



Fundamentos del Desarrollo Web

Yhoan Galeano

Agenda - Conceptos base

1. ¿Qué es el desarrollo web?
2. ¿Cómo funciona la web? Introducción a los protocolos HTTP, TCP/IP, DNS.
3. Arquitectura cliente-servidor: conceptos básicos y cómo se aplican en la web. (Arquitectura distribuida)
4. Navegadores web: historia, características y funcionamiento.
5. Tecnologías para el desarrollo de la web.

¿Qué es el desarrollo web?

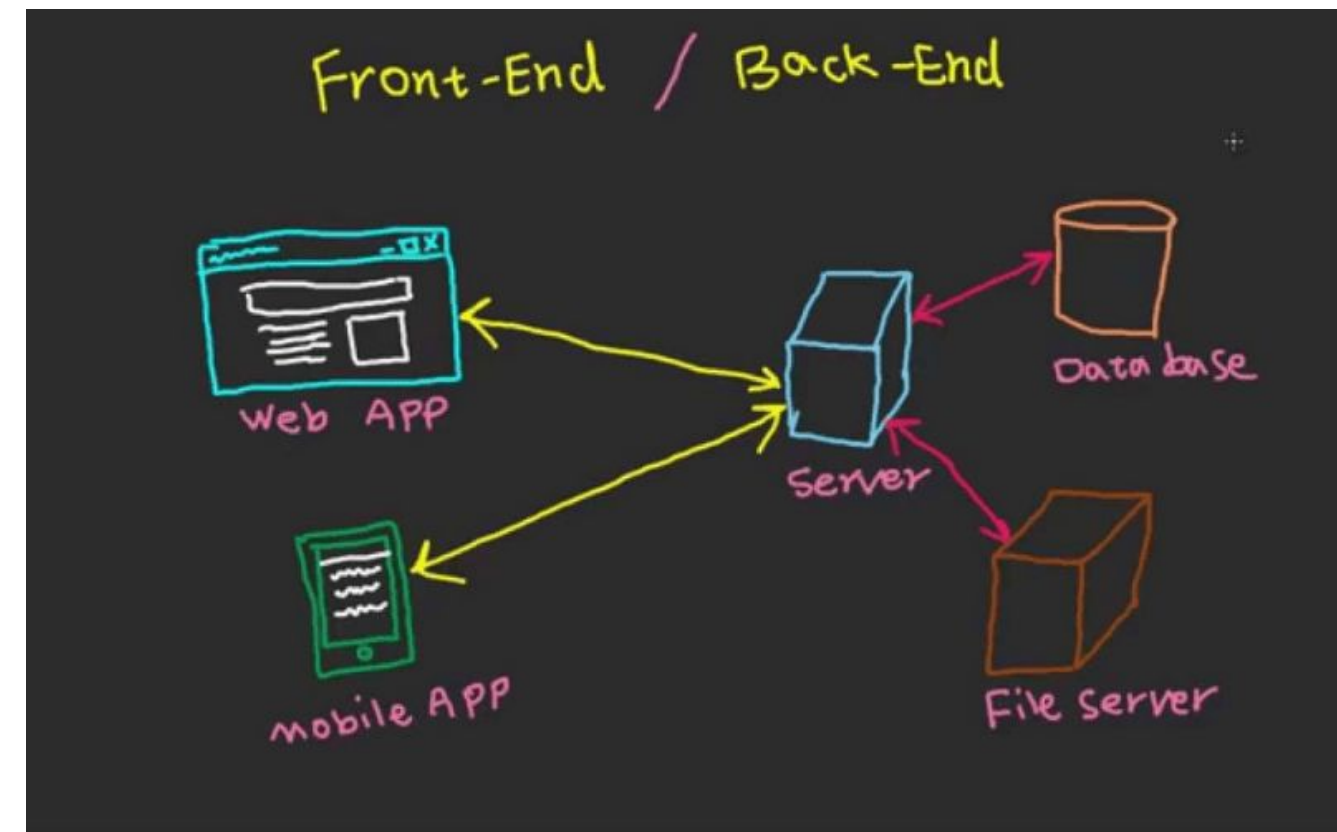
Proceso que combina múltiples habilidades para crear soluciones que permitan la interacción de usuarios por medio de un dispositivo con acceso a internet.



Front-end / Back-end

El desarrollo Front-end se encarga de la interfaz, la experiencia y la interacción del usuario.

El desarrollo Back-end se encarga de la lógica de la aplicación y la gestión de datos que responderá a las interacciones del usuario.



Empecemos a desarrollar ...

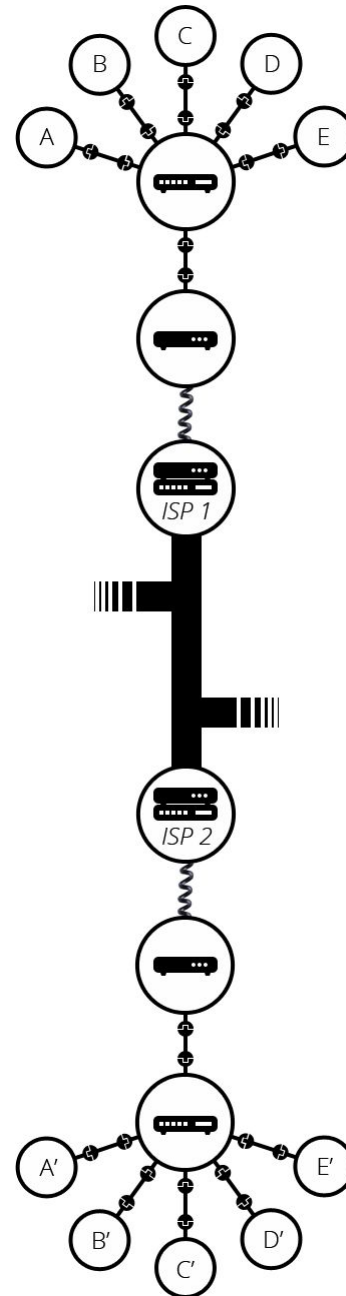


... pero antes, ¿será necesario conocer algo más ?

¿Cómo funciona la web?

La web es una red global de servidores y clientes interconectados que se comunican entre sí mediante diferentes protocolos y tecnologías.

https://developer.mozilla.org/es/docs/Learn/Common_questions/Web_mechanics/How_does_the_Internet_work



TCP/IP (Protocolo de Control de Transmisión y Protocolo de Internet): son los protocolos de comunicación que definen cómo deben viajar los datos a través de la web. Esto es, los medios de transporte que te permiten hacer un pedido, ir a la tienda y comprar los productos. En nuestro ejemplo, podría ser un coche, una bicicleta o tus propios pies.

