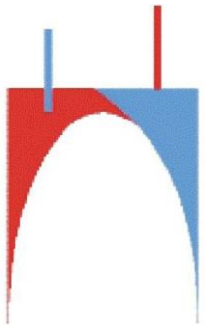


Instituto de Enseñanza Secundaria

Santiago Hernández

DESARROLLO WEB EN ENTORNO CLIENTE

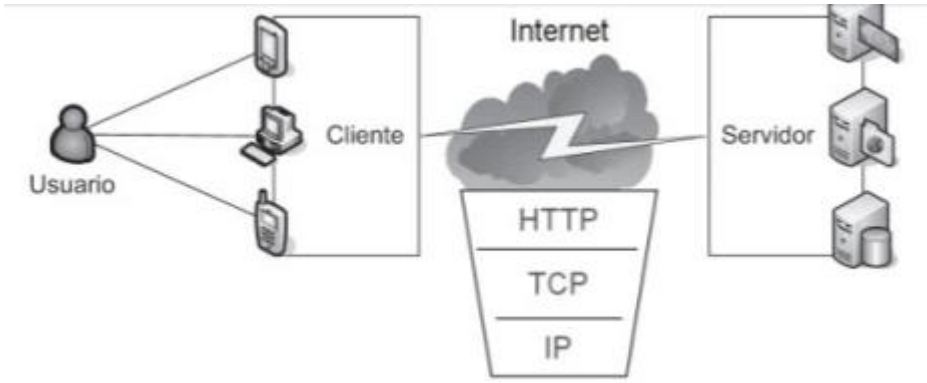
UD1
SELECCIÓN DE ARQUITECTURAS Y HERRAMIENTAS DE PROGRAMACIÓN

ÍNDICE

MODELOS DE PROGRAMACIÓN EN ENTORNO CLIENTE/SERVIDOR

Arquitectura Cliente-Servidor

- Se basa en la idea de **servicio**, en el que el cliente es un componente consumidor de servicios y procesos proporcionados por servidores.
- Habitualmente, el cliente inicia el intercambio de información, solicitando datos al servidor, que responde enviando uno o mas flujos de datos al cliente.
- Los usuarios se conectan y realizan peticiones utilizando navegadores.



MODELOS DE PROGRAMACIÓN EN ENTORNO CLIENTE/SERVIDOR

➢ Ejecución de código en el cliente

- El cliente solicita un contenido web
- El servidor envía al cliente un contenido que puede contener código
- El código se procesa en la máquina cliente
- El código se puede conectar a bibliotecas de terceros para proporcionar funciones avanzadas

➢ Ventajas

- Descargar de trabajo al servidor
- Ejecución de scripts más rápida
- Gestión interna de errores
- El usuario puede ver el código fuente

Inconvenientes

- Mayor tiempo de carga (bloqueos)
- Frameworks menos potentes que en servidor.
- El navegador debe entender el lenguaje de programación

MODELOS DE PROGRAMACIÓN EN ENTORNO CLIENTE/SERVIDOR

- Los servidores web albergan contenidos (locales o remotos)

Protocolo TCP/IP

- Los clientes (navegadores) solicitan contenidos

MODELOS DE PROGRAMACIÓN EN ENTORNO CLIENTE/SERVIDOR

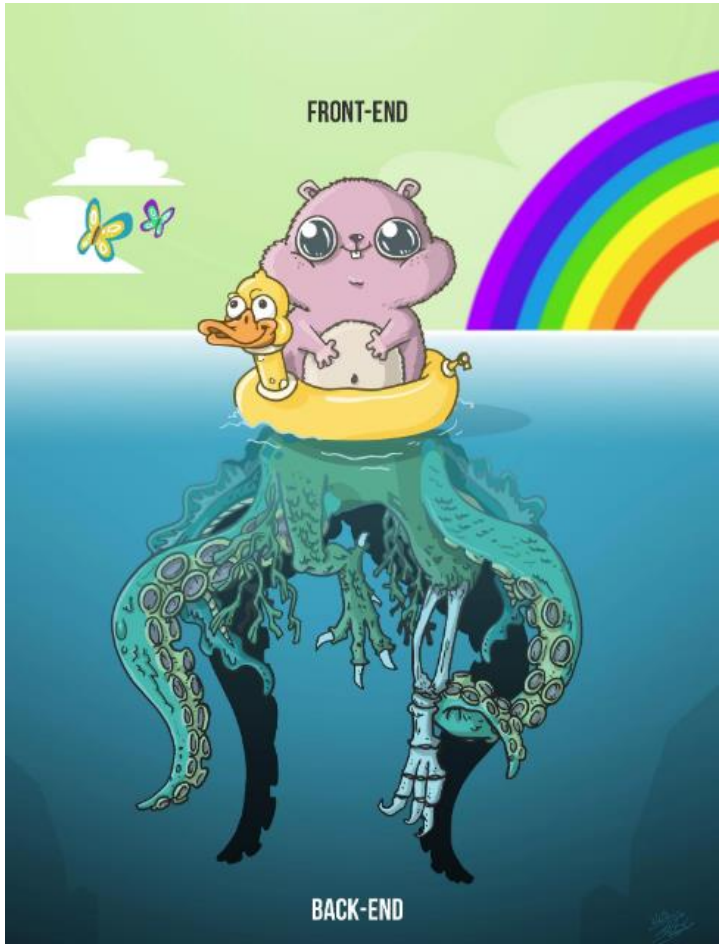
La **programación por capas** indica como se distribuye el software.

