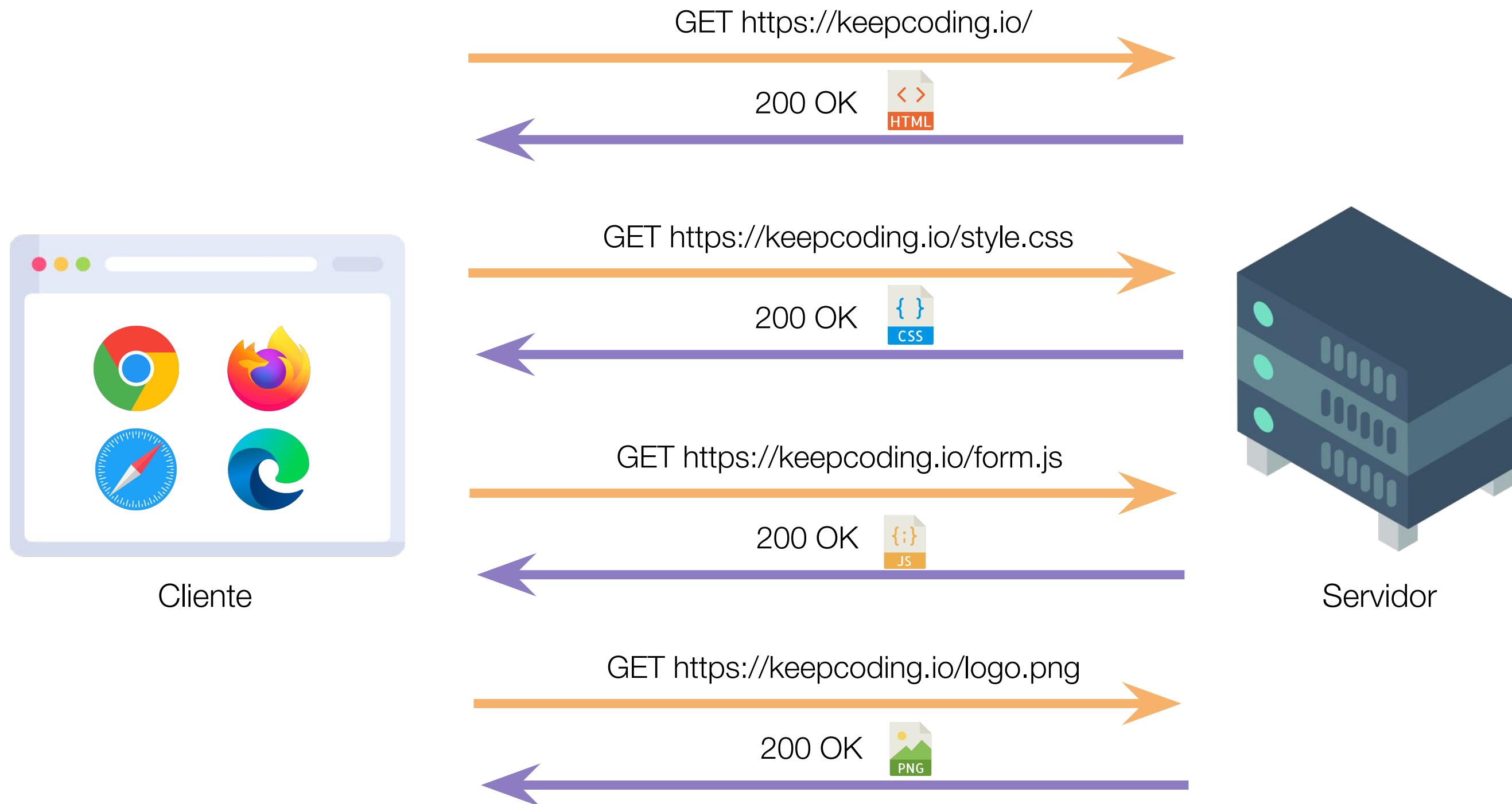


<https://www.edhotels.com/http-vs-https-the-difference-and-everything-you-need-to-know/>

¿Cómo funciona una página web?

¿Cómo funciona una página web?

keep coding

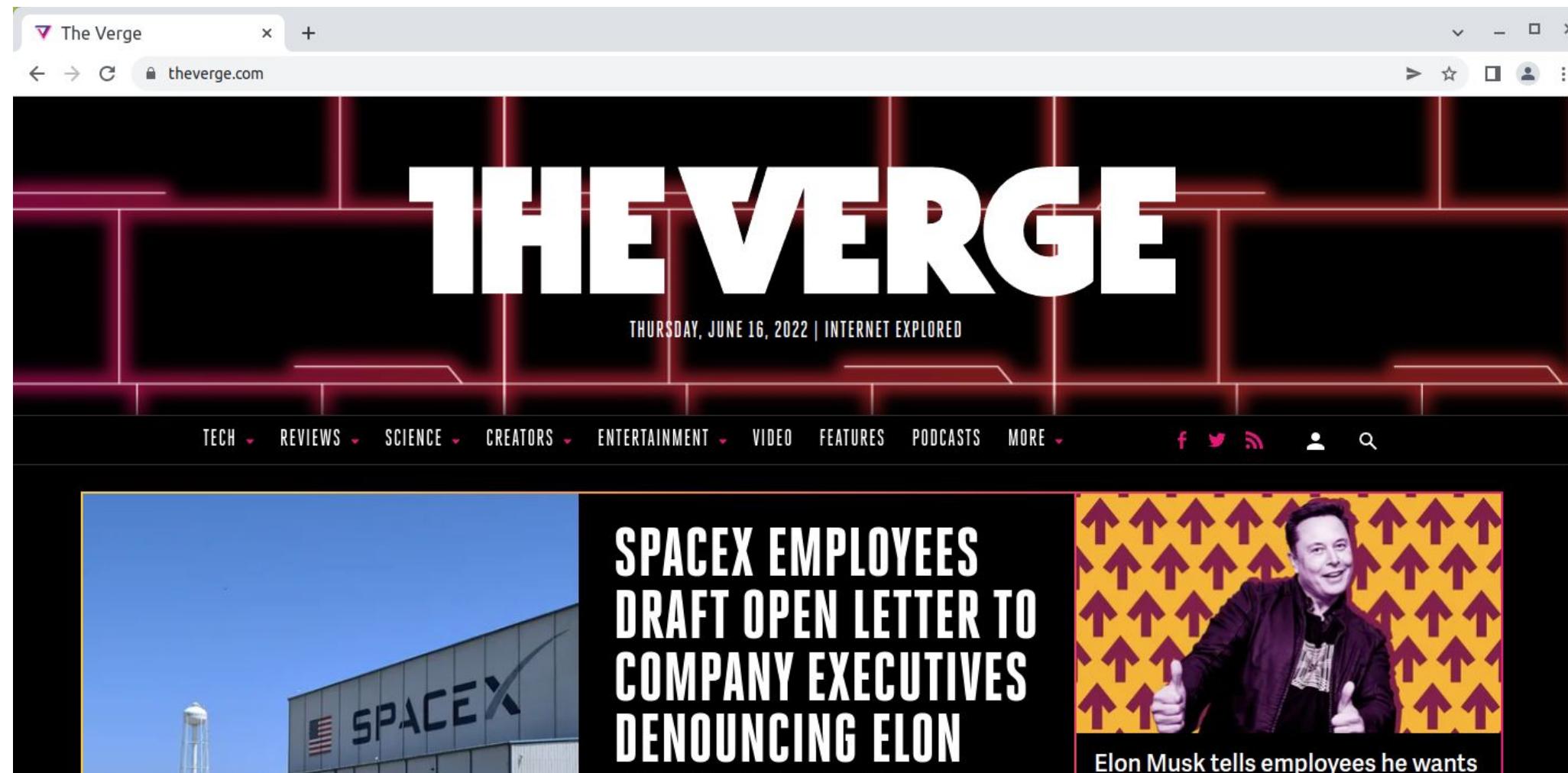
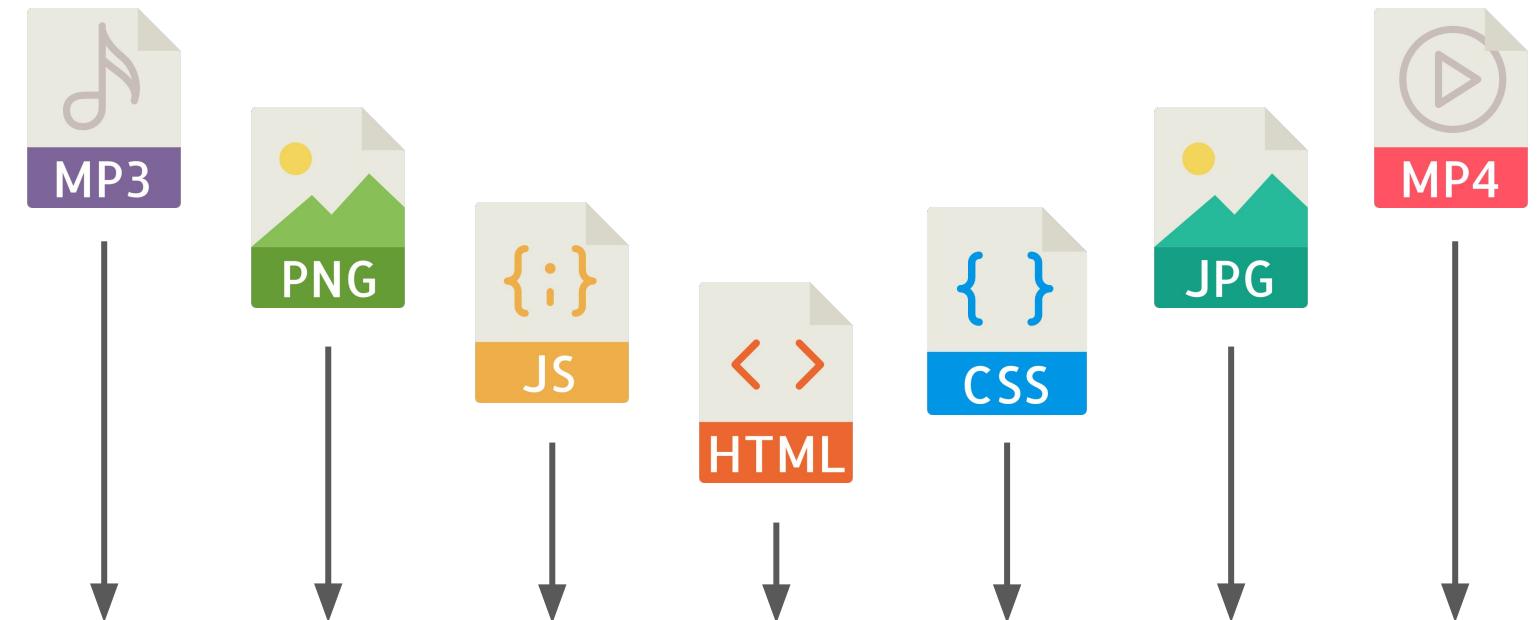


El navegador

keep coding

- Un navegador es un programa
- Que interpreta datos recibidos en **HTML**
- Que les da forma y color usando **CSS**
- Que en su interior, permite ejecutar otros programas escritos en **JavaScript**
- Que soporta infinidad de formatos multimedia (imagen, vídeo, audio, etc.)
- Y que recibe todos estos datos los recibe por **HTTP**

Por tanto: un buen desarrollador web tiene que dominar: HTML, CSS, JavaScript y HTTP/HTTPS.



