

Tablas, enlaces y multimedia



En este documento vamos a abarcar algunas etiquetas adicionales que nos van a permitir añadir más tipos de elementos en nuestras páginas. Veremos cómo estructurar la información en tablas, cómo definir enlaces para ir de un documento a otro, y cómo añadir diferentes tipos de contenido multimedia, como imágenes y recursos de audio y vídeo.

1. Tablas

Las tablas son un elemento muy útil en los procesadores de texto porque permiten estructurar la información en la página de diferentes formas, simulando columnas, o permitiendo textos a diferentes niveles o en diferentes orientaciones. Hasta hace no mucho, también se utilizaban con mucha frecuencia en las páginas web para estructurar de esta misma forma el contenido, pero la W3C las desaconseja o las considera obsoletas, porque hacen que el diseño de la página sea muy rígido (una tabla no va a cambiar su disposición para poderse ver mejor en un móvil, por ejemplo), y dificultan la lectura por los lectores de pantalla.

En cualquier caso, veremos brevemente cómo se crean tablas en HTML, tanto por si nos es necesario crearlas en algún momento, como para saberlas identificar en cualquier página web que estemos manipulando.

El contenido de una tabla va entre la pareja de etiquetas `table`. Dentro, distinguimos tres tipos de elementos:

- Cada fila de la tabla, se marca con un par de etiquetas `tr`.
- Cada celda dentro de esa fila se marca con un par de etiquetas `th` si es el encabezado de la tabla (normalmente primera fila y/o primera columna) y queremos darle un estilo diferente, o con etiquetas `td` si son celdas normales. El contenido de esa celda será lo que pongamos entre ese par de etiquetas, y puede ser cualquier cosa (texto llano, párrafos, listas, imágenes, otra tabla...).

Veamos un ejemplo:

```
<table>
  <tr>
    <th>Columna 1</th>
    <th>Columna 2</th>
  </tr>
  <tr>
    <td>Fila 1, Columna 1</td>
    <td>Fila 1, Columna 2</td>
  </tr>
  <tr>
    <td>Fila 2, Columna 1</td>
    <td>Fila 2, Columna 2</td>
  </tr>
</table>
```

Esto se vería más o menos así en el navegador:

Columna 1	Columna 2
Fila 1, Columna 1	Fila 1, Columna 2
Fila 2, Columna 1	Fila 2, Columna 2

1.1. Atributos de las tablas

Las etiquetas que usamos para construir tablas (`table`, `tr`, `td` ...) admiten una serie de atributos para añadir algunas propiedades. Muchos de estos atributos también pueden especificarse a través de estilos CSS.

Así, la etiqueta `table` admite los siguientes atributos:

- `border` : grosor del borde. Si no queremos tener borde, pondríamos `border="0"` .
- `width` : anchura de la tabla (en píxeles o en porcentaje sobre la anchura máxima)
- `cellspacing` : margen entre el interior del borde de cada casilla y el contenido
- `summary` : resumen del contenido de la tabla, útil para lectores de pantalla.

Por su parte, los elementos `td` admiten estos atributos:

- `align` : alineación horizontal del texto. Puede valer *left*, *right* o *center*.
- `valign` : alineación vertical del texto. Puede valer *top*, *bottom* o *middle*.
- `rowspan` : expande la celda X filas más hacia abajo
- `colspan` : expande la celda X celdas más hacia la derecha
- `width` : para indicar la anchura específica de esa columna (en píxeles o porcentaje).

