

TEMA 1. SELECCIÓN DE ARQUITECTURAS Y HERRAMIENTAS DE PROGRAMACIÓN

1.	INTRODUCCIÓN.....	2
2.	EVOLUCIÓN Y CARACTERÍSTICAS DE LOS NAVEGADORES WEB.....	2
3.	ARQUITECTURA DE EJECUCIÓN.....	4
4.	LENGUAJES Y TECNOLOGÍAS DE PROGRAMACIÓN EN ENTORNO CLIENTE.....	6
	4.1. HTML.....	6
	4.2. CSS.....	7
	4.3. JAVASCRIPT.....	7
5.	ALGO MÁS.....	7

1. INTRODUCCIÓN

En este primer capítulo presentamos los conceptos necesarios para comprender el contexto de ejecución de páginas web como forma de acceder a recursos de aplicaciones y sistemas de información web. Para ello, introducimos las características principales de los navegadores más comunes utilizados hoy en día. Describiremos, así mismo, los diferentes lenguajes y tecnologías de programación, del lado del cliente, aplicables en este tipo de entornos. Por último, se hace un recorrido introductorio por algunas de las técnicas de integración que el desarrollador tiene a su alcance para intercalar código con las etiquetas **HTML**.

2. EVOLUCIÓN Y CARACTERÍSTICAS DE LOS NAVEGADORES WEB

La *World Wide Web* (o “la Web”, como se conoce comúnmente) representa un universo de información accesible globalmente a través de la red internet. está formada por un conjunto de recursos interconectados que conforman el conocimiento humano actual. el funcionamiento de la Web es posible debido a la coexistencia de una serie de componentes software y hardware. estos elementos abarcan desde los componentes físicos de internet (*hubs*, repetidores, puentes, switches, routers, etc.) y los protocolos de comunicaciones (TCP, IP, HTTP, FTP, SMTP, etc.), hasta la utilización del sistema de nombres de dominio (DNS) para la búsqueda y recuperación de recursos o la utilización de software específico para proveer y consumir dichos recursos.

En este contexto, el desarrollo en entornos web debe tener en cuenta la distribución de los elementos y la función que tiene cada uno de ellos. la configuración arquitectónica más habitual se basa en el modelo denominado *Cliente/ Servidor*, basado en la idea de servicio, en el que el cliente es un componente consumidor de servicios y el servidor es un proceso proveedor de servicios. además, esta relación está robustamente cimentada en el intercambio de mensajes como el único elemento de acoplamiento entre ambos. en este libro, y en este capítulo en concreto, nos vamos a centrar en las características de los componentes software que se utilizan en el cliente.

Uno de los componentes más habituales en el cliente es el navegador web, que permite acceder al contenido ofrecido por los servidores de internet sin la necesidad de que el usuario instale un nuevo programa (con excepciones). Podemos encontrarnos muchos tipos de clientes en función de sus capacidades, los lenguajes soportados o las facilidades de configuración. los más livianos, o “ligeros”, son los que por sí solos no pueden ejecutar ninguna operación real más allá de la de conectarse al servidor. sin embargo, actualmente la tendencia es a disponer de clientes complejos, que utilizan lenguajes como Java o funciones avanzadas en **DHTML** para otorgar mayor funcionalidad y flexibilidad al usuario. estos navegadores pueden no solo conectarse al servidor, sino que también son capaces de procesar o sincronizar datos para su uso sin necesidad de que el usuario intervenga.

En cualquier caso, debemos entender que un navegador web, o explorador web (*browser*), es una aplicación, distribuida habitualmente como software libre, que permite a un usuario acceder (y, normalmente, visualizar) a un recurso publicado por un servidor web a través de internet y descrito mediante una dirección **URL** (*Universal Resource Locator*). Como ya hemos dicho, lo más habitual es que utilicemos los exploradores web para “navegar” por recursos de tipo hipertexto, comúnmente descritos en **HTML**, ofrecidos por servidores web de todo el mundo a través de internet.

