

UD1: Aplicacións web e linguaxes de marcas: HTML e CSS

ACTIVIDADE 2: Conceptos básicos de HTML.

1. HTML. Lenguaje para crear páginas Web

Antes de ver en que consiste HTML, vamos a ver qué es el *hipertexto*. El concepto de *hipertexto* responde, fundamentalmente, a una forma de acceso a la información. Cuando leemos un texto impreso como un libro o un periódico, prácticamente nadie lee desde la primera a la última página, sino que seleccionamos la información que más nos interesa a partir de lo que aparece en la portada y es a ese punto al que vamos en primer lugar. Esta posibilidad de seleccionar el itinerario que vamos a seguir es uno de los elementos que definen el *hipertexto*.

El hipertexto, también da la posibilidad de incluir, junto a la información escrita, una imagen fija, sonido e imágenes en movimiento. Por tanto, podemos considerar hipertexto como el *conjunto de información en soporte multimedia que se estructura mediante enlaces entre los diversos elementos, de forma que el usuario pueda acceder a ella rápidamente y teniendo la posibilidad de establecer su propio itinerario de "lectura"*.

Para escribir páginas web se usa HTML, iniciales de HyperText Markup Language (Lenguaje de Marcado de Hipertexto), que es una implementación específica de SGML para páginas Web y que durante años se mantuvo como la tecnología principal de la Web.

Un documento HTML es un archivo de texto al que se añaden una serie de etiquetas, delimitadas por los signos menor y mayor, que serán interpretadas por los visualizadores (browsers) a la hora de mostrar el contenido de los archivos en la pantalla. Las etiquetas son las que le indican al visualizador del documento cómo navegar, cómo cargar un gráfico, etc.

A lo largo del tiempo existieron varios tipos de HTML y dos tipos de sintaxis.

- Los tipos de HTML(HTML 3, HTML 4, HTML5 ...) se distinguen por las etiquetas, atributos y valores de atributos que soportan.
- Las dos sintaxis, HTML y XHTML, se distinguen en que la sintaxis XHTML es más *estricta* en el modo de escribir las etiquetas, atributos y valores, mientras que la sintaxis HTML es más *flexible*.

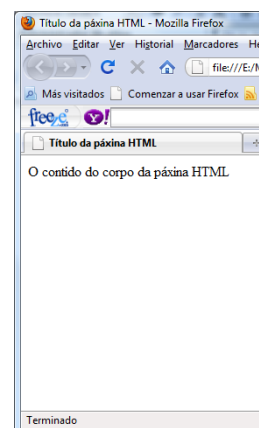
Un documento con errores de sintaxis (HTML o XHTML) se dice que es un documento *inválido*. Los navegadores consiguen mostrar la mayor parte de los documentos inválidos en HTML, lo que no quita que se aconseje validar y corregir los documentos para asegurar que los navegadores los muestren correctamente.

1.1 Creación de un documento básico HTML.

El documento HTML5 más sencillo tiene dos partes el *head* o cabecera y el *body* o cuerpo y tiene la siguiente estructura básica:

