

Escuela: **EPET N1 CAUCETE**

Docente: **Luis Fernando Zalazar**

Grado, Año, Ciclo y/o Nivel: **7º año CS**

Turno: **tarde**

Área Curricular: **Adaptación del Ambiente de Trabajo**

Título de la Propuesta: **HTML (Continuación)**

Director: **Mario Gómez**

Actividades semanas 3º y 4º

- **Lectura 1:**

El navegador web

Se trata de la puerta de acceso a los servicios que ofrece la Web. Como ya sabemos, la Web se basa en una arquitectura cliente-servidor, donde el servidor es el programa que atiende las peticiones que hacen los navegadores y les proporciona los recursos que solicitan mediante el protocolo HTTP. Además, cada servidor web dispone de programas servidores para la realización de páginas dinámicas (CGI, Servlets, ASP, JSP, PHP, etc.).

El cliente o navegador web (web browser) realiza las peticiones al servidor y presenta al usuario las páginas web que dicho servidor le suministra. Así pues, en su versión más simple, el navegador debe interpretar el código HTML en que están escritas y presentarlo al usuario para que este pueda interactuar con el contenido y navegar hacia otros lugares a través de hiperenlaces. Los navegadores actuales también interpretan las hojas de estilo en cascada (CSS), que indican cómo deben presentarse los datos, el código JavaScript que añade dinamismo a las páginas, además de otras extensiones, como Flash o Java, que aumentan las posibilidades de presentación.

Navegador	URL
Internet Explorer	http://windows.microsoft.com/es-ES/internet-explorer/products/ie/home
Mozilla Firefox	www.mozilla-europe.org/es/firefox
Google Chrome	www.google.com/chrome
Safari	www.apple.com/es/safari
Opera	www.opera.com

Tabla 1 *Páginas web de los principales navegadores.*

Navegadores web de escritorio

Los principales cinco navegadores que acaparan prácticamente más del 95 % del mercado son los siguientes:

- Internet Explorer (IE). Desarrollado por Microsoft, se halla integrado en Windows. Desde 1999 era el más utilizado (aunque no lo parezca hasta el 2019 gracias a que con Windows

10 no venía incorporado), en los últimos años ha perdido gran número de usuarios. Se trata de software propietario, aunque se ofrece de manera gratuita. Actualmente Windows ofrece el navegador Edge.

- Mozilla Firefox. Desarrollado por una comunidad independiente que colabora dentro de la Fundación Mozilla. Es muy innovador y adapta rápidamente todos los estándares que van surgiendo. Nació de la liberalización del código del navegador Navigator de la empresa Netscape. Por lo que es de código abierto y libre, lo que le permite estar disponible para muchos sistemas operativos: Windows, Mac y GNU/Linux.

- Google Chrome. Con una interfaz de usuario sencilla y eficiente, y garantizar una navegación más estable, rápida y segura. Es propiedad de Google, basado en componentes de código abierto. Está disponible para Windows, Mac y GNU/Linux (también para la mayoría de los smartphones). Actualmente el mas utilizado.

- Safari. Desarrollado por Apple para Mac OS X, aunque también existe una versión para Windows. Posee un diseño bastante atractivo, cómodo e intuitivo, y diversas características funcionales como el reproductor de Apple QuickTime.

- Opera. Creado por la empresa Opera Software, gran aceptación gracias a su velocidad, seguridad, soporte de estándares y reducido tamaño. Se encuentra disponible para GNU/Linux, Windows, Mac OS X, Solaris y FreeBSD.

Evolución HTML

El lenguaje HTML surgió de la necesidad de representar y relacionar documentos de hipertexto, y poder comunicarlos a través del protocolo HTTP. Al mismo tiempo que ha ido evolucionando la Web y el uso que se le ha dado, el lenguaje HTML ha evolucionado con ellos, dando lugar a diferentes versiones o estándares del mismo, introducidos inicialmente por el IETF y posteriormente por el W3C.

Estándar	Publicado en...	Algunas características...
HTML 2.0 (IETF)	1995	Estándar inicial (no existe versión 1.0). La declaración de elementos de estructura (html, head y body) es opcional. Sin soporte para tablas.
HTML 3.2 (W3C)	1997	Simplificación del HTML 3.0 propuesto por IETF. Su elaboración por parte de compañías como Netscape, Microsoft, IBM o Sun, dio lugar al nacimiento del W3C. Incluye soporte para applets, mayor versatilidad en la disposición del texto y las imágenes, y añade las tablas.
HTML 4.0	1997	Estandariza los marcos, las hojas de estilo CSS y los scripts, incluye más soporte multimedia y mejoras de accesibilidad, entre otras cosas.
HTML 4.01	1999	Es la más usada actualmente. Compatible con la mayoría de navegadores. Dispone de tres subestándares: Strict, Transitional, y Frameset.
HTML 4.01 Strict		No se aceptan etiquetas obsoletas (deprecated) de estándares anteriores.
HTML 4.01 Transitional		Permite el uso de todo tipo de etiquetas, aunque sean obsoletas.
HTML 4.01 Frameset		Da soporte a frames.
XHTML 1.0	2000	Surge como una combinación de XML y HTML.
XHTML 1.1	2001	Elimina el soporte para marcos y elementos obsoletos, siendo más o menos equivalente a HTML 4.01 Strict.
HTML 5.0	2014	Incluye muchas novedades, entre las que destacan nuevas etiquetas semánticas y un mayor soporte e integración de contenidos multimedia.

