

IFCD0110 CONFECCIÓN Y PUBLICACIÓN DE PÁGINAS WEB

Módulo formativo: MF0950_2: Construcción de páginas web

Unidad formativa: UF1302 Creación de páginas web con lenguajes de marcas

Unidad 1. Introducción a los lenguajes de marcas. Entorno de trabajo.

Navegadores.

FORMACIÓN PROFESIONAL PARA EL EMPLEO.

CIFP VIRGEN DE GRACIA



GOBIERNO
DE ESPAÑA

MINISTERIO
DE TRABAJO
Y ECONOMÍA SOCIAL



1. Introducción	2
2. Servidores Web	2
3. Configuración de un servidor.....	3
4. Nombres de dominio.....	3
5. El problema de compartir datos.....	4
6. Tipos de lenguajes de marcas.....	5
7. Funcionamiento de las aplicaciones web. HTML, CSS y JavaScript.	13
8. Front-end y back-end.....	13
9. Guerra de navegadores.....	14
10. Validar código HTML.....	16
11. Publicación de páginas web.....	16
12. Bibliografía.....	17

1. Introducción

- Los lenguajes de marcas son bastante antiguos, se popularizaron con la llegada de Internet.
- Origen: SGML (Standar Generalized Markup Language). Hoy día lo más usado es HTML (HyperText Markup Language).
- La versión más actual de HTML es HTML5.
- El fundamento de Internet y todas las tecnologías asociadas se basa en estándares abiertos e independientes de la tecnología. El organismo que los regula es W3C (World Wide Web Consortium).

2. Servidores Web

Un **servidor web** es un programa que “entiende” los protocolos HTTP y HTTPS y atiende peticiones de navegadores sirviendo páginas a medida que se van solicitando. Tener un servidor no implica necesariamente un nombre de dominio.

Cuando en un navegador escribimos «http://www.google.es» contactamos con un programa que se está ejecutando en un ordenador al que se le ha dado el nombre «www.google.es» y le pedimos establecer una conexión para intercambiar información usando el protocolo HTTP. Así, cuando un servidor recibe una petición examina sus directorios para ver si tiene ese archivo en el directorio que le han dicho, si es así, el fichero se transmite al que hizo la petición. Si no existe, el navegador devuelve un código 404 (recurso no encontrado) y nos mostrará la página web asociada a ese código 404. Si todo va bien nuestro navegador recibe un código 200.



Un servidor web puede ser privado o alquilado, existiendo grandes diferencias entre la forma de gestionar ambos.

Uno **privado**, tiene la ventaja de ofrecer un control absoluto. No siempre es fácil mantener un ordenador doméstico conectado 24x7. En especial, en España, las conexiones ADSL no suelen ofrecer las capacidades necesarias para un sitio web de tamaño mediano.

Los **alquilados** suelen conllevar un mayor precio cuando se necesita más espacio para alojar nuestros ficheros.

Ofrecen muchas garantías como por ejemplo anchos de banda muy aceptables a precios bastante competitivos y sobre todo que permiten al diseñador web liberarse de la gestión de los recursos de red.

3. Configuración de un servidor.

1. Se necesita un servidor web instalado en un equipo (Apache del paquete XAMPP)
2. Apuntar la IP del equipo en el que se instala Apache.
3. Entrar en el router y abrir el puerto 80 indicando que se debe reenviar el tráfico a la IP que se apuntó.
4. Se debe averiguar la IP pública del router (por ejemplo podemos visitar WhatsMyIp).
5. La IP que aparece es la que se puede dar a clientes o amigos para que naveguen por nuestro sitio web.
6. (Optativo) se puede alquilar un nombre de dominio y solicitar que ese nombre (loquesea.com) sea redirigido a nuestra IP pública.

4. Nombres de dominio.

El servicio de nombres de dominio es un sistema que convierte de direcciones tipo www.loquesea.com a direcciones IP. Esto permite que sea más fácil recordar direcciones de páginas. Sin embargo, DNS es un sistema muy complejo que funciona de forma distribuida entre distintos países.

Los nombres de dominio se resuelven de final a principio. La última parte se llama TLD o Top Level Domain o dominio de primer nivel.. Estos dominios son administrados por países ocupándose cada uno de ellos de los nombres o marcas que hay dentro de dichos países.

5. El problema de compartir datos.

Los problemas relacionados con el intercambio de información entre aplicaciones y máquinas informáticas es tan viejo como la propia informática.

