

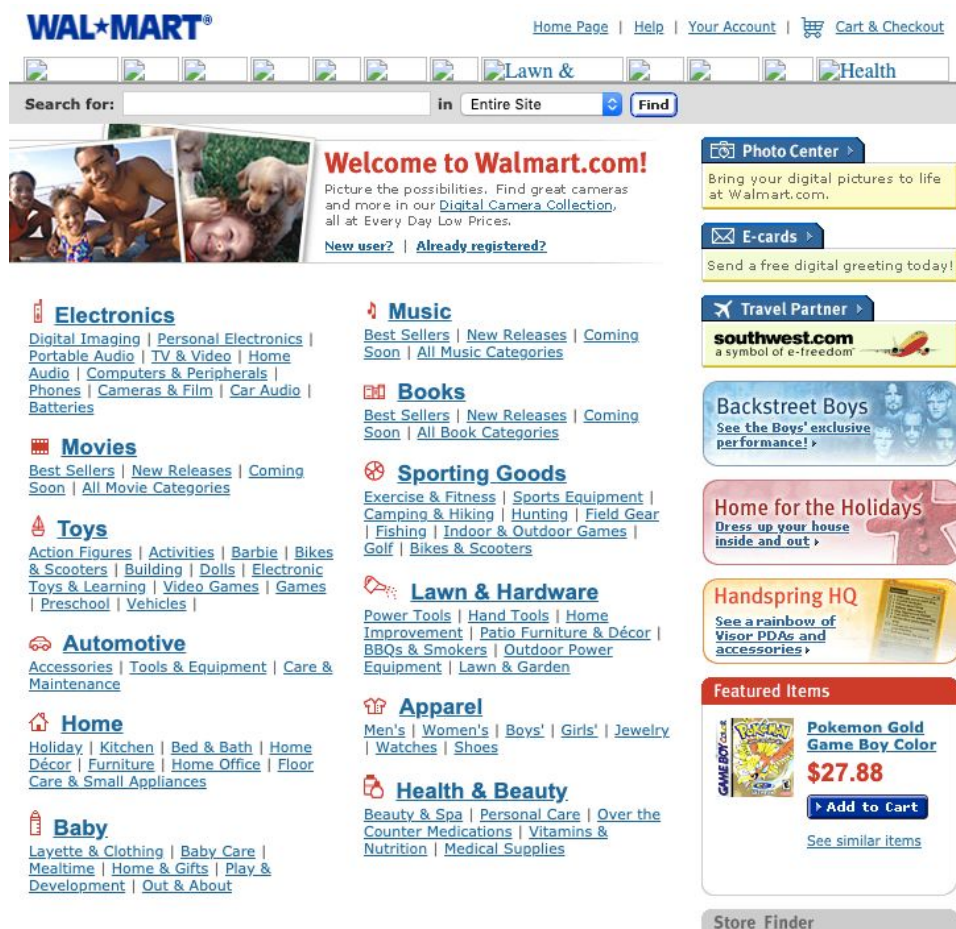
# Del big bang de la web hasta la actualidad

*"La web es más una creación más social que técnica. La diseñé para un efecto social —para ayudar a las personas que trabajan juntas— y no como un juguete técnico. El objetivo último de la web es apoyar y mejorar nuestra existencia en la telaraña mundial."*