Los diferentes navegadores

keep coding



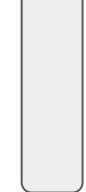
Apple Safari

Motor renderizado: Webkit
Motor JavaScript: Nitro



Google Chrome

Motor renderizado: Webkit -> Blink
Motor JavaScript: Nitro -> V8



Microsoft Edge

Motor renderizado: Blink
Motor JavaScript: V8



Mozilla Firefox

Motor renderizado: Gecko
Motor JavaScript: SpiderMonkey



Internet Explorer

Motor renderizado: MSHTML
Motor JavaScript: Chakra

Edge, hijo de Chrome, heredero de Safari

Internet Explorer



Internet Explorer 6



Internet Explorer 7-8



Internet Explorer 9



Internet Explorer 10-11

I WAS THERE GANDALF,



I WAS THERE 3000 YEARS AGO

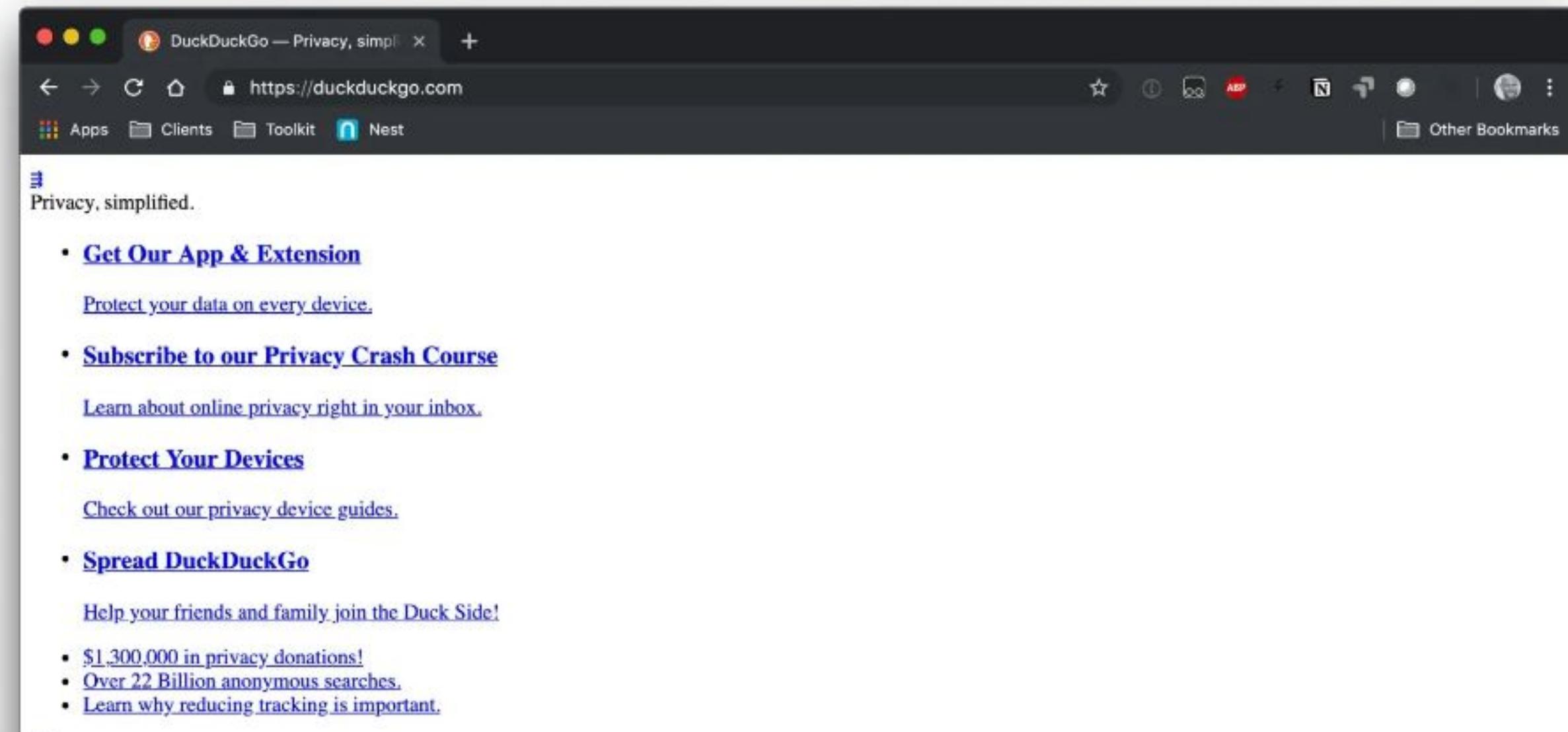
Internet Explorer

keep coding



¿Qué es HTML?

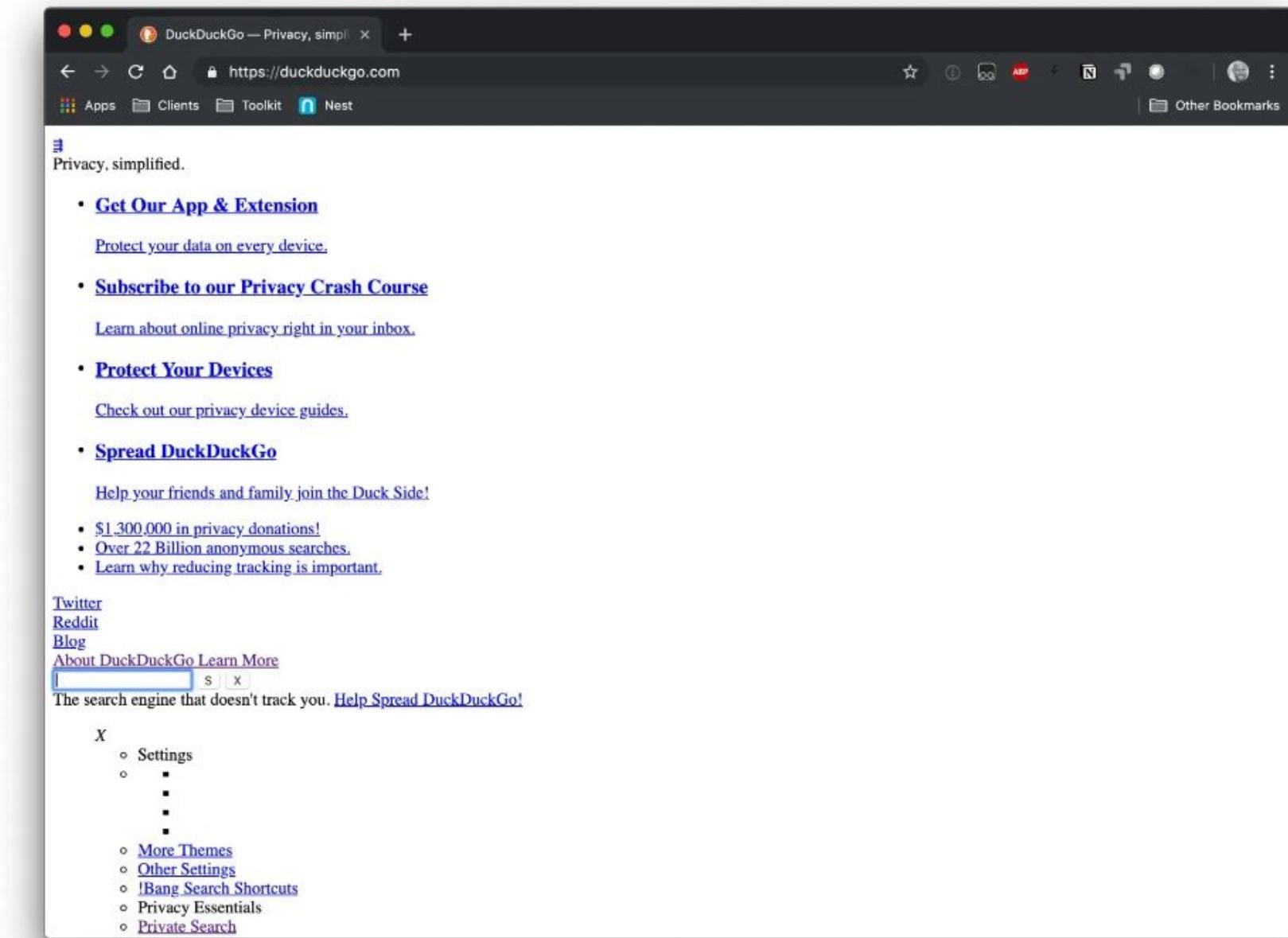
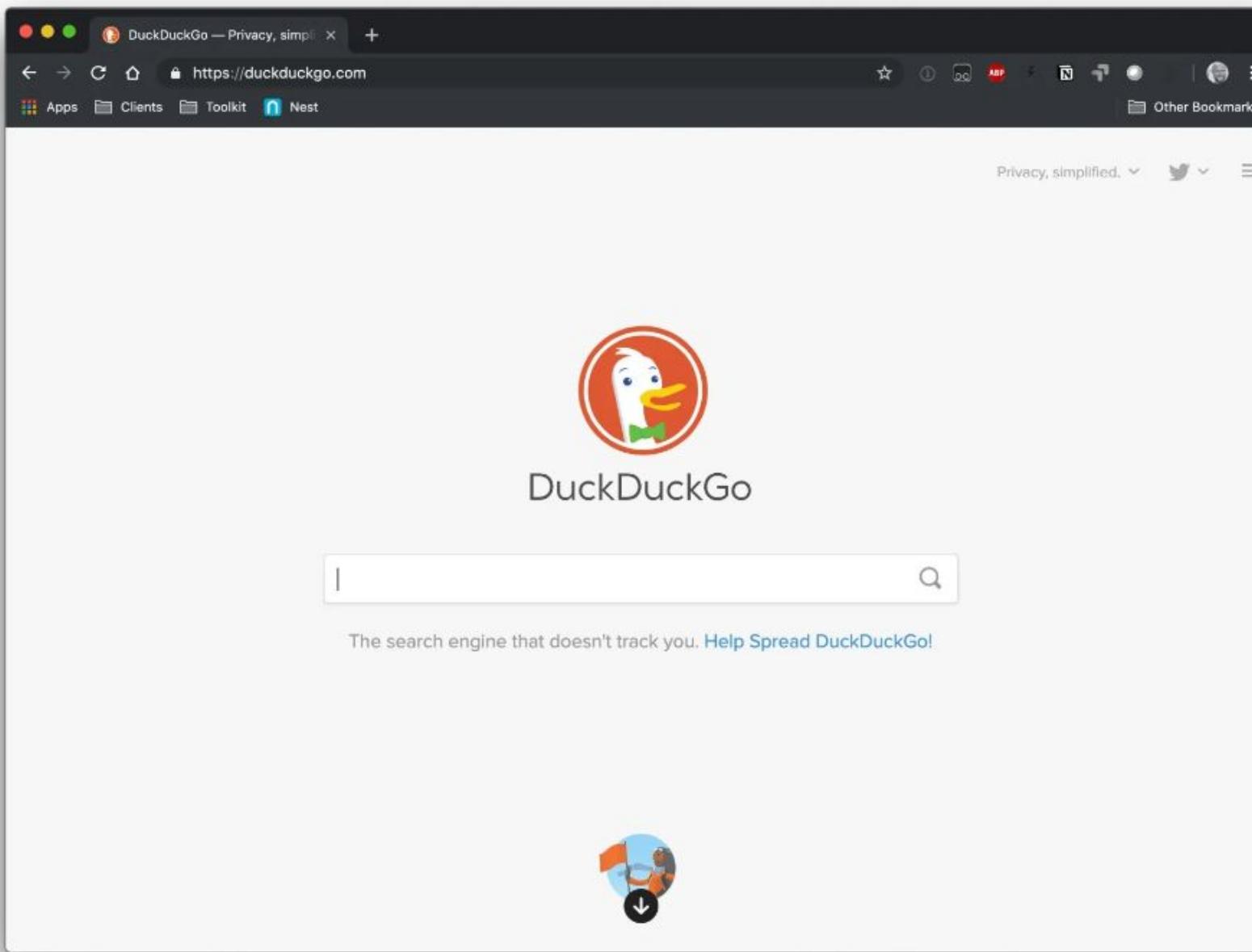
- Es un lenguaje de marcado, no de programación
- Se utiliza para estructurar los contenidos de una web