El problema parte del hecho de haber realizado un determinado trabajo con un software en un determinado ordenador y después querer pasar dicho trabajo a otro software en ese u otro ordenador.

Los archivos binarios tienen la complicación de que para hacer ese proceso, el origen y el destino de los datos deben comprender cómo codificar y decodificar la información. Eso, en muchos casos, ha sido un gran problema que ha obligado a que todos los trabajadores y trabajadoras hayan tenido que adaptarse al software de la empresa. Por otro lado dificulta que las empresas migren a otros sistemas por el miedo a perder los datos.

En la informática actual eso es aún más problemático al tener una necesidad de disponibilidad global del trabajo a través de dispositivos muy distintos como tabletas, smartphones y otros dispositivos portátiles.

Por ello, poco a poco han aparecido formatos binarios de archivo que han sido estándares de facto (aunque la mayoría no han sido reconocidos por ningún organismo de estándares) como por ejemplo el formato documental **PDF**, el formato de imagen **JPEG** o el formato de audio **MP3**.

Pero el problema fundamental sigue estando en el software empresarial. El caso típico podría ser una empresa que utiliza en sus oficinas el software **Apache Open Office** y luego una persona de la empresa quiere abrir el documento en el que está trabajando en un dispositivo que tiene instalado **Microsoft Office**. Es probable que no pueda ver el trabajo en ambos dispositivos.

Este problema puede ocurrir incluso con el mismo software pero con versiones distintas (por ejemplo intentar abrir un documento creado con **Microsoft Word 2016** en una máquina con **Microsoft Word 2001**).

De ahí que muchas veces la opción para exportar e importar datos sea utilizar **convertidores**. Software con capacidad de convertir los datos de un formato a otro (por ejemplo de **Word** a **Open Office**; de **MP3** a **MOV** de Apple, etc.).

EL TEXTO COMO FORMATO MÁS VERSÁTIL

Como ya se ha explicado, hay un formato de archivo que cualquier dispositivo es capaz de entender: el texto. La cuestión es que los archivos llamados de texto, solo son capaces de almacenar **texto plano**; es decir, solo texto sin indicar ningún formato o añadir información no textual.

Debido a la facilidad de ser leído con cualquier aparato, se intenta que el propio texto sirva para almacenar otros datos, es decir información que no es texto sin más.

Evidentemente hay tipos de información prácticamente imposibles de representar en un archivo de texto, pero sí hay trucos para poder representar información de diversos tipos.

Para ello dentro del archivo habrá contenido que no se interpretará como texto sin más, sino que dentro del archivo habrá **texto especial**, marcado de una forma que permita darle otro significado. Es lo que se conoce como **metadatos**: datos que sirven para describir otros datos. En el caso de los archivos de texto, son palabras marcadas de forma especial que sirve para describir el texto al que acompaña.

Desde hace muchos años hay tres campos en los que esta idea ha funcionado muy bien: en las bases de datos, en los procesadores de texto y, especialmente, en las páginas web. El éxito de Internet ha permitido espolear esta tecnología a otros muchos campos.

Hay que recordar un problema fundamental con el texto: al ser formato tan universal, y ser su contenido tan accesible, es peligroso como fuente para almacenar datos confidenciales, ya que queda fácilmente expuesto a cualquier persona.

6. Tipos de lenguajes de marcas.

- **Orientados a la presentación.** En ellos los metadatos permiten indicar el formato en el que se debe presentar el texto. Es el caso de **RTF**, en el que sus etiquetas especifican tipos de letra, tamaños de página, colores, etc. Las primeras versiones de **HTML** también se consideran así, ya que incluían etiquetas como **font** mediante la cual se especificaba el formato de fuente.
- **Orientados a la descripción.** En ellos las marcas especiales permiten dar significado al texto pero no indican cómo se debe presentar en pantalla el mismo. Sería el caso de **XML** (o de **SGML**), **JSON**, **Markdown** y de las versiones actuales de **HTML**. En estos lenguajes simplemente se indica el significado del contenido: si el texto es un título, un párrafo normal, un pie de ilustración, una dirección postal etc.
- **Orientados a procedimientos.** Se trata de documentos en los que el texto marcado, se interpreta como órdenes a seguir, y así el archivo en realidad contiene instrucciones a realizar con el texto (girarle, convertirlo en una fórmula, realizar una suma, etc.). Es el caso de **LaTeX** o **PostScript**.

CHARLES GOLDFARB Y EL GML

Se considera a **Charles Goldfarb** como al padre de los lenguajes de marcas. La razón para esta consideración es, precisamente, su ayuda en la creación del lenguaje GML (*Generalized Markup Language*).

