



GOGODEV / @JJRuizEmpresa



Autor: Juan José Ruiz a.k.a GOGODEV

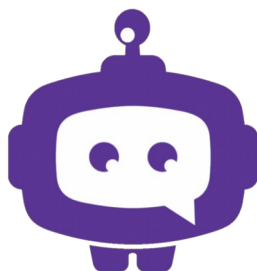
YouTube: <https://youtube.com/c/gogodev>

Twitter: <https://twitter.com/JJRuizEmpresa> (@JJRuizEmpresa)

#### IMPORTANTE

El contenido de este pdf lo tienes también en vídeo en el canal de YouTube <https://youtube.com/c/gogodev>, dentro de la lista de reproducción: Curso de HTML.

Todo el contenido formativo es gratuito, no olvides pasarte y suscribirte para no perderte nada, y así nos apoyas :)





GOGODEV / @JJRuizEmpresa

<CAPÍTULO .03>

# LA MAGIA DEL MULTIMEDIA

ETIQUETAS RELACIONADAS CON LA ESCRITURA

</CAPÍTULO .03>

<https://youtube.com/c/gogodev>



# LA MAGIA DEL MULTIMEDIA

## <CÓMO\_CONSTRUIR\_RUTAS\_RELATIVAS>

A lo largo de este capítulo vamos a comenzar a añadir contenido multimedia a nuestras construcciones. Esto es: imágenes, vídeos, audios... Por ende, vamos a necesitar indicar en dichas etiquetas dónde se encuentran dichos recursos. Es decir, vamos a tener que mostrarle a nuestro código la ruta para que el navegador pueda cargar correctamente la información.

Y por ello, vamos a comenzar este capítulo hablando sobre cómo se construye una ruta, los tipos de rutas existentes y cómo usarlas de forma apropiada para que, cuando entremos en materia, todo vaya como la seda.

### Tipos de rutas

Existen dos tipos de rutas principalmente:

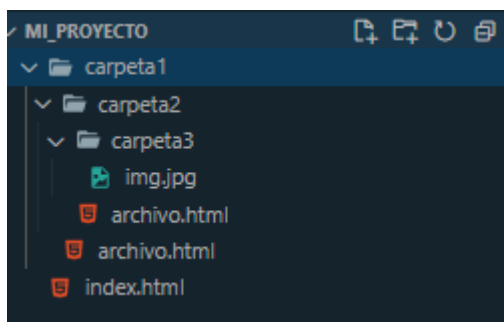
- **Rutas absolutas:** Estas son las más fáciles de escribir, ya que contienen el camino completo hacia el recurso. No importa dónde se encuentre el archivo que hace la llamada al recurso, la ruta absoluta siempre se escribe igual. Por ejemplo, si queremos hacer uso de una imagen que se encuentra en <https://picsum.photos/200/300>, su ruta absoluta sería exactamente esa: <https://picsum.photos/200/300>, expresando la dirección completa. Usaremos este tipo de rutas cuando llamemos a recursos externos que se encontrarán fuera de nuestro sitio web (una foto alojada en Google, por ejemplo). Para el resto de ocasiones (la gran mayoría) usaremos siempre las rutas relativas, ya que las absolutas pueden conducirnos a error.
- **Rutas relativas:** Son las rutas en las cuales no especificamos el camino completo para llegar a ellas, si no que expresamos cómo llegar al recurso desde la posición en la que se encuentra el archivo de código que estamos escribiendo. Un ejemplo de ruta relativa sería: *img/equipo/miembro1.jpg*. En esta ruta estamos indicando que el archivo se encuentra, desde la ubicación en la que se encuentra el documento que estamos escribiendo, accediendo a una carpeta llamada *img*. Una vez dentro accediendo a una carpeta llamada *equipo* y, dentro de esta, el archivo es la imagen que se llama *miembro1.jpg*.

### Cómo se construye una ruta

Profundicemos en cómo escribir una ruta relativa:

Para adentrarnos en una carpeta colocamos el nombre de la carpeta, y posteriormente usamos el símbolo /

Por ejemplo, en la siguiente estructura, donde la carpeta 3 está dentro de la carpeta 2, y la carpeta 2 dentro de la 1...