- **Capa de presentación**
  - Presenta una interfaz gráfica del recurso solicitado y sirve para recoger su interacción.
  - La programación de esta capa se centra en el formateo de la información enviada por el servidor y la captura de las acciones realizadas por el cliente.
- **Capa de negocio**
  - Capa donde se lleva a cabo toda la lógica de la aplicación.
- **Capa de datos**
  - Es donde residen los datos y la encargada de acceder a los mismos.

MODELOS DE PROGRAMACIÓN EN ENTORNO CLIENTE/SERVIDOR

Teniendo en cuenta en que lado se ubican las tecnologías existen distintos perfiles de desarrollo web:

- **Front-end**:
  - Se encarga del diseño y maquetación de la aplicación web utilizando tecnologías como HTML, CSS y Javascript (y sus frameworks).
  - También de la correcta presentación en cualquier tipo de dispositivo e incluso del posicionamiento en buscadores.
- **Back-end**:
  - Se encarga del lado servidor utilizando tecnologías como Java, .NET o Python.
  - También se encarga de la administración del servidor de aplicaciones y la Base de Datos.

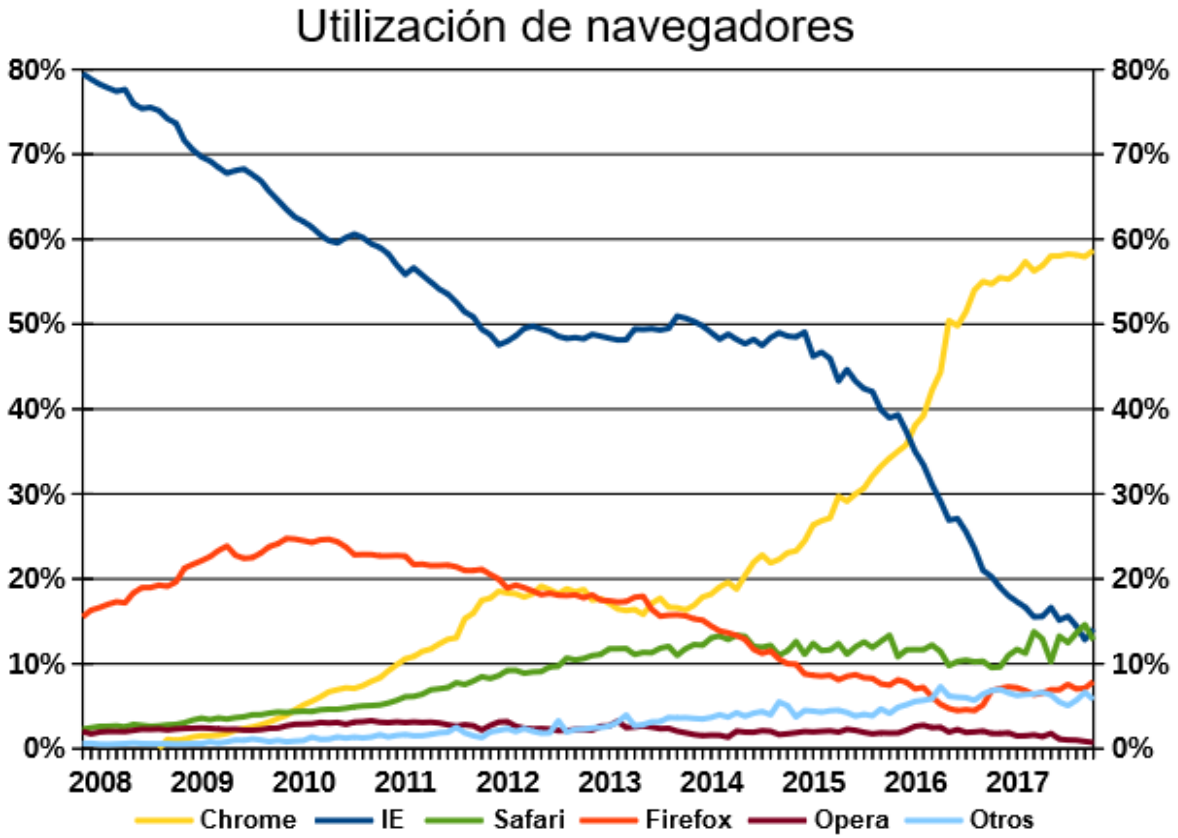


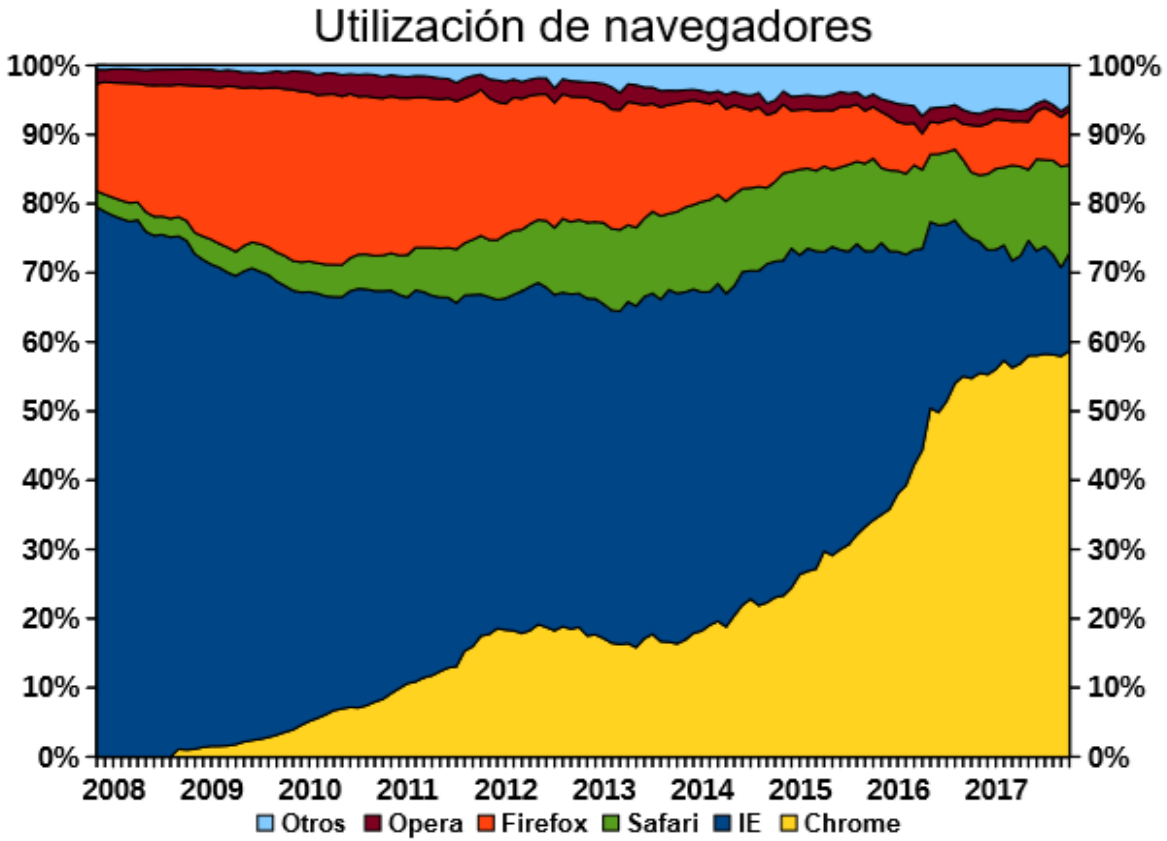
EL NAVEGADOR WEB

- Es un software que permite el acceso a internet, interpretando la información de distintos tipos de archivos y sitios web para que estos puedan ser visualizados.