HTTP (Hypertext Transfer Protocol): Es el protocolo que se utiliza para la transferencia de información en la web. Es el responsable de enviar y recibir datos entre los servidores y los navegadores web.

<https://howhttps.works/es/why-do-we-need-https/>

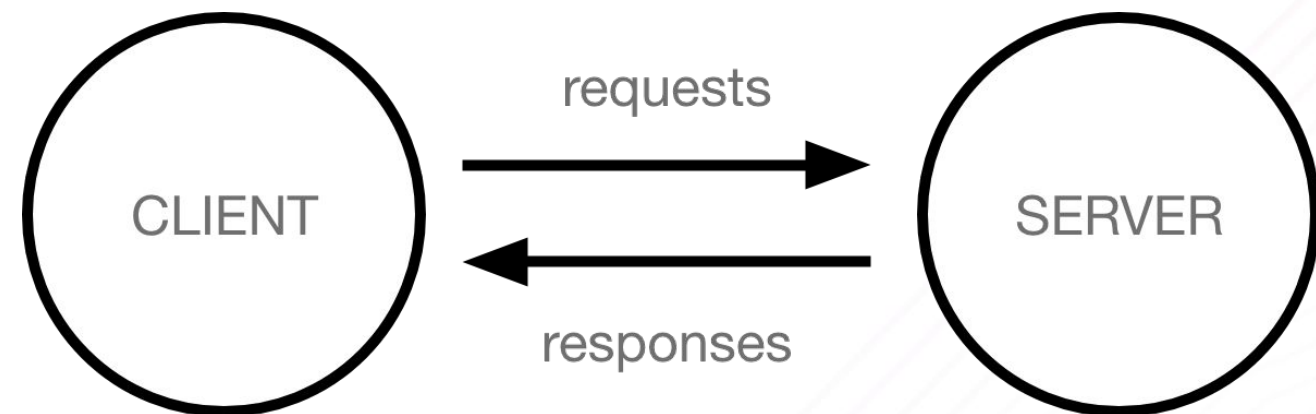
DNS (Domain Name System): Es el sistema que se encarga de convertir los nombres de dominio (por ejemplo, www.google.com) en direcciones IP (por ejemplo, 172.217.0.46) que se utilizan para acceder a los servidores.

<https://howdns.works/es/>

Arquitectura Cliente - servidor

El funcionamiento de la web se basa en la interacción entre los clientes (navegadores web) y los servidores. Cuando un usuario introduce una dirección web en el navegador, éste envía una solicitud al servidor correspondiente mediante el protocolo HTTP. El servidor responde a la solicitud enviando los datos solicitados al navegador, que los interpreta y muestra al usuario.

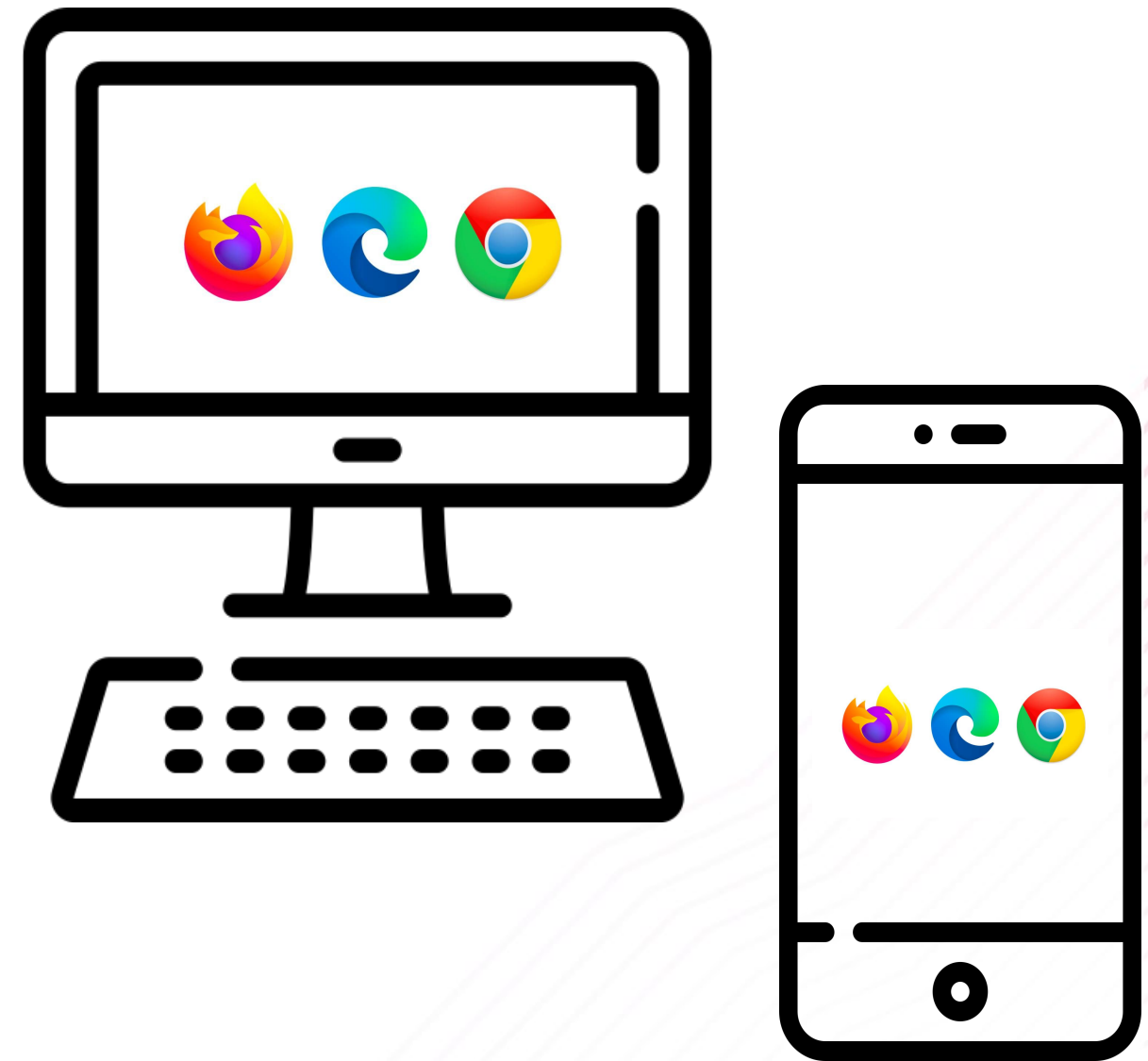
Los mensajes que envía el cliente, normalmente un navegador Web, se llaman *peticiones*, y los mensajes enviados por el servidor se llaman *respuestas*.