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title>Título de la página HTML</title>
  </head>
  <body>
    O contenido do corpo da página HTML
  </body>
</html>
```

donde:



<!doctype html>

- Indica que el tipo de documento y que sigue el estándar HTML5. Esto asegura que el navegador renderiza el documento usando el modo “Estándar” en lugar del modo “Quirks” (modo de compatibilidad), con lo que garantizamos la correcta visualización y comportamiento del sitio web siguiendo las normas de HTML5.

<html></html>

- Estas etiquetas deben ir SIEMPRE al inicio e al final de cada una de las páginas. Con ellas indicamos que estamos trabajando en lenguaje HTML.

Si queremos indicar el idioma del contenido del documento podemos incluir aquí el atributo lang. `<html lang="es">`

El *lenguaje español* se puede indicar con *es*, o con más detalle usando *es-ES* que es el símbolo internacional de español de España. Esta terminología sigue los códigos de dos letras definidos en la norma ISO 639-1 (que contiene dos letras para cada lenguaje) y el código de dos letras de país de la ISO 3166-1.

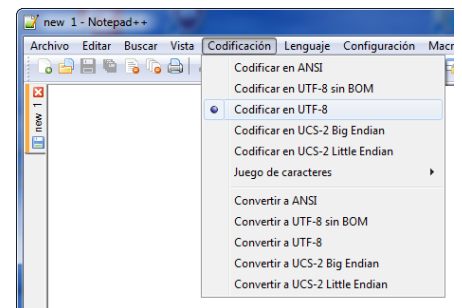
<head></head>

- En la cabecera se coloca el título de la página e información de carácter general (metadatos) que no se muestra en la ventana del navegador.

<meta charset="utf-8"/>

- indica la codificación que tendrá la página. La codificación **utf-8** permite usar símbolos como la "ñ", "ç" o acentos correctamente.

Si usamos la etiqueta meta para decir al navegador que la página va a utilizar una codificación UTF-8, *la página tiene que tener realmente esta codificación*. Esta codificación se tiene que indicar en el editor de html que estemos utilizando. Cada editor tiene sus características y sus propios menús. En la imagen vemos donde se ubica esta opción en **Notepad++**.



<title></title>

- El elemento *title* se encuentra anidado dentro del elemento *head*. En una página web formada por varias páginas independientes (a las que se accede mediante un menú, por ejemplo), cada una de estas páginas tendrá su propio elemento *title*. El texto marcado con esta etiqueta se visualiza en la pestaña activa del navegador.

Este título será el contenido más visible que devolverán los motores de búsqueda. Por lo que debe ser oportuno, conciso y claro. Por ejemplo, en la web de Formación Profesional de la Xunta de Galicia si tenemos seleccionada la pestaña Currículos el *title* sería:

<title>Currículos | Xunta de Galicia</title>



Si en un buscador escribimos “Currículos de Formación profesional da Xunta de Galicia” vemos que aparece destacado el texto que contiene esta etiqueta.

Currículos | Xunta de Galicia
www.edu.xunta.es/fp/curriculos ▼ Traducir esta página
O currículo das ensinanzas de formación profesional defínese como o conxunto de obxectivos, contidos, orientacións pedagóxicas, criterios de avaliación e ...

<body></body>

- La etiqueta *body* abarca todo el contenido de la página web que se visualiza en el navegador. (textos, imágenes, gráficos...).

1.2 El largo camino de HTML4 a HTML5

Tim Berners-Lee desarrolló el lenguaje HTML en 1989. Este se hizo muy popular ya que el concepto de etiquetas para crear páginas web era muy sencillo y fácil de aprender.

Para evitar que una compañía en particular obligue a los usuarios a utilizar una tecnología en concreto se crea el World Wide Web consortium (W3C), un organismo de normalización abierto constituido por representantes de diferentes compañías: Apple, Microsoft, Adobe, Sun, Google, IBM, Oracle, etc. Este consorcio ha creado muchas de las tecnologías que se utilizan hoy en día como HTML, XML, protocolos para servicios Web (SOAP), y gráficos en formato PNG entre otros. Cada uno de estos estándares son propuestos, definidos, ratificados y publicados con consentimiento de todo el grupo. Uno de los primeros estándares que siguió todo este proceso fue el lenguaje HTML.

La primera versión de HTML es muy diferente a la que usamos hoy en día, ya que no tenía en cuenta el diseño ni incluía herramientas para dar formato al texto. Entre los años 1989 y 1997, el lenguaje HTML pasó por cuatro ratificaciones importantes. El estándar de **HTML4** se publicó en 1997 y en él se proponía utilizar hojas de estilo en cascada nivel 1 (CSS1) para controlar el diseño de páginas, gráficos en formato PNG como estándar de mapa bits no sujetos a patentes, y normas DOM (Document Object Model, Modelo de objetos del documento) para que las aplicaciones JavaScript funcionaran coherentemente en todos los navegadores Web. También se introdujo la primera versión de **XML** para controlar la estructura de datos.

Los navegadores fueron adoptando este estándar, siendo Microsoft, con Internet Explorer 8 la que más tardó en respaldarlo (12 años). Hoy en día el estándar de referencia es **HTML5**. HTML 5: *permite la interacción del usuario y proporciona una experiencia muy similar a la que se obtendría con una aplicación de escritorio, pero sin necesidad de realizar instalaciones*

En la siguiente imagen podemos ver la evolución del lenguaje HTML.



En marzo de 2018 declararon obsoletas y reemplazadas (*Superseeded*) todas las recomendaciones del HTML y XHTML anteriores a HTML 5.1.

1.2.1 Eliminación de etiquetas y atributos no deseables

Los atributos de formato de muchas etiquetas han sido eliminados. Deben definirse en los estilos CSS. Así, se pretende separar el formato del contenido definitivamente. Prácticas que hasta ahora eran desaconsejadas, como el atributo background de <body> o los

atributos align y border, pasan a estar prohibidas. Algunos ejemplos más son los atributos cellpadding, cellspacing de las tablas, o las etiquetas <u>, , <center> o <strike>.

Se han eliminado etiquetas problemáticas como <frame>. Para mostrar una página dentro de otra se deberá utilizar <iframe>. También se eliminan otras etiquetas que habían caído en desuso <applet> (se utiliza <object>).

1.3 Organizar el texto

A continuación vamos a ver las etiquetas que vamos a utilizar para introducir texto en nuestra página. Algunas se usan desde la primera versión de HTML y otras se han introducido más recientemente en HTML5.

1.3.1 Párrafos y saltos de línea

El texto en una página web se teclea habitualmente en párrafos. En el lenguaje HTML la etiqueta <p> se usa para delimitar los párrafos. También se debe usar la etiqueta </p> al final del párrafo para que la sintaxis esté bien formada, aunque muchos navegadores mostrarán el párrafo correctamente aunque no la tenga.