¿Qué es CSS?

keep coding

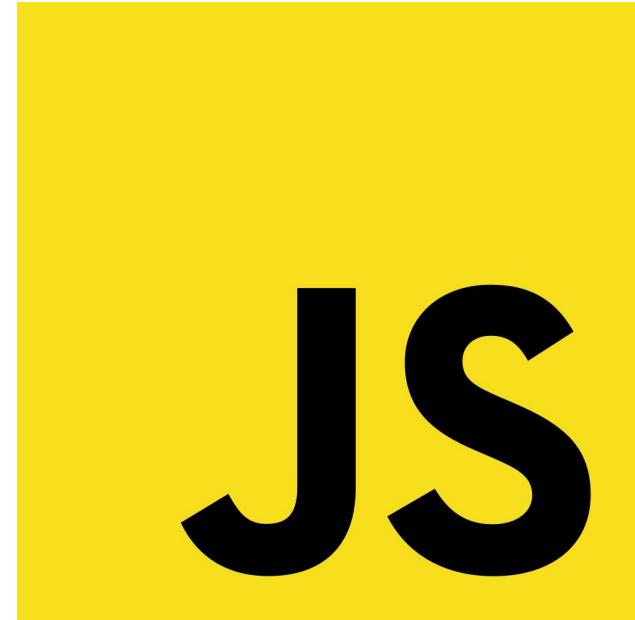
- Es un lenguaje de diseño gráfico, no de programación
- Permite definir y crear la presentación de un documento



¿Qué es JavaScript?

keep coding

- . Un lenguaje de programación interpretado
- . Usa la especificación de ECMAScript
- . Se puede usar también fuera de un navegador (Node)



¿Y TypeScript?

keep coding

- Extiende la sintaxis de JavaScript
- Es un JavaScript vitaminado
- Añade tipado estático
- Se puede usar también fuera de un navegador (Node)
- Acaba generando código JavaScript



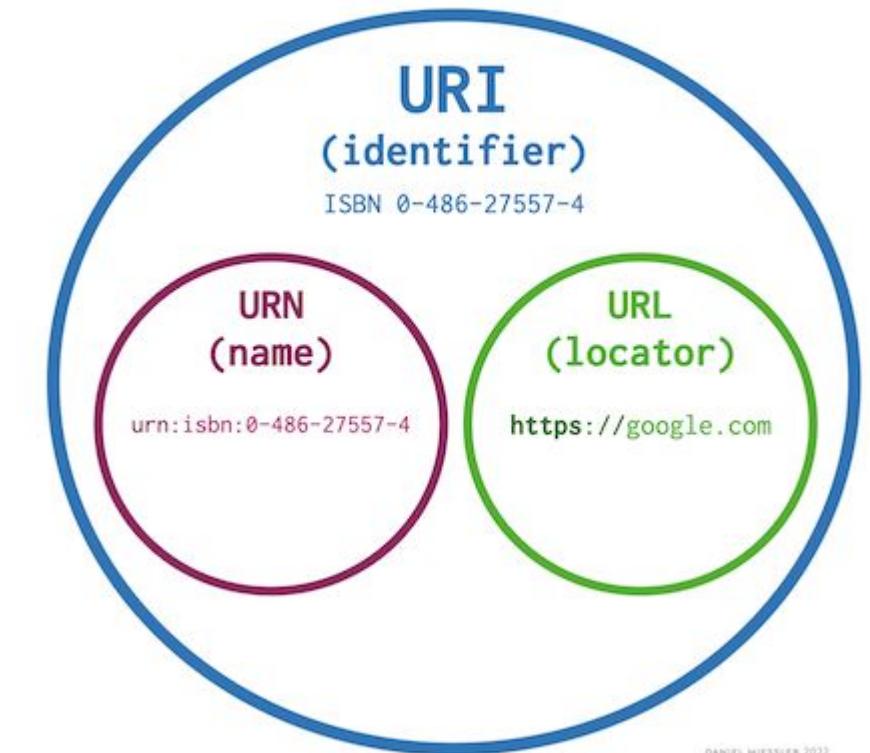
URI, URL y URN

keep coding

- **URI** es una cadena de caracteres que identifica de manera única un recurso en Internet: *keepcoding.io*
- **URL** es un tipo de URI que además dice cómo acceder a ese recurso (con el nombre de protocolo): *https://keepcoding.io*
- **URN** es un tipo de URI que se usa para referenciar recursos con un esquema específico: *urn:isbn:1-234-56789-0*

Tu nombre es un URI o URN, pero tu nombre junto con tu dirección, sería una URL.

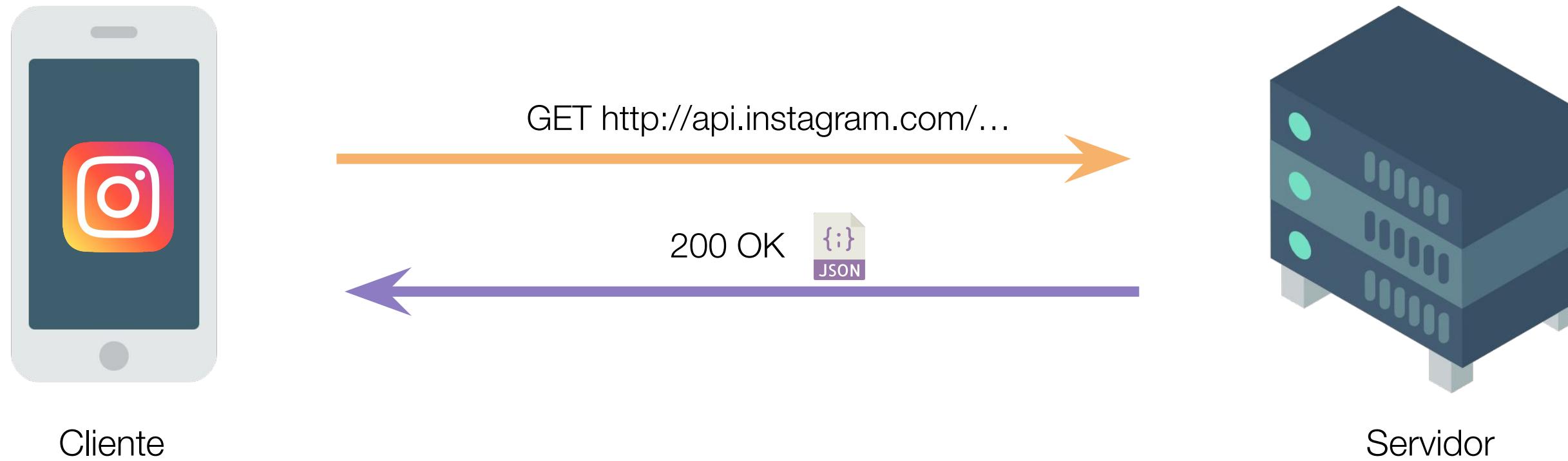
<https://danielmiessler.com/study/difference-between-uri-url/>



¿Cómo funciona una app móvil?

¿Cómo funciona una app?

keep coding



**Las apps hacen peticiones al servidor por HTTP/HTTPS,
pero no en formato HTML con los datos
(HTML es para los navegadores).**

**Los datos los suelen recibir en formato JSON
(aunque existen otros como XML).**

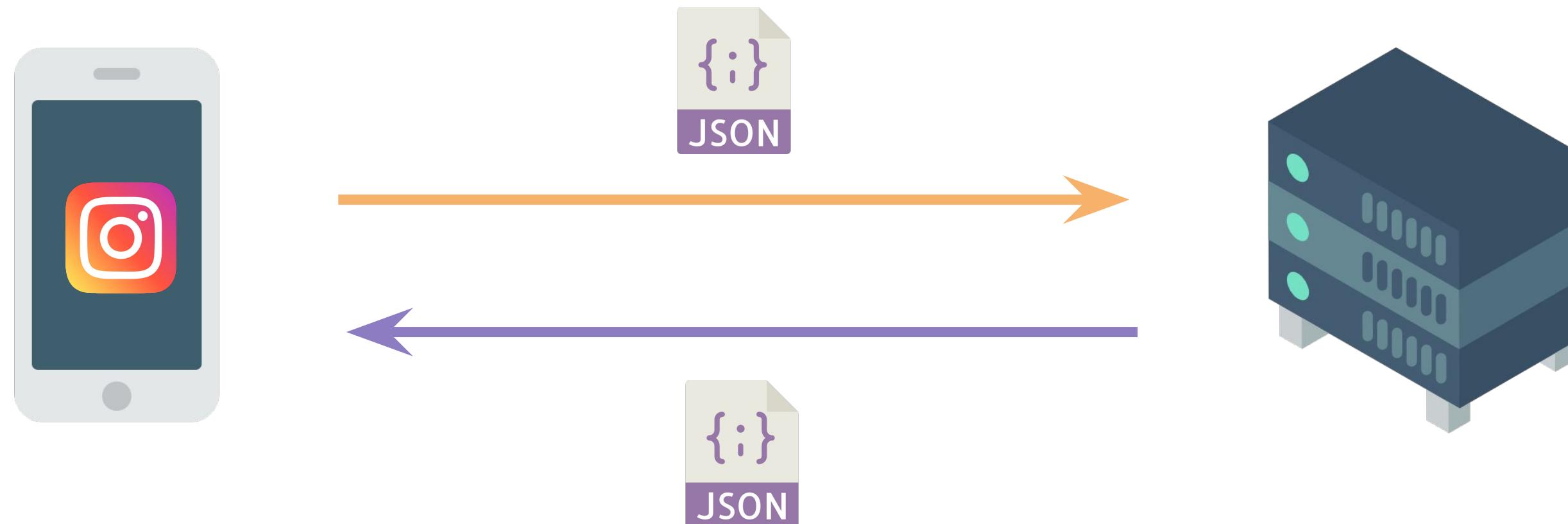
¿Qué es JSON?

