Desarrollo web en entorno cliente

UD – 1

Selección de arquitecturas y herramientas

Contenidos







ÍNDICE

1. DESARROLLO WEB

La web fue inicialmente concebida y creada por **Tim Berners-Lee**, un especialista del laboratorio europeo de partículas (CERN) en 1989. En sus mismas palabras, había una "necesidad de una herramienta colaborativa que soportara el conocimiento científico" en un contexto internacional. Él y su compañero **Robert Cailliau** crearon un prototipo web, lo mostraron a la comunidad para sus pruebas y comentarios.

Dicho prototipo estaba basado en el concepto de hipertexto (*Texto que cuando pulsamos en él nos conduce a otro texto, objeto, sonido, video, sección o documento relacionado*). Como resultado se crearon unos protocolos (*cuando pulsamos en él nos conduce a otro texto, objeto, sonido, vídeo, sección o documento relacionado*) y especificaciones que han sido adoptados universalmente e incorporados a Internet, gracias a aportaciones posteriores como el desarrollo por la NCSA de la popular interfaz MOSAIC.

Todos los prototipos y desarrollos posteriores crecieron bajo la guía del **consorcio W3C**, que es una organización con base en el MIT de Massachusetts, que se responsabiliza de desarrollar y mantener los estándares web

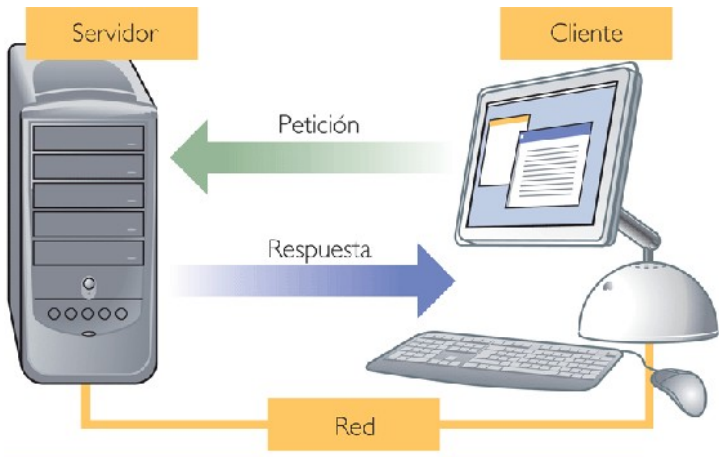
2. ESTRUCTURA CLIENTE SERVIDOR

Hoy en día los sitios web siguen un modelo basado en la programación cliente-servidor con tres elementos comunes:

- **El lado del servidor** ***[server-side]*** Incluye el hardware y software del servidor Web así como diferentes elementos de programación y tecnologías incrustadas. Las tecnologías pueden abarcar un rango amplio desde programas CGI escritos en PERL hasta aplicaciones multihilo (*También denominado multiproceso hace referencia a la posibilidad de ejecutar diferentes trozos de código de una misma aplicación de forma simultánea*) basadas en Java, incluyendo tecnologías de servidor de bases de datos que soporten múltiples sitios web.

- **El lado del cliente** ***[client-side]*** Este elemento hace referencia a los navegadores web y está soportado por tecnologías como HTML, CSS y lenguajes como JavaScript y controles ActiveX, los cuales se utilizan para crear la presentación de la página o proporcionar características interactivas. Es justamente aquí dónde nos vamos a centrar a lo largo de todo el módulo.

- **La red** ***[Network]*** Describe los diferentes elementos de conectividad (Capacidad que tiene un dispositivo para poder conectarse a otros. Aquí se detallan los diferentes protocolos y material utilizado para poder realizar dicha conexión) utilizados para mostrar el sitio web al usuario.



3. LENGUAJES DE PROGRAMACIÓN EN CLIENTES WEB

Cuando hablamos de tecnologías empleadas en lenguajes de programación web podemos citar dos grupos básicos: client-side y server-side. Las tecnologías ***client-side*** son aquellas que son ejecutadas en el cliente, generalmente en el contexto del navegador web. Cuando los programas o tecnologías son ejecutadas o interpretadas por el servidor estamos hablando de programación ***server-side***.