```
<p>Esto es un párrafo< /p>
```

Por defecto cada párrafo se separa con una línea respecto a los otros párrafos.

Espacios en blanco

El texto dentro de las páginas web no respeta los espacios en blanco, ni tabuladores ni saltos de línea que coloquemos en el código, a la hora de mostrar el contenido por pantalla. Solo se considera el primer espacio en blanco, el resto se eliminan.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Probando funcionamiento de espacios y salto de línea</title>
  </head>
  <body>
    <p>¡Hola!
      Vamos a probar como funcionan          los espacios
      los tabuladores          y saltos de línea    en html.
      Este texto   tiene un poco de todo..
    </p>
  </body>
</html>
```

¡Todo el texto se muestra en la misma línea aunque haya espacios, tabuladores y saltos de línea en el código! :

¡Hola! Vamos a probar como funcionan los espacios los tabuladores y saltos de línea en html. Este texto tiene un poco de todo ..

La razón es permitir una mayor legibilidad del código, permitiendo en él que coloquemos el código con sangrías y espacios que mejoren su lectura, para que el mantenimiento del mismo sea más cómodo.

Etiqueta de salto de línea

La etiqueta
 se usa para insertar saltos de línea. Es una etiqueta vacía que normalmente va dentro de un párrafo. Por ejemplo:

```
<p>¡Hola! <br />
    Vamos a probar como funcionan          los espacios
    los tabuladores          y saltos de línea    en html.</p>
<p>Este texto    tiene un poco de todo..
</p>
```

Se visualizaría:

```
¡Hola!
Vamos a probar como funcionan los espacios los tabuladores y saltos de línea en html.

Este texto tiene un poco de todo..
```

NOTA: Teóricamente se pueden varias etiquetas `
` una detrás de la otra para hacer varios saltos de línea, pero esto se considera una mala práctica porque dificulta el mantenimiento del código. Para cambiar un texto con más precisión se usa CSS, que es un lenguaje que complementa a HTML para dar formato al texto.