Desde la creación de la Web a principios de los años 90, los navegadores web han evolucionado desde meros visualizadores de texto que, aunque no ofrecían capacidades multimedia (visualización de imágenes), cumplían su propósito (LINKS, LYNX, W3M); hasta los actuales navegadores, totalmente preparados para soportar cualquier tipo de interacción y funcionalidad requerida por el usuario. A continuación, describimos una pequeña lista de algunos de los exploradores más relevantes a lo largo de la corta historia de los clientes de navegación web:

- ✓ **Mosaic.** se considera uno de los primeros navegadores web y el primero con capacidades gráficas. las primeras versiones se diseñaron para ser ejecutado sobre Unix, pero, debido a su gran aceptación, pronto fue portado a las plataformas de Windows y Macintosh. se utilizó como base para las primeras versiones de Internet Explorer y Mozilla. su desarrollo se abandonó en 1997.
 - ✓ **Netscape Navigator** (después **Communicator**). Fue el primer navegador en incluir un módulo para la ejecución de código *script* (Javascript). se le considera como el perdedor de la “Guerra de los navegadores”, que tuvo lugar entre netscape y Microsoft por el dominio del mercado de navegadores web a finales de los años 90. sus características, sin embargo, se consideran la base de otros navegadores, como Mozilla Firefox.
 - ✓ **Internet Explorer.** es el navegador de Microsoft. su cuota de distribución y uso ha sido muy elevada gracias a su integración con los sistemas Windows. en los últimos años su utilización ha ido descendiendo paulatinamente debido al aumento de usuarios que optan por otros navegadores, como Firefox o Chrome. a fecha de hoy es totalmente obsoleto y Microsoft lo ha sustituido por el nuevo Edge.
 - ✓ **Edge** es el nuevo navegador de Microsoft, que llegó con Windows 10, para recuperar el terreno perdido en el mercado de navegadores con Internet Explorer. Edge es mucho más limpio y simple que Internet Explorer. Se parece mucho a los navegadores Google Chrome y Opera. Trae funciones nuevas e integra a Cortana, y su motor de procesamiento funciona igual que el resto de los navegadores. Microsoft busca que Edge sea la alternativa frente a Chrome y Firefox.
 - ✓ **Mozilla Firefox.** se trata de un navegador de código abierto multiplataforma de gran aceptación en la comunidad de desarrolladores web. existen gran variedad de utilidades, extensiones y herramientas que permiten la personalización tanto del funcionamiento del navegador como de su apariencia. Fue uno de los primeros en incluir la navegación por pestañas. además, es un navegador multiplataforma, lo que le ha llevado a poder recortar parte de la cuota de distribución que desde los inicios de la década de los 2000 venía teniendo Internet Explorer.
- Google Chrome.** De reciente creación (septiembre de 2008), es el navegador de Google compilado a partir de componentes de código abierto. en boca de sus desarrolladores, sus características principales son la seguridad, velocidad y estabilidad. en numerosas pruebas comparativos este navegador ha demostrado ser uno de los más rápidos y seguros gracias, entre otras razones, a estar construido siguiendo una arquitectura multiproceso en la que cada pestaña es ejecutada de forma independiente.
- Safari.** es el navegador por defecto de los sistemas de apple, aunque también se han desarrollado versiones para su funcionamiento en las plataformas Windows. las últimas versiones incorporan las características habituales de navegación por pestañas, corrector ortográfico en formularios, almacenamiento de direcciones favoritas (“marcadores”), bloqueador de ventanas emergentes, soporte para motores de búsqueda personalizados o un gestor de descargas propio.
- ✓ **Dolphin Browser.** Debido al auge de los dispositivos móviles inteligentes (*smartphones* y *tablets*) y de los sistemas operativos para estos, tenemos que hacer referencia obligatoriamente a uno de los navegadores más populares para estas plataformas. específico para el sistema operativo android, fue uno de los primeros en incluir soporte para navegación multitáctil. Utiliza un motor de renderizado de páginas similar al de Chrome o safari.
 - ✓ **Opera.** Es un navegador web creado por la empresa noruega Opera Software. Usa el motor de renderizado Blink. Tiene versiones para computadoras de escritorio, teléfonos móviles y tabletas. Los sistemas operativos compatibles con Opera de escritorio son Microsoft Windows, macOS y GNU/Linux entre otros.
 - ✓ **Vivaldi.** Es verdaderamente único. No hay dos usuarios Vivaldi con la misma configuración. Destaca en personalización y posee características que mejoran la privacidad, por defecto las búsquedas son sin seguimiento (soporte duckduckgo).