keep coding

- Es un formato de intercambio de datos
- Se basa en la notación de objetos literales JavaScript
- Pero JSON no es JavaScript
- Nació para sustituir a XML (otro formato de intercambio de datos)
- Compatible con cualquier lenguaje de programación
- Con JSON, dos aplicaciones creadas en lenguajes de programación distintos pueden comunicarse entre sí

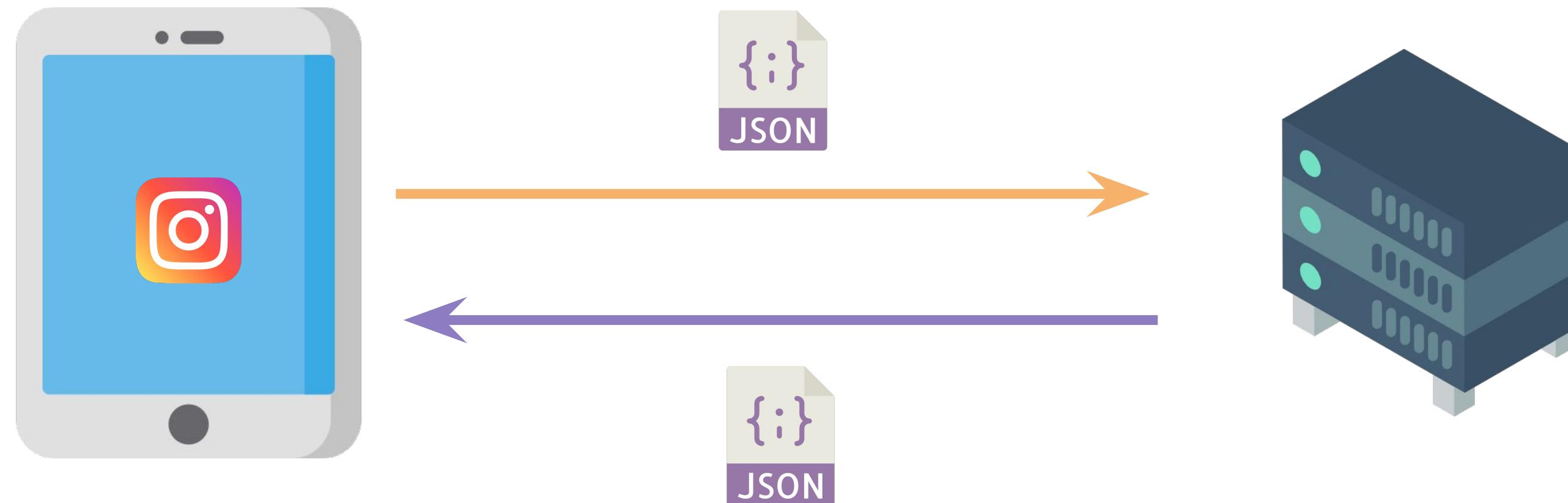
Comunicación app móvil con servidor

keep coding

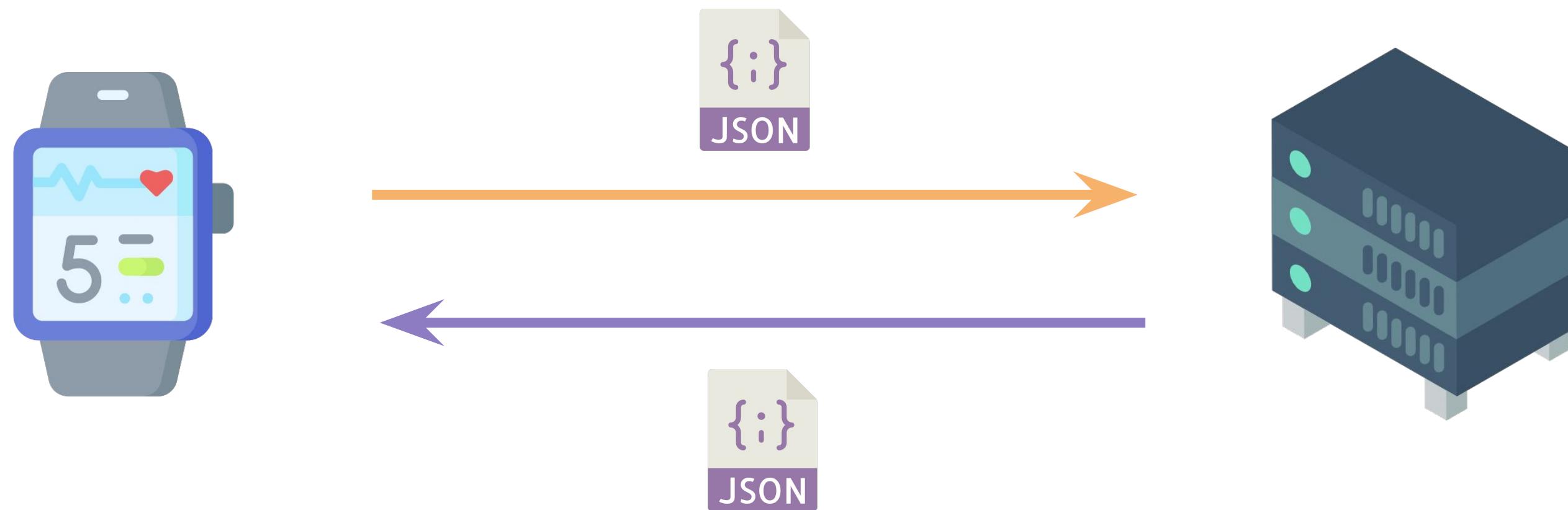


Comunicación app móvil con servidor

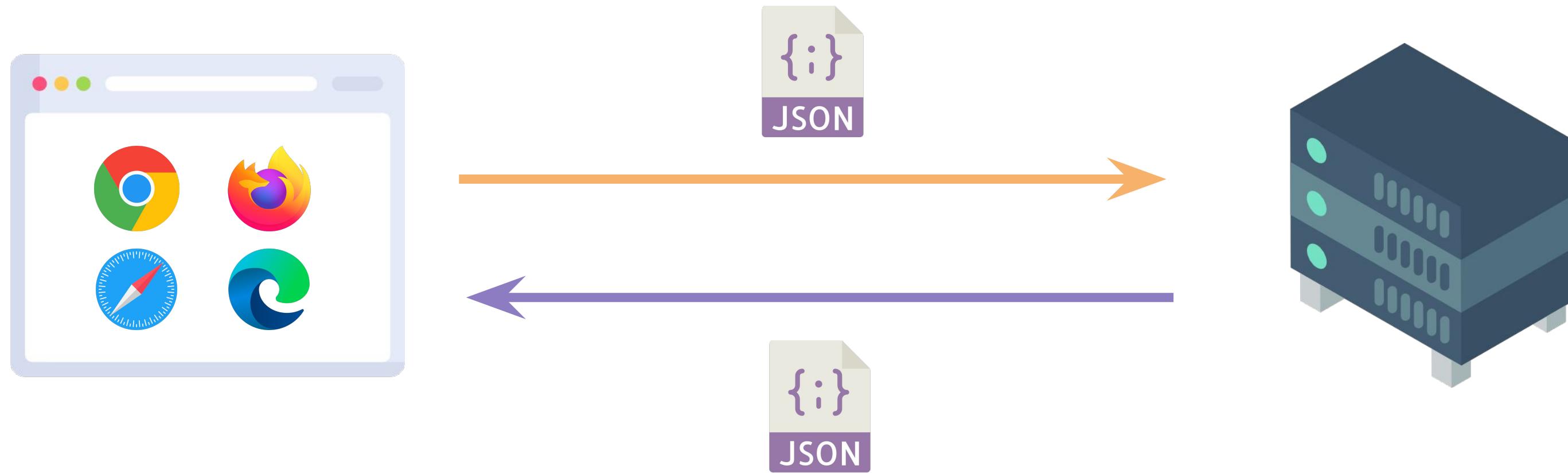
keep coding



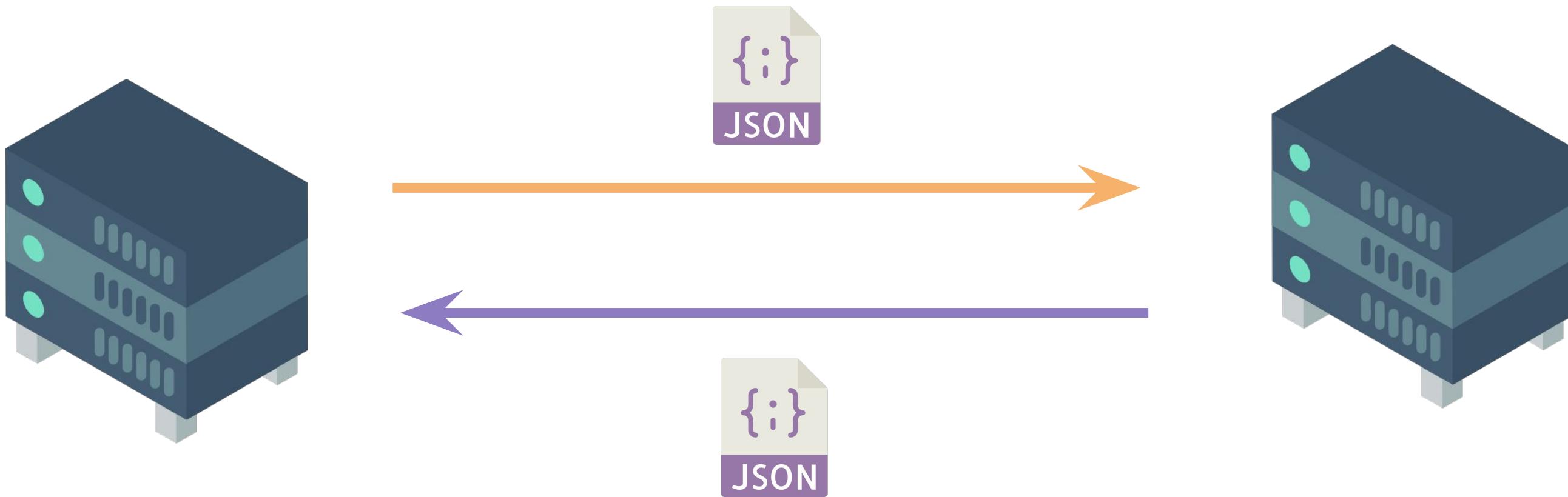
Comunicación app watch con servidor



Comunicación app web con servidor

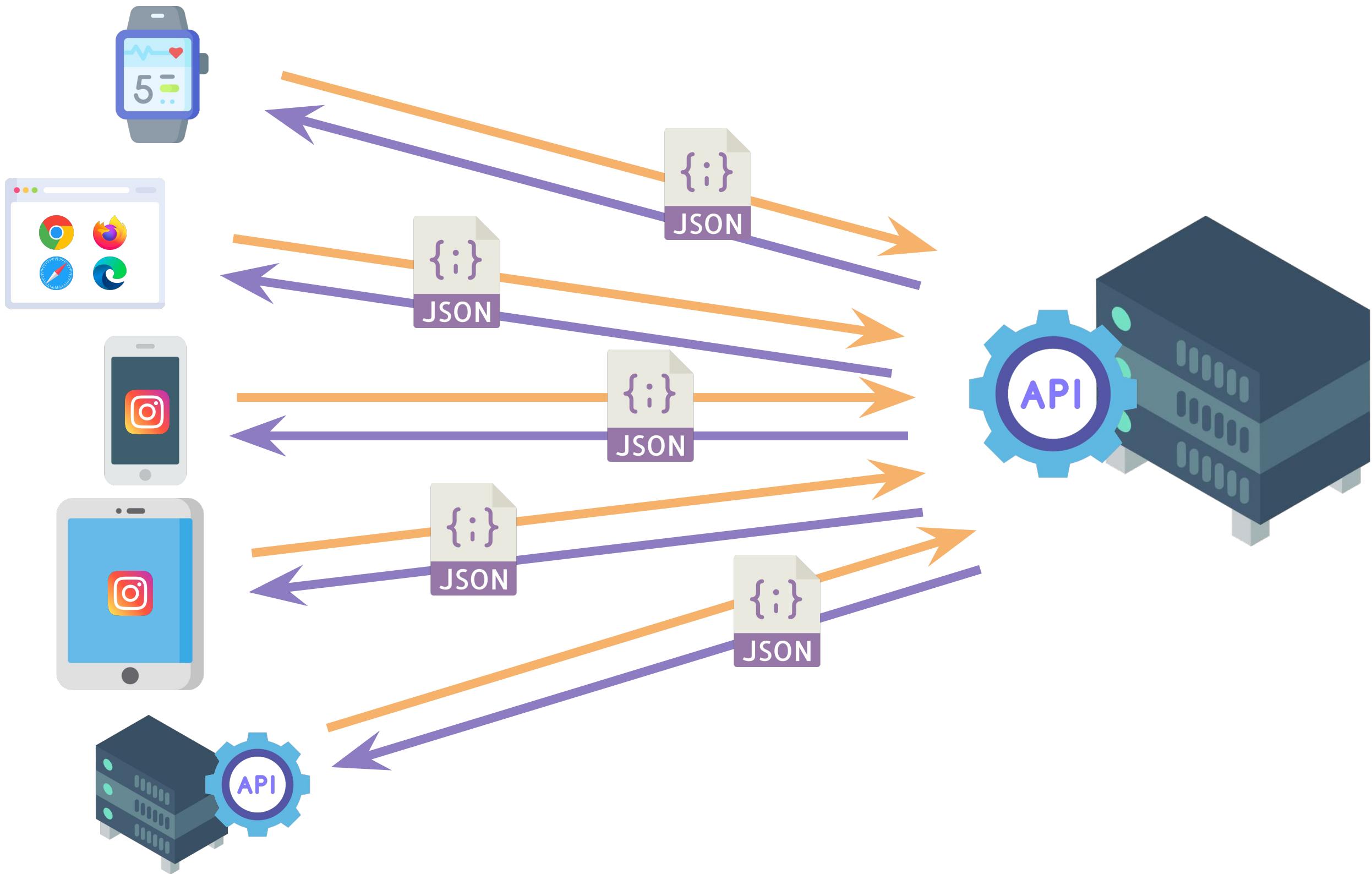


Comunicación servidor con servidor



Comunicación servidor con servidor

keep coding



¿Qué es un API REST?

keep coding

- API significa “Interfaz de programación”
- Los parámetros de una función o un método, es un API