1.3.2 Títulos

Existen seis niveles de títulos diferentes. Los elementos `h1`, `h2`, `h3`, `h4`, `h5` y `h6` pueden usarse para los distintos niveles de títulos y subtítulos. `h1` es el título principal y `h6` el de menor nivel.

El navegador muestra el título realmente importante (`h1`) en caracteres muy grandes y va disminuyendo el tamaño a medida que vamos bajando de nivel.

No debemos seleccionar la etiqueta del título en función del tamaño da al texto el navegador. Es fundamental estructurar la página correctamente y empezar con un título nivel 1 (`h1`), luego un título nivel 2 (`h2`), etc. ¡No tiene que haber ningún subtítulo sin un título principal! Para cambiar el tamaño del título usaremos CSS.

1.3.3 Elemento hr

El elemento HTML `hr` representa un cambio de tema entre párrafos (por ejemplo, un cambio de escena en una historia, un cambio de tema en una sección). En versiones previas de HTML representaba una línea horizontal. Aún puede ser representada como una línea horizontal en los navegadores visuales, pero ahora es definida en términos semánticos y no tanto en términos representativos, por tanto para dibujar una línea horizontal se debería usar el CSS apropiado.

En el siguiente ejemplo usamos `hr` para resaltar el contraste de sus temáticas:

```
<h1>Medio de comunicación</h1>
<p>La radio (entendida como radiofonía o radiodifusión, términos no estrictamente
    sinónimos) es un medio de comunicación que se basa en el envío de señales de
    audio a través de ondas de radio.</p>
<hr />
<p>La televisión es un sistema para la transmisión y recepción de imágenes en
    movimiento y sonido a distancia que emplea un mecanismo de difusión. </p>
```

1.3.4 Resaltar palabras

Algunas palabras en los párrafos son algunas veces más importantes que otras y queremos que aparezcan resaltadas. HTML proporciona varias formas para resaltar el texto.

- **Enfatizar:** Para enfatizar texto se usa la etiqueta `` ``. Por defecto, los navegadores muestran las palabras encerradas entre estas etiquetas en cursiva, aunque nosotros podremos cambiar la apariencia utilizando CSS.
- **Enfatizar considerablemente:** Si queremos resaltar palabras con un énfasis fuerte o importante se usa la etiqueta ``. Por defecto, el navegador las muestra en negrita.
- **Marcado de texto:** La etiqueta `<mark>` se usa para resaltar visualmente una parte del texto que no es necesariamente importante, pero queremos que destaque del resto del texto. Podemos cambiar en CSS la forma en la que se muestra.

Es importante comprender que la finalidad de estas etiquetas es indicar la relevancia del texto. Con `` se le dice al ordenador que “este texto es importante”, no que queremos que se visualice en negrita aunque la mayoría de los navegadores lo muestren así.

Muchos programas analizan el código fuente de las páginas web y aún más importante los robots de los motores de búsqueda. Estos robots rastrean la red y leen el código HTML de todos los sitios. Este es el caso de los robots de Google y Bing por ejemplo. Como ellos lo ven, las palabras clave «importantes» suelen tener más valor, así que si alguien hace una búsqueda con estas palabras es más probable que se encuentre con vuestro sitio web.

1.3.5 Listas

Las listas son un modo elegante de estructurar ciertos tipos de contenido en un documento HTML. Existen tres tipos de listas en HTML: listas sin orden, listas ordenadas y otras menos usadas que son las listas de definición.

1.3.5.1 Listas sin orden

Una lista sin orden es aquella que proporciona una estructura de lista sin numeración alguna que indique secuencia. En otros contextos recibe el nombre de lista de viñetas. El navegador mostrará cada elemento de la lista con una viñeta.

