

TEMA 5. ENLACES

1. URL.....	2
2. ENLACES RELATIVOS Y ABSOLUTOS.....	3
3. ENLACES BÁSICOS.....	4
4. ENLACES A UN EMAIL.....	6
5. ATRIBUTO TARGET.....	7

El lenguaje de marcado HTML se definió teniendo en cuenta algunas de las características que existían en ese momento para la publicación digital de contenidos. Entre los conceptos utilizados en su creación, se encuentra el mecanismo de "hipertexto".

De hecho, las letras "HT" de la sigla HTML significan "hipertexto" (hypertext en inglés), por lo que el significado completo de HTML podría traducirse como "lenguaje de marcado para hipertexto".

La incorporación del hipertexto fue una de las claves del éxito del lenguaje HTML, ya que permitió crear documentos interactivos que proporcionan información adicional cuando se solicita. El elemento principal del hipertexto es el "hiperenlace", también llamado "enlace web" o simplemente "enlace".

Los enlaces se utilizan para establecer relaciones entre dos recursos. Aunque la mayoría de los enlaces relacionan páginas web, también es posible enlazar otros recursos como imágenes, documentos y archivos.

1. URL

Antes de empezar a crear enlaces, 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. Las URL son esenciales para crear los enlaces, pero también se utilizan en otros elementos HTML como las imágenes y los formularios.

La URL de un recurso tiene dos objetivos principales:

- ✓ Identificar de forma única a ese recurso
- ✓ Permitir acceder de forma eficiente ese recurso

En primer lugar, las URL permiten que cada página HTML publicada en Internet tenga un nombre único que permita diferenciarla de las demás. De esta forma es posible crear enlaces que apunten de forma inequívoca a una determinada página.

Si se accede a la página principal de Google, la dirección que muestra el navegador es:

<https://www.google.com>

La cadena de texto <http://www.google.com> es la URL completa de la página principal de Google. La URL de las páginas es imprescindible para crear los enlaces, ya que permite distinguir una página de otra.

El segundo objetivo de las URL es el de permitir la localización eficiente de cada recurso de Internet. Para ello es necesario comprender las diferentes partes que forman las URL. Una URL sencilla siempre está formada por las mismas tres partes. Si por ejemplo se considera la siguiente URL:

<https://www.temarioDIW.es/tema5.html>

Las partes que componen la URL anterior son:

- ✓ Protocolo (<http://> o <https://>): el mecanismo que debe utilizar el navegador para acceder a ese recurso. Todas las páginas web utilizan <http://>. Las páginas web *seguras* (por ejemplo, las de los bancos y las de los servicios de email) utilizan <https://> (se añade una letra s).
- ✓ Servidor (www.temario.es): simplificando mucho su explicación, se trata del ordenador en el que se encuentra guardada la página que se quiere acceder. Los navegadores son capaces de obtener la dirección de cada servidor a partir de su nombre.
- ✓ Ruta ([./tema5.html](http://www.temario.es/tema5.html)): *camino* que se debe seguir, una vez que se ha llegado al servidor, para localizar el recurso específico que se quiere acceder.

Por tanto, las URL no sólo identifican de forma única a cada recurso de Internet, sino que también proporcionan a los navegadores la información necesaria para poder llegar hasta ese recurso.

La mayoría de URL son tan sencillas como la URL mostrada anteriormente. No obstante, existen URL complejas formadas por más partes.

<http://www.alistapart.com/comments/webstandards2008?page=5>

Las cinco partes que forman la URL anterior son:

- ✓ Protocolo (http://)
- ✓ Servidor (www.alistapart.com)
- ✓ Ruta (/comments/webstandards2008)
- ✓ Consulta (?page=5): información adicional necesaria para que el servidor localice correctamente el recurso que se quiere acceder. Siempre comienza con el carácter ? y contiene una sucesión de palabras separadas por = y &

Como las URL utilizan los caracteres :, =, & y / para separar sus partes, estos caracteres están reservados y no se pueden utilizar libremente. Además, algunos caracteres no están reservados, pero pueden ser problemáticos si se utilizan en la propia URL.