Cliente

Un cliente, es cualquier herramienta que actúe en representación del usuario. Esta función es realizada en la mayor parte de los casos por un navegador Web. Hay excepciones, como el caso de programas específicamente usados por desarrolladores para desarrollar y depurar sus aplicaciones.

El navegador es siempre el que inicia una comunicación (petición), y el servidor nunca la comienza (hay algunos mecanismos que permiten esto, pero no son muy habituales).



Navegadores web

El primer navegador web fue desarrollado por Tim Berners-Lee en 1990 y se llamó WorldWideWeb. Posteriormente, en 1993, Marc Andreessen creó el primer navegador gráfico, llamado Mosaic. Desde entonces, han surgido muchos navegadores web populares, como **Safari, Google Chrome, Microsoft Edge (Internet Explorer), Mozilla Firefox, Opera, Brave, Browzar, Maxthon y Vivaldi.**

Los navegadores web tienen muchas características útiles, como la posibilidad de abrir varias pestañas para navegar por múltiples sitios web, la capacidad de guardar contraseñas y datos de formulario, y la integración con motores de búsqueda para encontrar rápidamente información relevante en internet.

Los navegadores web tienen la capacidad de interpretar y ejecutar código HTML, CSS y JavaScript.

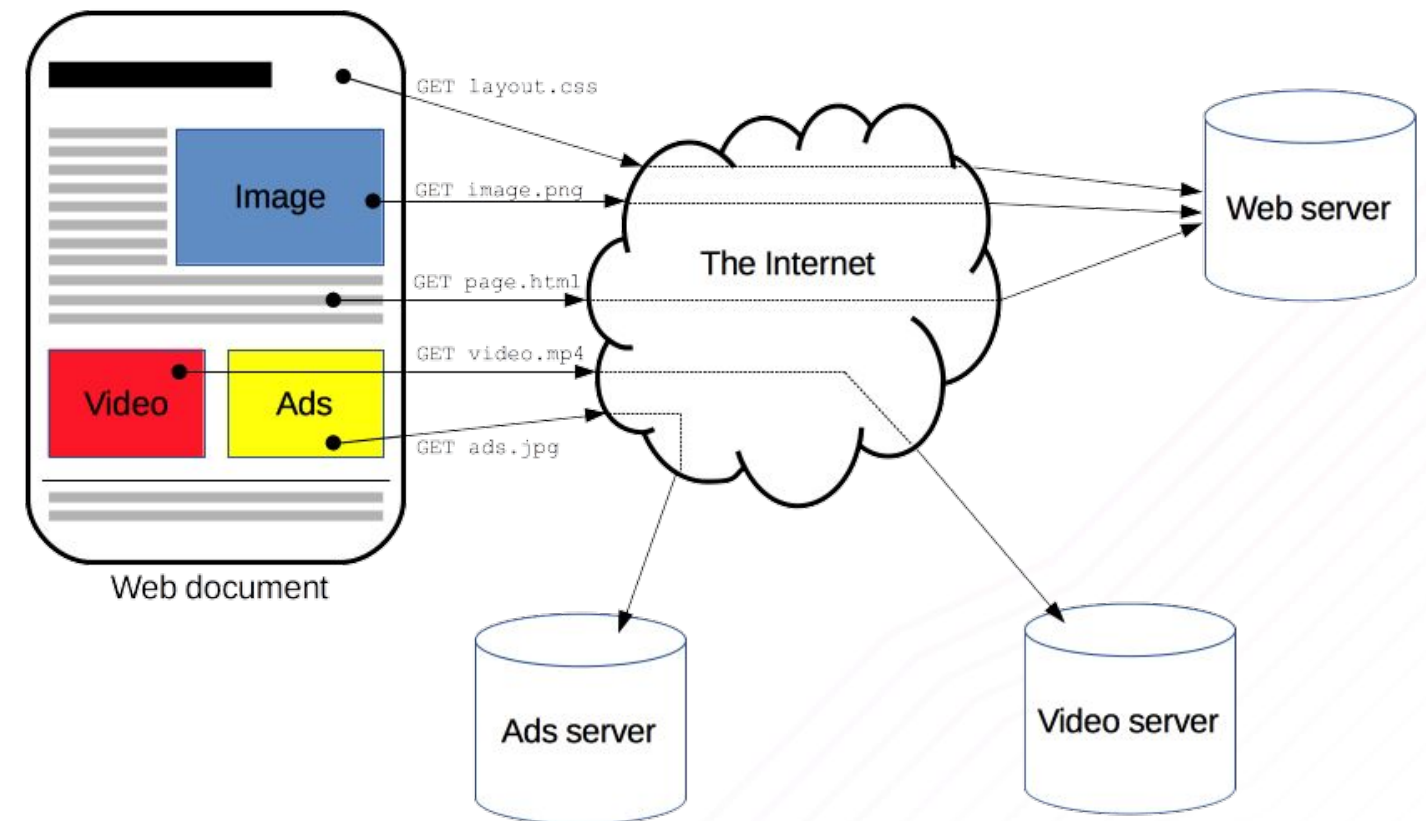


NAVEGADORES WEB

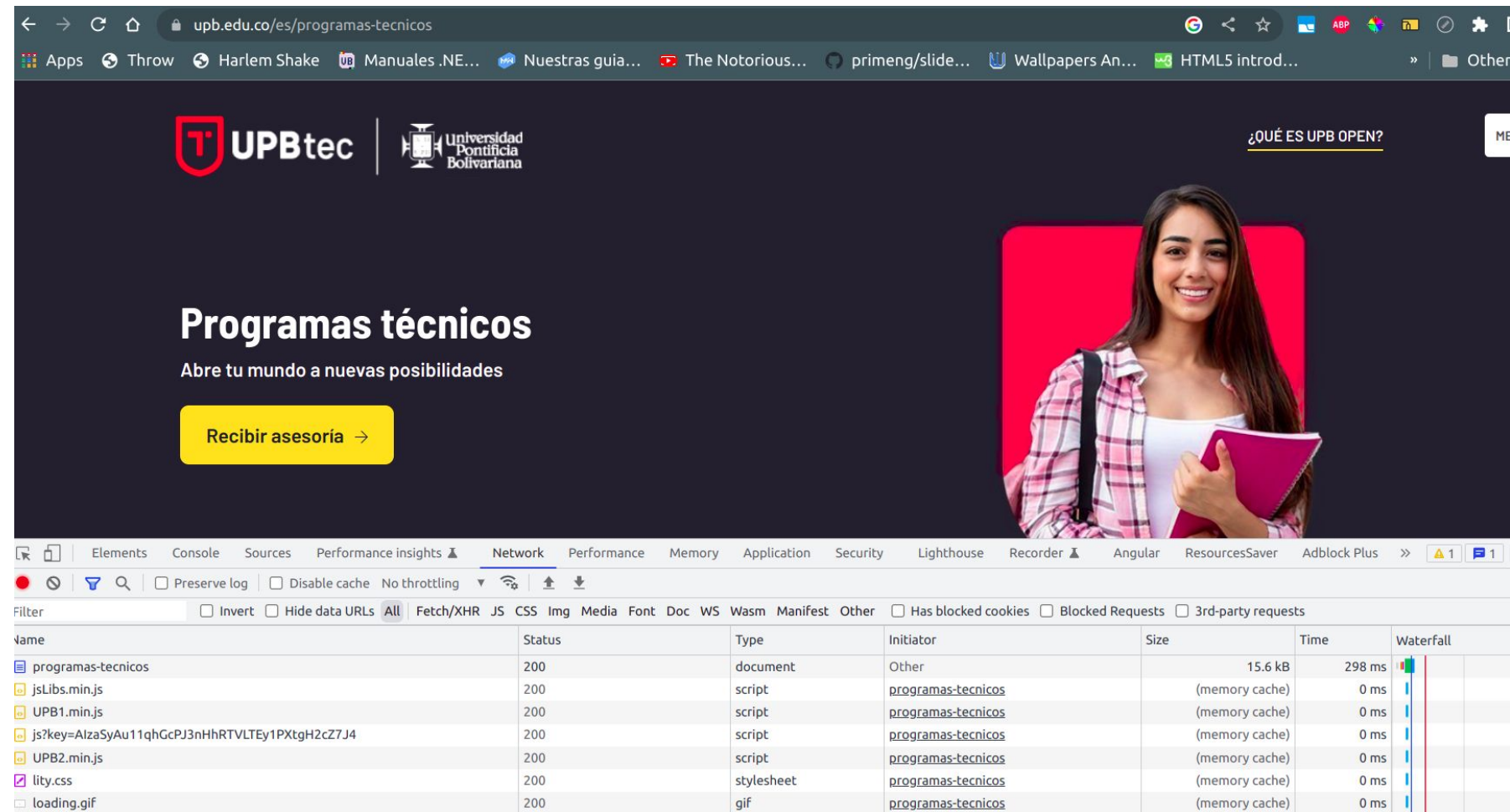


Para poder mostrar una página Web, el navegador envía una petición de documento [HTML](#) al servidor. Entonces procesa este documento, y envía más peticiones para solicitar scripts, hojas de estilo ([CSS](#)), y otros datos que necesite (normalmente vídeos y/o imágenes). El navegador une todos estos documentos y datos, y compone el resultado final: la página Web.

Los scripts los ejecuta también el navegador, y también pueden generar más peticiones de datos en el tiempo, y el navegador, gestionará y actualizará la página Web en consecuencia.



Veamos un ejemplo



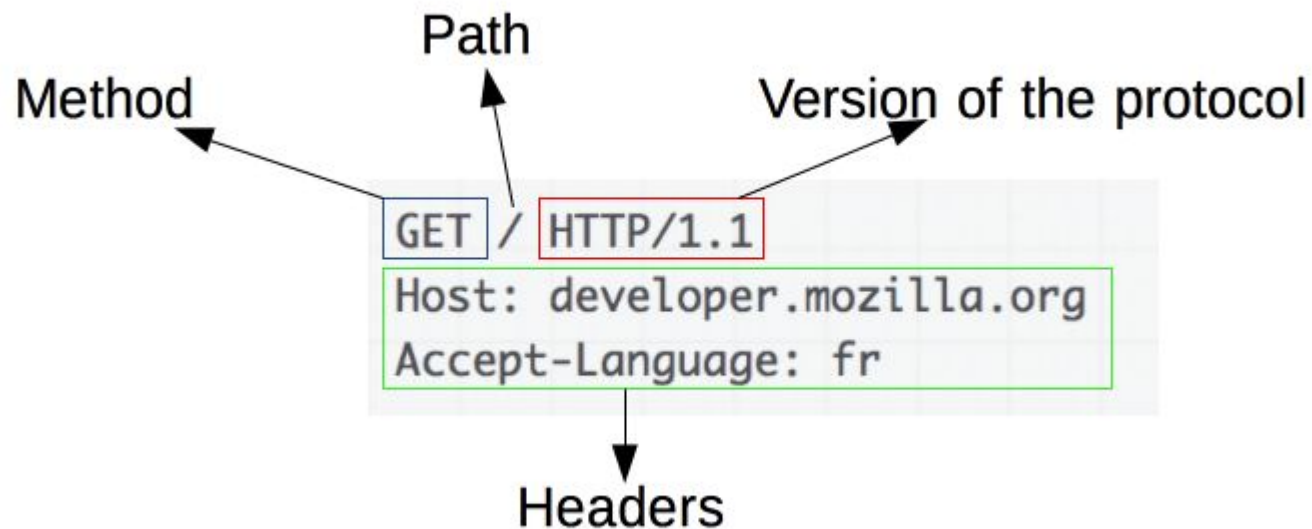
The screenshot shows the UPBtec website with the following content:

- Header: UPBtec logo and Universidad Pontificia Bolivariana logo.
- Section: **Programas técnicos** with the tagline "Abre tu mundo a nuevas posibilidades".
- Button: "Recibir asesoría →"
- Image: A smiling woman holding a pink folder.
- Network Developer Tool (Network tab):

