



Reconocimiento - NoComercial - Compartir Igual (by-nc-sa): No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

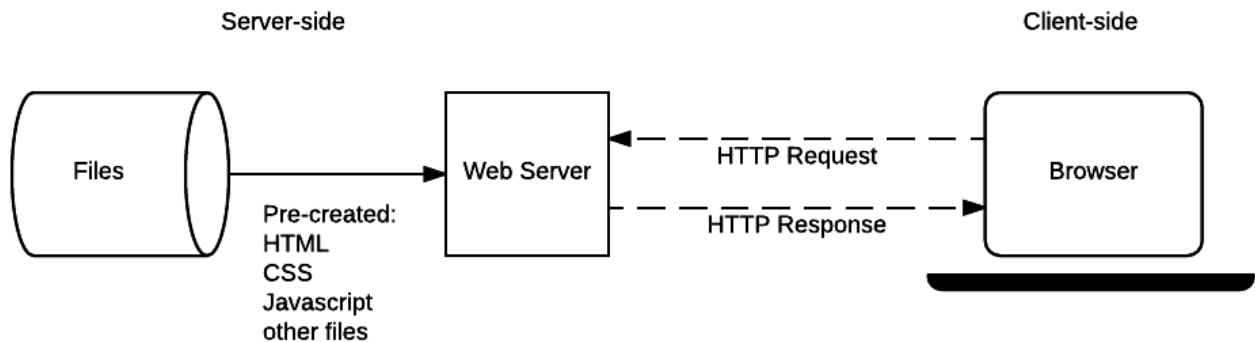
ÍNDICE DE CONTENIDO

1. TECNOLOGÍAS Y LENGUAJES ASOCIADOS.DESARROLLO FRONT-END.	2
1.1 Front End	2
1.2 Back End.....	3
1.3 Distribución de tareas.	3
2. Lenguajes de script	4
2.1 Java y JavaScript.....	4
3. Navegadores.	5
3.1 Introducción.	5
3.2 Definición de navegador.	6
3.3 Funcionamiento de los navegadores.	6
4. Principales navegadores.	7
4.1 ¿Qué navegador se recomienda para realizar el curso?	8
5. Herramientas útiles para el desarrollo: Consola Web.	8
5.1 Consola web	8
6. Entorno de desarrollo.	9
6.1 Visual Studio Code	9
6.2 Control de versiones en Visual Studio Code	10
7. Bibliografía.....	10

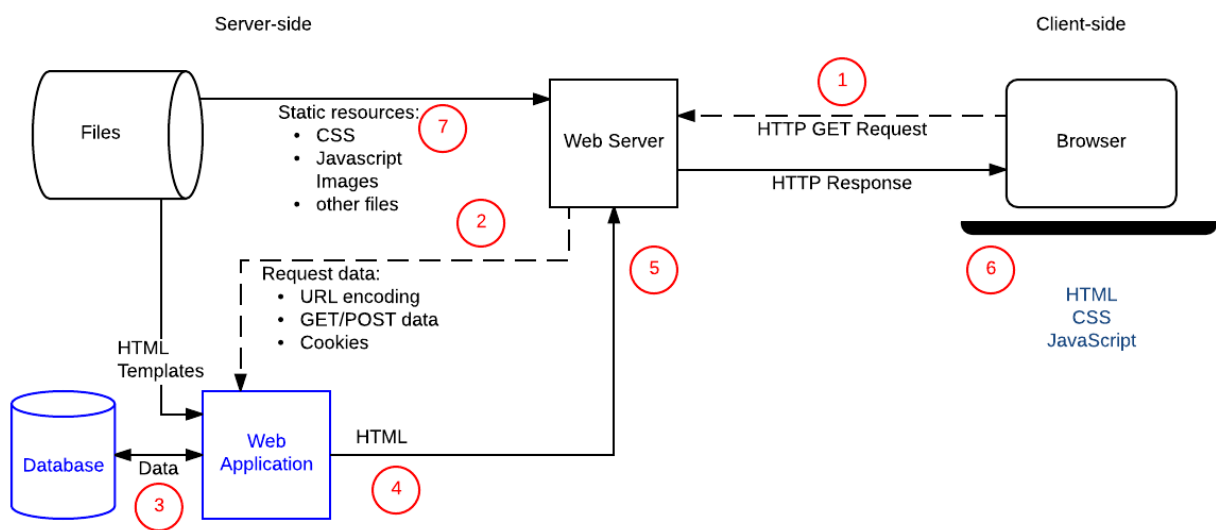
1. TECNOLOGÍAS Y LENGUAJES ASOCIADOS.DESARROLLO FRONT-END.