Si es necesario incluir estos caracteres reservados y especiales en una URL, se sustituyen por combinaciones de caracteres seguros. Esta sustitución se denomina *codificación* de caracteres y el servidor realiza el proceso inverso (*decodificación*) cuando le llega una URL con los caracteres codificados.

A continuación, se muestra la tabla para codificar los caracteres más comunes:

Carácter original	Carácter codificado	Carácter original	Carácter codificado
/	%2F	?	%3F
:	%3A	@	%40
=	%3D	&	%26
"	%22	\	%5C
'	%27	~	%7E
(espacio en blanco)	%20	#	%23

2. ENLACES RELATIVOS Y ABSOLUTOS

Las páginas web habituales suelen contener decenas de enlaces de diferentes tipos. La siguiente imagen muestra algunos de los tipos de enlaces de la página principal del sitio web www.thinkvitamin.com:



Figura 5.1 Ejemplo de diferentes tipos de enlaces en la página 456BereaStreet.com

En esa página, cuando se pincha sobre algunos enlaces, el navegador abandona el sitio web para acceder a páginas que se encuentran en otros sitios. Estos enlaces se conocen como "enlaces externos". Sin embargo, la mayoría de los enlaces de un sitio web apuntan a páginas del propio sitio web, por lo que se denominan "enlaces internos".

Además de internos/externos, la otra característica que diferencia a los enlaces (y por tanto, también a las URL) es si el enlace es absoluto o relativo. Las URL absolutas incluyen todas las partes de la URL (protocolo, servidor y ruta) por lo que no se necesita más información para obtener el recurso enlazado.

A continuación, se resumen los cuatro posibles casos de URL relativas y el procedimiento que sigue el navegador para convertirlas en URL absolutas:

Si la URL relativa...	El navegador la transforma en URL absoluta...
...sólo consiste en el nombre de un recurso	...añadiendo el protocolo, servidor y ruta completa del origen del enlace
...comienza por ../	...añadiendo el protocolo y servidor del origen del enlace, subiendo un nivel en la jerarquía de directorios y añadiendo el resto de la ruta incluida en la URL relativa
...comienza por /	...añadiendo el protocolo y servidor del origen del enlace
En cualquier otro caso	...añadiendo el protocolo, servidor y ruta completa del origen del enlace, a la que se añade la ruta incluida en la URL relativa

3. ENLACES BÁSICOS

Los enlaces en HTML se crean mediante la etiqueta `<a>` (su nombre viene del inglés "*anchor*", literalmente traducido como "ancla"). A continuación, se muestra la definición simplificada de `<a>` y más adelante se muestra su definición completa:

Etiqueta	<code><a></code>
Atributos propios	<p><code>name = "texto"</code> - Permite nombrar al enlace para que se pueda acceder desde otros enlaces.</p> <p><code>href = "url"</code> - Indica la URL del recurso que se quiere enlazar</p>
Tipo de elemento	En línea
Descripción	Se emplea para enlazar todo tipo de recursos

El atributo más importante de la etiqueta `<a>` es `href`, que se utiliza para indicar la URL a la que apunta el enlace. Cuando el usuario pincha sobre un enlace, el navegador se dirige a la URL del recurso indicado mediante `href`. Las URL de los enlaces pueden ser absolutas, relativas, internas y externas.

Con la definición anterior, para crear un enlace que apunte a la página principal de Google solamente habría que incluir lo siguiente en un documento HTML:

```
<a href="https://www.google.com">Página principal de Google</a>
```

Como el atributo `href` indica una URL, un enlace puede apuntar a cualquier tipo de recurso al que pueda acceder el navegador. El siguiente enlace apunta a una imagen, que se mostrará en el navegador cuando el usuario pinche sobre el enlace:

```
<a href="http://www.ejemplo.com/fondo_escritorio.jpg">Imagen interesante para un fondo de escritorio</a>
```

De la misma forma, los enlaces pueden apuntar directamente a documentos PDF, Word, etc.

```
<a href="http://www.ejemplo.com/informe.pdf">Descargar informe completo [PDF]</a>
<a href="http://www.ejemplo.com/informe.doc">Descargar informe completo [DOC]</a>
```

Un truco muy útil con los enlaces es el uso de URL relativas para poder volver al inicio del sitio web desde cualquier página web interior:

```
<a href="/">Volver al inicio</a>
```

El enlace anterior utiliza una URL relativa con una ruta que apunta directamente a la raíz del servidor. Para obtener la URL absoluta, el navegador añade el mismo protocolo y el mismo nombre de servidor de la página en la que se encuentra el enlace. El resultado es que cuando se pincha ese enlace, siempre se vuelve al inicio del sitio web, independientemente de la página en la que se incluya el enlace.