- Los navegadores Web procesan la etiquetas **HTML** e interpretan los lenguaje de **script** (javascript...).
- Su funcionalidad básica es **permitir la visualización** de documentos de texto, posiblemente con recursos multimedia incrustados.
- Los documentos que muestra el navegador pueden estar ubicados en la computadora del usuario o pueden ser **proporcionados por un servidor web**.
- Estos documentos o páginas web poseen hipervínculos que enlazan documentos. El seguimiento de estos enlaces se llama navegación, de ahí el nombre **navegador**.
- Los **estándares web** son un conjunto de recomendaciones dadas por el World Wide Web Consortium W3C) y otras organizaciones internacionales acerca de **cómo crear e interpretar** documentos basados en la web.
  - Para comprobar si un documento Web cumple el estándar: https://validator.w3.org/

EL NAVEGADOR WEB – PRINCIPALES NAVEGADORES

Uso de los navegadores 2008 - 2018





http://www.mclibre.org/consultar/htmlcss/otros/historia-navegadores.html

https://netmarketshare.com https://netmarketshare.com

https://www.w3schools.com/browsers/default.asp

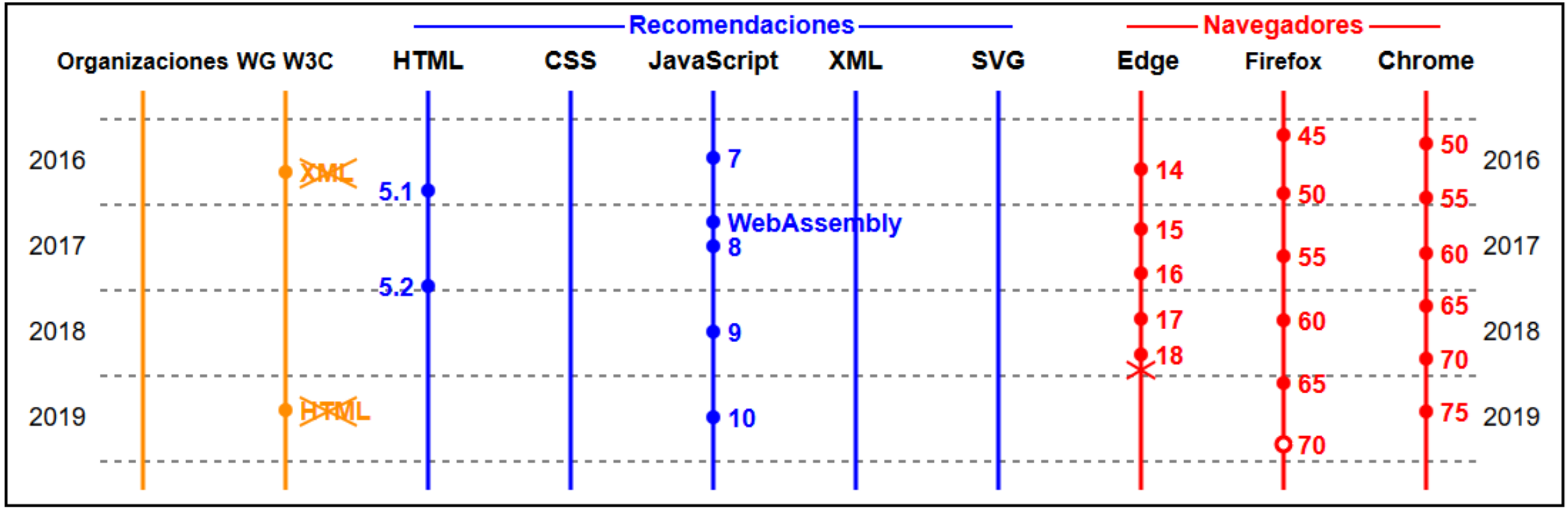
EL NAVEGADOR WEB – PRINCIPALES NAVEGADORES