Google, webkit, duckduckgo, yahoo, Yandex, Baidu

En la siguiente gráfica puede verse una tendencia de la utilización de navegadores

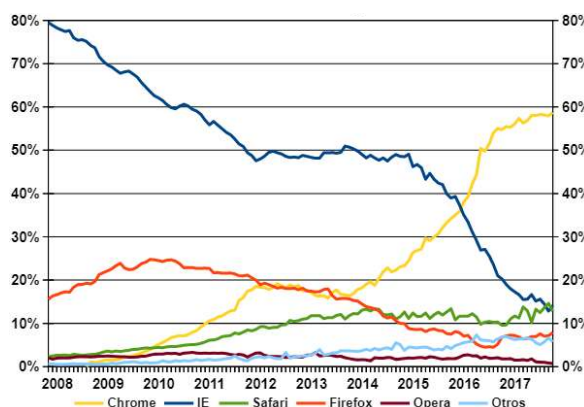


Figura 1.1. Estadísticas de uso de navegadores (2008-2017)

Podemos diferenciar los navegadores anteriores de acuerdo a una serie de criterios:

- ✓ **Plataforma de ejecución.** no todos los navegadores pueden ser ejecutados en cualquier sistema operativo. Safari, por ejemplo, es exclusivo de los sistemas de apple aunque tiene versiones para Windows.

Características del navegador. la mayoría de los navegadores ofrecen funcionalidades adicionales asociadas a la experiencia del usuario a la hora de navegar por la red. algunas de las características soportadas de forma nativa (sin necesidad de instalar extensiones) incluyen la administración de marcadores, gestores de descarga, almacenamiento seguro de contraseñas y datos de formularios, corrección ortográfica o definición de herramientas de búsqueda.

Personalización de la interfaz. las funciones de accesibilidad que definen la experiencia del usuario con un navegador web también son un aspecto diferencial. entre los aspectos más destacados podemos mencionar el soporte para la navegación por pestañas, la existencia de bloqueadores de ventanas emergentes, la integración de visualizadores de formatos de ficheros (como PDF), opciones de *zoom* o funciones avanzadas de búsqueda de texto en páginas web.

Soporte de tecnologías Web. actualmente, una de las mayores preocupaciones de los desarrolladores de navegadores web es el grado de soporte de los estándares de la Web. Por ello, podemos clasificar los navegadores de acuerdo con su nivel de soporte de tecnologías como **CSS** (hojas de estilo en cascada), Java, lenguajes de *scripting* del cliente (Javascript), RSS (sindicación de contenidos), **XHTML** (**HTML** con formato de **XML**), etc.

Licencia de software. existen navegadores de código libre, como Mozilla Firefox (licencia GNU GPL) o Google Chrome (licencia BSD), y navegadores propietarios Edge (Microsoft) o safari (apple). Salvo raras excepciones (omniWeb) todos los navegadores son gratuitos.

3. ARQUITECTURA DE EJECUCIÓN

Cada navegador web tiene su propia forma de interpretar la interacción con un usuario. El resultado de esta interacción, en cualquier caso, se inicia con el usuario indicando la dirección del recurso al que quiere acceder y termina con la visualización del recurso por parte del navegador en la pantalla del usuario (salvo interacciones posteriores del usuario con la página). La forma de realizar este proceso depende del propósito del navegador y de la configuración del mismo. De esta forma, un navegador puede estar más centrado en ofrecer una respuesta más rápida, en mostrar una respuesta más fiel al contenido del recurso obtenido, en priorizar los aspectos de seguridad de las comunicaciones con el servidor, etc.