El otro atributo básico de la etiqueta `<a>` es `name`, que permite definir enlaces dentro de una misma página web. Si una página es muy larga, puede ser útil mostrar enlaces de tipo "Saltar hasta la segunda sección", "Volver al principio de la página", etc.

Este tipo de enlaces son especiales, ya que la URL de la página siempre es la misma para todas las secciones y por tanto, debe añadirse otra parte a las URL para identificar cada sección.

```
<a name="primera_seccion"></a>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris id ligula eu felis adipiscing ultrices. Duis gravida leo ut lectus. Praesent condimentum mattis ligula.</p>
```

...

```
<a name="segunda_seccion"></a>
<p>Pellentesque malesuada. In in lacus. Phasellus erat erat, lacinia a, convallis eu, nonummy et, odio. Aenean urna elit, ultrices id, placerat varius, facilisis eget, dolor.</p>
```

El atributo `name` permite crear "enlaces vacíos" que hacen referencia a secciones dentro de una misma página. Una vez definidos los "enlaces vacíos", es posible crear un enlace que apunte directamente a una sección concreta de una página:

```
<!-- Enlace normal a la página -->
<a href="http://www.ejemplo.com/pagina1.html">Enlace a la página 1</a>

<!-- Enlace directo a la segunda sección de la página -->
<a href="http://www.ejemplo.com/pagina1.html#segunda_seccion">Enlace a la sección 2 de la página 1</a>
```

La sintaxis que se utiliza con estos enlaces es la misma que con los enlaces normales, salvo que se añade el símbolo `#` seguido del nombre de la sección a la que se apunta. Cuando el usuario pincha sobre uno de estos enlaces, el navegador accede a la página apuntada por la URL y baja directamente a la sección cuyo nombre se indica después del símbolo `#`.

También es posible utilizar este tipo de enlaces con URL relativas en una misma página. El siguiente ejemplo añade enlaces de tipo "Volver al inicio de la página" en varias secciones:

```
<a name="inicio"></a>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris id ligula eu felis adipiscing ultrices. Duis gravida leo ut lectus. Praesent condimentum mattis ligula.</p>
<a href="#inicio">Volver al inicio de la página</a>
```

...

```
<p>Pellentesque malesuada. In in lacus. Phasellus erat erat, lacinia
a, convallis eu, nonummy et, odio. Aenean urna elit, ultrices id,
placerat varius, facilisis eget, dolor.</p>
<a href="#inicio">Volver al inicio de la página</a>
```

...

Los enlaces directos a secciones también funcionan con el atributo id de cualquier elemento. El siguiente ejemplo es equivalente al ejemplo anterior:

```
<h1 id="inicio">Título de la página</h1>
```

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris id
ligula eu felis adipiscing ultrices. Duis gravida leo ut lectus.
Praesent condimentum mattis ligula.</p>
<a href="#inicio">Volver al inicio de la página</a>
```

...