Veamos todo esto en un ejemplo:

```
<table border="1">
  <tr>
    <td rowspan="2" valign="middle" width="50%">Uno</td>
    <td>Dos</td>
    <td>Tres</td>
    <td>Cuatro</td>
  </tr>
  <tr>
    <td>Cinco</td>
    <td colspan="2" align="right">Seis</td>
  </tr>
</table>
```

Esto se vería más o menos así en el navegador:

Uno	Dos	Tres	Cuatro
	Cinco	Seis	

Ejercicio 1:

Crema una página llamada **tablas.html**. En ella deberás crear dos tablas:

- Una sin bordes, con un encabezado de 4 columnas (*th*): Nombre y apellidos, E-mail, Dirección, Teléfono, y después 4 filas con esos datos rellenos para 4 personas que conozcas o te inventes.
- Otra tabla con borde de grosor 1 y anchura del 50%, donde la primera fila tenga 3 columnas combinadas y el texto en negrita "Trilogías". Después, habrá información sobre trilogías de cine. Para cada una, añadiremos una fila con las 3 columnas combinadas y el título de la trilogía (El Señor de los Anillos, El Hobbit, Matrix, etc), y después otra fila con los títulos de las 3 películas de la trilogía

Aquí tienes un ejemplo de cómo puede quedar:

Nombre	E-mail	Dirección	Teléfono
Juan García	jgarcia@gmail.com	C/La Puerta, 15, 1ºA	611223344
Adrián Llopis	all1979@hotmail.com	Av. Holanda 41, 6ºD	612345678
Aitana Martínez	a.mart@yahoo.com	C/Blasco Ibáñez, 4, 4ºK	699887766
Felipe Sánchez	fel.san.jim@gmail.com	C/Juan Ramón Jiménez, 51, Bajo	615263748

Trilogías		
El Señor de los Anillos		
La comunidad del Anillo	Las dos torres	El retorno del Rey
El Hobbit		
Un viaje inesperado	La desolación de Smaug	La batalla de los cinco ejércitos
Matrix		
Matrix	Matrix Reloaded	Matrix Revolutions

1.2. Otras etiquetas para tablas

Las etiquetas que hemos visto hasta ahora (`table`, `tr`, `td` y `th`) son las más habituales cuando queremos definir tablas en nuestras páginas HTML. Pero hay otras etiquetas que también podemos emplear, aunque no son tan frecuentes:

- `caption` permite asignar una breve descripción a la tabla, útil para lectores de pantalla en el caso de usuarios con problemas de visión, para así poder hacerse una idea rápida de lo que ofrece la tabla. Se coloca justo después de la etiqueta `table`

```
<table>
  <caption>Lenguajes de programación más populares</caption>
  <tr>
    ...
```

- `thead`, `tbody` y `tfoot` definen un encabezado, cuerpo y pie de la tabla, de forma similar al contenido de la página en general. Dentro de cada una de estas secciones podemos englobar una o varias filas.

```
<table>
  <caption>Lenguajes de programación más populares</caption>
  <thead>
    <tr>
      <th colspan="2">
        Lenguajes más populares
      </th>
    </tr>
    <tr>
      <th>Lenguaje</th>
      <th>Tipo</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Java</td>
      <td>Orientado a objetos</td>
    </tr>
    <tr>
      <td>Python</td>
      <td>Híbrido</td>
    </tr>
    <tr>
      <td>JavaScript</td>
      <td>Híbrido</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <td colspan="2">Actualizado a Agosto de 2021</td>
    </tr>
  </tfoot>
</table>
```

Lenguajes de programación más
populares

Lenguajes más populares	
Lenguaje	Tipo
Java	Orientado a objetos
Python	Híbrido
JavaScript	Híbrido
Actualizado a Agosto de 2021	

Ejercicio 2:

Crema una página llamada **horario.html** con tu horario de clases de este curso. La primera fila contendrá los encabezados (*th*) con los días de la semana de Lunes a Viernes, y la primera columna otros encabezados con las franjas horarias. Deberás tener en cuenta que los recreos serán una sola casilla que ocupará todos los días de la semana (*colspan*) y que algunos módulos o asignaturas pueden extenderse más de una franja horaria (*rowspan*).

2. Enlaces

Los enlaces (también llamados hipervínculos, hiperenlaces o *links*) son quizá el elemento más importante de las páginas web, ya que permiten "saltar" de una página a otra, haciendo clic en ellos. Es lo que se conoce como *hipertexto* (texto que permite acceder a otro texto), y es la razón de ser primitiva del lenguaje HTML (*lenguaje de marcado de hipertexto*), y del protocolo HTTP con el que nos comunicamos a través de los navegadores con los servidores web (*protocolo de transferencia de hipertexto*). Existen en casi cualquier página web; por ejemplo, si hacemos una búsqueda en Google, al hacer clic en alguno de los resultados que nos ofrece, estamos haciendo clic en un enlace.