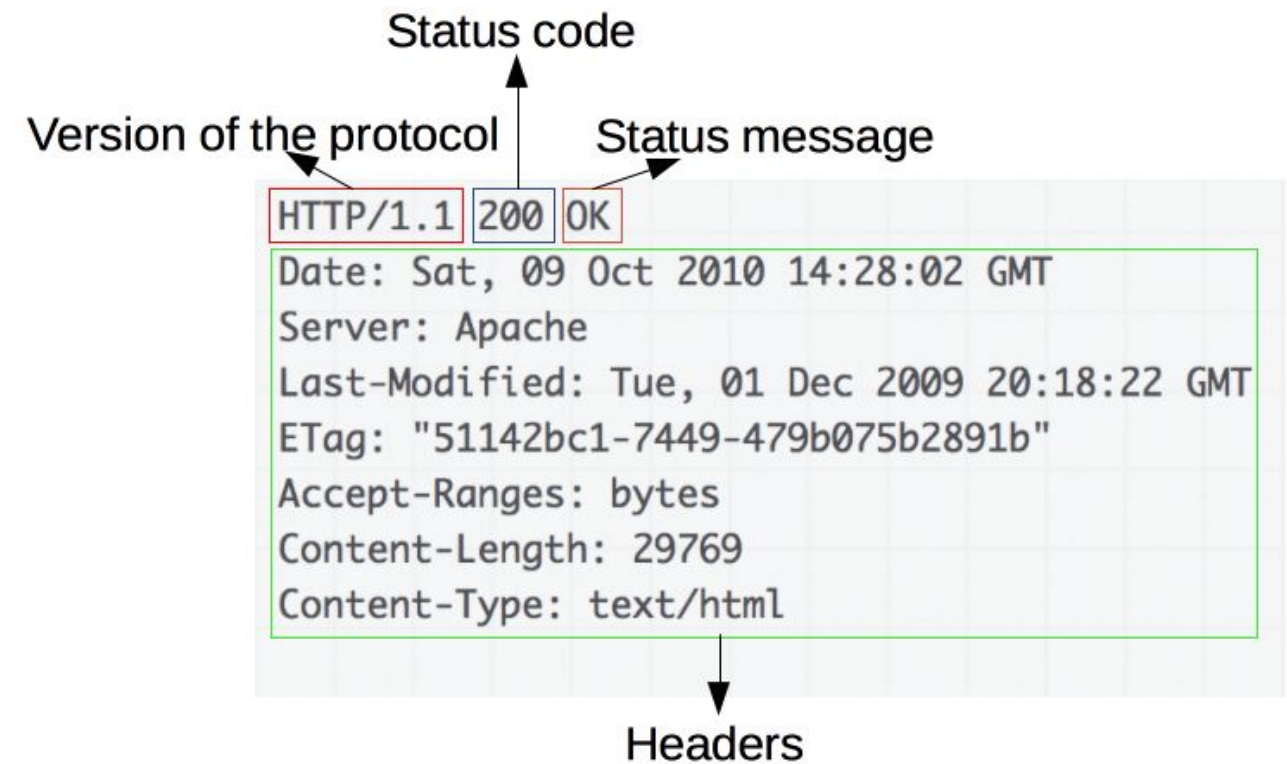
Name	Status	Type	Initiator	Size	Time	Waterfall
programas-tecnicos	200	document	Other	15.6 kB	298 ms	
jslibs.min.js	200	script	programas-tecnicos	(memory cache)	0 ms	
UPB1.min.js	200	script	programas-tecnicos	(memory cache)	0 ms	
js?key=AlzaSyAu11qhGcPJ3nHhRTLVEy1PXtgH2cZ7J4	200	script	programas-tecnicos	(memory cache)	0 ms	
UPB2.min.js	200	script	programas-tecnicos	(memory cache)	0 ms	
lity.css	200	stylesheet	programas-tecnicos	(memory cache)	0 ms	
loading.gif	200	gif	programas-tecnicos	(memory cache)	0 ms	

El detalle de una petición y una respuesta.

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

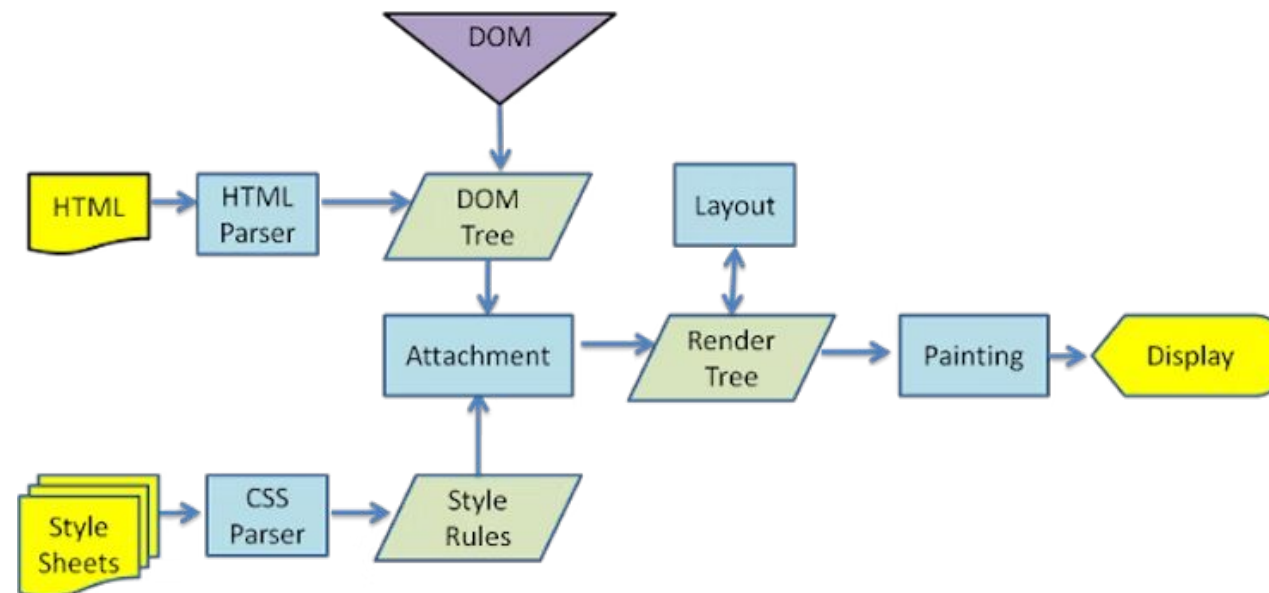


Request



Response

¿Cómo funciona el navegador?