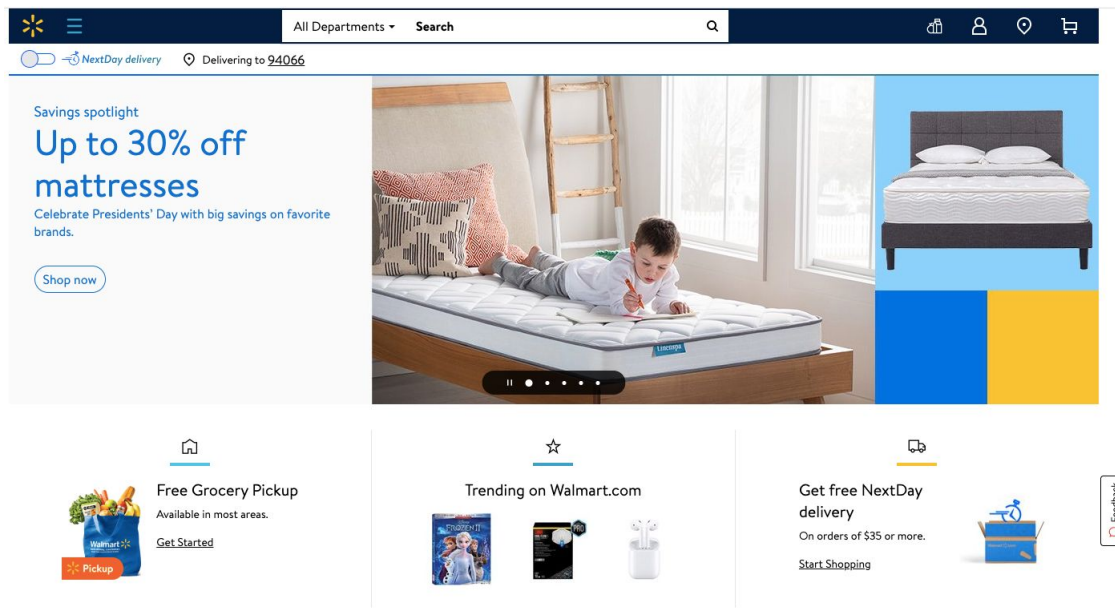
Tim Berners-Lee, creador de la web.

La web no siempre fue tal como la conocemos hoy en día: grandes modificaciones se han producido desde sus comienzos. Por ejemplo, observa cómo cambió el sitio web de Wal-Mart en los últimos 20 años:

www.walmart.com en 2000:



www.walmart.com en 2020:



Puedes buscar más ejemplos en esta gran biblioteca pública de archivos web actuales e históricos: [Wayback Machine](https://www.waybackmachine.org/).

Seis mujeres —Kathleen McNulty Mauchly Antonelli, Frances Bilas Spence, Ruth Lichterman Teitelbaum, Marlyn Wescoff Meltzer, Frances Snyder Holbertson and Betty Jennings Bartik— permitieron que internet se desarrolle como se desarrolló, ya que programaron la primera computadora 100% eléctrica en 1945 para un proyecto de la Armada de EE.UU. Puedes leer [aquí](#) otras contribuciones de mujeres en la computación que produjeron grandes transformaciones.

[Los orígenes de internet](#) se remontan a la Guerra Fría, cuando el Departamento de Defensa de EEUU creó [una Agencia de Investigación de Proyectos de Defensa](#) (ARPA, por sus siglas en inglés) que nucleaba a centros de investigación de distintas universidades.

[John Lickinder](#) comenzó a mediados de los '60 a pensar en una red que permitiera transmitir datos entre las universidades y poder centralizar las investigaciones. La llamó "*Intergalactic Computer Network*" pero no pudo concretarla. [Robert Taylor](#) tomó esta idea y formó un equipo junto a [Lawrence](#)

[Roberts](#), [Leonard Kleinrock](#) y [Donald Davies](#) para llevarla a cabo. El nombre de la red sería [ARPANET](#).

Para desarrollar ARPANET, tuvieron en cuenta una tecnología existente (la red de telefonía) y desarrollaron una forma de enviar paquetes a través de la misma, basada en la [teoría de colas](#) proveniente de la matemática y desarrollada por [Agner Erlang](#). Incluyeron además el concepto de multiplexación para poder hacer transmisiones simultáneas entre diferentes equipos.

Los equipos que desarrollaron se llamaron “*Interface Message Processor*”, y en **octubre de 1969 realizan la primera conexión de manera exitosa entre UCLA y Stanford.**

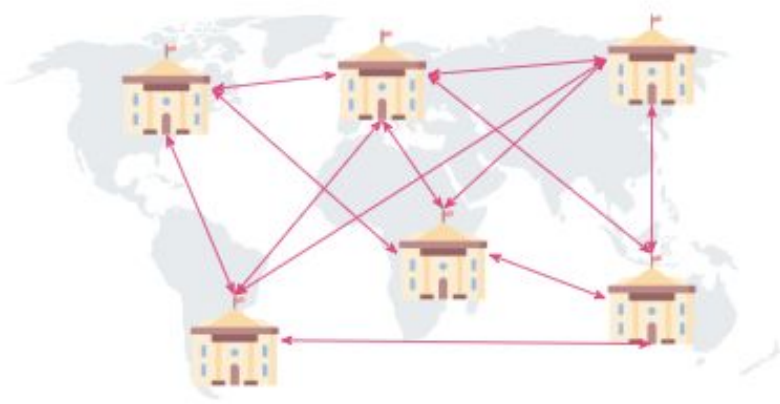
El uso masivo de la red hizo que gobiernos y universidades del mundo se sumaran y se crearan [protocolos](#). Como con cualquier lenguaje o comunicación entre personas, entes u organizaciones, es necesario establecer algunas normas de convivencia para facilitar la comprensión. ¡Era necesario estandarizar la red!