```
<p>Pellentesque malesuada. In in lacus. Phasellus erat erat, lacinia
a, convallis eu, nonummy et, odio. Aenean urna elit, ultrices id,
placerat varius, facilisis eget, dolor.</p>
<a href="#inicio">Volver al inicio de la página</a>
```

...

El nombre de la sección que se indica después del símbolo # puede utilizar el valor de los atributos id de cualquier elemento. De hecho, se recomienda utilizar los atributos id de los elementos ya existentes en la página en vez de crear "enlaces vacíos" de tipo ``.

A partir de la estructura de directorios y archivos indicada la ruta en la siguiente imagen:

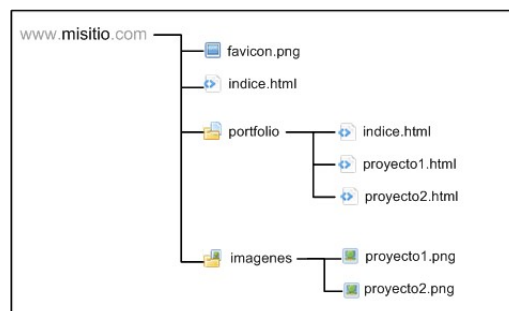


Figura 5.2 Estructura de archivos y directorios de un sitio web de ejemplo

4. ENLACES A UN EMAIL

```
<a href="mailto:nombre@direccion.com">Enviar correo a nombre</a>
```

Al pinchar sobre el enlace anterior, se abre automáticamente el programa de correo electrónico del ordenador del usuario y se establece la dirección de envío al valor indicado después de mailto: La sintaxis es la misma que la de un enlace normal, salvo que se añade el prefijo mailto:

La sintaxis de mailto: permite utilizarlo para otros ejemplos más complejos:

```
<!-- Envío del correo electrónico a varias direcciones a la vez -->
<a href="mailto:nombre@direccion.com,otro_nombre@direccion.com">
Enviar correo a nombre y a otro_nombre al mismo tiempo </a>

<!-- Añadir un "asunto" inicial al correo electrónico -->
<a href="mailto:nombre@direccion.com?subject=Solicitud de más
información"> Enviar correo a nombre con un asunto</a>

<!-- Añadir un texto inicial en el cuerpo del correo electrónico -->
```

```
<a href="mailto:nombre@direccion.com?body=Estaría interesado en solicitar más información sobre sus productos"> Enviar correo a nombre con un contenido </a>
```

Todas las opciones anteriores se pueden combinar entre sí uniendo con un & cada uno de los parámetros del correo (subject, cc, cco, body).

```
<a href="mailto: nombre@direccion.com?subject=Este es el asunto &body=Este es el texto del correo"> Enviar correo a nombre con body y subjec</a>
```

Si vas a poner una frase es recomendable que no dejes espacios, pero si los tienes que escribir sustituye cada espacio en blanco por el carácter %20

Aunque el uso de mailto: puede parecer una ventaja, su uso está desaconsejado. Si se incluye una dirección de correo electrónico directamente en una página web, es muy probable que en poco tiempo esa dirección de email se encuentre llena de correo electrónico basura o "spam", ya que existen programas automáticos encargados de rastrear sistemáticamente todas las páginas web de Internet para encontrar direcciones de correo electrónico válidas.

La forma de mostrar las direcciones de correo electrónico en las páginas web consiste en incluir la dirección en una imagen o indicarla de forma que solamente los usuarios puedan entenderlo:

```
<p>La dirección de correo es <strong>nombre (arroba) direccion.com</strong> </p>
```

```
<p>La dirección de correo es <strong> nombre_arroba_direccion.com</strong></p>
```

```
<p>La dirección de correo es <strong> nombreQUITAESTO@direccion.com</strong></p>
```

```
<p>La dirección de correo es <strong> nombre(ARROBA)direccion.com</strong></p>
```

```
<p> La dirección de correo es <strong> nombre @ direccion . com</strong> </p>
```

5. ATRIBUTO TARGET

Especifica en donde desplegar la URL enlazada, puede tomar los siguientes valores:

- ✓ `_self`: Carga la URL en el mismo contexto de navegación que el actual. Este es el comportamiento por defecto.
- ✓ `_blank`: Carga la URL en un nuevo contexto de navegación. Usualmente es una pestaña, sin embargo, los usuarios pueden configurar los navegadores para utilizar una ventana nueva en lugar de la pestaña.
- ✓ `_parent`: Carga la URL en el contexto de navegación padre (parent) del actual. Si no existe el padre, este se comporta del mismo modo que `_self`.
- ✓ `_top`: Carga la URL en el contexto más alto de navegación (el cual es un ancestro del actual, y no tiene padre (parent)). Si no hay padre (parent), este se comporta del mismo modo que `_self`.