<https://web.dev/howbrowserswork/>

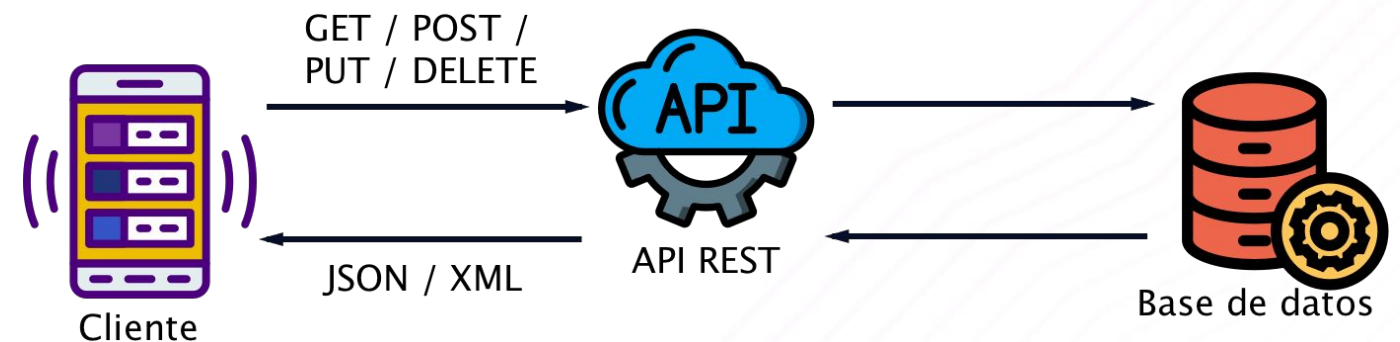
API REST

Es un acrónimo que significa "Interfaz de Programación de Aplicaciones de Transferencia de Estado Representacional" en inglés, que se traduce al español como "Representational State Transfer".

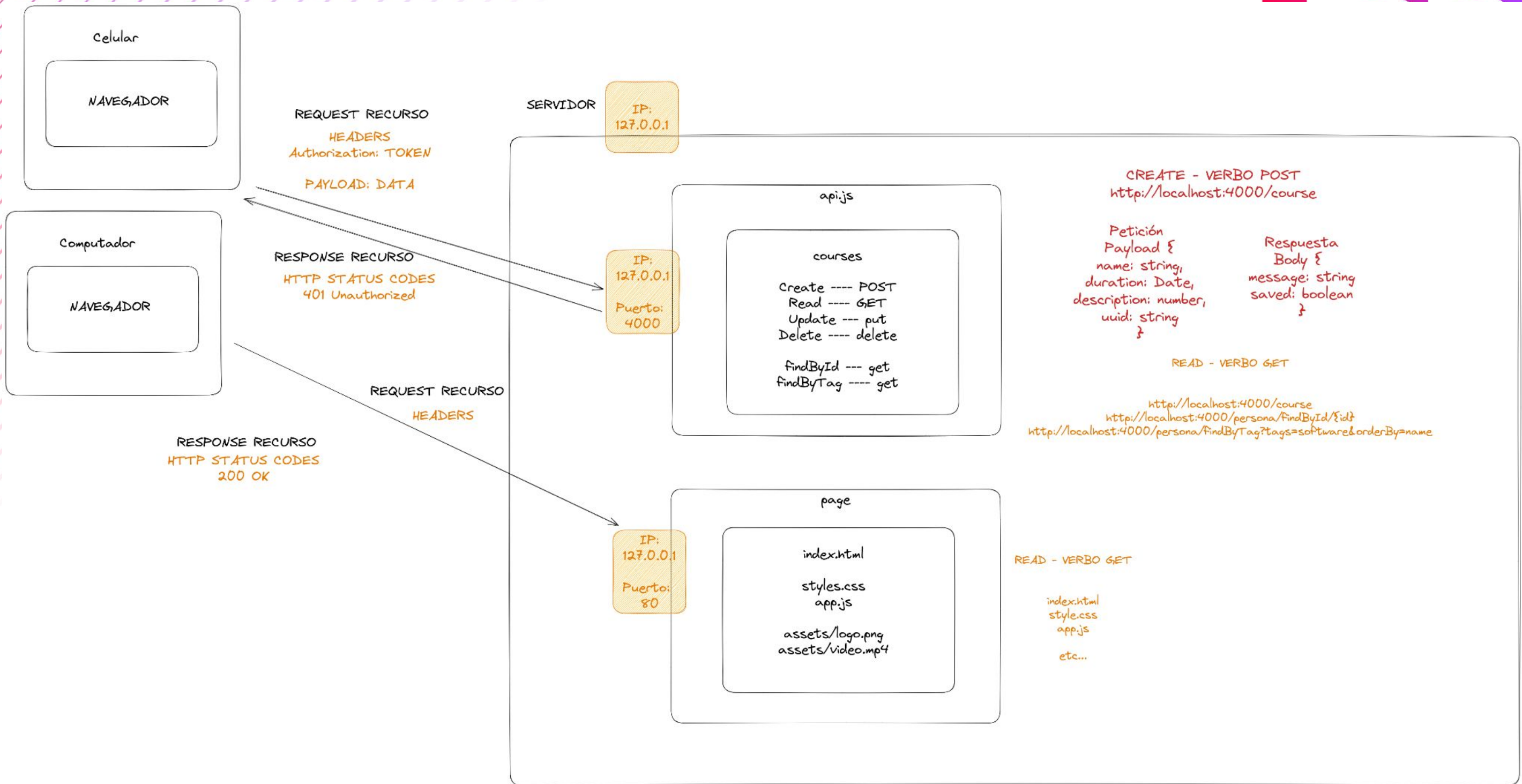
Una API REST es un conjunto de reglas y estándares que permiten que diferentes sistemas se comuniquen entre sí a través de Internet de manera eficiente y consistente. Es una forma de arquitectura de software que se utiliza comúnmente para construir aplicaciones web y móviles.

Las API REST permiten a los desarrolladores de software acceder y manipular los datos de una aplicación a través de una interfaz simple.

Utilizan el protocolo HTTP para enviar y recibir datos y se basan en la estructura de recursos (como URLs) y métodos (como GET, POST, PUT, DELETE) para definir las operaciones que se pueden realizar en los datos.



<https://blog.hubspot.es/website/que-es-api-rest>



¿Cómo hago que mi sitio sea visible en internet?