- **Microsoft Edge (Antiguo Internet Explorer)**
  - Está diseñado para ser un navegador web ligero con un motor de renderizado de código abierto construido en torno a los estándares web.
- **Mozilla Firefox**
  - Navegador de código abierto desarrollado por la Corporación Mozilla.
  - Usa el motor Gecko para renderizar páginas webs, que implementa estándares web.
- **Google Chrome**
  - Navegador web desarrollado por Chrome. Usa el motor de renderizado WebKit.
- **Safari**
  - Navegador web de código cerrado desarrollado por Apple Inc.
- **Opera**
  - Navegador web creado por la empresa noruega Opera Software. Usa el motor de renderizado Blink.

EL NAVEGADOR WEB

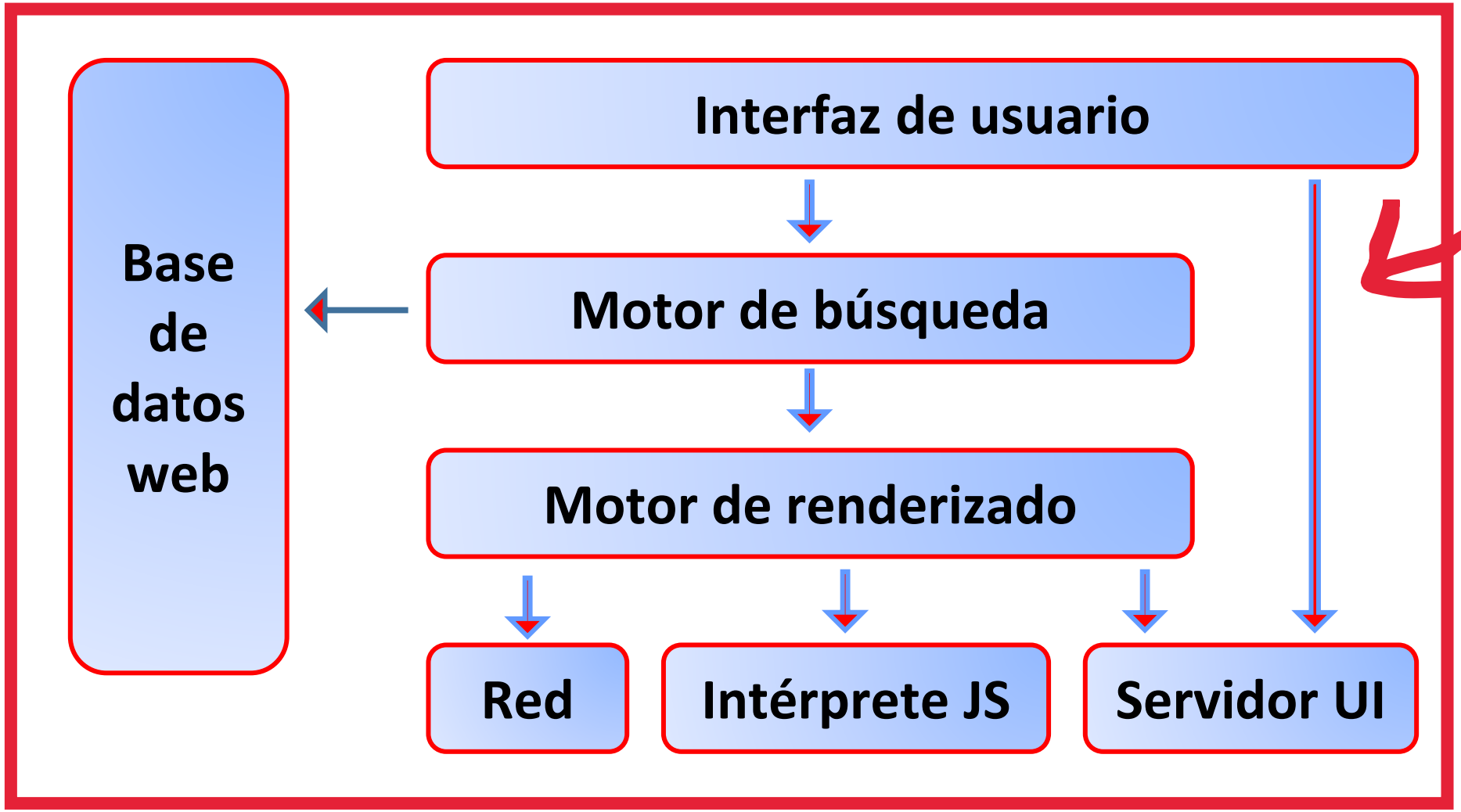
Estado del arte

El navegador es probablemente el software que más utilizamos



http://www.mclibre.org/consultar/htmlcss/otros/historia-resumen.html
https://netmarketshare.com