- Es la vía de comunicación con un programa
- Un API REST, es una vía de comunicación para comunicarnos con un programa que está en un servidor usando HTTP

```
function registerUser(username, password) {  
    ...  
}  
  
registerUser("Alberto", "supersegura")
```

Ejecución de una función en JavaScript

```
POST http://service.com/api/register-user  
  
{  
    "username": "Alberto",  
    "password": "supersegura"  
}
```

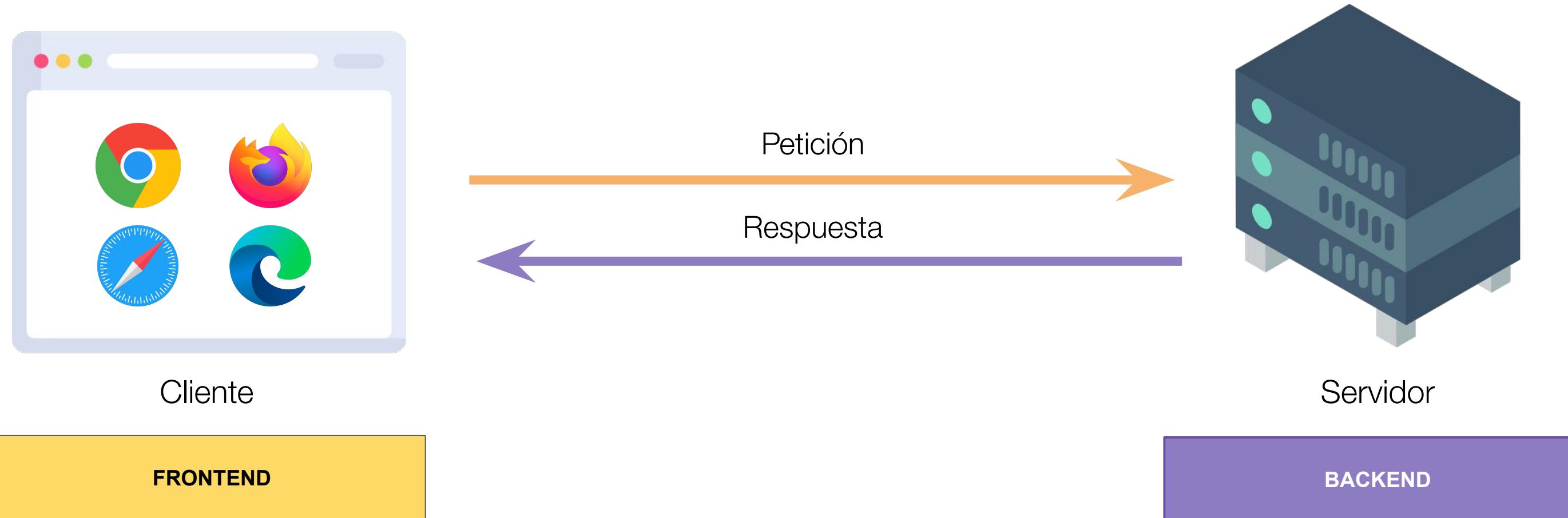
Ejecución de una petición a un API REST

keep coding

Frontend y Backend

Frontend y backend

keep coding



El frontend, es la parte de una plataforma que interactúa con el usuario y actúa como cliente.

Es lo que los usuarios usan directamente.

El backend, es la parte de la aplicación que se ejecuta en el servidor (o servidores).

Es la parte “que no se ve”.

Frontend y backend

keep coding



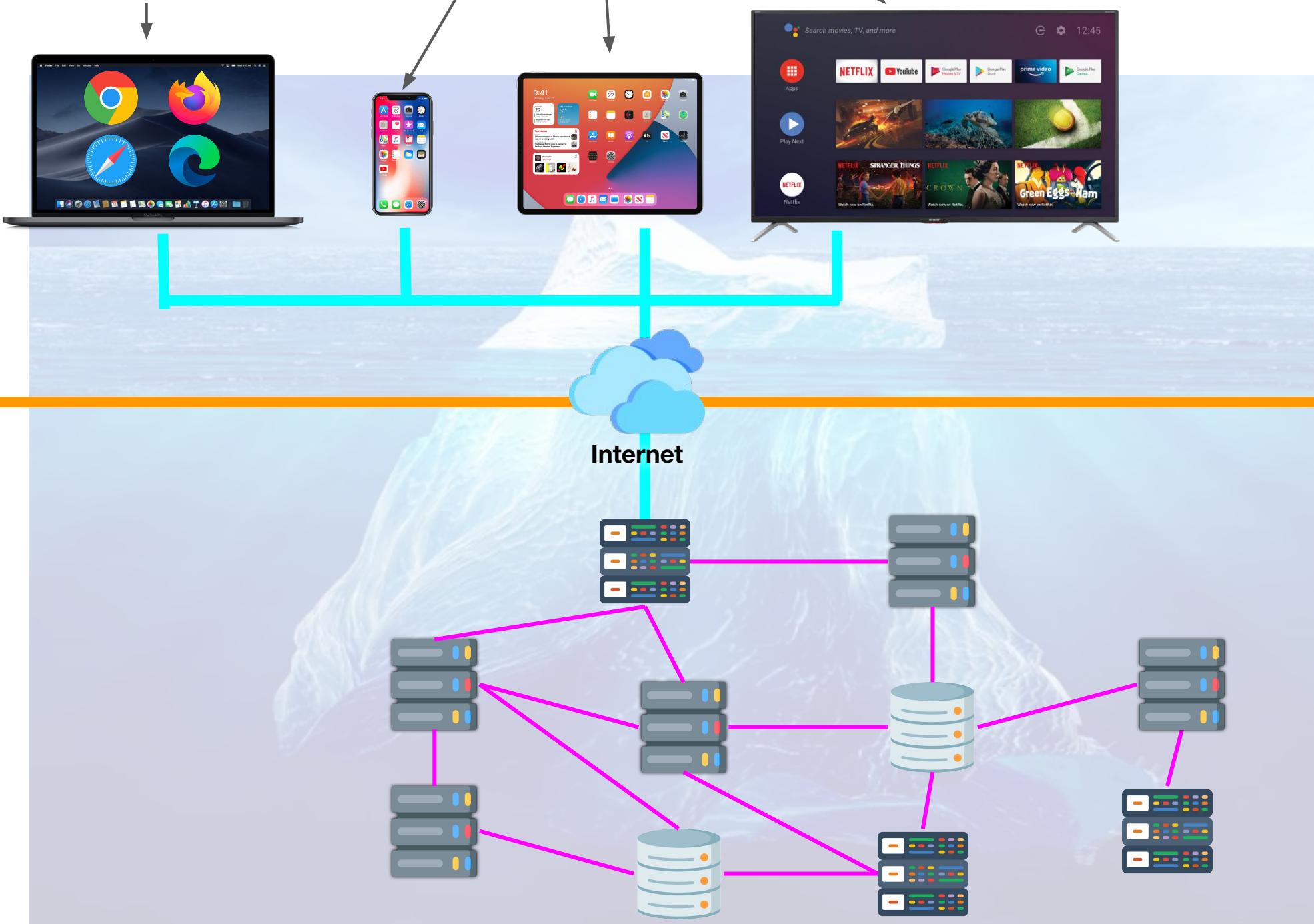
Frontend y backend

keep coding

En web, el frontend es lo que se ejecuta en el navegador.