Para poder llevar a cabo el proceso descrito anteriormente, cada navegador está formado por una serie de elementos y componentes determinados que conforman lo que se denomina *arquitectura del navegador*. a pesar de que cada navegador tiene su propia arquitectura, la gran

mayoría de ellos coinciden en una serie de componentes básicos y comunes en todos ellos, es lo que llamamos *arquitectura de referencia*.

Los componentes básicos incluidos en la arquitectura de referencia de un navegador web son los que pueden verse en la siguiente figura:

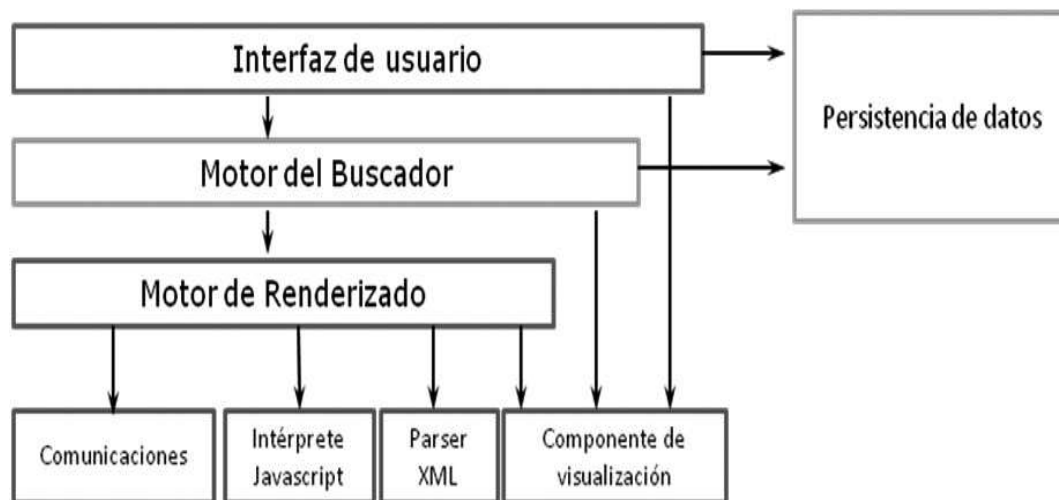


Figura 1.2. Arquitectura de referencia de un navegador web

Los componentes de esta arquitectura de referencia son:

Subsistema de interfaz de usuario: es la capa que actúa de interfaz entre el usuario y el motor del buscador (o de navegación). ofrece funcionalidades tales como la visualización de barras de herramientas, progreso de carga de la página, gestión inteligente de las descargas, preferencias de configuración de usuario o impresión. En algunos casos puede comunicarse con el sistema operativo para el manejo de sesiones de usuario o el almacenamiento de preferencias de visualización o configuración.

Subsistema del motor del buscador o motor de navegación: este subsistema es un componente que ofrece una interfaz de alto nivel para el motor de renderizado. su función principal es la de cargar una dirección determinada (URL o URI) y soportar los mecanismos básicos de navegación tales como ir a la página anterior o siguiente o la recarga de la página. Además, es el componente que gestiona las alertas de Javascript (mensajes de usuario) y el proceso de carga de una página (es quien le provee de información a la interfaz de usuario al respecto). Finalmente, es el encargado de consultar y administrar las preferencias de ejecución del motor de renderizado.

Subsistema de renderizado: este componente es el encargado de producir una representación visual del recurso obtenido a partir del acceso a una dirección web. el código de una página web es *interpretado* por este módulo. En función de los lenguajes, estándares y tecnologías soportadas por el navegador, este módulo será capaz de mostrar documentos **HTML**, **XML**, hojas de estilo **CSS** e incluso contenido embebido en la página (audio/vídeo) e imágenes. Además, este módulo establece las dimensiones exactas de cada elemento a mostrar y, en ocasiones, es el responsable de posicionar dichos elementos en una página.