Goldfarb era un investigador de IBM que propuso ideas para que los documentos de texto que incluyeran la posibilidad de marcar el formato del mismo. Al final ayudó a realizar el lenguaje **GML** de **IBM** el cual puso los cimientos del futuro **SGML** (padre de **HTML** y **XML**) ideado por el propio Goldfarb y padre de la mayoría de lenguajes de marcas actuales.

Ejemplo de código GML

```
:h0.El reino de los animales
:h1.Mamíferos
:p.Los mamíferos (:hp1.Mammalia:ehp1.) son una clase de
vertebrados :hp2.amniotas homeotermos:ehp2. que poseen
glándulas mamarias productoras de leche con las que
alimentan a las crías
:h1.Aves
:p. Las aves son animales vertebrados, de sangre
caliente, que caminan, saltan o se mantienen solo sobre
las extremidades posteriores
```

El reino de los animales

Mamíferos

Los mamíferos (**Mammalia**) son una clase de vertebrados *amniotas homeotermos*. que poseen glándulas mamarias productoras de leche con las que alimentan a las crías

Aves

Las aves son animales vertebrados, de sangre caliente, que caminan, saltan o se mantienen solo sobre las extremidades posteriores

Ilustración 1-1. Renderizado de código GML (al estilo de HTML)

TEXT Y LATEX

En la década de los 70 **Donald Knuth** (uno de los ingenieros informáticos más importantes de la historia, padre del análisis de algoritmos y premio Turing 1974) creó el lenguaje **TeX** para producir documentos científicos utilizando una tipografía y capacidades que fueran iguales en cualquier computadora, asegurando además una gran calidad en los resultados.

Para ello apoyó a TeX con tipografía especial (fuentes **Modern Computer**) y un lenguaje de definición de tipos (**METAFONT**). TeX ha tenido cierto éxito en la comunidad científica gracias a sus 300 comandos que permiten crear documentos con tipos de gran calidad. Requiere de software capaz de convertir el archivo TeX a un formato de impresión.

El éxito de TeX produjo numerosos derivados de los cuales el más popular es **LaTeX**. LaTeX fue definido en 1984 por **Leslie Lamport** (premio Turing 2003), aunque después ha sido numerosas veces revisado. Al utilizar comandos de TeX y toda su estructura tipográfica, adquirió rápidamente notoriedad y sigue siendo utilizado para producir documentos con expresiones científicas, de gran calidad. La idea es que los científicos se centren en el contenido y no en la presentación.

```

\documentclass[12pt]{article}
\usepackage{amsmath}
\title{\Ejemplo}
\begin{document}
Este es el texto ejemplo de \LaTeX{}
Con datos en \emph{cursiva} o \textbf{negrita}.
Ejemplo de f\,ormula

\begin{align}
E &= mc^2
\end{align}
\end{document}

```

Este es el texto ejemplo de \LaTeX Con datos en *cursiva* o **negrita**. Ejemplo de formula

$$E = mc^2 \quad (1)$$

Ilustración 1-2. Renderizado de código Latex

RTF

RTF es el acrónimo de **Rich Text Format** (*Formato de Texto Enriquecido*) un lenguaje ideado por Richard Brodie, **Charles Simonyi** y **David Luebbert** (miembros del equipo de desarrollo de **Microsoft Word**) en 1987 para producir documentos de texto que incluyan anotaciones del formato. Es un formato propiedad de **Microsoft**, pero reconocido por la mayoría de aplicaciones de proceso de texto.

Actualmente se trata de un formato aceptado para documentos de texto que contengan información sobre el estilo del texto. Se usa mucho como formato de intercambio entre distintos procesadores por su potencia. El procesador de texto **Word Pad** incorporado dentro del sistema operativo **Windows** lo utiliza como formato nativo.

Codifica el texto usando 8 bits, para caracteres fuera del ASCII requiere de secuencias de escape lo que, prácticamente, le hace inviable como formato legible de texto en la mayoría de lenguas del planeta. En las últimas versiones de RTF ya sí se ofrece un mayor soporte a Unicode.