El Back end y el Front end son dos partes fundamentales de la programación de una aplicación web.

Web estática.



Web dinámica.



1.1 Front End

Front End es la parte de una aplicación que interactúa con los usuarios, es conocida como el lado del cliente. Básicamente es todo lo que vemos en la pantalla cuando accedemos a un sitio web o aplicación: tipos de letra, colores, adaptación para distintas pantallas (RWD (Responsive Web Design)), los efectos del ratón, teclado, movimientos, desplazamientos, efectos visuales... y otros elementos que permiten navegar dentro de una página web. Este conjunto crea la experiencia del usuario.

Como hemos dicho, el desarrollador front end se encarga de la experiencia del usuario, es decir, en

el momento en el que este entra a una página web, debe ser capaz de navegar por ella, por lo que el usuario verá una interface sencilla de usar, atractiva y funcional.

Un desarrollador front end debe conocer los siguientes lenguajes de programación: HTML5, CSS3, JavaScript.

1.2 Back End

Como hemos dicho antes, Front end es todo con lo que el usuario se encuentra directamente en la web o aplicación, entonces cuando hablamos de “Back end” nos referimos al interior de las aplicaciones que viven en el servidor y al que a menudo se le denomina “el lado del servidor”.

El back end del sitio web consiste en un servidor, una aplicación y una base de datos. Se toman los datos, se procesa la información y se envía al usuario. Los desarrolladores de Front end y Back end suelen trabajar juntos para que todo funcione correctamente.

Un desarrollador Back end debe tener amplios conocimientos de los siguientes lenguajes: frameworks y los tipos de base de datos. No siendo necesario conocer todos los lenguajes, pero sí entender y saber trabajar con algunos de ellos.

Alguno de los lenguajes y/o tecnologías del backend serian;

ASP.NET, PHP, Python, Ruby, Node.js, Java, MySQL, SQL Server, PostgreSQL, Oracle, MongoDB.

1.3 Distribución de tareas.

Existen tareas que se pueden ejecutar tanto en el lado servidor, como en el lado cliente. Algunas de estas son:

1. Validación de formularios.
 - Cliente: Validación rápida con JavaScript antes de enviar el formulario (por ejemplo, que un campo no esté vacío).
 - Servidor: Validación obligatoria para garantizar la seguridad y consistencia de los datos (evita manipulación maliciosa).
 2. Renderizado de contenido (HTML/CSS)
 - Cliente: Renderizado dinámico con frameworks como React, Vue o Angular (Single Page Applications).
 - Servidor: Renderizado previo del HTML con tecnologías como PHP, Node.js (SSR - Server-Side Rendering), Next.js, etc.
 3. Autenticación de usuarios
 - Cliente: Manejo de tokens (como JWT), almacenamiento en localStorage o cookies, verificación de sesión activa.
 - Servidor: Validación real del token, gestión de sesiones, verificación de credenciales, seguridad.
 4. Visualización de datos
-

-
- Cliente: Uso de bibliotecas como Chart.js o D3.js para generar gráficas a partir de datos.
 - Servidor: Generación de gráficos o datos preprocesados listos para visualizar (por ejemplo, imágenes de gráficas generadas en Python o PHP).
5. Internacionalización (i18n)
- Cliente: Cambio de idioma dinámico sin recargar la página.
 - Servidor: Generación del contenido en el idioma correcto al servir la página, según la configuración del usuario.
6. Búsqueda y filtrado
- Cliente: Filtrado rápido de datos ya cargados (por ejemplo, búsqueda en una tabla).
 - Servidor: Consultas a base de datos con filtros avanzados, paginación, y rendimiento optimizado.
7. Carga y manipulación de archivos
- Cliente: Previsualización de archivos, compresión, renombrado antes de enviar.
 - Servidor: Validación, almacenamiento, análisis o procesamiento del archivo.
8. Comunicación en tiempo real
- Cliente: Uso de WebSockets, SSE o fetch/AJAX para recibir o enviar datos en tiempo real.
 - Servidor: Manejo de conexiones WebSocket, envío de datos en tiempo real, notificaciones push, etc.

2. LENGUAJES DE SCRIPT

Un lenguaje de script es un pequeño lenguaje de programación cuyo código se inserta dentro del documento HTML. Este código se ejecuta en el navegador del usuario al cargar la página, o cuando sucede algo especial como puede ser el pulsar sobre un enlace.

Estos lenguajes permiten variar dinámicamente el contenido del documento, modificar el comportamiento normal del navegador, validar formularios, realizar pequeños trucos visuales, etc... Sin embargo, conviene recordar que se ejecutan en el navegador del usuario y no en la máquina donde estén alojadas, por lo que no podrán realizar cosas como manejar bases de datos. Esto hace que los contadores (por ejemplo) se deban realizar de otra manera, utilizando programas CGI.