Crear una lista no ordenada es muy sencillo. Simplemente se usa la etiqueta `` para abrirla y se cierra cuando termina con ``. Cada uno de los elementos de la lista (List Item) se creará con una etiqueta ``. Se pueden crear tantos como sea necesario.

```
<ul>
  <li>elemento da lista</li>
  <li>outro elemento da lista</li>
  ...
</ul>
```

1.3.5.2 Listas ordenadas

En las listas ordenadas, los elementos de la lista tienen números o letras. El nombre de la etiqueta para crear una lista ordenada es ``, con `` de nuevo para cada elemento de la lista.

```
<ol>
  <li>elemento da lista</li>
  <li>outro elemento da lista</li>
  ...
</ol>
```

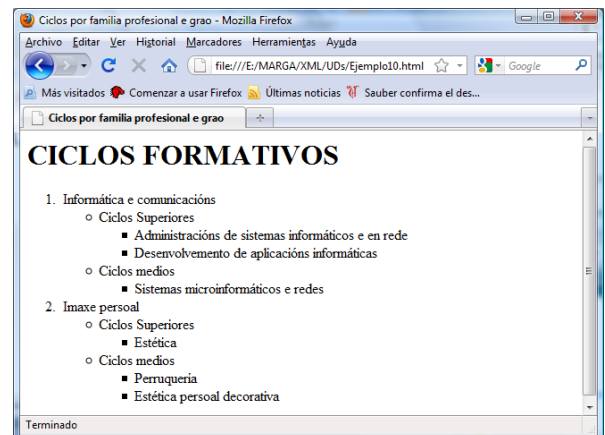
1.3.5.3 Listas anidadas

Las listas pueden estar anidadas y combinadas según sea contenido. Unicamente debemos tener algo importante en cuenta a este respecto y es que cualquier lista que esté anidada dentro de otra debe tener su propio elemento `li` como vemos en el ejemplo:

```
<ul>
  <li>elemento da lista principal</li>
  <li> comezo da lista ordenada
    <ol>
      <li>elemento da lista ordenada</li>
      </li>outro elemento da lista ordenada</li>
      ...
    </ol>
  </li>
  <li> outro elemento da lista principal</li>
  ...
</ul>
```

Vamos a ver esto con un ejemplo concreto:

```
<ol>
  <li>Informática e comunicacións
    <ul>
      <li>Ciclos Superiores
        <ul>
          <li>Administracións de sistemas informáticos e en rede</li>
          <li>Desenvolvemento de aplicacións informáticas</li>
        </ul>
      </li>
      <li>Ciclos medios
        <ul>
          <li>Sistemas microinformáticos e redes</li>
        </ul>
      </li>
    </ul>
  </li>
  <li> Imaxe persoal
    <ul>
      <li>Ciclos Superiores
        <ul>
          <li>Estética</li>
        </ul>
      </li>
      <li>Ciclos medios
        <ul>
          <li>Perruquería</li>
          <li>Estética persoal decorativa</li>
        </ul>
      </li>
    </ul>
  </li>
</ol>
</body>
</html>
```



1.3.5.4 Listas de definición

Las listas de definición son un tanto diferentes al resto de tipos de lista porque están estructuradas como un glosario de términos. El elemento externo de la lista de definición utiliza las etiquetas `<dl>`; y dentro de este elemento aparece uno o más pares de términos y sus definiciones. En cada par, el término está definido por un elemento `<dt>` y la definición aparece dentro de un elemento `<dd>`.

```
<dl>
  <dt>SGML</dt>
  <dd>Standard Generalized Markup Language</dd>
  <dt>HTML</dt>
  <dd>HyperText Markup Language</dd>
  <dt>XML</dt>
  <dd>eXtensible Markup Language</dd>
</dl>
```

Que se visualiza como se ve en la imagen:

```
SGML
  Standard Generalized Markup Language
HTML
  HyperText Markup Language
XML
  eXtensible Markup Language
```

