UT 1 - Tecnologías en Aplicaciones Web

Marina Navarro Pina

"I'm no expert, but I think it's some kind of cyber attack!"

World Wide Web

Sistema de distribución de documentos de hipertexto o hipermedia interconectados y accesibles a través de Internet. ¿Con qué aplicación?

Un usuario visualiza sitios web compuestos de páginas web que pueden contener texto, imágenes, vídeos u otros contenidos multimedia, y navega a través de esas páginas usando hiperenlaces. ¿Quién estandariza todo esto?

Componentes de la WWW

1. **Hipertexto -** lenguaje de marcado
2. **HTTP** (*HyperText Transfer Protocol*) - protocolo de transferencia de datos
3. **Servidores web -** servidores que alojan los sitios y responden a las peticiones
4. **DNS** (*Domain Name System*) - nombres de dominio, traducción a IP
5. **Recursos web** - documentos, imágenes, vídeos, etc. alojados en servidores
6. **Navegadores** - aplicaciones clientes visualizar/navegar por el sitio

Cada recurso está identificado por una **URL** (*Uniform Resource Locator*).

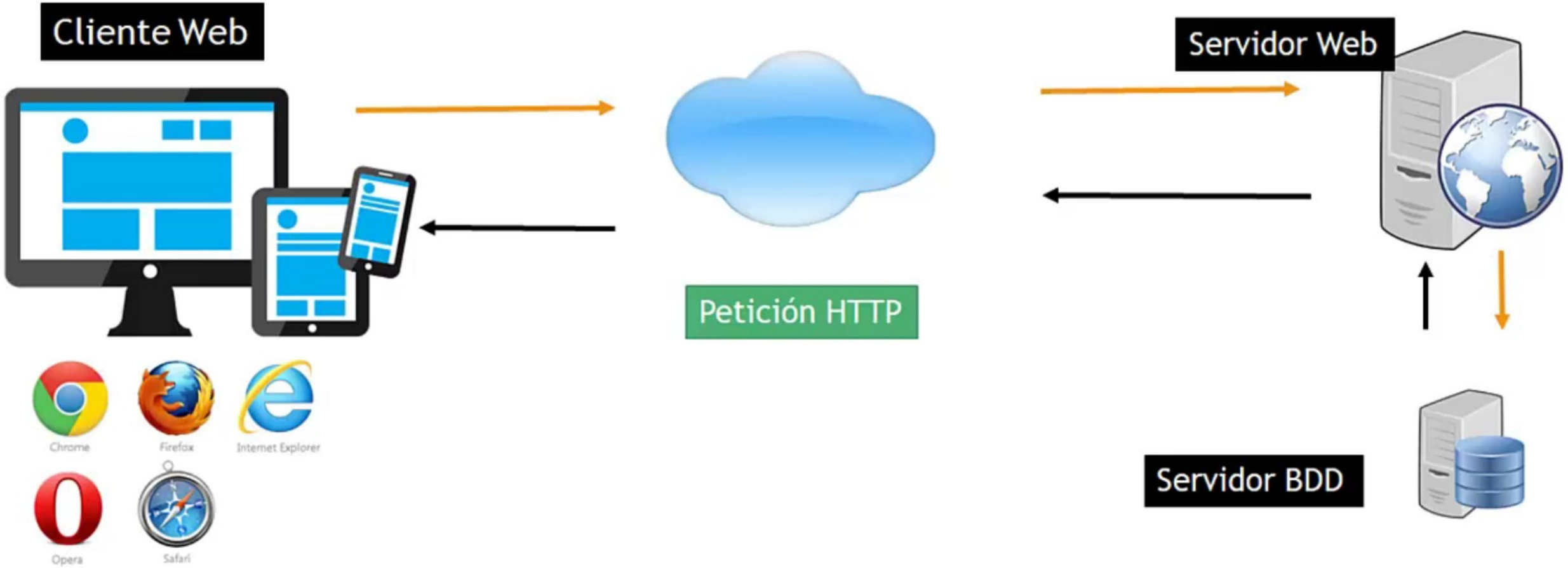
Aplicaciones Web

Aquella que se accede mediante el Protocolo de Transferencia de Hipertexto (HTTP o HTTPS).