El primer lenguaje de script que vió la luz fue el JavaScript de Netscape. Nacido con la versión 2.0 de este navegador y basado en la sintaxis de Java, su utilidad y el casi absoluto monopolio que entonces ejercía Netscape en el mercado de navegadores permitieron que se popularizara y extendiera su uso.

2.1 Java y JavaScript.

Aparte de esa raíz común en el nombre, Java y JavaScript son lenguajes totalmente diferentes. He aquí algunos ejemplos importantes de sus muchas diferencias:

- JavaScript tradicionalmente ha sido un lenguaje interpretado, y Java es compilado. Para
-

entendernos, y dicho muy básicamente, los programas JavaScript son archivos de texto que pueden leer tanto los ordenadores como las personas y que se interpretan a medida que se leen para luego ejecutarlos, mientras que los de Java se compilan a un archivo especial optimizado para que lo lea un ordenador y lo ejecute. Podríamos entrar en detalles de que Java se compila a un lenguaje intermedio llamado bytecode que es interpretado por un ejecutor JIT que facilita que sea multiplataforma, pero no es el objeto de este artículo y quiero mantenerlo intencionadamente simple. Por otro lado, y abundando en lo mismo, en la actualidad mucho código JavaScript es compilado (por ejemplo, el motor V8 de Chrome y Node.js compila a código nativo el código JavaScript y puede llegar a ser tan rápido como cualquier otro). Así que en la actualidad la distinción no es tan clara entrando en detalle, pero sí se puede afirmar que son de naturaleza diferente porque JavaScript no requiere una compilación explícita para ser ejecutado.

- Java se depura en dos fases y JavaScript en una. Como con todos los lenguajes interpretados, en JavaScript solamente sabes si has cometido inadvertidamente un error de sintaxis cuando vas a ejecutar el programa (salvo que uses herramientas especializadas como Visual Studio o WebStorm, que interpretan el código en segundo plano). En Java, sin embargo, primero se realiza la fase de compilación, en la que el compilador ya indica los posibles errores de sintaxis que existan. Luego al ejecutar podrían surgir errores de lógica o de otra índole. En JavaScript todos ellos se depuran al mismo tiempo, haciéndolo más complicado.
- Java es un lenguaje orientado a objetos puro, pero JavaScript está basado en prototipos. En realidad, JavaScript debe simular muchas de las características de orientación a objetos que se dan en la mayoría de los lenguajes. A cambio proporciona diversos paradigmas de programación (funcional, imperativo, dinámico y orientado a objetos) lo que lo convierte en un lenguaje tremendamente versátil. De hecho, en JavaScript es posible cambiar la "base" de la que hereda una clase en cualquier momento, afectando a todas, cosa imposible en la mayoría de los lenguajes orientados a objetos.
- Java es fuertemente tipado, y JavaScript es débilmente tipado. En Java todas las variables tienen un tipo determinado y una vez definidas no se pueden cambiar. En JavaScript una misma variable puede contener primero un texto, luego un número, luego una fecha o un objeto, etc... Esto es un arma de doble filo pues permite una gran flexibilidad a cambio de muchos posibles errores si no tenemos cuidado.

3. NAVEGADORES.

3.1 Introducción.

Para realizar cualquier desarrollo web, es imprescindible comprobar que el resultado que queremos es el adecuado con la mayor cantidad de navegadores posibles, especialmente aquellos más usados.

Además de procesar etiquetas HTML, los navegadores suelen interpretar lenguajes de script, siendo Javascript uno de los más populares.



3.2 Definición de navegador.

Un navegador o navegador web, o browser (en inglés), 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.

La funcionalidad básica de un navegador web es permitir la visualización de documentos de texto, posiblemente con recursos multimedia incrustados. Además, permite visitar páginas web y hacer actividades en ella, es decir, podemos enlazar un sitio con otro, imprimir, enviar y recibir correo, entre otras funcionalidades más.

Los documentos que se muestran en un navegador pueden estar ubicados en la computadora en donde está el usuario, pero también pueden estar en cualquier otro dispositivo que esté conectado en la computadora del usuario o a través de Internet, y que tenga los recursos necesarios para la transmisión de los documentos (un software servidor web).


Tales documentos, comúnmente denominados páginas web, poseen hipervínculos que enlazan una porción de texto o una imagen a otro documento, normalmente relacionado con el texto o la imagen.