A la hora de colocar un enlace en una página, tenemos que distinguir dos elementos:

- El **texto** (o imagen) sobre el que pinchamos en la página HTML, y que puede ser el que nosotros queramos.
- La **URL** (dirección web) a la que queremos ir al pinchar en ese enlace. Esta URL puede llevarnos a otra página HTML, una imagen, un documento PDF o cualquier otro tipo de recurso digital.

Para añadir enlaces en un documento HTML se utiliza la pareja de etiquetas `<a>`. Tienen un único atributo obligatorio, llamado `href`, que será la URL de la página a la que queremos ir. El texto (o la imagen) que queramos poner para hacer clic sobre él, deberá ir entre el par de etiquetas `<a>`. Veamos un ejemplo:

```
<p>Ir a la <a href="http://www.ua.es">web de la Universidad de Alicante</a>.</p>
```

que se vería así:

Ir a la [web de la Universidad de Alicante](http://www.ua.es).

2.1. Otros atributos útiles

La etiqueta `<a>` admite, además de `href`, otros atributos que pueden sernos útiles. Vamos a destacar dos de ellos:

- El atributo `title`, que sirve para que, cuando pasemos el ratón por encima del enlace (sin hacer clic en él), nos aparezca un recuadro explicativo sobre dónde vamos a ir con ese enlace. El texto del recuadro será el que pongamos en este atributo `title`.
- El atributo `target`, que sirve para indicar dónde queremos abrir el enlace. Si no ponemos este atributo, el enlace se abrirá en la misma ventana de la página que tenemos actualmente, con lo que dejaremos de ver la página actual y pasamos a la nueva (en cualquier caso, podemos recuperar la página

antigua pulsando el botón de *Atrás* del navegador). Pero podemos darle otros valores, como por ejemplo `_blank`, para que se abra en una ventana o pestaña nueva.

El siguiente ejemplo modifica el enlace anterior, añadiéndole un `title` y haciendo que se abra en una ventana nueva:

```
<p>Ir a la
  <a href="http://www.ua.es" title="Ir a la web de la UA" target="_blank">
    web de la Universidad de Alicante
  </a>.
</p>
```

Ejercicio 3:

Crea un documento HTML llamado **institutos.html**. Dentro, genera una lista (no ordenada) con el nombre de al menos 4 o 5 institutos de Ciclos Formativos o Secundaria que conozcas, y pon un enlace en cada uno que vaya a su correspondiente página web, haciendo que se abran en una ventana nueva (distinta a la actual con el listado).

2.2. Rutas relativas y absolutas

La URL del recurso o página al que queremos ir al pulsar en un enlace contiene la **ruta** hacia ese enlace. Cuando queremos ir a una URL de una web diferente a la nuestra, ponemos la dirección completa (como el caso del ejemplo anterior). Pero si queremos saltar a otra página de nuestra misma web, podemos indicar en el atributo `href` una ruta relativa o absoluta a la página donde estamos. Por ejemplo, si tenemos una aplicación web de varias páginas, repartidas en carpetas y subcarpetas, como en este ejemplo (indicamos las carpetas con los símbolos `<..>`):

```
<MiWeb>
  pagina1.html
  pagina2.html
  <otros>
    pagina3.html
```

Supongamos que estamos editando la *pagina1.html*. Si quisiéramos poner un enlace para ir a la *pagina2.html*, podríamos hacerlo de dos formas:

- Como una ruta **relativa** desde la carpeta donde está la *pagina1* que estamos editando. En este caso, como ambas páginas están en la misma carpeta, el enlace sería algo así:

```
<a href="pagina2.html">Ir a página 2</a>
```

- Como una ruta **absoluta** a la raíz de la aplicación. Como la *pagina2* está en la carpeta raíz de la aplicación *MiWeb*, el enlace quedaría así:

```
<a href="/pagina2.html">Ir a página 2</a>
```