Tabla 2 Versiones de HTML.

Entornos

Las páginas web son documentos de texto plano, cuyo código HTML es interpretado por el navegador web. Aunque, para la edición de estas páginas, bastaría con un simple editor de texto, existen aplicaciones y entornos específicos para tal fin, con diferentes grados de complejidad. Por lo tanto tenemos la siguiente clasificación de estos entornos:

- ✓ Editores de texto plano. Se trata de editores de texto bastante simples, como pueden ser el bloc de notas, SublimeText, NotePad++ en Windows, o gedit y Kate para GNU/Linux. Algunos de ellos incluyen características de coloreado de etiquetas, e incluso botones para insertar las etiquetas directamente. Entornos más completos, como Dreamweaver, Komodo Edit o Aptana nos permiten gestionar proyectos complejos, con múltiples ficheros de diferente tipos, e incluso soportar frameworks de desarrollo.

- ✓ Editor de texto con ventanas desdobladas. Muestran el resultado en tiempo real de la interpretación del código HTML en una ventana adyacente a la del código que estamos escribiendo. Es una opción bastante cómoda para no tener que guardar cada vez las páginas y abrirlas con el navegador. Ejemplos de estos editores los encontramos en Bluefish, BlueGriffon en Linux o Coda para MAC.

- ✓ Editores WYSIWYG (What You See Is What You Get, «lo que ves es lo que tienes»). Ofrecen la posibilidad de trabajar directamente con el resultado de la página, como si de un procesador de textos se tratase. El propio editor se encargará de la generación del documento HTML. Aunque ampliamente difundidos, estos editores han sido criticados por generar páginas de poca calidad, y han dado lugar a editores WYSIWYM (What You See Is What You Mean, «lo que ves es lo que quieres decir»), donde el usuario introduce los contenidos de forma estructurada, siguiendo su valor semántico (un título, una sección...), en lugar de su representación final. Algunos editores WYSIWYM son WYMean Editor o WebCS Editor. Dentro de los editores WYSIWYG podemos destacar Amaya (creado por la W3C), KompoZer o Dreamweaver (para entornos MS, con licencia), entre muchos otros.

- ✓ Editores on-line. Otro tipo de editores son los que nos permiten la edición de código HTML directamente sobre el navegador, facilitando así la subida al servidor y colaborar con otras personas. Generalmente utilizan coloreado de código y autotabulación, y aunque no se trate de entornos para desarrollar grandes sitios son de gran utilidad para experimentar con HTML, CSS y JavaScript, viendo el resultado a medida que se escribe, así como para compartir código. Algunos de estos sitios son Tinkerbin (<http://tinkerbin.com/>) que permite editar y combinar HTML, CSS y Javascript, CSSDesk (<http://cssdesk.com/>), orientado a la edición HTML y CSS, JSFiddle (<http://jsfiddle.net/>), que ofrece además librerías como Mootools o jQuery, JSBin (<http://jsbin.com/>), también con la posibilidad de incrustar diferentes librerías JQuery, o Dabblet (<http://dabblet.com/>), que permite guardar nuestras muestras directamente en GitHub.

- **Lectura 2**

Enlaces

Los links, vínculos, hipervínculos o enlaces se utilizan para establecer relaciones entre dos recursos. Como HTML es un lenguaje de HiperTexto entonces los enlaces son denominados comúnmente **Hiperenlaces**. La mayoría de enlaces relacionan páginas web, pero también es posible enlazar otros recursos como imágenes, documentos y archivos. Antes de empezar a incluir enlaces en las páginas HTML, es necesario comprender y dominar el concepto de URL. El acrónimo URL (del inglés Uniform Resource Locator) hace referencia al identificador único de cada recurso disponible en Internet. Un URL se construye de la siguiente manera: **protocolo://maquina/ruta**. Donde **protocolo** puede ser http://, https://, ftp://, etc., la **máquina** es la dirección del equipo donde se encuentra el recurso y la **ruta** es el camino a seguir dentro de los directorios de la máquina para llegar hasta el recurso.

Un ejemplo típico de una URL sería:

`http://www.miservidor.com/documentos/fisica/index.html`

La home page o index de cualquier servidor web, se carga con solo escribir en el navegador la dirección del servidor. Por ejemplo, para acceder a la página de bienvenida en nuestro ejemplo anterior basta con ir a la dirección

`http://www.miservidor.com/documentos/fisica` para acceder a la página de bienvenida.