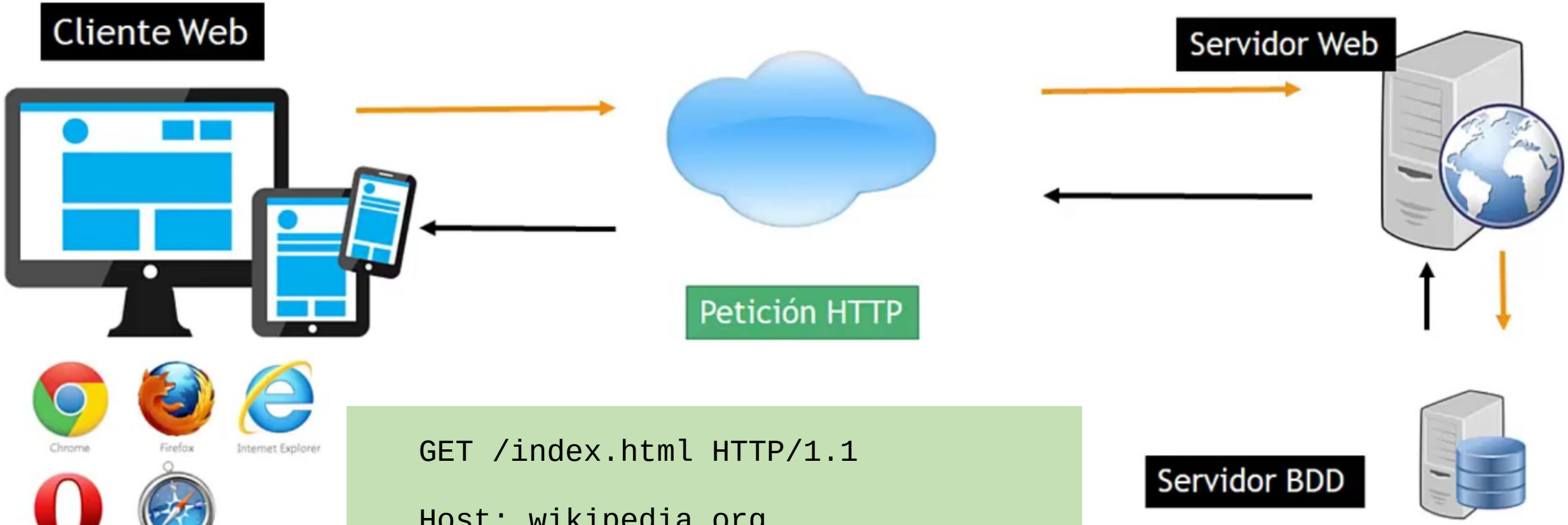
Están escritas en distintos tipos de lenguajes y se pueden ejecutar en muchos Sistemas Operativos.

El resultado se formatea en HTML y las entradas de datos se comunican mediante GET, POST y métodos similares.