En cambio, si desde *pagina1* quisiéramos poner un enlace a *pagina3.html*, los enlaces relativo y absoluto quedarían, respectivamente:

```
<!-- Relativo -->  
<a href="otros/pagina3.html">Ir a página 3</a>  
<!-- Absoluto -->  
<a href="/otros/pagina3.html">Ir a página 3</a>
```

En el caso contrario (un enlace en la *pagina3* para ir a *pagina1*), los enlaces relativo y absoluto, respectivamente, serían así:

```
<!-- Relativo -->  
<a href="../pagina1.html">Ir a página 1</a>  
<!-- Absoluto -->  
<a href="/pagina1.html">Ir a página 1</a>
```

Hacemos uso del símbolo `..` para salir a la carpeta anterior a la que estamos. Podemos enlazar esta expresión varias veces para salir varias carpetas atrás, como por ejemplo `../../..` para salir tres carpetas.

En general, ambos tipos de rutas son equivalentes, aunque las rutas absolutas pueden darnos problemas en servidores de pruebas, ya que la raíz de la aplicación web no suele coincidir con la raíz del servidor y al poner la barra `/` en la ruta absoluta los servidores interpretan la raíz del servidor.

Ejercicio 4:

Descarga [esta plantilla de ejemplo](#). Dentro verás distintos archivos distribuidos en diferentes carpetas. Se pide que completes el atributo `href` de los enlaces que encontrarás en el archivo `index.html` de la carpeta principal y del archivo `index.html` de la subcarpeta *projects*, para que enlacen con el documento correcto.

2.3. Puntos de anclaje y enlaces internos

Además de poder enlazar con otras páginas y otras webs, los enlaces también permiten saltar a otras zonas dentro de una misma página. Para ello, debemos seguir dos pasos:

1. Ir a la zona a la que queremos saltar, y definir un punto de anclaje (en inglés, anchor). Para ello, asignamos un atributo `id` con el nombre que queramos (siempre que no se repita) a la zona donde

queremos ir. Por ejemplo, a un encabezado de nivel 1 o 2:

```
<body>
...
<h1 id="introduccion">Introducción</h1>
<p>blablablabla...</p>
```

2. Luego, vamos donde queramos poner el enlace, y utilizamos la misma etiqueta `a` de los enlaces. En el atributo `href` pondremos, en lugar de un nombre de archivo o una URL, el nombre del anclaje que hemos usado antes, precedido de una almohadilla "#". Para el ejemplo anterior, podríamos ir al principio de la página y poner un enlace como este:

```
<body>
<a href="#introduccion">Introducción</a>
...
<h1 id="introduccion">Introducción</h1>
...
```

Ejercicio 5:

Crea una página llamada **harrypotter.html** con la siguiente estructura:

- Título (*title*): Harry Potter
- Encabezado de nivel 1 que diga *Saga Harry Potter*
- Introducción (encabezado nivel 2)
- Un pequeño párrafo o dos que digan de qué va la saga
- Episodios (encabezado nivel 2)
- Harry Potter y la piedra filosofal (nivel 3)
- Harry Potter y la cámara secreta (nivel 3)
- Harry Potter y el prisionero de Azkaban (nivel 3)
- Harry Potter y el cáliz de fuego (nivel 3)
- Harry Potter y la orden del Fénix (nivel 3)
- Harry Potter y el misterio del Príncipe (nivel 3)
- Harry Potter y las reliquias de la muerte (nivel 3)
- Dentro de cada apartado de nivel 3, añade 1 o 2 párrafos que cuenten de qué va el episodio
- Resumen de personajes (encabezado nivel 2). Dentro tendrá una lista no numerada con los nombres de 5 de los actores y, entre paréntesis, a qué personaje interpretan
- Pon anclajes en cada encabezado (de nivel 2 y 3) y al principio de todo un índice de enlaces que permitan, pinchándolos, ir a cada apartado de la página. Al final, deberá quedarte algo como esto:

1. [Introducción](#)
2. [Episodios de la saga](#)
 1. [Harry Potter y la piedra filosofal](#)
 2. [Harry Potter y la cámara secreta](#)
 3. [Harry Potter y el prisionero de Azkaban](#)
 4. [Harry Potter y el cáliz de fuego](#)
 5. [Harry Potter y la orden del Fénix](#)
 6. [Harry Potter y el misterio del príncipe](#)
 7. [Harry Potter y las reliquias de la muerte](#)
3. [Resumen de personajes](#)

Introducción

blablabla...