Su éxito procede de que las indicaciones de formato son potentes y son más legibles por las personas que el formato nativo de los procesadores de textos, aunque es, como lenguaje de marcado, uno de los más críticos.

```
{\rtf\ansicpg1252\deff0\deflang3082
{\fonttbl
{\f0\fcharset0\froman Times New Roman}
{\f1\fcharset0\fswiss Arial Black}
}
{\pard \f1\fs48
El reino de los animales
\par}
{\pard \f1\fs40
Mamíferos
\par}
```

```
{\pard \f0\fs25
Los mamíferos ({\b Mammalia}) son una clase de
vertebrados {\i amniotas homeotermos} que poseen
glándulas mamarias productoras de leche con las que
alimentan a las crías
\par}
{\pard \f1\fs40
Aves
\par}
{\pard \f0\fs25
Las aves son animales vertebrados, de sangre caliente,
que caminan, saltan o se mantienen solo sobre las
extremidades posteriores
\par}
}
```

El reino de los animales

Mamíferos

Los mamíferos (**Mammalia**) son una clase de vertebrados *amniotas homeotermos* que poseen glándulas mamarias productoras de leche con las que alimentan a las crías

Aves

Las aves son animales vertebrados, de sangre caliente, que caminan, saltan o se mantienen solo sobre las extremidades posteriores

Ilustración 1-3. Renderizado de código GML (al estilo de HTML)

SGML

Se trata de una mejora muy notable del lenguaje de **GML** que estandarizaba el lenguaje de marcado y que fue definida finalmente por ISO como estándar mundial en documentos de texto con etiquetas de marcado. Su responsable fue **Charles Goldfarb**.

Su importancia radica en que es el padre del lenguaje XML y la base sobre la que se sostiene el lenguaje HTML, dos de los lenguajes de marcas más populares de la historia.

En SGML los elementos que contienen indicaciones para el texto se colocan entre símbolos < y >. Las etiquetas se cierran con el signo /. Es decir las reglas fundamentales de los lenguajes de etiquetas actuales ya las había definido SGML.

En realidad (como XML) no es un lenguaje con unas etiquetas concretas, sino que se trata de un lenguaje que sirve para definir lenguajes. Entre los lenguajes definidos mediante SGML, sin duda HTML es el más popular.

```
<articulo>
  <titulo1>El reino de los animales</titulo1>
  <titulo2>Mamíferos</titulo2>
  <normal>Los mamíferos (<negrita>Mammalia</negrita>) son una clase de
  <titulo2>Aves</titulo2>
  <normal>Las aves son animales vertebrados, de sangre caliente, que
</articulo>
```

Como veremos más adelante, este documento es muy parecido a un documento realizado en XML, de hecho XML es un subconjunto de SGML más restrictivo (es un lenguaje que tiene normas más estrictas).

SGML necesitará definir cómo se debe mostrar los elementos *titulo1*, *titulo2*, etc. Ya que son nombres de elementos que habrá que definir. Esa es la prueba de que es un lenguaje para definir tipos de documento.

SGML aportó las etiquetas tal cual las conocemos actualmente gracias al éxito de HTML.

POSTSCRIPT

Se trata de un lenguaje de descripción de páginas. De hecho es el más popular para ese fin, siendo el lenguaje más utilizado por los sistemas de impresión de alta gama.

Permite crear documentos en los que se dan indicaciones potentísimas sobre como mostrar información en el dispositivo final. Se inició su desarrollo en 1976 por **John Warnock** y dos años más tarde se continuo con la empresa **Xerox**, hasta que en 1985 el propio Warnock funda **Adobe Systems** y desde esa empresa se continua su desarrollo.

Es en realidad todo un lenguaje de programación que indica la forma en que se debe mostrar la información que puede incluir texto y el tipo de letra del mismo, píxeles individuales y formas vectoriales (líneas, curvas). Sus posibilidades son muy amplias.

```
%colocar el cursor
100 100 moveto
%dibuja cuadrado
100 200 lineto
200 200 lineto
200 100 lineto
100 100 lineto
%relleno
stroke
```

HTML

Tim Bernes Lee utilizó SGML para definir un nuevo lenguaje de etiquetas que llamó **Hypertext Markup Language** (lenguaje de marcado de hipertexto) para crear documentos transportables a través de Internet en los que fuera posible el hipertexto; es decir la posibilidad que determinadas palabras marcadas de forma especial permitieran abrir un documento relacionado con ellas.

A pesar de tardar en ser aceptado, HTML fue un éxito rotundo y la causa indudable del éxito de Internet. Hoy en día casi todo en Internet se ve a través de documentos HTML, que popularmente se denominan **páginas web**.