ARQUITECTURA DE UNA APLICACIÓN WEB

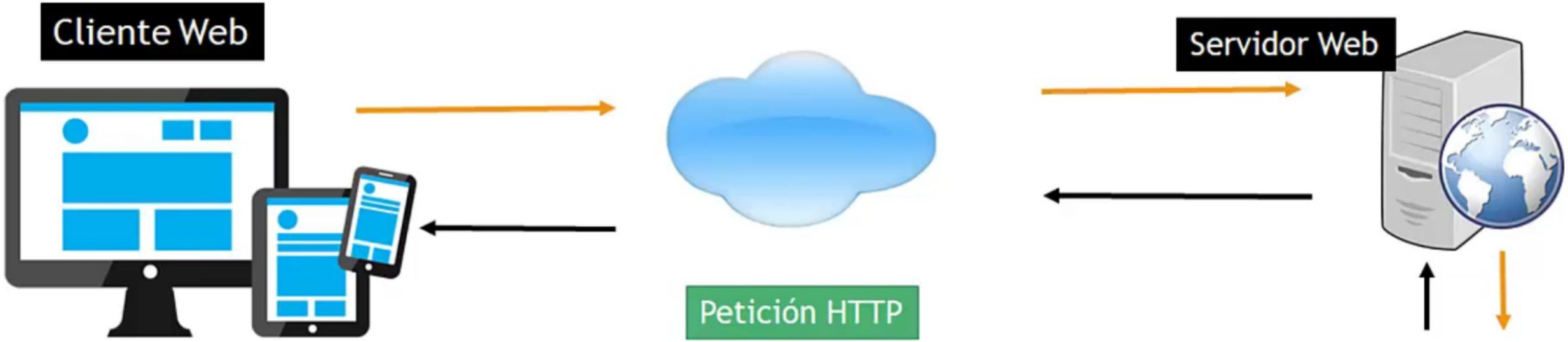


ARQUITECTURA DE UNA APLICACIÓN WEB



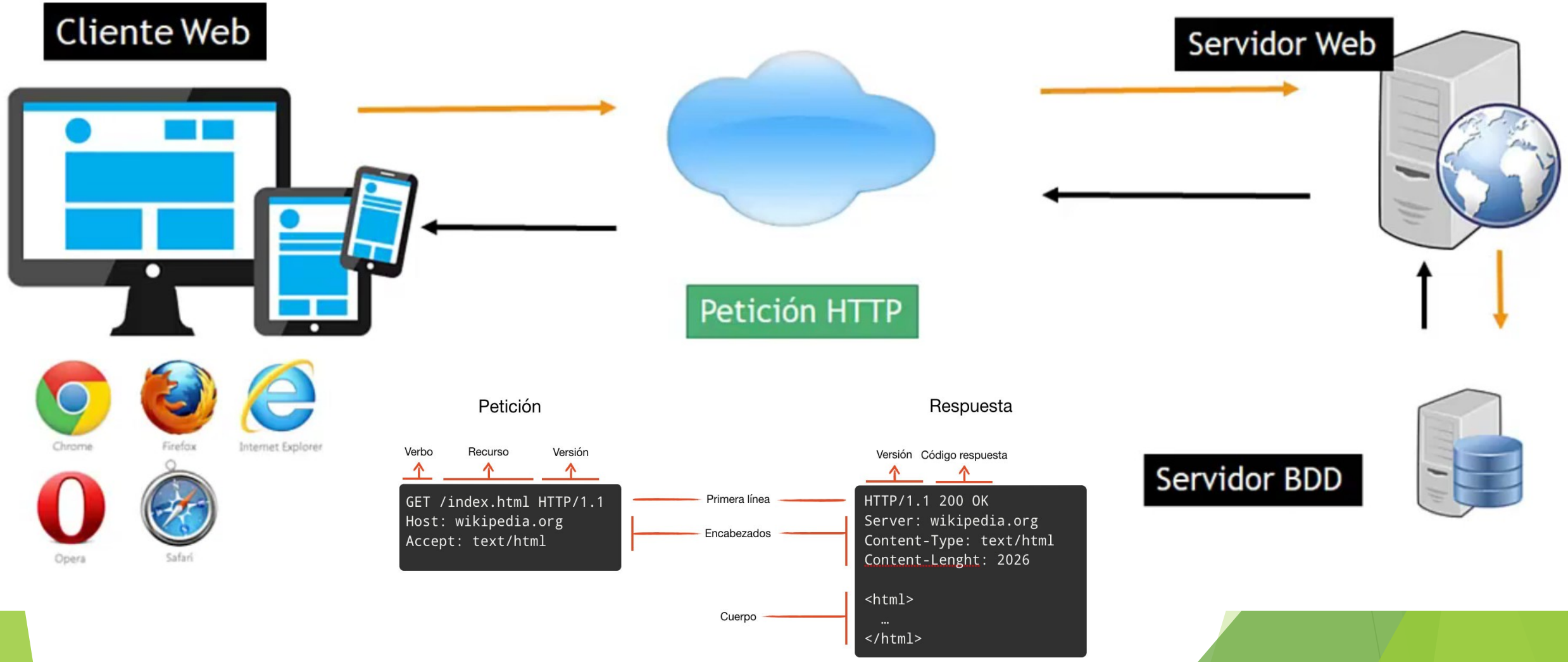

```
GET /index.html HTTP/1.1

Host: wikipedia.org

Accept: text/html
```