Algunos de los motores de renderizado más utilizados son:

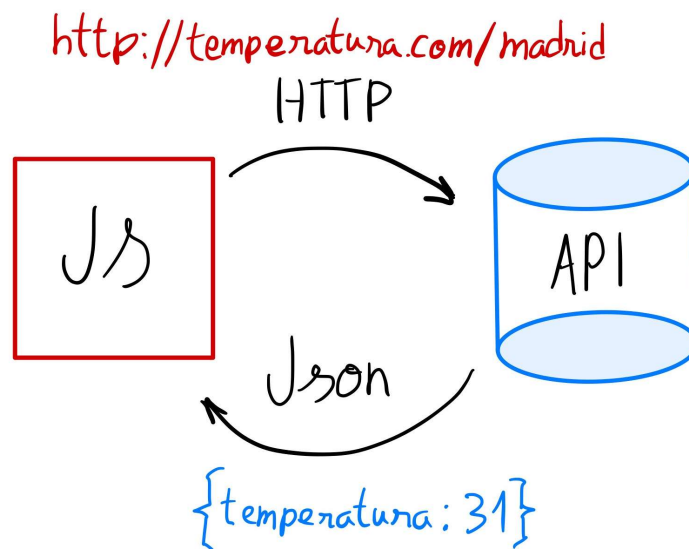
Servo, utilizado en Mozilla con el apoyo de Samsung.

WebKit, el motor de Google Chrome para Mac y Safari.

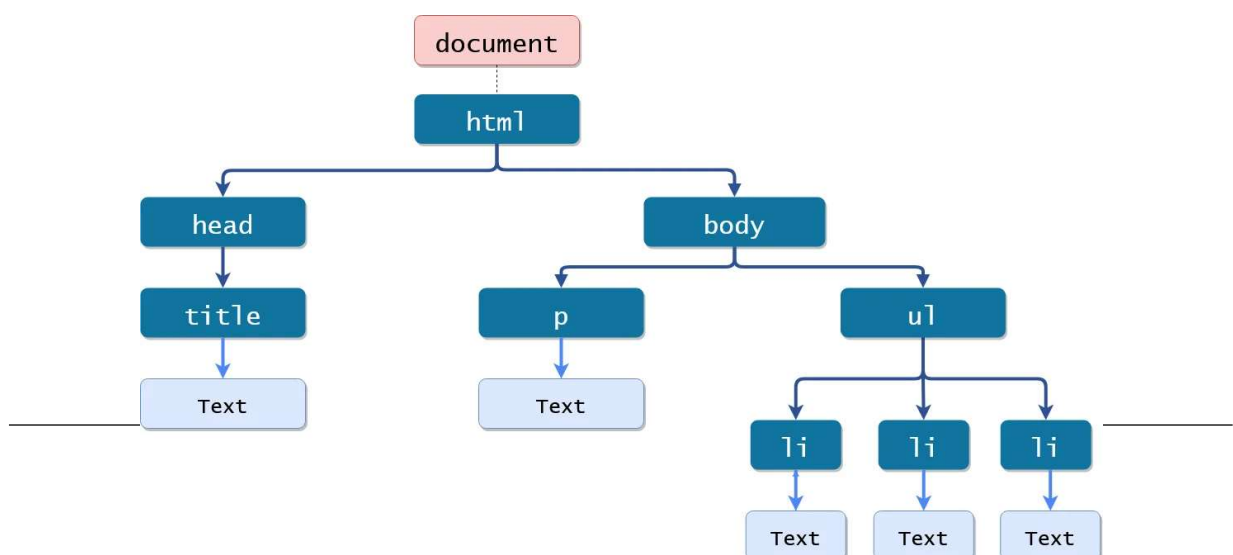
Blink, motor de búsqueda de Chrome, Edge y Brave.