Dominio y Hosting (Servidor)

El dominio sirve para identificar recursos de Internet, como computadoras, redes y servicios, con una etiqueta basada en texto que es más fácil de memorizar que las direcciones numéricas (conocidas como IP) utilizadas en los protocolos de Internet.

Puedes tener todos los dominios que quieras sin la necesidad de tener hosting, esto solo te ayudaría evitar que nadie más lo tenga y protegerlos.

Un hosting es un servicio que permite a las personas o empresas almacenar sus páginas, archivos, información, correo, imágenes, vídeos o cualquier contenido accesible desde la web. Los proveedores de Hosting (Ejm: <https://www.godaddy.com/es>) suelen ser empresas que proporcionan un espacio en sus servidores y conexión a Internet para sus datos a sus clientes.



Esta teoría no es esencial para escribir código web a corto plazo, pero en poco tiempo empezarás a beneficiar realmente al entender lo que está sucediendo en el fondo

https://developer.mozilla.org/es/docs/Learn/Common_questions/Web_mechanics/Pages_sites_servers_and_search_engines

Tecnologías Front-end



¿Qué es HTML?

Hypertext Markup Language o en español, **Lenguaje de marcado de hipertexto**, es un lenguaje de marcado, no un lenguaje de programación. La diferencia principal entre un lenguaje de programación y un lenguaje de marcado es que un lenguaje de programación se utiliza para crear programas que pueden realizar tareas complejas, mientras que un lenguaje de marcado se utiliza para estructurar y presentar información.

HTML se utiliza para crear documentos web que se pueden ver en un navegador web. Con HTML, los desarrolladores web pueden definir la estructura de una página web, así como el texto, imágenes, videos y otros elementos que aparecen en la página.

A través del uso de etiquetas de HTML, los desarrolladores pueden definir el significado y la estructura de los elementos de la página, como títulos, párrafos, listas y tablas.

HTML





1989: Tim Berners-Lee, un físico británico que trabajaba en el CERN (Organización Europea para la Investigación Nuclear), desarrolla el concepto de la World Wide Web (WWW) y crea el primer navegador web, llamado WorldWideWeb.

1991: Berners-Lee crea la primera versión de HTML, que constaba de solo 18 elementos básicos.

1993: Se publica la especificación de HTML 2.0, que incluye mejoras como tablas, formularios y enlaces internos.

1995: Se publica la especificación de HTML 3.0, que incluye soporte para hojas de estilo y tablas más avanzadas.

1997: Se publica la especificación de HTML 4.0, que incluye soporte para marcos, scripts y hojas de estilo en línea.

1999: Se publica la especificación de XHTML 1.0, que es una versión más estricta de HTML que sigue las reglas de XML.

2000: Se publica la especificación de HTML 4.01, que corrige algunos errores y se centra en la accesibilidad y la internacionalización.

2004: Se publica la especificación de XHTML 2.0, que es una versión completamente nueva de XHTML.

2008: Se publica la especificación de HTML5, que introduce muchas nuevas características como etiquetas de video y audio, soporte para gráficos vectoriales, y mejoras en la semántica y accesibilidad.

2011: HTML5 se convierte en la recomendación oficial de la W3C (World Wide Web Consortium), la organización encargada de definir los estándares de la web.

2014: Se publica la especificación de HTML5.1, que introduce mejoras en la accesibilidad, multimedia, formularios y estilos.

2016: Se publica la especificación de HTML5.2, que incluye mejoras en la semántica, el procesamiento de datos y la accesibilidad.

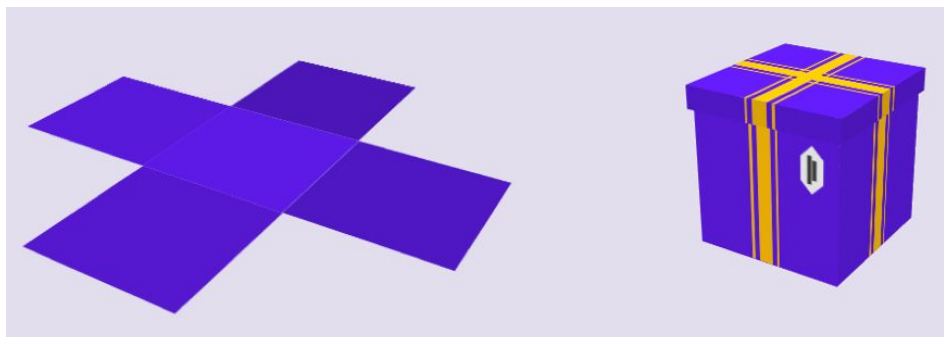
Actualidad: Se espera que se publique la especificación de HTML6, que se centrará en mejorar la estructura semántica y la accesibilidad.

¿Qué es CSS?

Cascading Style Sheets o en español, **Hojas de estilo en cascada**, tampoco es un lenguaje de programación, es un lenguaje de estilo utilizado para definir la apariencia visual de un documento HTML.

CSS se utiliza para controlar la presentación, el diseño y el formato de un documento HTML, incluyendo el color de fondo, el tamaño y tipo de letra, el espaciado y la posición de los elementos en una página web, con el objetivo de mejorar la experiencia de usuario y hacer que el contenido sea más legible y atractivo.

Ha evolucionado tanto, que nos permite realizar animaciones, transiciones y hasta visualizar elementos en 3D.



CSS





1994: Håkon Wium Lie, un estudiante noruego, presenta la idea de hojas de estilo en un documento llamado "Cascading HTML Style Sheets" a su supervisor, Tim Berners-Lee, inventor de la World Wide Web.

1996: El World Wide Web Consortium (W3C) forma un grupo de trabajo para desarrollar un estándar para las hojas de estilo. El grupo estaba liderado por Lie y se llamaba "Cascading Style Sheets (CSS)".

1997: Se publica la especificación de CSS nivel 1, que permite a los desarrolladores web definir estilos para los elementos de una página web.

1998: Se publica la especificación de CSS nivel 2, que introduce nuevas características como hojas de estilo en cascada, posicionamiento y diseño de tablas.

1999: Microsoft introduce su propia implementación de CSS llamada "Cascading Style Sheets, Level 1" en Internet Explorer 5.

2001: Se publica la especificación de CSS nivel 2.1, que corrige algunos errores en la especificación anterior y añade algunas nuevas características.

2005: Se publica la especificación de CSS nivel 3, que introduce características más avanzadas como selecciones avanzadas, sombras y degradados.

2007: Se publica la especificación de CSS nivel 3 como una recomendación oficial del W3C.