1.3.6 Entidades

Cuando estemos insertando texto en nuestros documentos HTML puede darse el caso de que necesitemos insertar símbolos o caracteres no incluidos en el juego de caracteres que estamos utilizando. Para ello tenemos las entidades. La estructura de la entidad HTML es un ampersand(&) seguido del código o nombre de la entidad y terminado en un punto y coma. Por ejemplo, para insertar el símbolo del yen (¥) sería:

`¥` ó `¥`

Las entidades más utilizadas en HTML son para cinco caracteres que tienen un significado especial (<, >, &, ", '), ya que si queremos mostrar estos caracteres en una página web, escribirlos directamente puede hacer que la página no sea válida o que el navegador no muestre el resultado esperado:

Carácter	Entidad
<	<
>	>
&	&
"	"
'	'

1.3.7 Comentarios

Un comentario en HTML no modifica lo que se visualiza en la página. Se usa para dejar instrucciones sobre la forma en la que funciona la página. De este modo, ayuda a recordar cómo funciona la página cuando se vuelve al código fuente después de que haya pasado un tiempo sin verlo.

Los comentarios en HTML tienen el aspecto que se ve a continuación:

```
<!--Esto es un comentario -->
```



Haz las tareas 2, 3, 4 y 5 para comenzar a diseñar páginas sencillas.

1.4 Los metadatos

Los Metadatos son datos que describen datos, y HTML tiene una forma "oficial" de agregar metadatos a un documento: el elemento `<meta>`.

Ya hemos visto que podemos usar este elemento para indicar la codificación de la página, pero hay muchos otros tipos de elementos meta. Por ejemplo, podríamos usarlo para indicar el autor y la descripción de la página. Para ello debemos usar los atributos `name` y `content`.

- `name` especifica el tipo de información que contiene

- `content` especifica el contenido meta real

Así para indicar el autor y la descripción de la página pondríamos:

```
<meta name="author" content="Pablito" />
```

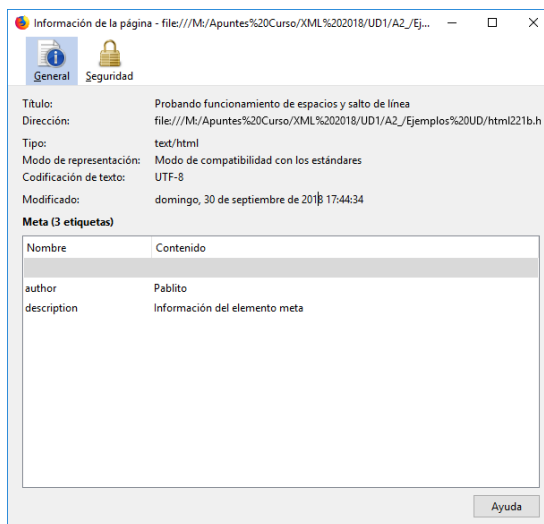
```
<meta name="description" content="Información del elemento meta" />
```

En Firefox, si pinchamos con el botón derecho del ratón y seleccionamos “Ver Información de la página” abre un cuadro de diálogo donde podemos ver los metadatos:

Especificar una descripción que incluya palabras clave relacionadas con el contenido de la página es útil ya que tiene el potencial de hacer que la página aparezca más arriba en las búsquedas relevantes realizadas en los motores de búsqueda.

La descripción también se usa en las páginas de resultados del motor de búsqueda ya que es la descripción que se muestra en los resultados del buscador.

Ya habíamos visto la importancia del elemento `title` y ahora podemos ver donde aparece esta descripción:



[Currículos | Xunta de Galicia](#)

www.edu.xunta.es/fp/curriculos ▼ Traducir esta página

O currículo das ensinanzas de formación profesional defínese como o conxunto de obxectivos, contidos, orientacións pedagóxicas, criterios de avaliación e ...