En móviles, tablets, relojes, TVs el frontend son las apps.

FRONTEND



El frontend son las aplicaciones o webs que usa el cliente.

En frontend, **sólo podemos usar HTML, CSS y JavaScript**.

BACKEND

El backend, está formado por un conjunto de programas que colaboran entre sí para dar servicio a los frontends.

Podemos crear backends con **cualquier lenguaje de programación**.

Tecnologías frontend

Si sólo podemos usar HTML, CSS y JavaScript....

- **¿Qué pasa con TypeScript?**
 - TypeScript realmente se acaba convirtiendo en JavaScript.
- **¿Qué pasa con React, Angular o Vue?**
 - Son librerías/frameworks de JavaScript, por tanto, al final lo que generan es código JavaScript ya sólo JavaScript funciona en los navegadores.

Librería? Framework? Diferencias?

keep coding

- Una librería es una herramienta que tú utilizas como quieras.

Es como un hacha, que es para cortar madera, pero si quieres, puedes clavar clavos con ella.

- Un framework es una herramienta que te obliga a funcionar como dicen sus instrucciones.

Una motosierra sirve también para cortar madera, pero sólo funcionará si la pones en marcha como te dice el manual de instrucciones. Y...no creo que puedas clavar un clavo con una motosierra.

¿React, Angular o Vue?

keep coding

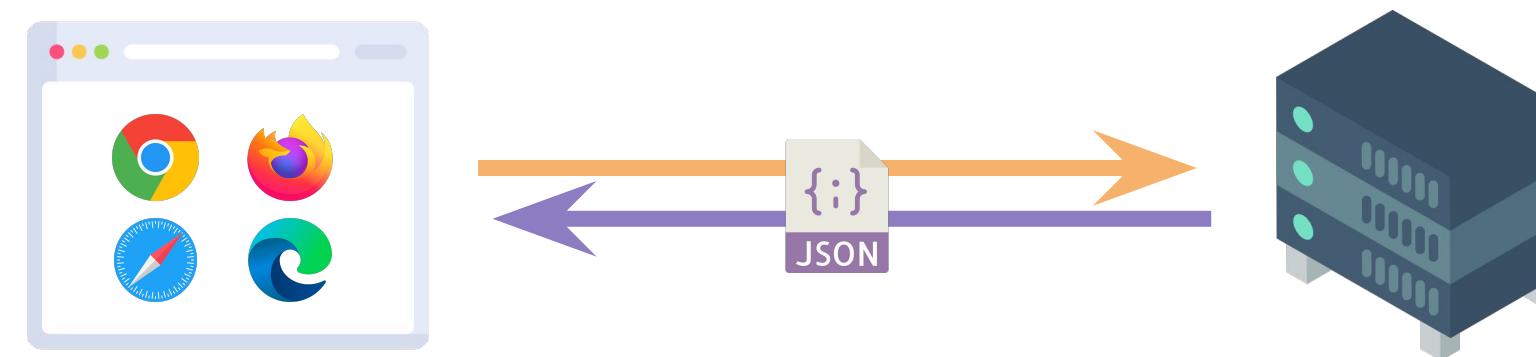


Librería	Framework	Librería
JavaScript	TypeScript	JavaScript
Facebook	Google	Open
Más fácil	Difícil	Fácil

¿Qué es una Web App?

keep coding

- Una web app, es una página web que se carga una sola vez y, tras la primera carga, usando JavaScript sólo actualiza porciones de la pantalla.
- Cuando hacemos click en un enlace, no se recarga la página web por completo.
- Las SPA (Single Page Application), son Web Apps que simulan la navegación de una web normal, pero en realidad siempre estás en la misma página.



¿Y una Progressive Web App?

- Una Progressive Web App (PWA), es una aplicación web que además los navegadores permiten instalarla como si fuera una aplicación “normal”.
- Las PWA se crean usando HTML, CSS y JavaScript, porque al fin y al cabo son páginas web.

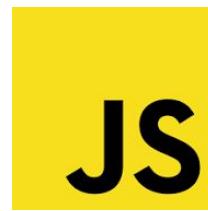
PWA

Tecnologías Backend

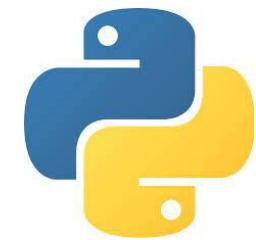
keep coding



+



JavaScript



Python



PHP



Java



Ruby



Go



C#



Swift



Express

django



spring

RAILS

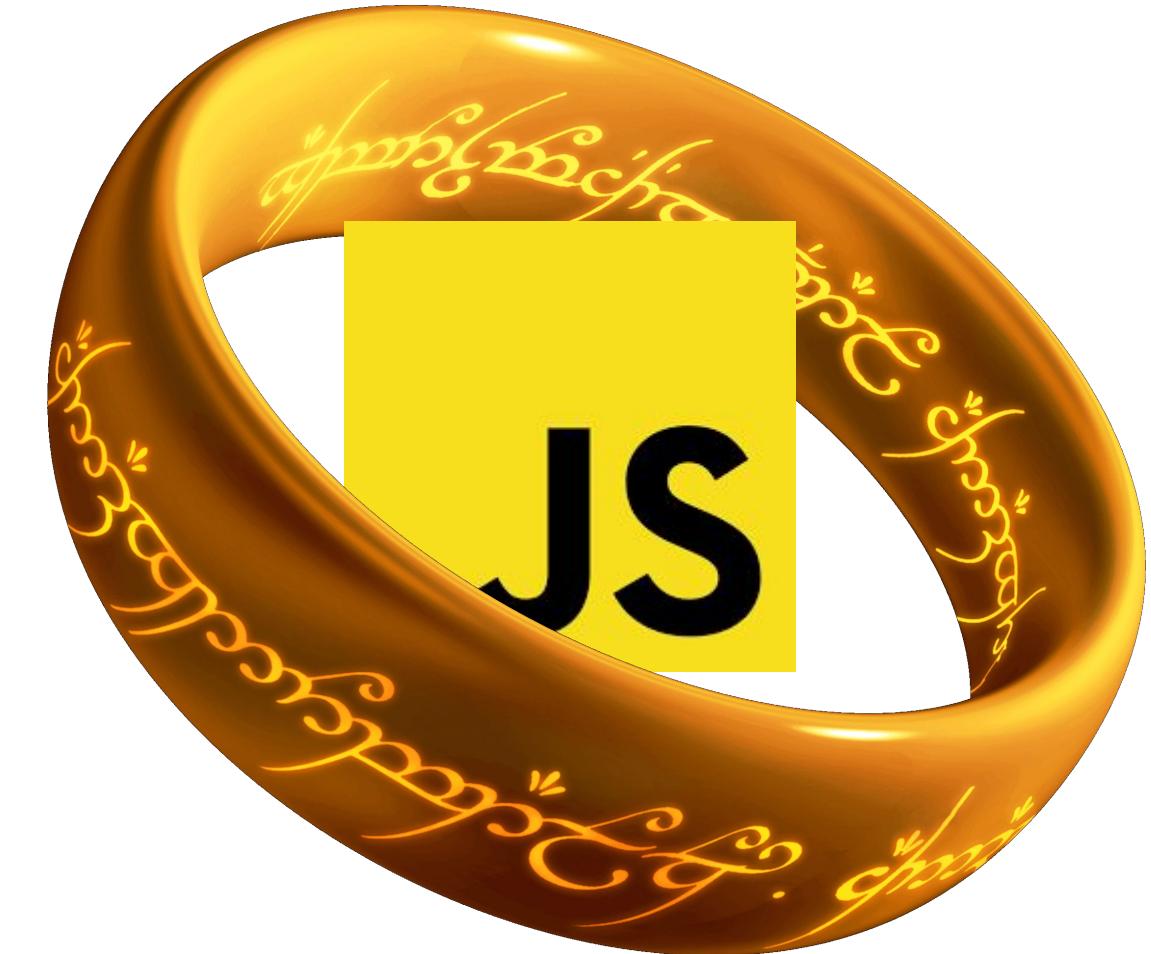


ASP.NET

VAPOR

Espera, JavaScript en frontend...y en backend?