ARQUITECTURA DE UNA APLICACIÓN WEB

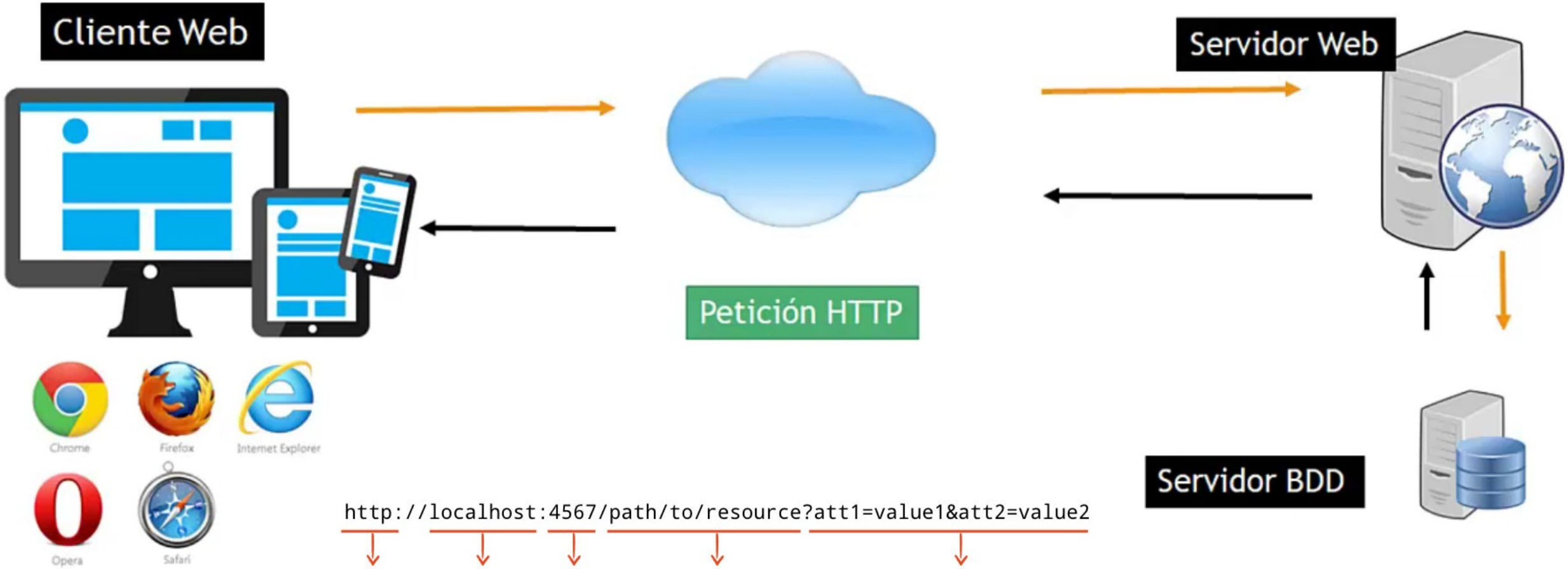



```
HTTP/1.1 200 OK
Server: wikipedia.org
Content-Type: text/html
Content-Lenght: 2026
<html>
  …
</html>
```

