

# La evolución de la web: borrador 1

Kevin Esguerra Cardona

Junio 2 del 2022

En sus incios el internet era un sistema de mensajería. Realmente no había mucho que se pudiera hacer si es que uno no trabajaba para el gobierno, pues al ser una red estrictamente hecha para solventar una excepción militar, tendía a tener poco atractivo para el público en general. Eso cambió a principios de la década de los 90, pues Tim Bernes-Lee desarrolló una tecnología la cuál dotaría al internet de las capacidades que le permitieron cambiar el mundo. Estoy hablando de la web. El sistema de transmisión de información por excelencia de internet.

No vamos a entrar en detalles técnicos, solo es indispensable saber que a día de hoy, cualquier sitio que visites en internet es considerado una web. Parte de la WWW. Este génio de la computación, si me permiten llamarlo así, también desarrolló una primera y primitiva forma de HTML, lenguaje por medio del cuál se estructura la web.

La web tiene actualmente una increíble capacidad, desde páginas de información, cómo wikipedia; hasta servicios de transmisión en tiempo real. Las redes sociales, los servicios de desarrollo en línea. Todo eso nos es posible gracias a la web. Pero, ésta no siempre fue así pues en sus inicios solo contaba con una herramienta para construir contenido: HTML. El cuál fue muy útil al principio, y lo sigue siendo. Pero se quedaba corto para lo que los desarrolladores buscaban crear. Luego, llegaron las hojas de estilos y los scripts. Es decir, CSS y JavaScript.

Cabe mencionar que de estos tres solo se considera cómo un lenguaje de programación a JavaScript, HTML es un lenguaje de marcado de hipertexto y CSS es un lenguaje para hojas de estilo en cascada.

Con la llegada de estos la web creció de forma exponencial. Los desarrolladores que sabían estas tecnologías "tenían el mundo en sus manos" pues poseían la capacidad de construir todo cuánto imaginaran. Pero, eso no era

suficiente, la web necesitaba más. Los desarrolladores no eran capaces de llevar el ritmo, la web crecía muy rápido. Fue entonces cuando se dió un paso adelante con un sistema llamado CGI o Puerta de Enlace Común.

El CGI cambió la web para siempre, pues este sistema permitía a un desarrollador utilizar programas externos al navegador para generar HTML de forma dinámica. La capacidad de los programadores creció de forma exponencial. Ahora podían escribir programas que les automatizaran los tediosos trabajos que supone escribir una web con solo HTML. Sin embargo, CGI seguía siendo repetitivo y muchas veces se hacía difícil reutilizar el código por lo que se necesitaba una nueva solución.

Para esto llegó PHP (PreProcesador de Hipertexto) el cuál es a día de hoy el lenguaje de programación del lado del servidor más popular en todo el mundo. Ahora, te preguntará que a qué me refiero con "lenguaje de programación del lado del servidor". Pues, resulta que en la web tenemos dos mundos separados en los que poder trabajar. Por un lado, el más bonito, tenemos el llamado Frontend. Este es un conjunto de tecnologías que permiten desarrollar la interfaz con la que va a interactuar el cliente (humano que usa el computador). Un ejemplo muy claro de esto es la tupla CSS y JavaScript juntos estos dos le permiten a un desarrollar las interfaces más hermosas y útiles. Estos programas siempre son ejecutados en la máquina del cliente lo que se conoce como procesamiento y ejecución en local. Por consiguiente, su opuesto es el Backend, este comprende todo el conjunto de tecnologías que se ejecutan y procesan sus datos dentro del servidor. Por ejemplo, PHP.

Ahora puede que estes un poco confundido con todo esto de servidores y clientes y web. Para hacerte las cosas más fáciles te voy explicar cómo funciona la web: Tenemos un cliente, normalmente es tu ordenador el que se denomina cliente. Luego tenemos un servidor, este es un computador dedicado exclusivamente a recibir, procesar y devolver datos. A diferencia de los clientes estos no suelen contar con una interfaz gráfica. Y, en medio de estos se encuentra el internet, haciendo posible su comunicación. La idea es simple, tal cuál la explicación brindada en el capítulo anterior. Un cliente inicia una petición, supón que quieres entrar a <https://google.com>, por ahora no te distraigas con lo demás y concéntrate en el nombre el sitio **google.com**; pues, para ello necesitas tener una copia del código fuente del sitio web. Para eso está el servidor. Una vez que le pidas a tu máquina abrir **google.com** esta se comunicará con una red de servidores en busca del archivo. Esto lo hace por medio del internet. Una vez que haya localizado el archivo este será descargado por el cliente y podrá ser visualizado por el

usuario.

Cómo supongo que te podrás imaginar, este proceso no es eficiente, pues en una red con millones de millones de nodos cómo lo es el internet actual no es viable que cada cliente busque por la red el o los archivos necesarios para visualizar la web solicitada. Los tiempos de espera para las personas se irían por los cielos. Para este problema se inventó una solución y es, de forma poco intuitiva, añadir otro servidor en medio. Repasemos, originalmente teníamos a dos equipos conectados por medio de internet, cliente y servidor. Pues, se propuso añadir un servidor extra en medio de estos dos. Este servidor es conocido como DNS y su objetivo no es otro que ser un directorio. Con este nuevo servidor en medio, el cuál está lleno de registros que almacenan la ubicación de los archivos más comunmente solicitados los clientes solo tienen que preguntar a estos donde está la información que necesitan y luego solicitarle directamente al servidor indicado.

Ahora tenemos una gran cantidad de componente, servidores, clientes, servidores DNS, HTML, CSS, JavaScript y PHP. Pero, nos surge un nuevo problema cómo el lector supondrá es muy común que el ser humano quiera superar los límites de las cosas. Y la web no iba a ser la excepción. Con la llegada de internet y la web llegaron también los mal llamados Hackers. Esas personas que por gusto o profesión se hacen expertas en temas de las web y que gracias a sus conocimientos son capaces de vulnerar sistemas y adentrarse en los almacenes de información más recónditos. Por supuesto que no estamos indefensos antes ellos, existe una rama de la ingeniería de la información llamada seguridad informática, la cuál se encarga de idear e implementar contramedidas para este tipo de atacantes. Sin embargo, no es algo fácil de aprender y pues los desarrolladores suelen tener bastante para estudiar solo con las tecnologías anteriormente mencionadas. Imagina tener que aprender, a demás, a implementar técnicas y algoritmos para proteger un sitio web. Se hace insostenible. Y especialmente PHP es un lenguaje muy propenso a dejar huecos de seguridad, esto debido a su versatilidad y el hecho de que no suele ser muy estricto con los programadores, lo que promueve la creación de código mal hecho.

Para solventar este problema, llegaron los Frameworks, estos son conjuntos de programas, los cuales buscan facilitarle al desarrollador todo el proceso de creación de una herramienta. Por ejemplo, existen Frameworks como Django, el cuál le permite al desarrollador centrarse en el diseño y la funcionalidad dejando de lado los aspectos más técnicos y profundos como la seguridad del sitio web. Este fue un gran avance y a día de hoy de lo más

útil que puede aprender un aspirante a desarrollador web. Por esta razón, he optado por añadir una serie de capítulos dirigida únicamente al Framework Django.

Piensa en un framework cómo en un asistente, el cuál se encarga de todo el papeleo aburrido mientras tú te centras en diseñar lo que quieres que se haga.

Siguiente capítulo →