En 1981 nació el [protocolo TCP/IP](#), **que es la tecnología que hoy en día se utiliza para interconectar a las computadoras del mundo, a través de la red de INTERNET.** De esta manera se estandarizó la comunicación punto a punto para que todas las computadoras pudieran “hablar” entre sí en el mismo lenguaje.

Pasamos de una comunicación así:



A una comunicación así:

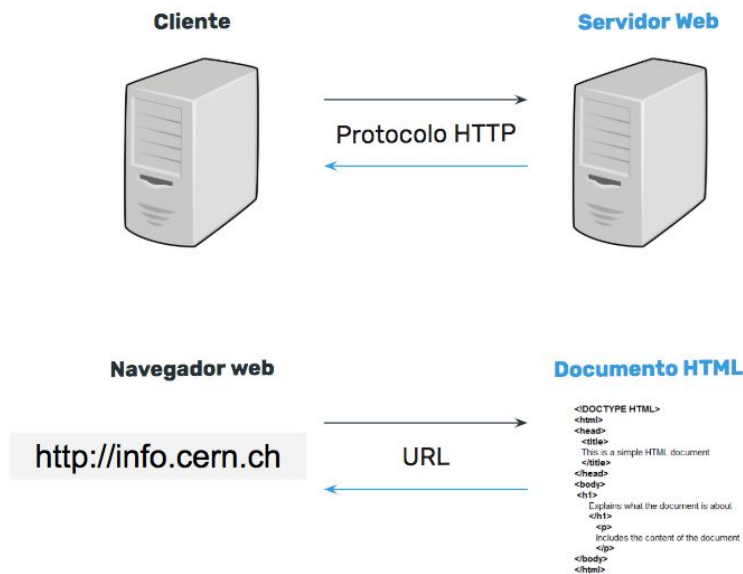


Durante los '80, la red de internet se estandarizó y se empezó a utilizar en la mayoría de los centros de conocimiento, entre ellos, el [CERN](#) (Comisión Europea de Energía Atómica), que tiene el nodo de internet más grande de Europa.

Desde dentro del CERN, el físico [Tim Berners-Lee](#) comenzó a pensar que era necesario que se pudiera acceder a las investigaciones de las distintas sedes desde cualquier punto de la red. En 1989 presentó ante el CERN [Cómo funciona internet](#), el proyecto de [World Wide Web](#).

Creó para esto un protocolo, un lenguaje, una arquitectura de comunicación y un intérprete de documentos HTML (Navegador).

En este [artículo](#) del diario digital "OKDIARIO" se destaca sobre Tim Berners-Lee: *"Tanto él como su grupo de trabajo crearon las siglas conocidas como lenguaje HTML, es decir, las etiquetas de hipertexto, así como el sistema de localización URL y el protocolo HTTP. Así que, de no ser por él y por sus proyectos Memex y Xanadú, tal vez ahora no estarías leyendo este texto."*



La **arquitectura web** que crea Tim Berners-Lee es la de cliente-servidor (que vimos en la bitácora anterior).