Episodios de la saga

Harry Potter y la piedra filosofal

blablabla...

3. Imágenes

En un procesador de textos al estilo de Word u OpenOffice, para colocar una imagen que nos guste, basta con copiarla y pegarla en el documento, si no la tenemos disponible en alguna carpeta de nuestro ordenador. Pero para añadir imágenes en documentos web, debemos tenerlas localizadas en la misma carpeta (o una subcarpeta) de nuestra web.

Si son pocas imágenes, podemos tenerlo todo junto en una misma carpeta, pero si la aplicación web es grande, o hay varias imágenes, lo más normal es crear una carpeta llamada *imagenes* o *img*, y dejar dentro las imágenes que necesitemos.

Una vez tengamos ya guardada la imagen que queremos poner, usaremos la etiqueta `img` para colocarla en la página HTML. Esta etiqueta tiene un único atributo obligatorio, llamado `src`, que funciona igual que el atributo `href` visto para los enlaces: debemos indicar la ruta (absoluta o relativa) hacia la imagen que queremos añadir. También podemos poner una URL hacia una imagen de otra web, pero corremos el riesgo de que la imagen deje de verse si la página cierra o está caída, o que tarde en cargar mientras conectamos con esa otra web.

Por ejemplo, si queremos añadir la imagen *foto.png* que está dentro de la subcarpeta *imagenes*, podríamos añadirla con algo como:

```

```

Observa que la etiqueta *img* es una etiqueta vacía, que no tiene pareja. Ella sola se abre y cierra. Dentro, además del atributo `src`, podemos tener otros **atributos** opcionales:

- `width`: especifica la anchura que queremos que tenga la imagen. Se la podemos dar en píxeles o en porcentaje. Conviene que el tamaño que le demos no difiera mucho del original, para que no se distorsione, y para que no ocupe demasiado espacio si luego la queremos ver mucho más pequeña. También podemos establecer esta y otras propiedades por CSS.
- `height`: especifica la altura que queremos que tenga la imagen (normalmente en píxeles). Si no ponemos nada, se redimensiona según la anchura que hayamos puesto.
- `alt`: es un texto alternativo que mostrar en el caso de que el navegador no pueda cargar la imagen, o ésta tarde en cargar. Para según qué versiones de HTML o XHTML, este atributo también es obligatorio (de lo contrario, la página no será correcta).
- `title`: hace referencia al título de la imagen, y funciona de forma similar al *title* de los enlaces: es el texto que se muestra cuando pasamos el ratón por encima de la imagen. En este caso no es recomendable utilizarla, porque muchos lectores de pantalla se comportan de forma imprevisible con ella, al tener un propósito parecido al del atributo *alt*.

3.1. Imágenes y enlaces

Como hemos comentado anteriormente, dentro del contenido de un enlace, además de texto, podemos colocar imágenes, de modo que al hacer clic sobre la imagen vayamos a la ruta indicada por el enlace. Aquí tenemos un ejemplo de ello:

```
<a href="https://www.ua.es">  
    
</a>
```

3.2. Icono de la aplicación web

De un tiempo a esta parte se ha puesto de moda que las aplicaciones (tanto en móviles como en ordenadores e incluso las aplicaciones web) tengan su propio icono que las identifique. De hecho, en muchas webs, cuando cargamos la página, aparece su icono en la barra de direcciones, o en la pestaña del navegador donde hemos abierto la web (dependiendo del navegador). ¿Cómo conseguimos esto?