1.5 Enlaces

Los enlaces de hipertexto (o hiperenlaces) son el elemento característico de la web porque permiten que las páginas se relacionen entre sí. Tiene dos partes:

- el *enlace*, es decir, el elemento (texto o imagen) sobre el que se puede hacer clic con el ratón,
- el *destino*, es decir, el elemento (página web, imagen, archivo, etc) que se muestra al hacer clic en el enlace. El destino de un enlace puede ser un directorio, un archivo o un elemento (texto o imagen) situado en una página web:
 - Si el destino es un directorio, se puede configurar el servidor para que muestre el contenido del directorio, un archivo determinado (por ejemplo, el archivo `index.html` o `home.html`) o para que rechace la petición.
 - Si el destino es un archivo, el navegador mostrará el archivo si es capaz de mostrarlo (p.e. si es una página web o una imagen), o abrirá otro programa que sea capaz de mostrarlo (por ejemplo si es un archivo PDF), o preguntará al usuario qué debe hacer con el archivo (guardarlo en el disco o abrirlo con el programa que elija el usuario).
 - Si el destino es un elemento (texto o imagen) situado en una página web, el navegador abrirá la página web y se desplazará hasta el elemento enlazado (salvo que el elemento se encuentre al final de la página o cerca del final, en que entonces el navegador se desplaza todo lo que puede hacia abajo). También puede ser un elemento situado en la propia página web.

En el código fuente HTML los enlaces se identifican mediante la etiqueta `<a>` y su atributo `href`, que contiene el URI del destino al que conduce el enlace. Por ejemplo, para crear un enlace que abra el documento `Doc.htm`, que está en el directorio Documentos de la dirección `www.tal.net`, sería:

```
<a href = "http://www.tal.net/Documentos/Doc.htm">Hacer clic aquí</a>
```

Donde el texto que el usuario verá será “Hacer clic aquí”.

Si en el atributo `href` se escribe simplemente el carácter almohadilla (`#`), casi todos los navegadores se desplazan al principio de la página.

Vínculos dentro de un mismo documento.

En documentos extensos se pueden utilizar enlaces que conducen a distintas partes del documento para facilitar la navegación. Para construir un enlace dentro del mismo documento, es preciso definir dos etiquetas: el ancla y el enlace.

El *ancla* identifica la parte del documento a la que se va a ir al hacer clic encima del vínculo. En el vínculo se debe especificar el nombre del ancla en lugar del nombre de un documento. Esto se hace añadiendo el atributo `id` a la etiqueta correspondiente.

```
<p id = "destino">Párrafo al que vamos a saltar cuando pinchemos en el enlace </p>
```

En el *enlace* indicamos en el atributo `href` el nombre del marcador indicado, anteponiendo al nombre el símbolo de la almohadilla (`#`). Ejemplo:

```
<a href = "#destino">Ir al párrafo destino</a>
```

Por otro lado es posible indicar direcciones de enlace que salten a una página y dentro de ella se coloque en una posición marcada, por ejemplo:

```
<a href = "http://www.colegio.edu/sec1/actividades.html#verano">Actividades de verano</a>
```

Este enlace salta a la página `actividades.html` de la ruta indicada y además se coloca en la posición indicada por la etiqueta marcada con el valor `verano` (en su atributo `id`).

Enlaces no HTTP

Los navegadores son capaces de gestionar otros protocolos distintos a HTTP, por ejemplo el protocolo FTP:

```
<p>Enlace al <a href="ftp://glup.uv.es/pub/cdlibre/">servidor FTP de cdlibre</a>.</p>
```

Un enlace con el atributo href con el valor mail-to:dirección_de_correo_electrónico abre automáticamente el cliente de correo electrónico (Outlook, Thunderbird, Evolution, etc.) con un correo dirigido a la dirección indicada. El uso cada vez más extendido del correo web hace que esta opción sea un engorro más que una ayuda.

1.6 Imágenes

En informática se distinguen dos tipos de imágenes:

- Imágenes de mapa de bits, formadas por puntos de colores.
- Imágenes vectoriales, formadas por formas geométricas.