Los enlaces permiten vincular partes del documento con otros documentos o con otras partes del mismo documento. Por ejemplo, que al pulsar con el ratón sobre un texto o sobre una imagen se nos redirija a una nueva Web con un contenido diferente.

href

Para crear un enlace se utiliza la etiqueta `` cuyo atributo href establece la dirección URL a la que apunta el enlace. Por ejemplo, un enlace a la Wikipedia sería de la forma:

```
<a href=http://es.wikipedia.org target="_blank" title="La Wikipedia">Wikipedia</a>
```

Partes (propiedades)

href es donde colocamos la ruta y el nombre del archivo de la página de destino, se pueden enlazar otras páginas web, lugares dentro de la misma página, archivos de música, archivos de imágenes, etc.

- Ejemplo ir a otra página fuera de mi sitio: un enlace a una página web en internet:

```
href=http://es.wikipedia.org
```

- Un ejemplo de enlace a una página dentro de mi sitio web:

href="noticias/noticias-de-actualidad.html" que es una página también de mi web y que está dentro de una carpeta llamada "noticias" y cuyo nombre de archivo es "noticias-de-actualidad.html"

target que puede tomar los siguientes valores:

target="_top": Se usa esta opción cuando queremos que el archivo enlazado se muestre en una pantalla completa de la ventana eliminando los frames o marcos si los hay. Esta opción se utiliza solo cuando nuestra página web esté hecha con frames o marcos.

target="_parent": Con esta opción la página destino se muestra en marco anterior al marco o frame en el que está el enlace.

target="_self": Con esta tercera opción, la página de destino a la que apunta en vínculo se mostrará en la misma ventana del navegador del visitante, es decir, no se abrirá en una ventana aparte. Esta opción es la que se toma por defecto

target="_blank": Con esta opción la página enlazada se abrirá en una ventana nueva del navegador. Resulta útil cuando queremos enviar a las visitas a una página externa, fuera de nuestra web. Así se mantendrá nuestra web abierta en otra ventana y no perdemos esa visita, pues tiene nuestra página a mano. También la puedes usar cuando quieres mostrar otra página de tu web, pero sin que el visitante pierda de vista la página en la que estaba.

alt="Texto Alternativo": Se puede definir un texto alternativo con esta propiedad.

title="Texto Alternativo": En caso de usarlo será este el texto que aparecerá al dejar el cursor del ratón sobre el enlace.

Anclajes

En el apartado anterior se dijo que se podía ir a lugares dentro de la misma página, dado que en ocasiones, tenemos páginas con un contenido tan grande que nos interesa enviar a estas visitas no a una página sino a una "parte" concreta de una página de nuestra web. Para estos casos Html define un enlace especial que denomina Ancla o Anclaje. Vamos a ver cómo podemos colocarlos en nuestro sitio.

Usando las anclas o anclajes puedes hacer que cuando las visitas pulsen sobre ese enlace, la ventana del navegador corra hacia esa parte concreta de la misma página al comienzo, a la parte media o a la parte que tú desees de la misma página.

Si defines un ancla o un anclaje en cada uno de sus títulos puedes luego colocar vínculos de modo que al pulsarlos se dirija justo a esa parte de la página. Si defines un ancla al principio del contenido y colocas un enlace en la parte inferior indicando "Ir al principio de la página", al pulsarlo se dirigirá automáticamente a la parte superior, aún siendo la misma página.

Para empezar hay que definir esa parte a la que quieres enviar a las visitas cuando hagan clic sobre ese enlace que pondremos después. Por ejemplo, veamos cómo se define un ancla en la parte superior.

Funcionamiento de las anclas

Colocar esta línea de código (ancla) en la parte correspondiente (arriba justo al comienzo del body):

```
<a name="arriba"></a>
```

Con esto definimos un lugar al que podemos enviar a las visitas si pulsan el enlace

Ahora en la parte inferior colocamos otra línea (ancla) como esta:

```
<a name="abajo"></a>
```

Tendremos la posibilidad de ir a la parte baja de esa página web.

Ahora falta colocar el propio enlace. Si queremos enviar a las visitas a la parte superior, lo normal es colocar el enlace en la parte inferior (al final de la página). Esto se puede hacer:

```
<a href="#arriba" title="Ir Arriba">Ir arriba</a>
```

Del mismo modo, podríamos colocar arriba (al principio) de la página otro enlace que permitiera ir al parte inferior:

```
<a href="#abajo" title="Ir Abajo">Ir abajo</a>
```

Generalmente esto es muy útil cuando hay mucho texto y se tienen partes diferenciadas. Se define un ancla en cada uno de esos títulos, y luego un menú en la parte inferior, o en la parte superior, o en ambas, dando la opción a las visitas a dirigirse, dentro de esa misma página, a la sección que deseen.

Fuentes:

Introducción a la Programación Web – Antonio Javier Gallego Sánchez; Fundamentos Básicos de HTML + PHP - Wil A. Alvarez R.; Aplicaciones web - Raúl V. Lerma-Blasco José Alfredo Murcia Andrés Elvira Mifsud Talón; <https://www.comocreartuweb.com/>

- **Actividades**

1. Crear un archivo llamado ejercicio3.html cuyo contenido sea la lectura 1 de esta guía, utilizando todos los conocimientos adquiridos hasta ahora.
2. Colocar las anclas a todos los títulos para colocar al principio y final del archivo unos menús (links a los títulos) que permitan recorrer el archivo.

Consultas: luizone890@gmail.com