En primer lugar, como cualquier otra imagen, tendremos que tener el icono localizado en alguna carpeta o subcarpeta de nuestra web. Normalmente los iconos tienen extensión *.ico* o *.png*, y en el caso del icono de la web, se le suele llamar *favicon* (por lo que el archivo entero suele llamarse `favicon.ico` o `favicon.png`). El icono suele tener unas dimensiones de 16x16 o 32x32 píxeles. Una vez tengamos el icono localizado, deberemos añadir la siguiente línea en el *head* de la(s) página(s) web donde queramos añadirlo:

```
<link rel="shortcut icon" href="favicon.ico" type="image/x-icon">
```

Existen algunas variantes. Por ejemplo, si el icono cumple los estándares de tamaño, podemos poner `rel="icon"` y `type="image/png"` (en el caso de que sea un archivo PNG). También para algunos navegadores es suficiente con dejar un archivo llamado *favicon.ico* en la raíz de nuestra aplicación web, y lo detectan automáticamente.

Ejercicio 6:

Modifica el ejercicio de Harry Potter que hemos hecho al hablar de enlaces, y haz los siguientes cambios:

- Haz que aparezca la portada de cada película justo tras su título. Busca las portadas en Internet y cópialas en una subcarpeta *imgs* dentro del ejercicio.
- Opcionalmente, añade también una foto de cada personaje de la lista del resumen de personajes, a tamaño 100x100, junto al nombre del personaje.

3.3. Imágenes con pie de foto. *figure* y *figcaption*

En ocasiones nos puede interesar añadir un pie de foto a la imagen que incorporamos a una página. Para tratarlo como un todo (imagen más pie de foto) podemos emplear la etiqueta `figure` para que englobe todo el conjunto (imagen y pie) y la etiqueta `figcaption` para definir dicho pie de foto. Podría quedar algo así:

```
<figure>
  
  <figcaption>Imagen reciente de la Sagrada Familia</figcaption>
</figure>
```

Ejercicio 7:

Crema una página llamada **biografia.html**. Vas a crear en ella la biografía de un personaje famoso (el que quieras), de forma que te quede algo como esto:

Groucho Marx



Groucho Marx (n. Nueva York, 2 de octubre de 1890 - m. Los Ángeles, 19 de agosto de 1977) fue un actor, comediante y escritor estadounidense, conocido principalmente por ser uno de los miembros de la familia cómica Hermanos Marx.

Julius Henry Marx nació en Nueva York, creció en en el seno de una modesta familia de inmigrantes alemanes judíos. Su madre era Miene (Minnie) Schoenberg (1864-1929) que emigró a Estados Unidos desde Alemania junto a sus padres y sus hermanos y su padre era Simon (Samuel) Marx (nacido como Marris) (1860?-1933), inmigrante alemán, nacido en Alsacia cuando formaba parte de Francia y anexada posteriormente al Imperio Alemán tras la Guerra Franco-Prusiana en 1871, con lo que Samuel pasó de ser ciudadano francés a ciudadano alemán. Orgulloso de su pasado francés fue conocido toda su vida como Frenchy.

Marx falleció en Los Ángeles a causa de una neumonía, dejando atrás tres ex-esposas, Ruth, Kay y Eden, tres hijos, Arthur, Miriam y Melinda, 18 películas, 14 de ellas junto a sus hermanos, y millones de admiradores. Una de sus célebres frases fue: "No deseo pertenecer a ningún club que acepte como socio a alguien como yo". Groucho Marx fue incinerado, sus cenizas se guardan en el Eden Memorial Park, siendo falso el epitafio "Perdonen que no me levante" que popularmente se creía que estaba en su tumba.

Como ves, deberás crear una tabla, de borde 1, con una columna y dos filas. En la primera fila, centrada, deberá aparecer la imagen del personaje que hayas elegido, a tamaño 300 x 400. Opcionalmente, puedes añadir un pie de foto asociado a la imagen. En la segunda fila, deberán aparecer al menos un par de párrafos que cuenten algo de su vida.

Además, la página debe tener como título (title) el texto "Biografía de XXXXX", siendo XXXXX el nombre del personaje que hayas elegido. Antes de la tabla, debe haber un encabezado de primer nivel (h1) con el nombre de ese personaje.