keep coding



Un lenguaje para dominarlos a todos

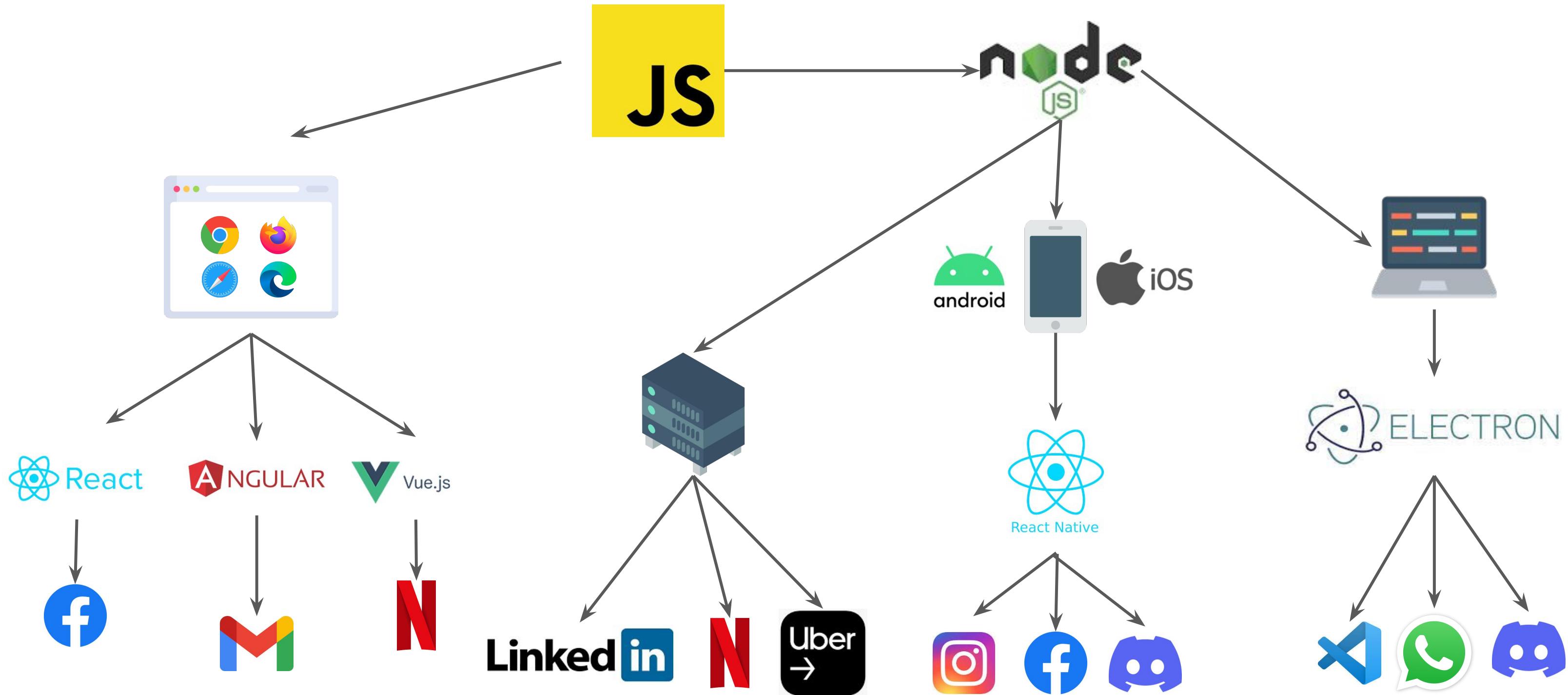
Node: JavaScript fuera del navegador



- En 2009, a Ryan Dahl se le ocurrió sacar el motor V8 de Chrome, fuera de Chrome para ejecutar JavaScript fuera del navegador.
- El resultado de esto fue NodeJS
- A partir de entonces, JavaScript se ha convertido en uno de los lenguajes más usados en todos lados.
- ¿Qué es Deno? (noDe, al revés). Es un proyecto similar a Node.

JavaScript: un lenguaje para dominarlos a todos

keep coding

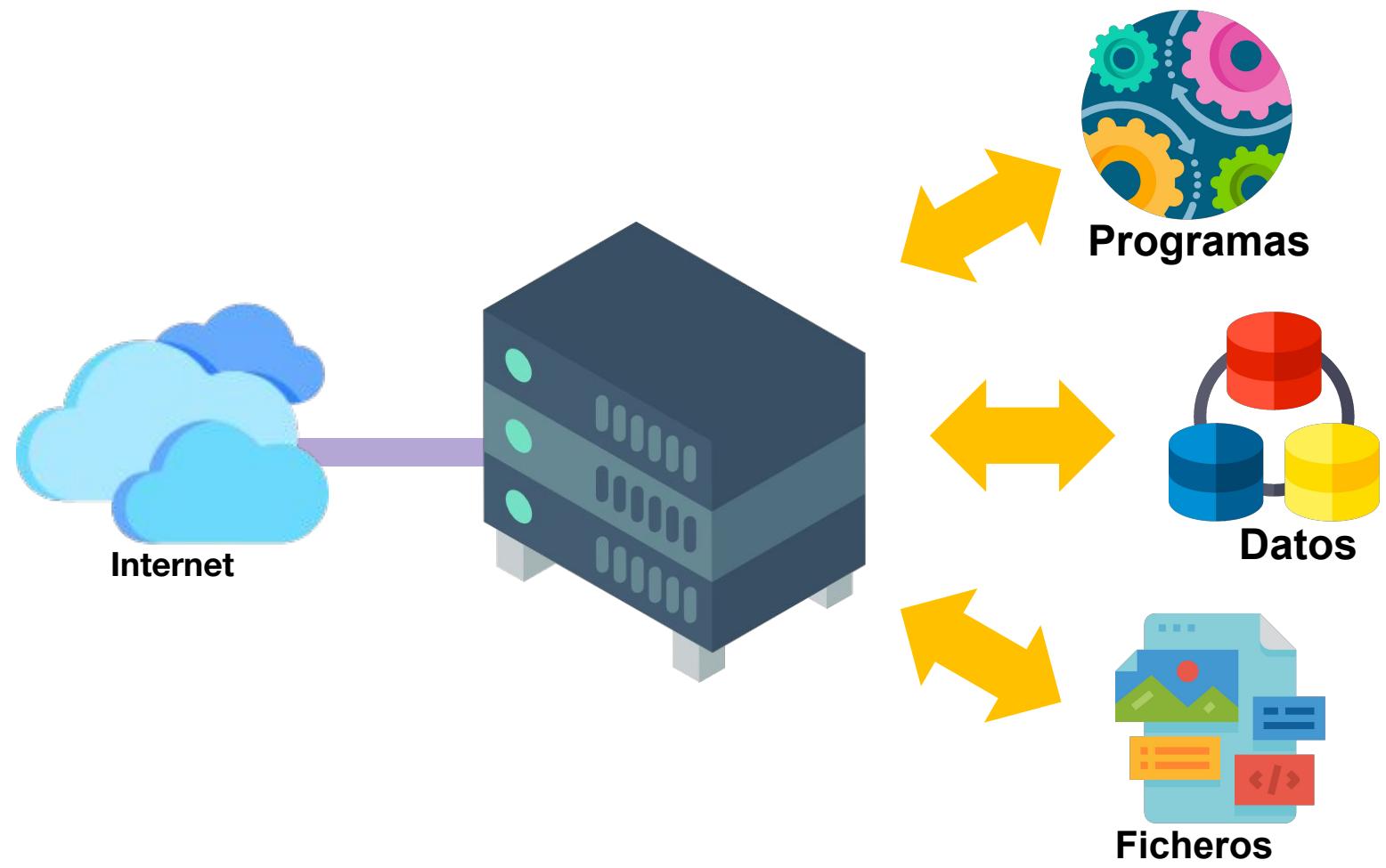


¿Cómo funciona el backend?

¿Cómo funciona el backend?

keep coding

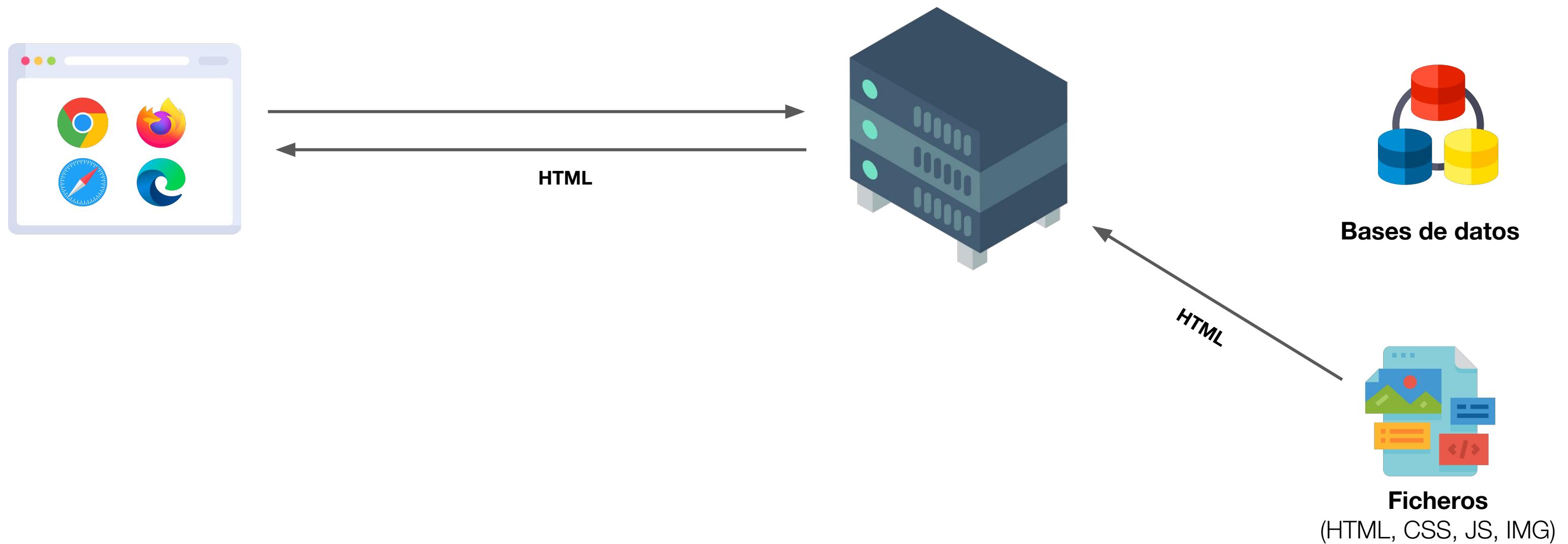
- Un servidor es un ordenador que siempre está funcionando y algunos accesibles por internet para dar servicios.
- Estos ordenadores almacenan datos, ficheros (imágenes, vídeos, etc.) y ejecutan programas.
- Estos programas, son “software de backend” o programas backend.
- Esos programas se encargan de gestionar los datos (usando bases de datos) y recursos (archivos) que hay en el servidor.



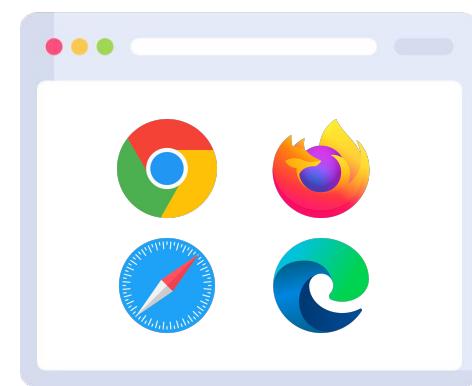
OK, ¿pero cómo funciona la web?