Inicialmente estos documentos se veían con ayuda de intérpretes de texto (como por ejemplo el **Lynx** de **Unix**) que simplemente coloreaban el texto y remarcaban el hipertexto. Después el software se mejoró y aparecieron navegadores con capacidad más gráfica para mostrar formatos más avanzados y visuales.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
</head>
<body>
  <h1>El reino de los animales</h1>
  <h2>Mamíferos</h2>
  <p>Los mamíferos (<strong>Mammalia</strong>) son una clase de vertebrados.
  <h2>Aves</h2>
  <p>Las aves son animales vertebrados, de sangre caliente, que caminan o vuelan.
</body>
</html>
```

XML

Se trata de un subconjunto de SGML ideado para mejorar el propio SGML y con él definir lenguajes de marcado con sintaxis más estricta, pero más entendible.

Ha sido enormemente popular desde finales de los 90 y ha conseguido incorporar numerosos lenguajes a su alrededor para conseguir documentos muy dinámicos y con

gran capacidad de formato. Es uno de los formatos de documentos más populares para exportación e importación de datos.

Actualmente está siendo sobrepasado en la mayoría de sus usos por JSON

```
<?xml version="1.0" encoding="UTF-8"?>
<nombre>Jorge</nombre>
<apellido1>Sánchez</apellido1>
<dirección>
  <calle>C/ Falsa nº 0</calle>
  <localidad>Palencia</localidad>
  <código_Postal>34001</código_Postal>
  <pais>España</pais>
</dirección>
<teléfonos>
  <teléfono tipo="fijo">999 999 999</teléfono>
  <teléfono tipo="móvil">666 666 666</teléfono>
</telefonos>
```

MARKDOWN

Se trata de un formato de marcado simple que permite crear documentos sencillos y convertirlos en documentos HTML.

Fue creado por **John Gruber** con la ayuda de **Aaron Schwartz**. La pretensión de este lenguaje es definir unas normas muy sencillas para crear documentos semejantes a los que se crean mediante el lenguaje HTML.

Ha tenido un éxito muy notable, especialmente desde que fue adoptado por sitios tan populares como [GitHub](#), [Reddit](#) o [StackExchange](#) para que los usuarios publicaran contenido con formato.

Ejemplo de texto con formato Markdown:

```
# El reino de los animales
## Mamíferos
Los mamíferos (Mammalia) son una clase de vertebrados amniotas homeotermos.
## Aves
Las aves son animales vertebrados, de sangre caliente, que caminan, saltan
```

JSON

Abreviatura de *JavaScript Object Notation*, Se trata de una notación de datos procedente del lenguaje JavaScript estándar (concretamente en la versión **ECMAScript** de 1999). En el año 2002 se le daba soporte desde muchos de los navegadores y su fama ha sido tal que ahora se ha convertido en una notación independiente de JavaScript que compite claramente con XML en funcionalidad.

Las razones de su éxito se deben a su versatilidad, ya que permiten definir datos complejos, como arrays o código de funciones, elementos pertenecientes al mundo de la programación de aplicaciones. El éxito de JavaScript junto a la versatilidad comentada, le han convertido en el lenguaje de marcado más popular para almacenar datos.

En JSON, el texto se divide en datos y metadatos. De modo que el símbolo de los dos puntos separa el metadato del dato. Por otro lado, los símbolos de llave y corchete permiten agrupar de diversas formas los datos.

Ejemplo de código JSON:

```
{
  "nombre": "Jorge",
  "apellido1": "Sánchez",
  "dirección": {
    "calle": "C/ Falsa nº 0",
    "localidad": "Palencia",
    "código Postal": 34001,
    "país": "España"
  },
  "teléfonos": [
    {
      "tipo": "fijo",
      "número": "999 999 999"
    },
    {
      "tipo": "móvil",
      "number": "666 666 666"
    }
  ]
}
```


7. Funcionamiento de las aplicaciones web. HTML, CSS y JavaScript.

Una aplicación web, es una aplicación creada usando como base el lenguaje HTML. Por lo tanto, se trata de una aplicación que se ejecuta en un navegador de Internet.

Las aplicaciones web utilizan lenguajes que apoyan la labor de HTML y que son traducibles por parte de los navegadores. Esencialmente las aplicaciones web actuales utilizan:

- **HTML.** Para dar significado a los contenidos de la aplicación web. Permite indicar qué textos son títulos, cuáles son párrafos normales, cuáles son celdas de una tabla, cuáles son imágenes, etc.
- **CSS.** Lenguaje que permite dar formato y maquetación a los contenidos. Color, tamaño de letra, posición, etc.
- **JavaScript.** Permite diseñar la interactividad de la página. Permite que las acciones del usuario se puedan capturar y que la página reaccione a ellas. Ejemplos de sus posibilidades que cuando el usuario arrime el ratón a una imagen esta se ilumine, o que cuando el usuario haga clic en un título se muestre el contenido relacionado si estaba oculto o que el usuario pueda aumentar el tamaño de la letra de la página haciendo clic en un icono.

8. Front-end y back-end.

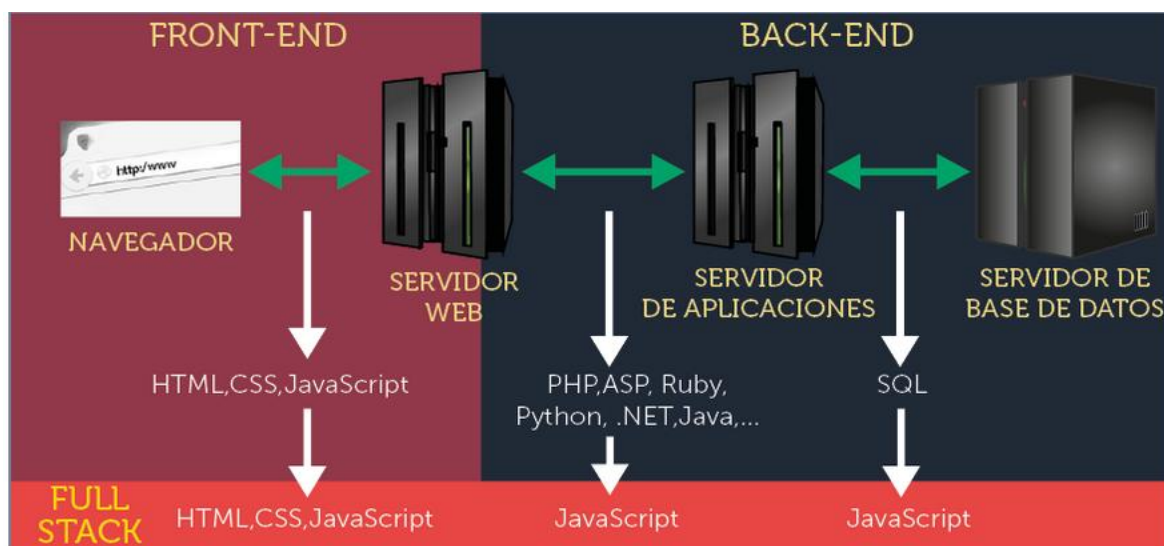
En este apartado se explica brevemente el funcionamiento de una aplicación web compleja. El único propósito es indicar que las aplicaciones web utilizan numerosas tecnologías.

En la mayoría de aplicaciones web complejas, cuando un navegador requiere de dicha aplicación, el servidor necesita ejecutar acciones como: interpretar código del lado del servidor (por ejemplo en **PHP** o **ASP.Net**), pedir recursos a otros servidores (bases de datos, mapas, streaming de vídeo, etc.) o almacenar datos de sesión.

Estas acciones son totalmente opacas al usuario. Tras su ejecución, el navegador del usuario recibe código que sí es capaz de interpretar (normalmente HTML, CSS y JavaScript).

Esto implica que una aplicación web puede realizar tareas en el llamado **lado del servidor**. Es decir, tareas que se ejecutan en el servidor y cuyo código no podemos obtener. Por lo tanto, cuando se crea una aplicación web hay personas dedicadas a programar en código traducible en el lado del servidor y código en el lado del cliente,

El lado del servidor se asocia al desarrollo **back-end**, su programadores son denominados programadores back-end. Mientras que los profesionales encargados de la apariencia en el navegador, son llamados programadores front-end.



9. Guerra de navegadores.

Los programas capaces de traducir el código HTML y producir una salida en pantalla de los mismos son los **navegadores** (*browsers* en inglés). Se trata de un software gráfico que se inició con la creación de **Mosaic** a principios de los 90 y que ha continuado con otros muchos hasta ahora.

El más popular de los navegadores empezó siendo **Netscape Navigator**; pero a finales de los 90 se habló de una **guerra de navegadores** entre **Internet Explorer** de **Microsoft** y el propio **Navigator** de **Netscape**. Poco a poco Explorer (que apareció instalado por defecto en los sistemas Windows de forma gratuita) ganó la batalla. A principios del siglo XXI incluso se dio por zanjada la guerra.