El seguimiento de enlaces de una página a otra, ubicada en cualquier computadora conectada a Internet, se llama navegación, de donde se origina el nombre navegador.

Para acceder a estos recursos, se utiliza un identificador único llamado URL (Uniform Resource Locator).

El formato general de una URL es “protocolo://máquina/directorio/archivo”.

- Si no se especifica el directorio, toma como directorio la raíz.
- Si no se especifica el fichero, toma alguno de los nombres por defecto (“index.html”, “index.php”, etc....)

 Un ejemplo muy típico es <https://www.google.es> donde se accede al recurso www.google.es usando el protocolo https.

3.3 Funcionamiento de los navegadores.

La comunicación entre el servidor web y el navegador se realiza mediante el protocolo HTTP, aunque la mayoría de los navegadores soportan otros protocolos como FTP y HTTPS (una versión cifrada de HTTP basada en Secure Socket Layer o Capa de Conexión Segura (SSL)).

La función principal del navegador es obtener documentos HTML e interpretarlos para mostrarlos

en pantalla. En la actualidad, no solamente descargan este tipo de documentos, sino que muestran con el documento sus imágenes, sonidos e incluso vídeos streaming en diferentes formatos y protocolos. Además, permiten almacenar la información en el disco o crear marcadores (bookmarks) de las páginas más visitadas.

Los primeros navegadores web sólo soportaban una versión muy simple de HTML. El rápido desarrollo de los navegadores web propietarios condujo al desarrollo de dialectos no estándares de HTML y a problemas de interoperabilidad en la web. Los más modernos (como Google Chrome, Mozilla, Netscape, Opera e Internet Explorer / Microsoft Edge) soportan los estándares HTML y XHTML (comenzando con HTML 4.01, los cuales deberían visualizarse de la misma manera en todos ellos).

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. Su objetivo es crear una web que trabaje mejor para todos, con sitios accesibles a más personas y que funcionen en cualquier dispositivo de acceso a Internet.

Se puede comprobar de manera online si un documento Web cumple el estándar definido por W3C mediante <https://validator.w3.org/>

Actualmente la mayoría de navegadores aceptan páginas no estándar, pero cuanto más estándar se la aplicación web desarrollada, mayor probabilidad que funcione correctamente en todos los navegadores.

Es una práctica imprescindible el comprobar que cualquier desarrollo Web funcione correctamente en los principales navegadores.

4. PRINCIPALES NAVEGADORES.



- **Microsoft Edge (Antiguo Internet Explorer)**
 - URL Oficial: <https://www.microsoft.com/es-es/windows/microsoft-edge>
 - Antiguamente se llamaba Internet Explorer. Microsoft Edge 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**
 - URL Oficial: <https://www.mozilla.org/es-ES/firefox/new/>
 - Mozilla Firefox es un navegador web libre y de código abierto desarrollado por la Corporación Mozilla y la Fundación Mozilla. Usa el motor Gecko para renderizar páginas
-

webs, el cual implementa actuales y futuros estándares web.

- Posee una versión para desarrolladores: “Firefox Developer Edition”
https://www.mozilla.org/en-US/firefox/developer/?utm_source=firebug&utm_medium=lp&utm_campaign=switch&utm_content=landingpage
- **Google Chrome**
 - URL Oficial: <https://www.google.com/chrome/>
 - Google Chrome es un navegador web desarrollado por Google y compilado con base en varios componentes e infraestructuras de desarrollo de aplicaciones (frameworks) de código abierto, como el motor de renderizado Blink (bifurcación o fork de WebKit). Está disponible gratuitamente bajo condiciones específicas del software privativo o cerrado.
- **Safari**
 - URL oficial: <http://www.apple.com/es/safari/>
 - Safari es un navegador web de código cerrado desarrollado por Apple Inc. Está disponible para OS X, iOS (el sistema usado por el iPhone, el iPod touch y iPad) y Windows (sin soporte desde el 2012).
- **Opera**
 - URL oficial: <http://www.opera.com/es>
 - Opera es un navegador web creado por la empresa noruega Opera Software. Usa el motor de renderizado Blink. Tiene versiones para escritorio, teléfonos móviles y tabletas.

4.1 ¿Qué navegador se recomienda para realizar el curso?