Subsistema de comunicaciones. Es el subsistema encargado de implementar los protocolos de transferencia de ficheros y documentos utilizados en internet (HTTP, FTP, etc.). además, es el responsable de identificar la codificación de los datos obtenidos en función de su tipo, de tal forma que es capaz de identificar si el recurso obtenido es de tipo texto, audio, vídeo, etc. (codificado en estándar [MIME](#), *Multipurpose Internet Mail Extensions*). Dependiendo de las capacidades personalizadas para el navegador, este subsistema puede almacenar una caché de elementos accedidos recientemente.

Intérprete de Javascript. Las páginas **HTML** habitualmente llevan código intercalado para la provisión de ciertas funcionalidades al usuario como puede ser la respuesta a ciertos eventos del ratón o del teclado. el lenguaje comúnmente aceptado para la programación de este código embebido es Javascript. El intérprete de Javascript será el encargado de analizar y ejecutar dicho código. este módulo puede ser configurado (e incluso deshabilitado) por cuestiones de seguridad o facilidad de navegación desde el motor de navegación o el motor de renderizado (por ejemplo, para evitar que aparezcan ventanas emergentes). La existencia de módulos de interpretación de código difiere de un navegador a otro. Por ello, es posible que existan subsistemas intérpretes de otros lenguajes, como *applets* de Java o **AJAX**.



Parser XML. Con el fin de poder acceder más fácilmente a los contenidos definidos en un documento **HTML** (en realidad **XHTML**), los navegadores web suelen incluir un módulo (*parser*) que permite cargar en memoria una representación en árbol (árbol **DOM**, *Document Object Model*) de la página. De esta forma, el acceso a los diferentes elementos de una página por parte del navegador es mucho más rápido.



Componente de visualización. este subsistema ofrece funcionalidades relacionadas con la visualización de los contenidos de un documento **HTML** en una página web. ofrece primitivas de dibujo y posicionamiento en una ventana, un conjunto de componentes visuales predefinidos (*widgets*) y un conjunto de fuentes tipográficas a los subsistemas principales del navegador web. suele estar muy relacionado con las librerías de visualización del sistema operativo.

Subsistema de persistencia de datos. Funciona como almacén de diferentes tipos de datos para los principales subsistemas del navegador. Estos datos suelen estar relacionados con el almacenamiento de historiales de navegación y el mantenimiento de sesiones de usuario en disco. otros datos de alto nivel que también son gestionados por este subsistema incluyen las preferencias de configuración del navegador (de barras de herramientas, por ejemplo) o el listado de marcadores. A bajo nivel, este sistema administra también los certificados de seguridad y las *cookies*

4. LENGUAJES Y TECNOLOGÍAS DE PROGRAMACIÓN EN ENTORNO CLIENTE

Los lenguajes de programación del entorno de cliente son aquellos que se ejecutan en el navegador web, dicho de otro modo, en el lado del cliente dentro de una arquitectura Cliente/servidor. El lenguaje cliente principal es **HTML** (lenguaje de marcado de hipertexto, *HyperText Markup Language*), ya que la mayoría de las páginas del servidor son codificadas siguiendo este lenguaje para describir la estructura y el contenido de una página en forma de texto. existen algunas alternativas y variaciones de este lenguaje tales como **XML** (lenguaje de marcas extensible, *eXtensible Markup Language*) o **XHTML** (*eXtensible HTML*).

4.1. HTML

El *Hyper Text Markup Language* (lenguaje de marcado de hipertexto) es el lenguaje de marcas de texto más utilizado en la *World Wide Web*. Fue creado en 1989 por Tim Berners-Lee a partir de dos elementos previos para crear dicho lenguaje: por un lado, el concepto de *hipertexto* como herramienta básica para conectar dos elementos (documentos o recursos) entre sí; y SGM, como lenguaje básico para colocar etiquetas o marcas en un texto. Debemos tener en cuenta que **HTML** no es propiamente un lenguaje de programación como puede ser Java, sino que se basa en la utilización de un sistema de etiquetas cerrado aplicado a un documento de texto. además, este lenguaje no necesita ser compilado, sino que es interpretado (ejecutado a medida que se avanza por el documento **HTML**). Una característica particular de **HTML** es que, ante algún error de sintaxis que presente el texto, **HTML** no lo detectará y seguirá con la interpretación del siguiente fragmento de documento.