En esos años ya apreció el problema fundamental de la falta de estandarización, ya que el código HTML que entendía Explorer, no era el mismo que entendía Navigator. A veces incluso los diseñadores hacían dos versiones de la misma página.

Con la guerra “supuestamente” terminada, los diseñadores fundamentalmente probaban las páginas web para Explorer. Pero Tras recoger la función **Mozilla** el código fuente de Navigator, lanzó un nuevo navegador llamado Firefox, comenzando una nueva guerra que se volvió más fuerte al aparecer también el navegador **Chrome** de **Google**, o **Safari** de **Apple** (sin olvidar al ya veterano **Opera**).

Es importante conocer el uso de navegadores por parte de los usuarios, así el sitio w3schools (<https://www.w3schools.com/browsers/>) que posee 45 millones de visitas posee estas visitas:

- Google Chrome. 79,6 %
- Mozilla Firefox: 10,6%%
- Internet Explorer y Microsoft Edge: 3,7%
- Apple Safari: 3,3%
- Opera: 1,5%

Sin embargo el sitio **statcounter** analiza más variedad de sitios para sus estadísticas. Están disponibles en <http://gs.statcounter.com/> y ofrece estos otros datos de uso:

- Google Chrome: 60,6 %
- Apple Safari: 14,85%
- UC Browser 5,42%
- Mozilla Firefox: 5,01%
- Opera 3,38%
- Samsung Internet 2,82%
- Internet Explorer/Microsoft Edge 1,99%

Finalmente **netmarketshare** (<http://netmarketshare.com/>), otro sitio reconocido de estadísticas, ofrece estos datos

- Google Chrome: 64,10 %
- Apple Safari: 16,38%
- Mozilla Firefox: 4,83%
- Microsoft Internet Explorer/Edge: 4,26%
- Opera: 1,81%
- UC Browser: 1,3%

Aunque parece claro el dominio de Chrome en todas las estadísticas, el resto de datos varían bastante. Por ello conviene examinar estadísticas sobre nuestros propios visitantes en nuestra página web. Herramientas como **Google Analytics**, **Urchin** u otras, nos ayudarán a conseguir información detallada al respecto y así tener en cuenta qué navegadores usan nuestros usuarios.

Estandarización

Las incompatibilidades existentes en los navegadores han supuesto un problema desde el inicio de la web. La solución pasó por intentar estandarizar el lenguaje. Por ello, el propio Tim Bernes Lee fundó la **World Wide Web Consortium** (abreviado **W3C**) como organismo de estandarización del lenguaje HTML ante la industria. Este organismo se encarga de proponer estándares para el lenguaje HTML con la esperanza de que sea aceptado por los fabricantes de navegadores.

Con esta finalidad W3C trabaja con recomendaciones, a las que pone un número. Así ha habido HTML 2, HTML 3, HTML 4, etc. La W3C también se encarga de estandarizar otras tecnologías y lenguajes relacionados con HTML como CSS, JavaScript, SVG o el propio XML.

Las directrices de W3C son seguidas por la mayoría de navegadores aunque no al 100%, lo que sigue generando problemas a los creadores de páginas web. Por si fuera poco, este organismo ha perdido peso al apostar por XHTML, que es un HTML que usa las normas de XML y que ha perdido la batalla de ser el estándar definitivo en favor de HTML 5, que, inicialmente, no fue apoyado por este organismo.

Las tres últimas familias de estándares han sido