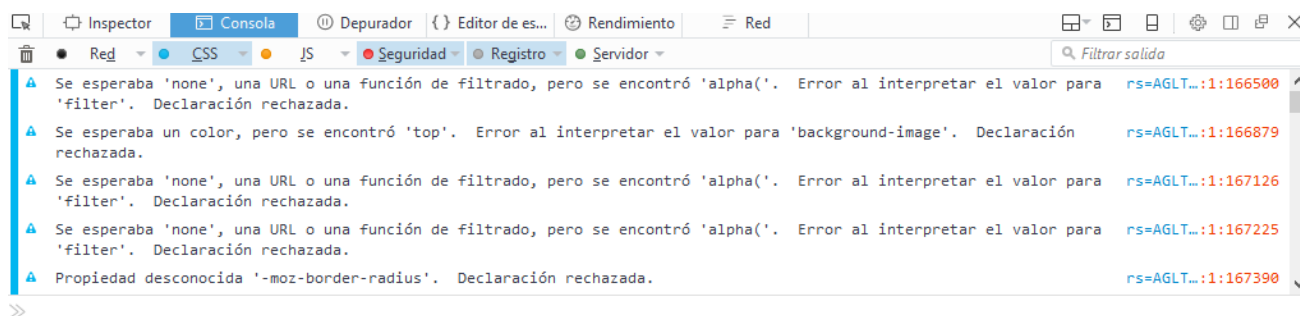
Para realizar las actividades del módulo, recomiendo utilizar **Mozilla Firefox** (aunque es posible usar el que se desee).

El motivo de usar este es la gran cantidad de herramientas para depuración que posee incluso en su versión estándar. Para la mayoría de acciones con este será suficiente, pero está disponible una versión que amplía las herramientas de desarrollo llamada “Firefox Developer Edition”.

5. HERRAMIENTAS ÚTILES PARA EL DESARROLLO: CONSOLA WEB.

Los navegadores incorporan de manera nativa herramientas para facilitar el desarrollo, entre la que destacamos la “Consola Web”. Asimismo, también mediante ampliaciones (extensiones, plugins, etc.) se amplían características para facilitar el desarrollo y la depuración de código.

5.1 Consola web



✈ 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 (que incluyen una consola similar a la de Firefox) con la combinación de teclas “Ctrl+shift+i”

✈ En las ultimas versiones se a estandarizado con la tecla F12

Esta consola incluye varias pestañas:

- Red: registro de Peticiones HTTP.
- CSS: registra análisis y errores CSS.
- JS: registra análisis y errores Javascript
- Seguridad: registra advertencias o fallos de seguridad.
- Registro: registra mensajes enviados al objeto “window.console”
- Servidor: registrar mensajes recibidos del servidor Web.

El resultado de las peticiones HTTP se muestra de color negro, CSS de color azul, JavaScript amarillo y los errores o advertencias de seguridad de color rojo, registro objeto “window.console” en gris y Servidor en verde.

📖 Más información del uso de la “Web console” en https://developer.mozilla.org/en-US/docs/Tools/Web_Console

6. ENTORNO DE DESARROLLO.

Existen diversos entornos de desarrollo, desde los más sencillos (Brackets, Notepad++, Sublime, Brackets, Visual Studio Code, etc...) a interfaces más complejas (Aptana, Eclipse, etc...).

En principio usaremos Visual Studio Code.

6.1 Visual Studio Code

Recomendamos **Visual Studio Code**. Es software libre y muy potente <https://code.visualstudio.com/>

Aquí algunos manuales libres de uso de Visual Studio Code en castellano:

<http://www.mclibre.org/consultar/informatica/lecciones/vsc-instalacion.html>

<http://www.mclibre.org/consultar/informatica/lecciones/vsc-personalizacion.html>

6.2 Control de versiones en Visual Studio Code

Durante el curso, se utilizarán repositorios Git tanto para la entrega de prácticas como para facilitaros el disponer de un repositorio con control de versiones.

Utilizaremos una cuenta Git en BitBucket <https://bitbucket.org/>.

Podéis instalarlo en:

- Ubuntu:
 - `sudo apt-get update`
 - `sudo apt-get install git`
- Windows: <https://git-for-windows.github.io/>

Para facilitar la tarea del uso de Git es recomendable instalar alguna extensión o entorno que os facilite su uso.

- Para usar Git en Visual Studio Code
 - <https://code.visualstudio.com/docs/editor/versioncontrol>
 - <http://www.mclibre.org/consultar/informatica/lecciones/vsc-git-repositorio.html>
- Aquí un ejemplo del uso de Git en Visual Studio Code.
 - <https://code.visualstudio.com/docs/introvideos/versioncontrol>

7. BIBLIOGRAFÍA

[1] Navegador Web

https://es.wikipedia.org/wiki/Navegador_web

[2] Git

<https://es.wikipedia.org/wiki/Git>

[3] Servidor Web

https://es.wikipedia.org/wiki/Servidor_web

Apuntes del Centre Específic d'Educació a Distància de la Comunitat Valenciana.