EL NAVEGADOR WEB - COMPONENTES



EL NAVEGADOR WEB - COMPONENTES

- **Interfaz de usuario**
  - Capa que actúa de interfaz entre el usuario y el motor del buscador.
  - Visualiza el proceso de carga.
  - Gestiona las descargas de forma inteligente.
  - Visualiza barras de herramientas.
- **Motor de búsqueda**
  - Coordina el interfaz con el motor de renderizacdo.
  - Carga la dirección URL.
  - Soporta los mecanismos básicos de navegación (página anterior o siguiente, recarga de página…).
  - Gestiona las alertas de javascript.
  - Gestiona y administra las preferencias de ejecución del motor de renderizado.

importante!!

EL NAVEGADOR WEB - COMPONENTES

❑ Motor de renderizado

- Encargado de producir una representación visual del recurso obtenido.
- Interpreta el código de la página Web.
- En función de las tecnologías soportadas será capaz de mostrar documentos HTML o XML, hojas de estilo CSS, imágenes o contenido embebido (audio/vídeo).
- Establece las dimensiones exactas de cada elemento a mostrar y la posición de estos.
- Los motores de renderizado más conocidos son:
  - Gecko (Mozilla)
  - Trident (Internet Explorer)
  - WebKit (Chrome y Safari)
  - Blink (Opera)
  - Tasman (Internet Explorer)

importante

edge --- blink (desde 2019)

EL NAVEGADOR WEB - COMPONENTES

❑ Red o subsistema de comunicaciones

- Implementa los protocolos de transferencia de ficheros y documentos.
- Identifica la codificación de los datos obtenidos en función de su tipo (texto, audio, vídeo, ...)
- Puede almacenar una caché de elementos accedidos recientemente.

❑ Intérprete de JavaScript

- Analiza y ejecuta el código de JavaScript intercalado en HTML.
- Puede ser configurado, e incluso deshabilitado desde el motor de renderizado.
- Cada navegador tiene sus propios módulos de interpretación, por lo que es posible que existan subsistemas intérpretes de otros lenguajes, como applets de Java, Ajax o ActionScript.

❑ Parser XML

parser = procesador

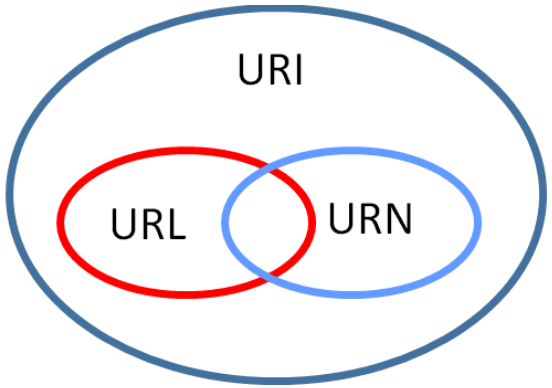
- Permite cargar en memoria una representación de árbol (árbol DOM, Document Object Model) de la página.
- El acceso a los diferentes elementos de una página por parte del navegador es mucho más rápido.

EL NAVEGADOR WEB - COMPONENTES

- **Servidor de la interfaz**
  - Ofrece funcionalidades relacionadas con la visualización de los contenidos de un documento HTML en una página web.
  - Utiliza métodos de UI del SO en segundo plano.
- **Almacenamiento de datos**
  - Funciona como almacén de diferentes tipos de datos para los principales subsistemas del navegador.
  - Suelen estar relacionados con el almacenamiento de historiales de navegación y mantenimiento de sesiones de usuario en disco.
  - Incluye preferencias de configuración del navegador o la lista de marcadores.
  - A bajo nivel, este sistema administra también los certificados de seguridad y cookies.

EL NAVEGADOR WEB – IDENTIFICACIÓN DE RECURSOS

- **URI**: Universal Resource Identifier
  - Cadena de caracteres que **identifica** un recurso ya sea usando ubicación, nombre o ambos.
  - URI puede representar la URL y el URN del recurso al mismo tiempo.
- **URL**: Uniform Resource Locator
  - Cadena de caracteres que hace referencia a una dirección.
  - Es la forma más utilizada para **localizar** recursos en la web.
  - ftp://ftp.is.co.za/rfc/rfc1808.txt
- **URN**: Uniform Resource Name
  - Nombre único para un recurso
  - No da ninguna información para su localización, solo lo identifica.
  - urn:isbn:0451450523