Cada formato tiene sus ventajas e inconvenientes:

- Las imágenes vectoriales tienen la ventaja de que ocupan poco espacio y se pueden ampliar a cualquier escala sin pérdida de calidad. Su inconveniente es que no permiten representar imágenes sin formas definidas, como ocurre en las fotografías.

El formato de imágenes vectoriales más utilizado es SVG.

- Las imágenes de mapa de bits permiten representar imágenes sin formas definidas, pero a cambio pierden calidad al ampliarlas y ocupan mucho espacio en disco.

Para reducir el tamaño de las imágenes de mapa de bits se utilizan diferentes algoritmos de compresión y por ello existen numerosos formatos de mapa de bits. Cada formato está optimizado para un determinado tipo de imagen (número de colores y degradados).

Existen numerosos formatos de mapa de bits, enfocados a diferentes tipos de imagen:

- para imágenes fotográficas (muchos colores), el formato más extendido es JPEG.
- para dibujos (pocos colores, formas bien definidas) los formatos más extendidos son GIF y PNG.
- para dibujos con animación, el formato más extendido ha sido siempre el GIF animado, aunque también se usa bastante el formato APNG.
- para dibujos sin compresión, el formato más extendido es BMP de Windows, pero apenas se utiliza, ni se recomienda hacerlo, ya que los archivos son muy grandes.

img

Es la etiqueta que permite mostrar imágenes vectoriales o de mapa de bits en una página web. Es una etiqueta **huérfana** (como
).

Tiene los siguientes atributos:

- **src** → contiene el camino absoluto o relativo a la imagen desde la página web. Así que si la imagen está en la subcarpetaimagenes, tienes que introducir:src="imagenes/flor.png"

Por ejemplo, para cargar la imagen *logo.png*, que se encuentra en la carpeta *imagenes* situada dentro de la carpeta en la que está el documento html, sería:

```
<img src ="imagenes/logo.png" />
```

■ **alt** → que significa "texto alternativo", esto es, un texto corto que describe lo que contiene la imagen y que se mostrará en lugar de la imagen si esta última no se puede descargar, o en los navegadores de personas con discapacidad visual que no pueden "ver" la imagen. También ayuda a los robots de motores de búsqueda a buscar imágenes. Para nuestra imagen, por ejemplo, deberíamos escribir:

```

```

■ **title** → que contiene el texto que se muestra en forma de "tip" (rectángulo amarillento que aparece cuando se sitúa el ratón encima de la imagen). Es un atributo optativo.

■ **width y height** → establecen la anchura y altura de la imagen. Si estos atributos no están definidos, la imagen se muestra con su tamaño original, pero si los atributos están definidos, la imagen se adapta a esos valores. Los valores numéricos se interpretan como píxeles.

No es aconsejable usar estos atributos para cambiar el tamaño de la imagen, ya que si la ampliamos no se verá en buena calidad y si la reducimos estaremos cargando una imagen grande para luego mostrarla en pequeño; sería más inteligente reducirla primero con un editor de imágenes. En cualquier caso es importante utilizar este atributo para que el navegador sepa de antemano el tamaño de la imagen y así que maquete la página correctamente.

Si los valores no son proporcionales a la imagen original, la imagen de mapa de bits se deforma mientras que la imagen vectorial se ajusta al menor tamaño sin deformarse.

En HTML5 no está permitido utilizar porcentajes en estos atributos.

Mostrar la imagen en miniatura

Si la imagen es muy grande, es recomendable mostrar su vista en miniatura en tu sitio web. Después se añade un enlace a la vista en miniatura para que los visitantes puedan ver la imagen en su tamaño original.¹

Tendríamos las dos imágenes en la carpeta. Mostramos la versión pequeña y creamos un enlace a la grande para que se muestre la versión ampliada al hacer clic sobre la vista en miniatura.

```
<a href="imagenes/flor.jpg"></a>
```



Resuelve los puntos 6, 7, 8, 9 y 10 de la lista de tareas.

¹ Existen muchos programas para crear miniaturas de imágenes, como por ejemplo Easy Thumbnails.