Cada tipo general de programación tiene su propio lugar y la mezcla es generalmente la mejor solución. Cuando hablamos de lenguajes de programación en clientes web, podemos distinguir dos variantes:

- ✔ ***Lenguajes que nos permiten dar formato y estilo a una página web***
  - ***HTML, CSS, etc.***
- ✔ ***Lenguajes que nos permite aportar dinamismo a páginas web***
  - ***Lenguajes de scripting.***

En este módulo nos vamos a centrar principalmente en estos últimos, los lenguajes de scripting, y en particular en el lenguaje JavaScript que será el lenguaje que utilizaremos a lo largo de todo este módulo formativo.

Tabla comparativa de lenguajes de programación web cliente – servidor.	
Lado del Cliente (client-side)	**Lado del servidor (server-side)**
✓ Aplicaciones de Ayuda. ✓ Programas del API del navegador. ➔ Plug-ins de Netscape. ➔ Controles ActiveX. ➔ Applets de Java. ✓ Lenguajes de scripting. ➔ JavaScript. ➔ VBScript.	✓ Scripts y programas CGI. ✓ Programas API del servidor. ➔ Módulos de Apache. ➔ Extensiones ISAPI y filtros. ➔ Servlets de Java. ✓ Lenguajes de scripting. ➔ PHP. ➔ Active Server Pages (ASP/ASP.NET). ➔ ColdFusion. ...

Hemos escogido JavaScript porque es el lenguaje de script (Lenguaje de guiones o lenguaje de órdenes que se almacena por lo general en archivos de texto plano y que será ejecutado por un programa intérprete) más utilizado en la programación en el lado del cliente, y está soportado mayoritariamente por todas las plataformas (Sistema operativo utilizado por un determinado dispositivo). Por lo tanto a partir de ahora todas las referencias que hagamos estarán enfocadas hacia JavaScript.

A continuación te mostramos un esquema de las 4 capas del desarrollo web en el lado del cliente, en la que se puede ver que JavaScript se sitúa en la capa superior gestionando el comportamiento de la página web.

Tabla de las 4 capas del desarrollo web en el lado del cliente.	
Comportamiento (JavaScript)	
Presentación (CSS)	
Estructura (DOM / estructura HTML)	Contenido Estructurado (documento HTML)
Contenido (texto, imágenes, vídeos, etc.)	

4. NAVEGADORES

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.

4.1 Definición de navegador

Definición: 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 el 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.

4.2 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 Edg*e) soportan los estándares HTML y XHTML.

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.

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

5. 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.

5.1 ¿Que navegador se recomienda para realizar el curso?

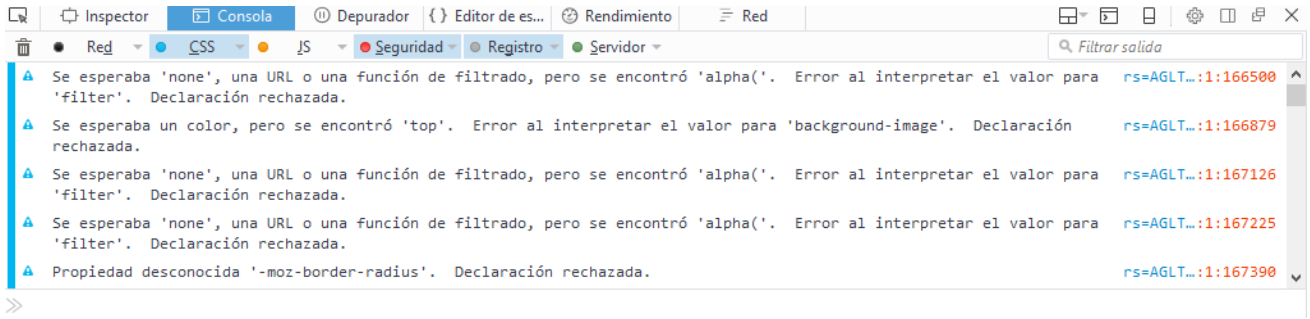
Para realizar las actividades del módulo, recomiendo utilizar **Mozilla Firefox o Google Chrome**

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 amplia las herramientas de desarrollo llamada "Developer Edition".

6. HERRAMIENTAS PARA EL DESARROLLO: CONSOLA WEB

Los navegadores incorporan de manera nativa herramientas para facilitar el desarrollo, entra 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.

6.1 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"

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.

Mas información del uso de la "Web console" en https://developer.mozilla.org/en-US/docs/Tools/Web_Console

7. ENTORNO DE DESARROLLO

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

En principio podéis usar aquel que queráis.