EL NAVEGADOR WEB - ESTRUCTURA DE UNA URI

Información contenida en una URI

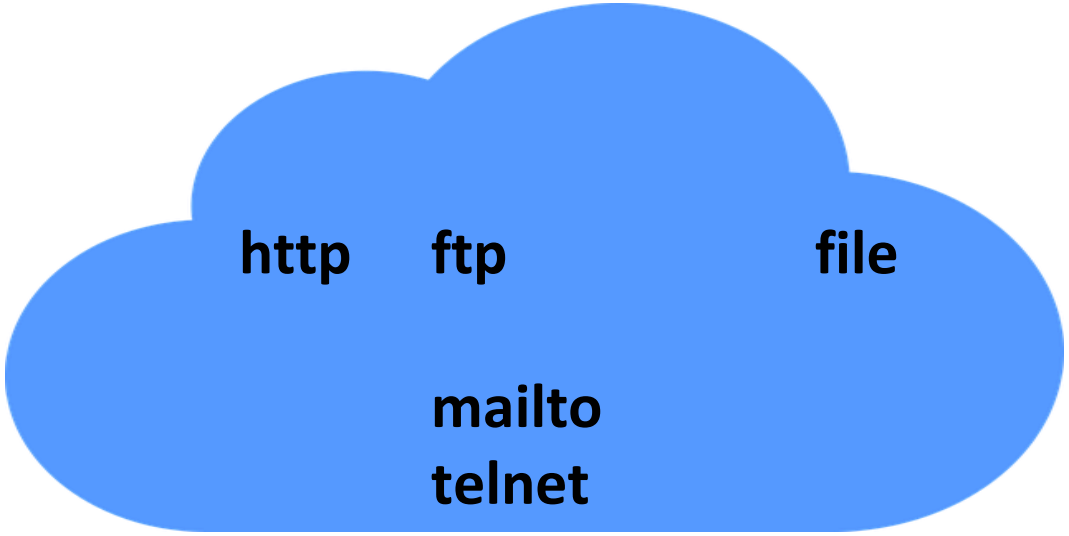
Esquema: Parte jerárquica ? Solicitud # Fragmento

EXMAEN!!

EL NAVEGADOR WEB - ESTRUCTURA DE UNA URI

Esquema:

- Identifica el protocolo a utilizar al solicitar el recurso:
  - http
  - ftp
  - file
  - mailto
  - telnet
- No distingue mayúsculas y minúsculas



esquema:// Parte jerárquica ? Solicitud # Fragmento

ESTRUCTURA DE UNA URI

protocolo usuario contraseña en dominio puerto ruta en servidor

Parte jerarquica : //<user>:<password>@<host>:<port>/<url-path>

- User: algunos esquemas permiten especificar un nombre de usuario.
- Password: Si aparece, debe estar separado del user por **:** y debe ir seguido de @
- Host: nombre de dominio (completo) o su dirección IP.
- Port : número del puerto al cual conectarse. La mayoría de los protocolos tienen ya un número de puerto asignado por defecto.
- url-path: Ruta en el servidor para acceder al recurso
  - http://usuario:clave@miembros.sitio.com:80
- Caracteres reservados:
  - / : ? # [ ] @ ! $ & ‘ ( ) * + , ; =
- No distingue mayúsculas y minúsculas

http:// Parte jerárquica ? Solicitud # Fragmento

EL NAVEGADOR WEB - ESTRUCTURA DE UNA URI

Solicitud o consulta:

- Variables que se pasan al solicitar el recurso
- Separada de la ruta mediante el carácter reservado ? Y del fragmento si lo hubiera mediante el carácter reservado #
  - /miruta.html?variable=valor&variable2=valor2
- Dentro de la cadena de consulta, el símbolo "+" se reserva como una abreviación del espacio (" ")

esquema:// Parte jerárquica ? Solicitud # Fragmento

EL NAVEGADOR WEB - ESTRUCTURA DE UNA URI

Fragmento:

- Identifica una subdirección dentro del recurso al que apunta la dirección
- Separado del resto de la estructura mediante el carácter reservado #
  - /miruta.html#subdireccion
- Diferencia entre URI y URL. La URL NO identifica fragmentos

esquema:// Parte jerárquica ? Solicitud # Fragmento

EL NAVEGADOR WEB – PROTOCOLO HTTP

- La comunicación entre el servidor web y el navegador se realiza mediante el **protocolo HTTP**.
  - La mayoría de los navegadores soportan otros protocolos como FTP y HTTPS

- El servidor ejecuta un programa que devuelve el **código** del recurso

- GET: solicitud genérica. Al usar un enlace o teclear una URL
- HEAD: solicita información de un fichero (tipo, tamaño, fecha modificación)
- POST: Envía información al servidor (formulario)

EL NAVEGADOR WEB - FLUJO DE LA SOLICITUD

1. El usuario selecciona una URL mediante texto o enlace.
2. El cliente Web (navegador) decodifica la URL separando protocolo de acceso y parte jerárquica.
3. Se abre una conexión TCP/IP con el servidor llamando al puerto correspondiente.
4. Se envía la petición que contiene:
   - Comando + Contenido URL + Versión del protocolo +
     + Información diversa (ej. capacidades del navegador)
5. El servidor envía la respuesta que contiene:
   - Línea de estado + Tipo de dato + información
6. Se cierra la conexión TCP/IP.
7. El navegador interpreta la información y la muestra en base a las especificaciones correspondientes.