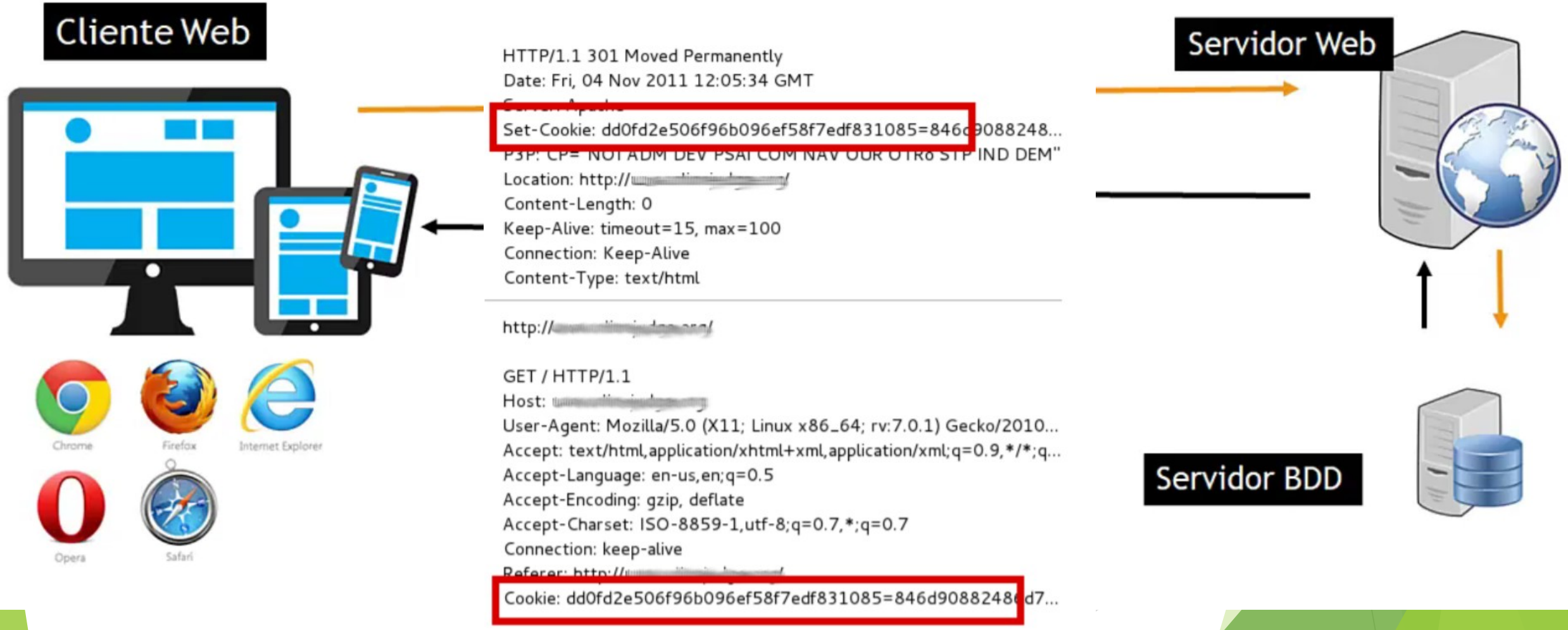
ARQUITECTURA DE UNA APLICACIÓN WEB



ARQUITECTURA DE UNA APLICACIÓN WEB



ARQUITECTURA DE UNA APLICACIÓN WEB



Arquitectura cliente-servidor

La arquitectura cliente-servidor se basa en el modelo de comunicación petición-respuesta.

En el lado servidor (server side) se ejecutan las aplicaciones que generan las páginas web dinámicas.

En el lado cliente (client side) se ejecutan aplicaciones que se ejecutan en el navegador web y que interactúan con el usuario.

¿SPAs? ¿AJAX?

Server-side

El servidor genera las páginas web dinámicas y las envía al cliente. El cliente recibe la página y no tiene que hacer nada más que mostrarla.

Los elementos que encontramos en el lado servidor son el hardware (servidores y elementos de red), el software (servicios web como Apache, Nginx, etc) y lenguajes del lado servidor (PHP, Perl, C, Python, Javascript con NodeJS, etc).

Server-side

Tareas comunes en el lado servidor:

1) Acceder y guardar los datos - base de datos
2) Enviar correos electrónicos
3) Procesar archivos
4) Generar páginas web dinámicas
5) Validación de formularios - también en el cliente
6) Autenticación y autorización - parcialmente en el cliente
7) Procesar datos y realizar cálculos

Client-side

La lógica de la aplicación se ejecuta en el navegador web del cliente. El servidor envía al cliente los recursos necesarios para ejecutar la aplicación y el cliente se encarga de mostrar la interfaz de usuario y de interactuar con el usuario.