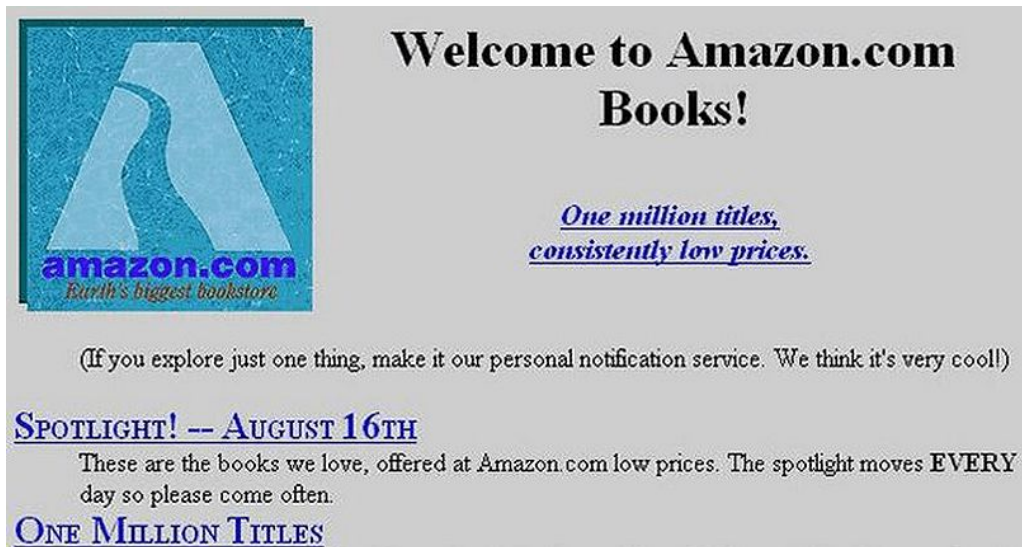
## De los sitios web a las aplicaciones web

En 1991 se lanzó la **World Wide Web** de uso público. Comenzaron a aparecer los primeros sitios web, y distintos navegadores que interpretaban **HTML**. El problema era que cada navegador tenía su propia forma de interpretar el lenguaje. Nuevamente, el mismo problema que había surgido en 1981: ¡había que estandarizar!

A medida que se masificó el uso de internet para fines comerciales, empezó a ser necesario agregar **estilos** y **diseños** con fines publicitarios.

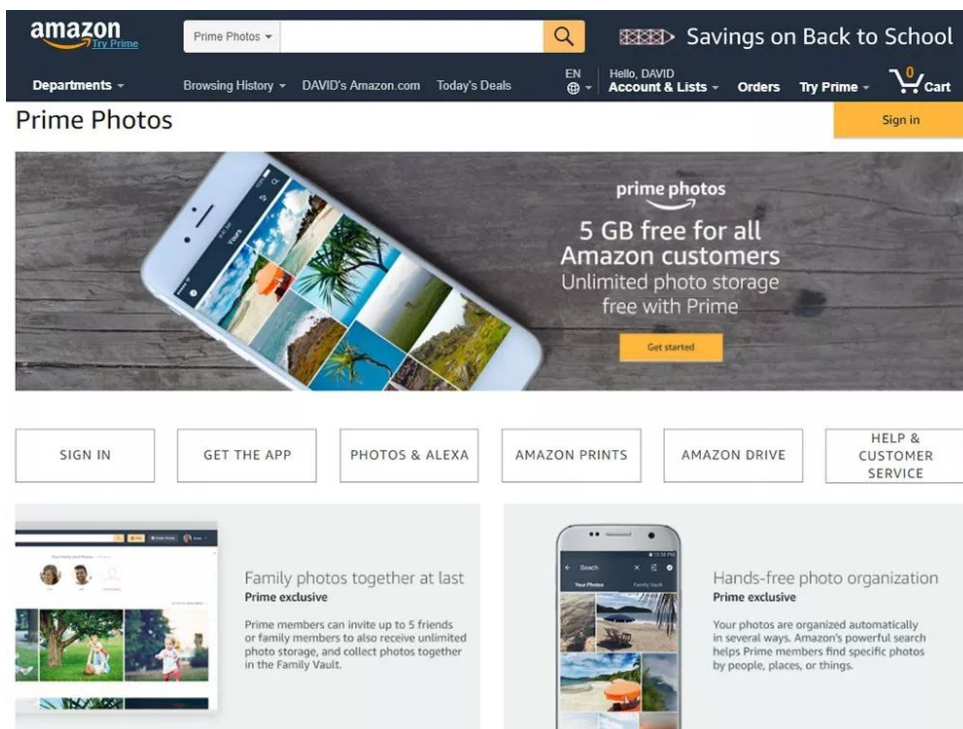
HTML proveía sólo algunas **etiquetas** y **atributos** para tal fin, pero provocaba que fuera necesario trabajar con cada etiqueta de forma individual.