(Lo subrayado son como los pasos que sigue)

LENGUAJES DE PROGRAMACIÓN EN ENTORNO CLIENTE

Lenguajes de programación / tecnologías

- HTML: agregar contenido a una página web. Componentes básicos.
- CSS: especificar diseño, estilo y alineación de la página web.
- JavaScript / TypeScript / Coffeesprit: dar vida a las páginas web.

Frameworks

- Frameworks y librerías JavaScript: jQuery, Angular, Vuejs y React
- Frameworks y librerías CSS: Bootstrap y Material Design
- Gestores de paquetes para gestionar librerías y plugins: npm y yarn.
- Herramientas para construcción y desarrollo del código: NPM Scripts, Gulp, Webpack y Rollup.

LENGUAJES DE PROGRAMACIÓN EN ENTORNO CLIENTE

- Son aquellos que se ejecutan en el navegador web.
- Definen la estructura, comportamiento y estilo.

LENGUAJES DE PROGRAMACIÓN EN ENTORNO CLIENTE - HTML

Versiones

- **HTML**: Es el lenguaje que usan los desarrolladores de los navegadores y está en continua evolución. Depende del grupo WHATWG.
- **HTML5**: Es el estándar aceptado globalmente. Su responsable es el W3C. Adopta los cambios de más éxito qué están siendo implementados en los navegadores del grupo WHATWG en HTML.
- Los navegadores principales se adaptan automáticamente a los cambios que integra HTML.
  - https://caniuse.com documenta qué funcionalidades soporta cada navegador.
  - https://validator.w3.org/ chequea si una web está diseñada acorde a los estándares de w3c.

LENGUAJES DE PROGRAMACIÓN EN ENTORNO CLIENTE - HTML5

HTML anteriores

-
- Especificar lenguaje de scripting
- Etiqueta de bloque (div) / Etiqueta en línea (span)
  - Todavía se usan
- Sin tags para vídeo o audio

HTML5

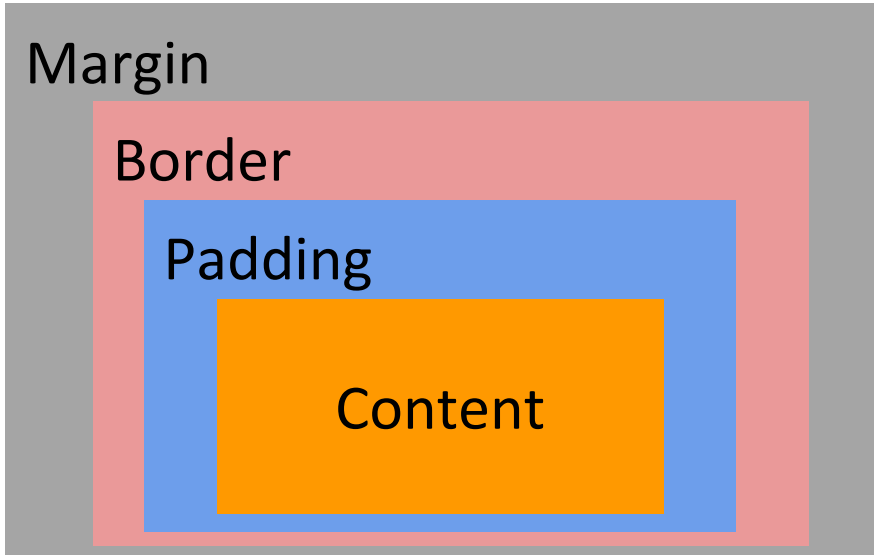
-
- JS es el lenguaje de scripting por defecto
- Flow Content / Phrasing Content (y algunas categorías más):
  - https://www.w3.org/TR/2011/WD-html5-20110525/content-models.html
- Nuevos tags para elementos multimedia, audio y vídeo.

LENGUAJES DE PROGRAMACIÓN EN ENTORNO CLIENTE - CSS

A tener en cuenta

- Selectores: se utilizarán también en JavaScript.
  - Etiqueta HTML: el selector se nombra igual (h1, p, a…)
  - Atributo id HTML → selector #id
  - Atributo class: → selector .nombredelaclase.
  - Atributo determinado → selector [atributo]

Modelo de cajas



LENGUAJES DE PROGRAMACIÓN EN ENTORNO CLIENTE - LENGUAJES DE SCRIPT

Script

- o guión: programa usualmente simple, almacenado en un archivo de texto plano.
- interpretados: se ejecutan instrucción a instrucción
- Tipos: de SO o de diseño web

Tipos de scripts en diseño web

- Del lado de cliente
  - Generalmente escritos en JavaScript
  - Definidos con la etiqueta <script> y el atributo type=MIME
  - Objetivo: dotar de mayor funcionalidad a la web
- Del lado de servidor

JavaScript

- Lenguaje de programación de scripting, débilmente tipado y orientado a objetos.

Ajax

- Mantiene una comunicación asíncrona con el servidor en segundo plano.