Con el lenguaje **HTML** se pueden hacer gran variedad de acciones, desde organizar simplemente el texto y los objetos de una página web, pasando por crear listas y tablas, hasta llegar a la esencia de la Web: los hipervínculos. Un hipervínculo es un enlace de una página web o un archivo a otra página web u otro archivo. Cuando un usuario hace clic en el hipervínculo, el destino se mostrará en un explorador web, se abrirá o se ejecutará, en función del tipo de recurso destino. El destino es con frecuencia otra página web, pero también puede ser una imagen, un archivo multimedia, una dirección de correo electrónico, un programa, etc.

Podríamos decir que **HTML** es el lenguaje más importante en el ámbito de la *World Wide Web* puesto que casi todos los lenguajes utilizados en la Web, de acuerdo con la tendencia mostrada en la última década, terminan confluyendo hacia una representación en **HTML** para que nuestro navegador lo pueda leer y visualizar en la pantalla del cliente.

4.2. CSS

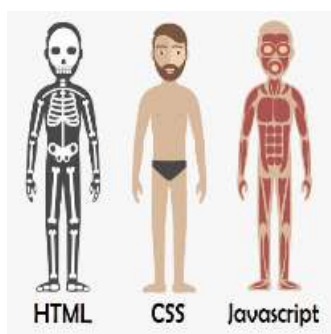
Las **CSS** (Hojas de estilo en Cascada, *Cascade Style Sheets*) sirven para separar el formato que se quiere dar a la página web de la estructura de la página web y las demás instrucciones. Utilizamos **CSS**, por ejemplo, cuando queremos que en determinados párrafos de nuestra página web se use

un determinado tipo y tamaño de letra, un color de fuente y un color de fondo. en vez de tener que definir párrafo por párrafo todos los atributos del formato que queremos dar, solo hace falta que lo definamos una vez, en la hoja de estilo (**CSS**). Nos basta con poner una referencia en nuestro documento **HTML** que la dirija al formato que queramos darle, definido en la hoja de estilo. De esta forma, solo debemos poner esa referencia en cada párrafo, en vez de especificar el formato uno por uno.

4.3. JAVASCRIPT

Javascript es un lenguaje de programación de *scripting* (interpretado) y, normalmente, embebido en un documento **HTML**. Se define como orientado a objetos y con características dinámicas. Se utiliza principalmente su forma del lado del cliente, con un intérprete implementado como parte de un navegador web. su objetivo principal es el de permitir realizar mejoras en la interfaz de usuario y, de esta forma, crear páginas web dinámicas. existe, no obstante, una forma de Javascript del lado del servidor.

Inicialmente, se diseñó con una sintaxis similar al lenguaje C, aunque adopta nombres y convenciones propias del lenguaje de programación Java. Sin embargo, tenemos que dejar claro que Java y Javascript no están relacionados y propósitos diferentes.



5. ALGO MÁS...

- ✓ Durante el tema hemos hablado de la “Guerra de los Navegadores” ¿Qué sucedió exactamente? ¿Pensáis que actualmente con la cantidad de navegadores disponibles estamos ante una “Segunda Guerra de los Navegadores”?
- ✓Cuál es el motor que utiliza Microsoft Edge y cuál es el motor que utiliza Opera.
- ✓ Crees que Microsoft Edge puede convertirse en una alternativa frente a Chrome o Firefox.
- ✓ Qué es la W3C
- ✓ Qué es Developer Mozilla. (<https://developer.mozilla.org/es/>)