Nace entonces **CSS**, para poder aplicar estilos a diferentes documentos **HTML** y por lo tanto a diferentes etiquetas, de manera centralizada. De esta manera, se abstrae la capa de estructura de la capa de presentación de los sitios web.

¡Gracias CSS!



La siguiente necesidad fue la de poder modificar los documentos HTML y sus estilos desde el navegador, sin la necesidad de traer un nuevo documento HTML con los cambios necesarios.

Para tal fin, en 1995 [Netscape](#), a través de [Brendan Eich](#), creó el lenguaje de programación [JavaScript](#). **El objetivo primario del lenguaje fue interceptar eventos que el usuario realiza sobre el sitio web (por ejemplo “un click”), y realizar una acción sin necesidad de recargar todo el documento HTML.**

## ¡Bienvenidos/as a la web interactiva!

Tim Berners-Lee crea entonces el **World Wide Web Consortium**, el cual hoy sigue presidiendo, y es el encargado día a día de mantener el estándar de los lenguajes de la web. Pasamos de desarrollar páginas web a aplicaciones web que puedan responder a peticiones de usuarios personalizadas.



HTML, CSS y JavaScript —tres lenguajes que habíamos mencionado en nuestra primera bitácora— componen el stack de tecnologías que utilizamos hasta hoy para desarrollar un sitio web y las que veremos en esta carrera como parte del Front-End. Lo que ves a la derecha en el gráfico (node.js y MySQL son lenguajes que se utiliza en Backend, y que abordaremos en el Bloque 3).

## Un lenguaje para diseñar y desarrollar páginas web

Ahora que ya sabes sobre el nacimiento de internet y sus funciones, nos vamos a sumergir en HTML.

**HTML es un lenguaje de Marcas de Hipertexto de la web.** Se basa en etiquetas y atributos que luego los browser se van a encargar de interpretar para mostrar el contenido jerarquizado de determinada manera.

En [este artículo](#), Dan nos aclara: "HTML no se considera un lenguaje de programación porque no "programa" comportamientos, simplemente se encarga de estructurar información, ni siquiera se encarga de manejar su propia imagen, de eso se encarga el CSS."

Podemos verlo en un ejemplo de Roy Batty en su artículo ["Qué es HTML?"](#):

*Imaginemos una web donde sólo se nos plantea una pregunta:*

***"¿Quieres ir a Google.com?"***

*Y debajo, dos posibles respuestas: SÍ y NO; y dependiendo de la opción que seleccionemos, el hipervínculo de rigor nos lleva a Google.com o a otra web con otra lista de navegadores. Nuestro documento tendría esta estructura:*

```
<body>

  ¿Quieres ir a Google.com?</br>

  - <a href="http://www.google.com">Sí</a></br>
  - <a href="listanavegadores.html">No</a></br>
</body>
```

*¿Es esto un programa? No, ¿verdad?*

*Las líneas que hemos escrito en HTML tan sólo son "formas de mostrarnos las cosas", nada más, no ejecutamos ninguna rutina, subrutina, programa, subprograma, ni nada... nuestro código HTML no tiene que interpretar funciones simples o complejas, sólo llevarnos a uno u otro sitio, que hemos definido previamente.*



Si aún no viste [Introducción a HTML](#) ¡estás a tiempo de hacerlo! En este bloque, aprenderemos primero a escribir y entender este lenguaje que nos permite estructurar el contenido que vamos a brindarle a los/as usuarios/as.

## Cómo escribir en un lenguaje tecnológico

La sintaxis HTML se basa en [etiquetas](#) que son las encargadas de comunicarle al browser qué contenido debe mostrar. HTML dispone de etiquetas para imágenes, linkear documentos, hipervínculos que nos permiten dirigirnos a otras páginas, saltos de línea, listas, tablas, etc.

¿Cómo comenzar a escribir en HTML? Existen dos formas:

1. Las etiquetas que tienen un [inicio y cierre](#).

Se escriben abriendo el símbolo “menor que” (<) y se cierran con el símbolo “mayor que” (>):

```
<etiqueta>
```

Y se cierran con una etiqueta igual a la de apertura pero con una barra hacia la derecha (/) delante:

```
<etiqueta>
  Contenido
</etiqueta>
```

2. Otras etiquetas no tienen cierre, son únicas, y tienen [atributos](#).

Se cierran con una barra inclinada hacia la derecha al final (/) luego del último signo:

```

```

Los atributos son elementos que al completarse dan una funcionalidad extra a la etiqueta.