3.4. Estructurando nuestra web

Como hemos comentado, a medida que una web se hace algo más compleja, no es viable mantener todos los archivos (páginas HTML, CSS, imágenes, etc) en una misma carpeta, y conviene definir una estructura de carpetas más o menos "estándar" y fácil de gestionar. Por ejemplo, podríamos utilizar esta estructura:

- Los archivos CSS que definamos (uno o varios) pueden ubicarse en una carpeta `CSS` de nuestra aplicación

- Las imágenes podemos ubicarlas en una carpeta `img` o `imgs`. En el caso de que sean muchas, también podemos estructurarlas en subcarpetas dentro de esa carpeta inicial.
- Las páginas pueden colocarse en la carpeta raíz del proyecto o, si son muchas, dejar en la carpeta raíz la página inicial o punto de entrada a la web (que típicamente suele llamarse `index.html`) y luego colocar en subcarpetas el resto de páginas, agrupadas por categorías según la web en cuestión. Por ejemplo, todos los formularios en una carpeta, o todas las páginas que tengan que ver con clientes, etc.

Ejercicio 8:

Aquí tienes la estructura de archivos e imágenes de una web de prueba. Como podrás ver, está todo en una misma carpeta, y hay algunas cosas por hacer. En primer lugar, se pide que hagas los siguientes cambios:

1. Mueve todas las imágenes a una subcarpeta llamada **img**
2. Mueve los archivos `java.html` y `python.html` a una subcarpeta llamada **lenguajes**
3. Te habrá quedado únicamente el archivo `index.html` en la carpeta raíz después de los pasos anteriores.

Ahora, completa estos pasos:

1. Añade dos enlaces en `index.html` en los lugares indicados para ir a las páginas `java.html` y `python.html`
2. Añade un enlace en cada una de las otras dos páginas (`java.html` y `python.html`) para volver a `index.html`, en el lugar habilitado para ese enlace.
3. Corrige los atributos `src` de las imágenes que se usan en las tres páginas para cargar las imágenes correspondientes de la carpeta `img`.

Prueba a abrir ahora `index.html` y comprueba que desde ella puedes navegar a las otras dos páginas y volver atrás, y que las imágenes se muestran correctamente.

4. Audio y vídeo

Desde HTML 5, se han añadido un par de etiquetas específicas para añadir contenido de audio y vídeo a nuestras páginas, y poderlos reproducir directamente sobre ellas. En algunos casos, será necesario acudir a una etiqueta auxiliar llamada `iframe` para poder visualizar el archivo de ciertas fuentes.

4.1. La etiqueta `iframe`

La etiqueta `iframe` permite incrustar contenido externo en una web. Es la abreviatura de *inline frame*, o marco incorporado. Básicamente define una especie de ventana o marco interno con el que poder "asomarse" o visualizar contenido de otra web, como si estuviera en la nuestra.

Entre sus aplicaciones podemos destacar la posibilidad de incrustar vídeos de YouTube o mapas de Google Maps en nuestras webs, como veremos a continuación. Para utilizarlo correctamente, debemos tener en cuenta los atributos que maneja:

- `src` : URL del recurso incrustado
- `width` y `height` : anchura y altura del *iframe*
- `name` : nombre que le damos al *iframe* (opcional)
- `allowfullscreen` : si está presente, permitimos que el recurso mostrado se pueda ampliar a pantalla completa
- `frameborder` : borde del *iframe* (0 para ninguno)
- `allow` : conjunto de *plugins* o funcionalidades que se permiten sobre el *iframe*, tales como *autoplay* para reproducir automáticamente un vídeo, *gyroscope* para girar el vídeo, etc.