7.1 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

7.2 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 Github :

https://www-github.com

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*

8. CONTROL DE VERSIONES: GIT – GITHUB

8.1 Introducción

Git es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente. Su propósito es llevar registro de los cambios en archivos de computadora y coordinar el trabajo que varias personas realizan sobre archivos compartidos.

Si estás en una distribución basada en Debian como Ubuntu, puedes usar apt-get:

```
apt-get install git
```

Para instalar git en otros sistemas operativos puedes utilizar como referencia la información de la web:

https://git-scm.com/book/es/v2/Inicio---Sobre-el-Control-de-Versiones-Instalación-de-Git

8.2 Órdenes básicas

Crear un subdirectorio nuevo llamado .git, el cual contiene todos los archivos necesarios del repositorio

```
git init
```

Descargar información de una determinada rama(*unifica los comando fetch y merge*).

```
git pull
```

Comienza a trackear el archivo “nombre_archivo”.

```
git add <nombre_archivo>
git add .
```

Confirma los cambios realizados. El “*mensaje*” generalmente se usa para asociar al commit una breve descripción de los cambios realizados.

```
git commit -am "<mensaje>"
git commit -m "<mensaje>" .
```

Commitea los cambios desde el branch local origin al branch “nombre_rama”.

```
git push origin <nombre_rama>
```

Para consultar más comando, puedes visitar la web:

https://git-scm.com/docs

8.3 Github

GitHub es una plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de ordenador. El software que opera GitHub fue escrito en Ruby on Rails
El 4 de junio de 2018, Microsoft compró GitHub por la cantidad de 7.500 millones de dólares, este cambio de propietarios provocó la salida de varios proyecto desde este repositorio debido la posibilidad de acceso a códigos fuentes por parte de una compañía que su negocios es el software,

http://www.github.com

9. INTEGRACIÓN DE CÓDIGO JAVASCRIPT CON HTML .

Ahora que ya conoces las herramientas que puedes utilizar para comenzar a programar en JavaScript, vamos a ver la forma de integrar el código de JavaScript en tu código HTML.

9.1 Etiqueta <script>

Los navegadores web te permiten varias opciones de inserción de código de JavaScript. Podremos insertar código usando las etiquetas **<script> </script>** y empleando un atributo type indicaremos qué tipo de lenguaje de script estamos utilizando, Por ejemplo:

```
<script type="text/javascript">
    // El código javascript vendrá aquí
</script>
```

9.2 Fichero externo

Otra forma de integrar el código de JavaScript es incrustar un fichero externo que contenga el código de JavaScript. Ésta sería la forma más recomendable, ya que así se consigue una separación entre el código y la estructura de la página web y como ventajas adicionales podrás compartir código entre diferentes páginas, centralizar el código para la depuración de errores, tendrás mayor claridad en tus desarrollos, más modularidad, seguridad del código y conseguirás que las páginas carguen más rápido.

Para ello tendremos que añadir a la etiqueta script el atributo src , con el nombre del fichero que contiene el código de JavaScript. Generalmente los ficheros que contienen texto de JavaScript tendrán la extensión .js .

Por ejemplo:

```
<script type="text/javascript" src="tucodigo.js"> </script>
```

9.3 En código html

Otra alternativa consiste en incorporar javascript directamente en elementos html. Numerosos autores no recomiendan esta opción, con el principal motivo de que dificulta considerablemente el mantenimiento de aplicaciones.

```
<body>
  <p onclick="alert('un mensaje de prueba')">Párrafo de texto</p>
<body>
```

9.4 Etiqueta <noscript>

El lenguaje HTML define la etiqueta ***<noscript>*** para mostrar un mensaje al usuario cuando su navegador no puede ejecutar JavaScript. La etiqueta ***<noscript>*** se debe incluir en el interior de la etiqueta **<body>** (normalmente se incluye al principio de <body>). El mensaje que muestra puede incluir cualquier elemento o etiqueta XHTML.

```
<head>  … </head>

<body>

     <noscript>

     <p> Bienvenido a Mi Sitio</p>

              <p> La página que estás viendo requiere para su
              funcionamiento el uso de JavaScript. Si lo has
              deshabilitado intencionadamente,

              por favor vuelve a activarlo</p>


     <noscript>

<body>
```