🧐 *TIP: Es una buena práctica escribir las etiquetas HTML y los atributos en minúscula. Aunque pueden funcionar igual, no es costumbre escribirlas en mayúsculas.*

## ¿Qué hay dentro de HTML?

Para comenzar, nuestra página web o aplicación web va a tener como archivo central al documento HTML que se complementa con los otros lenguajes. Es importante prestarle mucha atención y aprender a manejarlo bien, ya que es la forma en la que “encontrarán” el archivo que creamos.

Vamos a conocer la [estructura básica](#) de un documento HTML que se mantiene siempre igual.

```
<!DOCTYPE html>

<html>

  <head>

    <title> Título aquí </title>

  </head>

  <body>

    Contenido del sitio web aquí.

  </body>

</html>
```

¿Qué son todas estas palabras [en azul](#) que estamos viendo? ¡No te asustes! Al principio suenan extrañas, pero se van a ir aclarando. Estos elementos son una serie de etiquetas que el navegador interpreta y da forma en la pantalla.

Nuestro documento inicia con la declaración `<!DOCTYPE html>` que se escribe muy diferente de las demás. DOCTYPE le va a comunicar al browser la manera en la que el navegador procesará el documento.

La etiqueta `<html>` contendrá todo el documento. Observa que todas las etiquetas tienen una apertura, que se escribe `<etiqueta>` y un cierre, que se escribe `</etiqueta>`. Adentro hay dos etiquetas fundamentales:

- La etiqueta `body` es una de las más importantes ya que contendrá todas las etiquetas que compondrán el documento, y por lo tanto, la información que visualizará el usuario en la ventana del navegador.
- La etiqueta `head` por otro lado, tiene el rol de contener toda información, pero que no visualizará el usuario. La utilizamos para que el navegador pueda tomar decisiones sobre cómo presentar el documento al usuario. Dentro de esta etiqueta se encuentra la etiqueta `title` que va a mostrar en el browser el título de nuestro sitio. Dentro del `head` también podemos utilizar la etiqueta `meta`, y allí podremos describir nuestro sitio para ayudar a los buscadores a encontrarnos.

Hay que diferenciar elementos en línea y bloque de HTML para no generar confusiones:

- Las etiquetas `inline` no afectan al flujo de los demás elementos. Dejan que el texto y otros fluyan en la misma línea, colocándose uno al lado del otro.

```
<div>hola <span>mundo</span></div>
```

- Las etiquetas `block` cortan el flujo y se renderizan en el navegador en líneas independientes.

```
<div>hola <div>mundo</div></div>
```

No te preocupes por entender en detalle esto, pero es importante que lo vayas viendo para cuando lo trabajemos en los encuentros.

¿Conoces otras etiquetas? El conocimiento que traes es importante, puedes investigar hasta nuestro próximo encuentro. ¡Ya estás más cerca de tu primer sitio web!

🔊 **Recuerda cargar todas las preguntas** que te hayan surgido sobre los temas cubiertos en la presente bitácora en el **Trello** de tu grupo de estudio. Tu equipo docente armará el repaso en clase a partir de esto.

---

## ¡Prepárate para el próximo encuentro!

### Profundiza

📖 [Comenzando con HTML](#)

📖 [Sintaxis HTML](#)

📖 [Etiquetas HTML](#)

### Herramientas

🔧 [Markup Validation Service \(Servicio de validación de marcado W3C\)](#) es un validador creado por el W3C que permite a los usuarios analizar la si documentos HTML y XHTML fueron redactados siguiendo las reglas gramaticales que están utilizando. La validación del marcado es un paso importante para asegurar la calidad técnica de las páginas web. En el encuentro la utilizaremos, por lo que puedes comenzar a explorarla para llegar más preparado/a.

### Challenge

📝 ¡Llegó la hora de investigar! Genera un listado de los 10 tags HTML más utilizados. Busca diferentes artículos a través de Google y arma tu propia lista. Tenla a mano para el próximo encuentro, entre todos/as crearemos nuestra lista de tags más utilizados.