Por ejemplo, así incrustaríamos un vídeo de YouTube en nuestra web para poderlo ver directamente (y no tener que ir a YouTube para verlo):

```
<iframe width="560" height="315" src="https://www.youtube.com/embed/QH2-TGU1wu4" frameborder="1" allow="autoplay; gyroscope" allowfullscreen></iframe>
```

4.2. Añadir vídeo

La etiqueta `video` permite añadir vídeos a nuestras páginas. En general, se utiliza del mismo modo que la etiqueta *img* para imágenes, indicando en el atributo `src` la URL del vídeo a reproducir. Dicho vídeo, igual que ocurre con las imágenes, podrá estar alojado en alguna subcarpeta propia de la web, o en una web externa.

```
<video src="videos/prueba.mp4" controls></video>
```

El atributo `controls`, si está presente, muestra los controles para reproducir o pausar el vídeo. También podemos especificar una anchura (`width`) y/o altura (`height`), típicamente en píxeles. A diferencia de las imágenes, el vídeo no se distorsiona si el tamaño de la zona de reproducción no coincide con las proporciones del vídeo. El espacio que quede sin llenar se dejará con un fondo negro.

```
<video src="videos/prueba.mp4" width="720" height="400" controls></video>
```

En ocasiones, puede que el navegador no disponga de los *codecs* necesarios para poder reproducir el vídeo. En este caso, podemos ofrecer diferentes alternativas con diferentes formatos, y que el navegador utilice la primera que pueda reproducir. Esto se consigue añadiendo distintos bloques `source` dentro del bloque `video` y que cada uno apunte a su propia URL de vídeo:

```
<video width="720" height="400" controls>
  <source src="videos/prueba.mp4" type="video/mp4">
  <source src="videos/prueba.webm" type="video/webm">
</video>
```

Recuerda que para ver vídeos existentes en ciertas webs, como por ejemplo YouTube, se deben emplear *iframes* en la página (ver ejemplo en subapartado anterior).

4.3. Añadir audio

La etiqueta `audio` nos permite añadir archivos de audio a una página. Se emplea del mismo modo y con los mismos atributos y subetiquetas que *video* (`src`, `controls`, `source`, etc), salvo algunos elementos, como *width* o *height*, que aquí no tienen sentido al carecer un audio de dimensiones visuales.

```
<audio controls>
  <source src="canciones/cancion1.mp3" type="audio/mp3">
  <source src="canciones/cancion1.wav" type="audio/wav">
</audio>
```

4.4. Añadir otros elementos

Desde versiones anteriores de HTML 5, podemos emplear una etiqueta llamada `object` para hacer referencia distintos tipos de contenido multimedia. Por ejemplo, podemos emplearlo para cargar imágenes, como alternativa a `img` (aunque es preferible utilizar esta última):

```
<object data="img/logo.svg" width="100%"></object>
```

Además, podemos incluir otros tipos más variados de elementos (como, por ejemplo, un mapa de Google Maps) a través de la etiqueta `iframe`. De este modo añadiríamos un mapa de Google con una ubicación predeterminada:

```
<iframe src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d3126.6024620057733!2d-0.5317505844743932!3d38.4044398796498!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0xd6233fea3991575%3A0xf28fcef8c48c1513!2sIES%20San%20Vicente!5e0!3m2!1ses!2ses!4v1601811180214!5m2!1ses!2ses" width="600" height="450" frameborder="0" style="border:0;" allowfullscreen="" aria-hidden="false" tabindex="0"></iframe>
```

En el caso de Google Maps, para obtener el código HTML que colocar en el *iframe*, una vez tengamos ya la ubicación deseada, pulsamos en el botón de Compartir, y elegimos *Insertar un mapa*. Después, podemos *Copiar HTML* para guardarlo en el portapapeles y poderlo pegar en la página que queramos:

Compartir



Enviar un enlace

Insertar un mapa

Mediano ▾

`<iframe src="https://www.google.com/maps/embed?pb=!1m18!1m`

COPIAR HTML

Al insertar este mapa, aceptas las [condiciones de servicio](#).**Ejercicio 9:**

Crema una página llamada **estreno.html**. Piensa en una película que te gustaría ver en el cine. Busca un trailer de la película, y la ubicación de un cine donde la vayan a poner, y añade en la página:

- Una etiqueta *video* para ver el trailer de la película
- Una etiqueta *iframe* para ver la localización del cine