Ejemplos de tecnologías client-side son HTML, CSS y JavaScript.

Los elementos que encontramos en el lado cliente son el navegador web (Firefox, Chrome, Safari, Edge, Opera, …), lenguajes de marcas (HTML, CSS) y lenguajes de programación, sobre todo JavaScript.

Client-side

Tareas comunes en el lado cliente:

1) Interacción con el usuario - interfaz
2) Validación de formularios
3) Autenticación y autorización
4) Procesar datos y realizar cálculos

Javascript

Lenguaje de programación que se utiliza para crear aplicaciones web interactivas.

Es un lenguaje de programación interpretado y *débilmente tipado*. JavaScript se utiliza para añadir interactividad a las páginas web.

Limitaciones JS

- No puede leer ni escribir archivos en el disco duro del cliente
- No puede lanzar aplicaciones ni modificar las preferencias del navegador
- No puede enviar emails, transmitir en streaming, etc
- No puede acceder a la información de otros dominios (política del mismo origen)
- No puede manipular ventanas que no haya abierto
- Se ejecuta en el navegador del cliente, por lo que puede ser modificado por el usuario. *Esto puede ser un problema de seguridad si no se toman las medidas adecuadas*.
- Es un lenguaje interpretado, por lo que puede ser más lento que otros lenguajes de programación.
- JavaScript no es compatible con todos los navegadores.
- Depende del navegador del cliente, por lo que puede haber diferencias en la forma en que se ejecuta en diferentes navegadores. *Es importante probar la aplicación en diferentes navegadores para asegurarse de que funciona correctamente en todos ellos.*

Framework y librerías JS

Opciones que tenemos:

- Javascript "Vanilla": se refiere a JavaScript puro
- Bibliotecas: son colecciones de funciones y métodos que se pueden utilizar para realizar tareas comunes en JavaScript. Algunas bibliotecas populares son *jQuery* (facilita Ajax y la manipulación de DOM, se usa cada vez menos), *Bootstrap* (para mejorar el diseño), *Lodash*, *Moment.js*, etc.
- Frameworks: son conjuntos de herramientas y librerías que se utilizan para desarrollar aplicaciones web. Algunos de los frameworks más populares son *Angular, React, Vue.js, Ember.js*, etc.

Evolución aplicaciones web

Muchas opciones.

¿Cuál es la mejor? Dependerá de las necesidades de la aplicación. En todos los casos la aplicación tendrá que ser *responsive* para que se adapte a cualquier dispositivo (escritorio, móvil, tablet, …).

Framework y librerías JS

Opciones:

- Páginas web estáticas: no cambian
- Páginas web dinámicas: cambian en función de la interacción del usuario o de otros factores. Cambios en el server-side (acceso a datos, etc) o en el client-side
- Aplicaciones web: aplicaciones que permiten al usuario interactuar con ellas. La mayor parte de la programación se ejecuta en el server-side.
- Aplicaciones de una sola página (SPAs): se cargan una sola vez y que permiten al usuario interactuar sin tener que recargar. Gmail, Google Maps, Facebook, Twitter, etc.
- Aplicaciones web progresivas (PWAs): se comportan parecido a las aplicaciones nativas en los móviles. Twitter Lite, Flipkart, Starbucks, etc. - *Ionic*
- Aplicaciones web híbridas: aplicaciones que se ejecutan en el navegador web y que pueden también comportarse como aplicaciones nativas, aunque su rendimiento es menor. Para ello se utilizan herramientas como PhoneGap, Cordova, etc.

Despliegue aplicaciones

Ahora lo normal es utilizar herramientas llamadas “bundlers” que:

- unifican en uno todos los ficheros js y css
- minimizan el código js resultante lo que reduce el tamaño del fichero y mejora la velocidad de carga
- transpilan el código js a una versión que todos los navegadores entiendan
- eliminan código duplicado o innecesario de las librerías que se usan