LENGUAJES DE PROGRAMACIÓN EN ENTORNO CLIENTE - JAVASCRIPT

Ventajas e Inconvenientes

- No necesita compilarse, los navegadores lo interpretan como HTML
- Funciona en casi todos los navegadores
- Fácil de aprender
- Permite hacer las páginas webs más interactivas e interesantes
- Rápido y ligero

Ejecución de código malicioso

No en todos

Vulnerable a amenazas

Puede generar inconsistencias en dispositivos diferentes

Fragmentos de código grandes

LENGUAJES DE PROGRAMACIÓN EN ENTORNO CLIENTE - CONSIDERACIONES

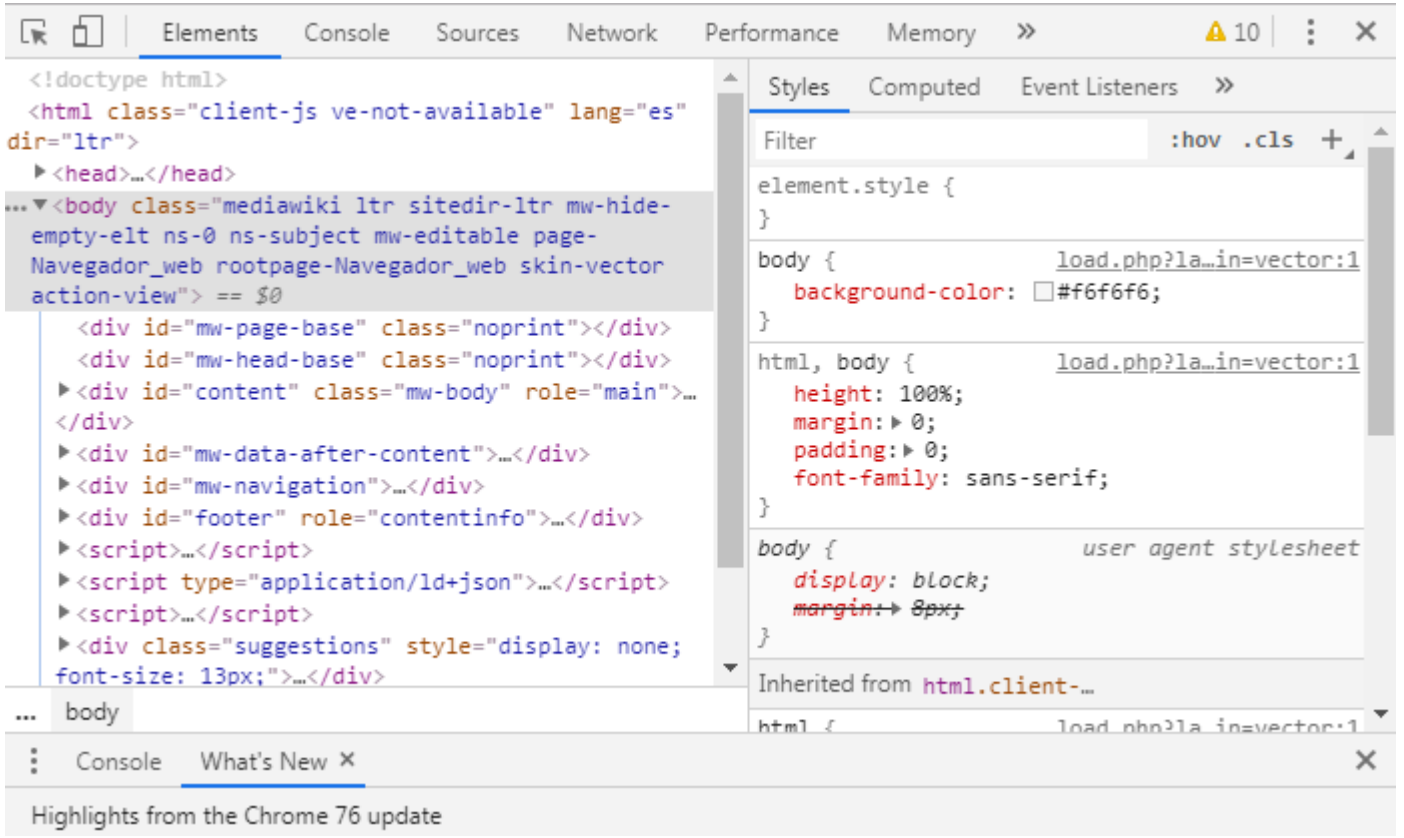
Compatibilidad de navegadores

- No todos los navegadores implementan todas las funciones de un lenguaje.
- Elegir para qué navegadores se quiere desarrollar:
  - Número de usuarios: https://www.w3schools.com/browsers/default.asp
  - Tipo de dispositivo
  - Otras consideraciones
- Existen tablas para saber en qué navegadores funciona correctamente nuestro código:
  - https://www.w3schools.com/cssref/css3_browsersupport.asp
  - https://html5test.com/results/desktop.html

HERRAMIENTAS ÚTILES PARA EL DESARROLLO

- ✓ En Mozilla Firefox, se puede acceder a la **consola web** con la combinación de teclas "Ctrl+shift+k"

- ✓ En Google Chrome, se puede acceder a las herramientas de desarrollo con la combinación de teclas "Ctrl+shift+i"



ENTORNOS DE DESARROLLO

Navegador:

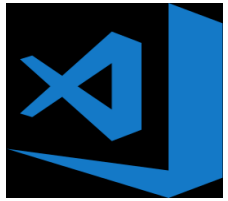
- Herramientas de desarrollo

Pruebas:

- Codepen
- Browser-sync (NodeJs)

Editores de código:

- VS Code
- Sublime Text
- Brackets
- Atom



Compilador:

- Closure Compiler