HTML4: muy permisivo, lo que dificulta a los navegadores el procesar el HTML dando lugar a que fuera bastante difícil para ellos el mostrar correctamente y de igual forma todos los HTML.

XHTML: es HTML con las estrictas reglas que impuso XML. Esto simplificó el desarrollo de navegadores y se avanzó en facilidad para mostrar páginas en distintos navegadores.

HTML5: es una nueva revisión de XHTML en el que se han incluido nuevas posibilidades como etiquetas <audio> y <video> así como posibilidad de hacer muchas cosas desde JavaScript.

10. Validar código HTML.

Los navegadores no son estrictos con las normas HTML; es decir, aunque tengamos algunos fallos, van a ignorarlos en aras de mostrar la mejor versión del código que hagamos. A los navegadores les interesa hacer funcionar el código, sea válido o no.

Sin embargo, como creadores de páginas web, debemos cumplir las normas del lenguaje aprendiendo así realmente el lenguaje y además impidiendo que el navegador se equivoque al interpretar nuestros fallos. Además, hace que otros creadores de páginas comprendan el código perfectamente, facilitando el trabajo colaborativo.

El validador oficial de la W3C (<http://validator.w3.org/>) permite validar todas las versiones de HTML, incluida la versión 5.

Los errores graves se muestran en rojo y puede haber avisos (*warnings*) que son fallos leves, pero que hay que tener en cuenta.

Por otro lado, la mayor parte de los entornos de trabajo dispone de validador incorporado (en el propio entorno se marcan los fallos a la vez que escribimos el código) y algunos otros se puede instalar un plugin.

También podemos instalar validadores en línea de comandos como el **htmltidy**, disponible en <http://www.html-tidy.org/>, que permiten validar desde la línea de comandos, aunque esta opción cada vez se usa menos.

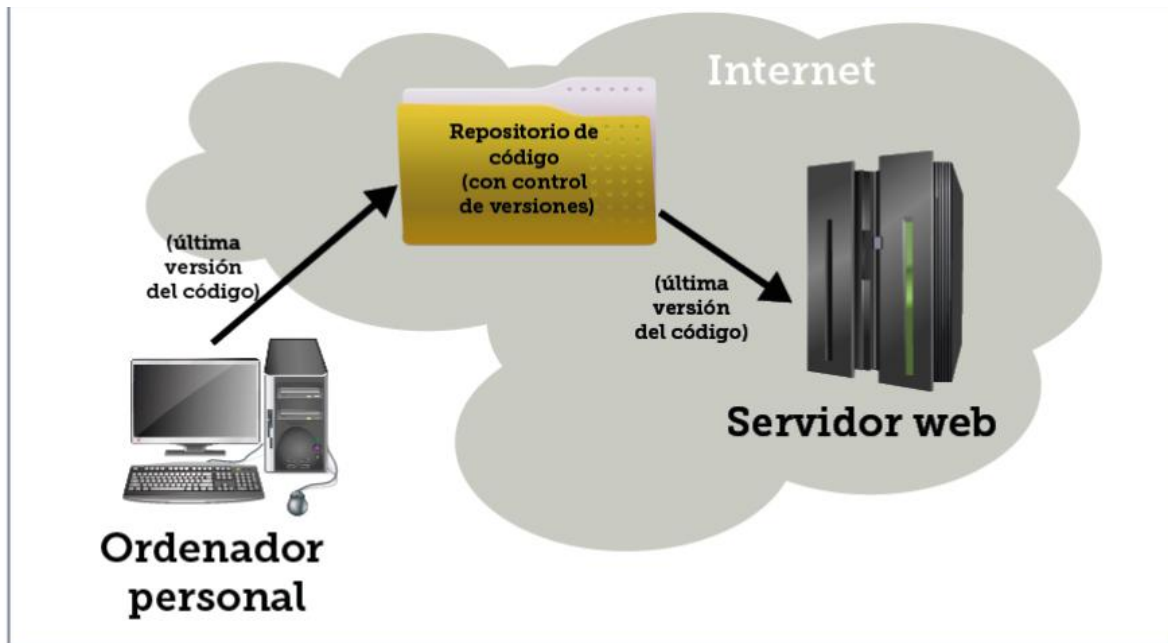
11. Publicación de páginas web.

Se denomina **sitio web** al conjunto de páginas web y recursos de las mismas que contienen toda la información asociada a una determinada dirección de inicio en Internet.

Cuando una persona desea crear un nuevo sitio web, inicialmente le crea en su ordenador de trabajo y para ello debe crear una carpeta y en ella almacenar todas las páginas y recursos necesarios (imágenes, sonidos, vídeos, archivos auxiliares,...). Esa carpeta se deberá enviar al servidor web que hayamos contratado o del que dispongamos para publicar nuestra página en Internet.

Para ello normalmente se utiliza el protocolo de transmisión de ficheros conocido como **FTP**, aunque es posible que esa transmisión se haga con otros protocolos como **WebDAV** o **RDS**. Con copiar la carpeta en el sitio adecuado de nuestro servidor, la página estará publicada. Normalmente para ello se nos pide un usuario y contraseña que verifica que realmente somos los propietarios del espacio.

Otra opción, más habitual hoy en día para los desarrolladores, es utilizar sistemas de control de versiones en la nube (el más conocido es **GitHub**) y descargar la última versión del código directamente al servidor web. Es decir, desde nuestro ordenador subimos la última versión de nuestro código al servidor en la nube y luego actualizamos el código en el servidor web.



12. Bibliografía.

- <https://www.w3schools.com/>
- <https://es.wikipedia.org/wiki/HTML>
- <https://desarrolloweb.com/>
- <http://jorgesanchez.net>
- <https://www.campusmvp.es>
- https://oscarmaestre.github.io/lenguajes_marcas/index.html