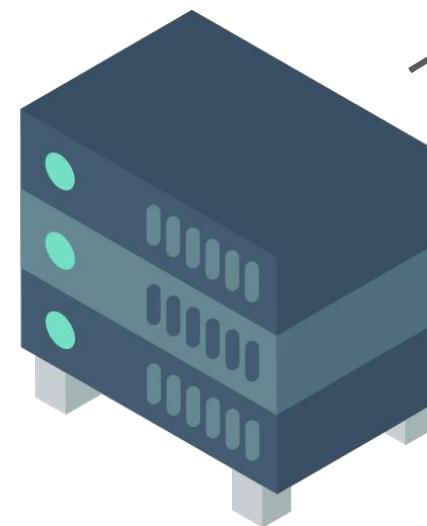
Static Website



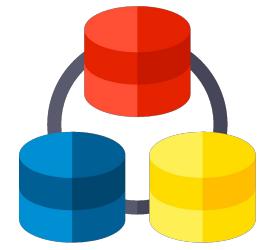
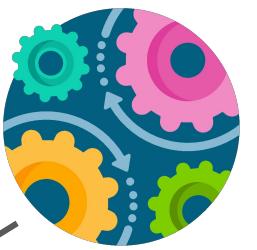
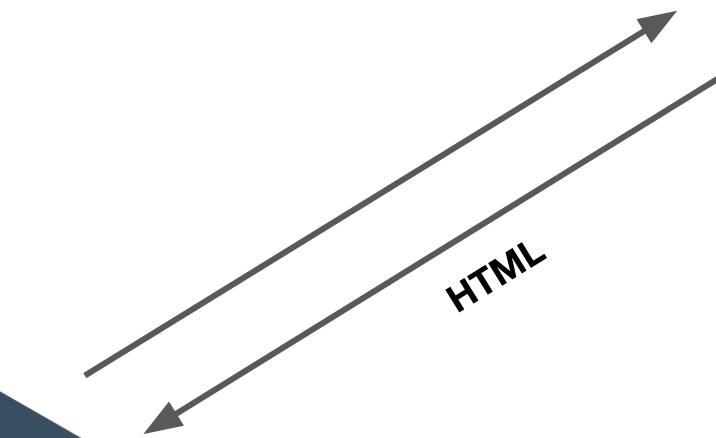
Server Side Rendering



HTML



HTML



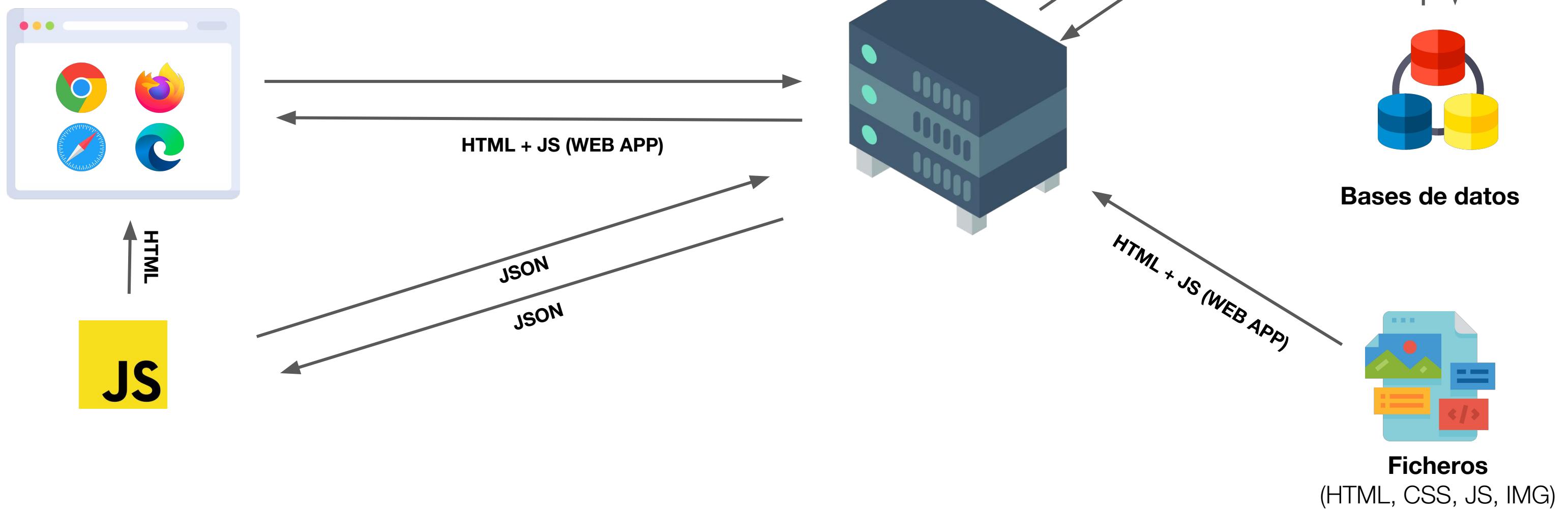
Bases de datos



Ficheros

(HTML, CSS, JS, IMG)

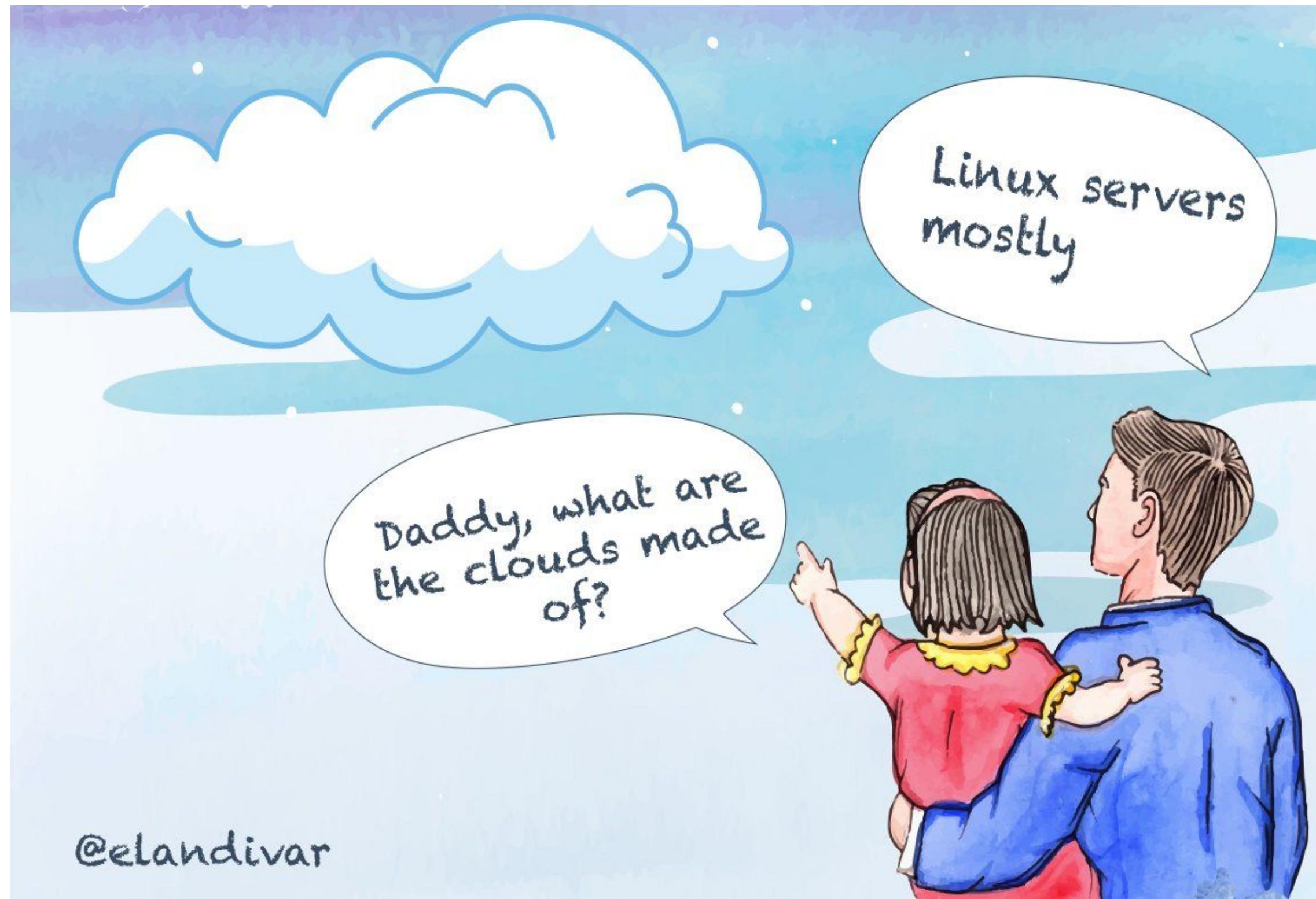
Client Side Rendering (Web App)



¿Qué es el Cloud Computing?

¿Qué es el Cloud Computing?

keep coding



¿Qué es el Cloud Computing?

keep coding

- Cloud computing es la disponibilidad bajo demanda de recursos de computación como servicios a través de Internet.
- Evita que las empresas tengan que encargarse de aprovisionar, configurar o gestionar los recursos (servidores) y permite que paguen únicamente por los que usen.

Ejemplo:

- Necesito un servidor de 1 CPU, 4 GB de RAM y 120 GB de disco.
- Oh vaya, el servidor se me queda pequeño
- Sin la nube, tendría que preparar otro servidor y migrar información
- Con la nube, en cuatro clicks mi servidor ha pasado a tener 2 CPU, 8GB de RAM y 240 GB de disco.

¿Qué son las cookies?

¿Qué son las cookies?

keep coding

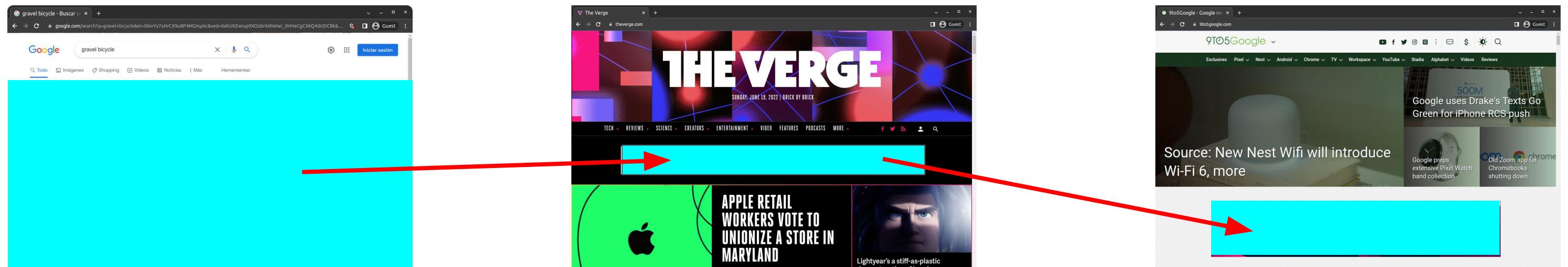
- Son archivos que las webs “dejan” en nuestro ordenador y que cada vez que volvemos a visitar una web, nuestro ordenador se los vuelve a enviar.
- No son malas...si no se usan para hacer el mal.
- Un buen uso: saber que el usuario que visita la web ya se ha autenticado para no volver a pedirle el login.
- Entonces...¿por qué hay tanto rollo con las cookies?



Cookie tracking

keep coding

- Las cookies también se pueden utilizar para trackear a un usuario a través de diferentes webs gracias a las tecnologías de anuncios.
- Al visitar una web, algunas usan tecnologías de otras web (Google) para mostrar anuncios o contabilizar visitas.
- Esas otras webs te guardan una cookie también que les permite trackearte por todas las webs que usen su tecnología (la mayoría).





keep coding



GRACIAS
www.keepcoding.io