2011: Se publica la especificación de CSS nivel 4, que introduce nuevas características como animaciones y transiciones.

2014: Se publica la especificación de CSS Grid Layout, que permite a los desarrolladores web crear diseños complejos y flexibles en la web.

2017: Se publica la especificación de CSS Grid Layout como una recomendación oficial del W3C. Se publica la especificación de Flexbox como una recomendación oficial del W3C.

¿Qué es Javascript?

Es un lenguaje de programación de alto nivel, interpretado y orientado a objetos. Es utilizado principalmente en el desarrollo web para agregar interactividad y dinamismo a las páginas web.

JavaScript se ejecuta en el lado del cliente, es decir, en el navegador web del usuario, y se utiliza para manipular el contenido HTML y CSS de una página, validar formularios, crear animaciones y efectos visuales, y realizar solicitudes a servidores web.

También puede ser utilizado **en el lado del servidor**, por ejemplo, con el entorno de ejecución **Node.js**.

JavaScript es uno de los lenguajes de programación más populares y ampliamente utilizados en la actualidad. Además del desarrollo web, también se utiliza en la creación de aplicaciones móviles, juegos, aplicaciones de escritorio y muchas otras áreas de la informática.





1995: Brendan Eich, un desarrollador de Netscape, crea el primer prototipo de JavaScript en 10 días. Fue diseñado originalmente para ser utilizado en Netscape Navigator, un popular navegador web de la época.

1996: Netscape envía JavaScript a la ECMA (European Computer Manufacturers Association) para su estandarización. La ECMA crea un grupo de trabajo llamado TC39 para desarrollar una especificación estándar para el lenguaje.

1997: Se publica la primera versión de la especificación de ECMAScript, que es la base del lenguaje JavaScript. Esta versión se conoce como ECMAScript 1.

1998: Se publica ECMAScript 2, que corrige algunos errores en la especificación anterior y añade algunas nuevas características.

1999: Se publica ECMAScript 3, que se convierte en la especificación dominante durante muchos años y se utiliza ampliamente en todos los principales navegadores web.

2005: Se publica ECMAScript 3.1, que se convierte en ECMAScript 4. Sin embargo, esta versión nunca se adoptó debido a la falta de consenso entre los miembros de la comunidad.

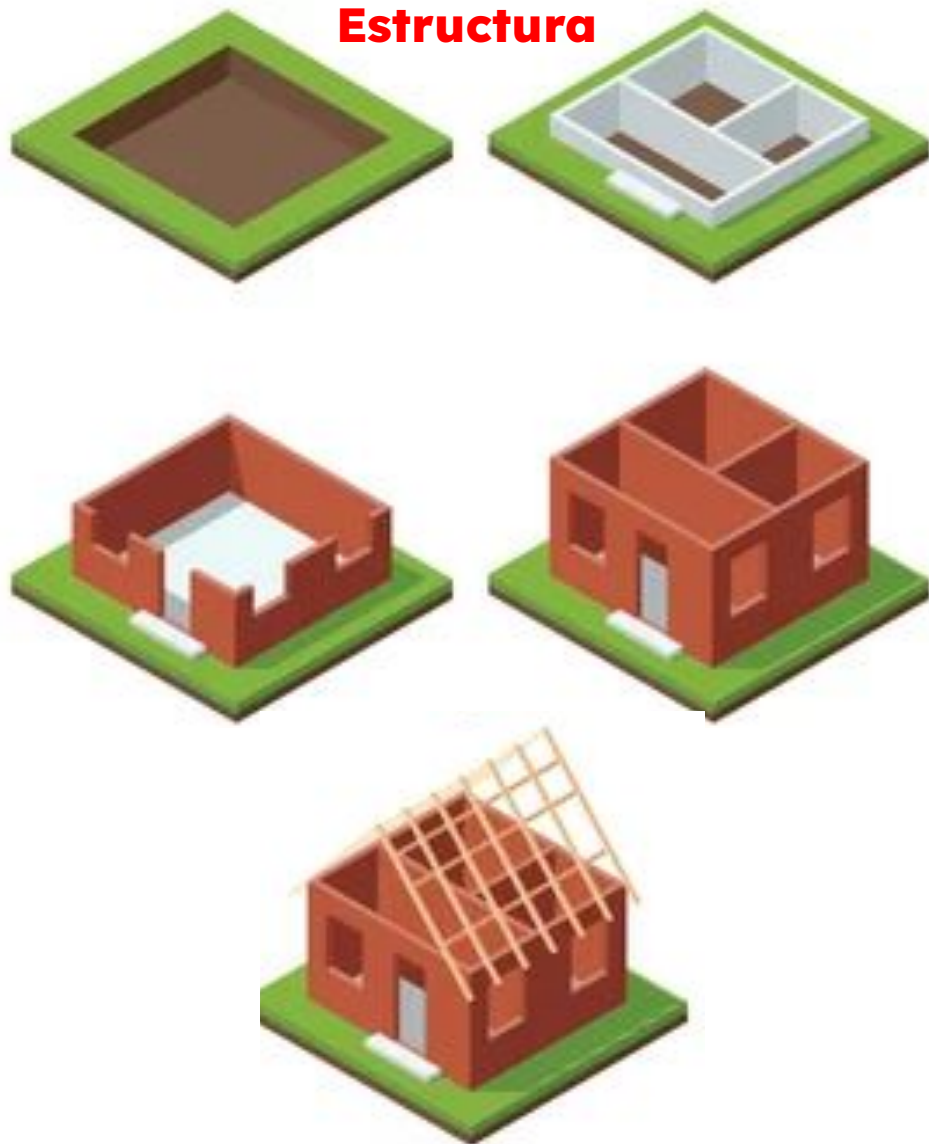
2009: Se publica ECMAScript 5, que añade nuevas características como getters y setters, y mejora la compatibilidad entre navegadores.

2015: Se publica ECMAScript 6 (también conocido como ECMAScript 2015), que introduce nuevas características importantes como la sintaxis de clases, las variables let y const, y las promesas.

2016-2022: Se publican versiones sucesivas de ECMAScript, incluyendo ECMAScript 2016, ECMAScript 2017, ECMAScript 2018, ECMAScript 2019, ECMAScript 2020, ECMAScript 2021 y ECMAScript 2022.

Cada versión añade nuevas características y mejoras al lenguaje, lo que lo hace más poderoso y flexible.

.html
Estructura



.css
Estilo



.js
Interacción