Si queremos usar la imagen img.jpg que se encuentra en la carpeta3 en el archivo index.html que se encuentra en el directorio raíz del proyecto, la ruta será:

`carpeta1/carpeta2/carpeta3/img.jpg`

Sin embargo, si queremos usar ese mismo recurso desde el documento archivo.html que se encuentra en la carpeta2, la ruta será:

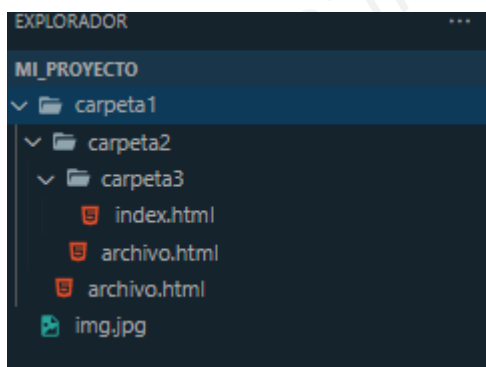
`carpeta3/img.jpg`

Ya que siempre escribimos la ruta *saliendo* desde la posición del archivo en el que estamos escribiendo la ruta.

Muy bien, ya sabemos entrar en carpetas. Ahora la duda que podría surgir sería ¿y cómo salimos de ellas?

Muy sencillo. Para salir de una carpeta, usamos la expresión `../`

Por ejemplo, supongamos la siguiente estructura de archivos:



En esta ocasión, es el index.html el que se encuentra dentro de la carpeta3, mientras que el recurso (img.jpg) se encuentra en el directorio raíz. Por tanto, para construir la ruta correctamente deberíamos indicar:

*"Sal de la carpeta3, sal de la carpeta2, sal de la carpeta1, el archivo se llama img.jpg"*

Entonces, si para salir de una carpeta al nivel superior usamos `../` la solución sería:



.././../img.jpg

Y así, combinando el acceso y la salida de los diferentes directorios, podemos localizar de forma relativa cualquier recurso de nuestro proyecto.

*RECOMENDACIÓN: Crea un proyecto nuevo y practica escribiendo cómo serían las diferentes rutas relativas antes de continuar.*

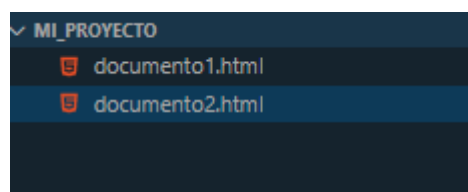
</CÓMO\_CONTRUIR\_RUTAS\_RELATIVAS>

<ENLACES>

Ahora que ya sabemos escribir rutas, es hora de afrontar nuestra primera herramienta de navegación: los enlaces.

Los enlaces, o hipervínculos, o links, o anclas, son la base de toda estructura web, ya que nos permiten navegar entre los diferentes documentos. Cuando pulsamos sobre un enlace, el navegador saldrá del documento actual, y cargará el documento al que referencia dicho enlace.

Creemos un proyecto con dos documentos html:



Cada uno de estos documentos, contendrá la estructura básica de un documento HTML, tal y como vimos en el capítulo 01.

Ahora, vamos a crear en el cuerpo del documento1.html un enlace que, al ser pulsado, nos lleve al documento2.html.

En el documento2.html escribiremos en su cuerpo un título <h1> (por ejemplo) que diga: "Hola, soy el documento2". Así, podremos comprobar que nuestro enlace funciona de una forma rápida y visual. Es hora de realizar el enlace. Para ello, usamos la etiqueta <a>. Además, completaremos dicha etiqueta usando el atributo **href**, donde indicaremos la ruta relativa al documento que debe abrir. Así pues, y siguiendo el ejemplo, en el documento1.html escribiremos:

```
<a href="documento2.html">Pulsa aquí para ver el documento 2</a>
```

Guarda ambos documentos y comprueba. ¡Felicidades! Has creado tu primer enlace. Por último, vamos a ampliar un poco más el concepto de enlace con un segundo atributo, el atributo **target**. Este atributo nos permite indicar dónde queremos que se abra el documento. Si no especificamos nada, como hemos podido comprobar, el enlace se abre en el lugar donde se encontraba el documento anterior. Sin embargo, podemos especificar a través de este atributo que éste se abra, por ejemplo, en una pestaña nueva del navegador, sin cerrar la pestaña actual:



- **target="\_blank"** Abre el documento en una nueva pestaña.
- **target="\_self"** El valor por defecto. Abre el enlace en el marco actual.
- **target="\_parent"** Abre el enlace en el marco padre donde ocurre la acción.
- **target="\_top"** Abre el enlace en el cuerpo completo de la página.

En la mayoría de ocasiones, se usará su valor por defecto, o el valor "\_blank". Las otras opciones se encuentran en desuso y no son aconsejables.

```
<a href="documento2.html" target="_blank">Abrir en una nueva ventana.</a>
```

**IMPORTANTE:** El href indica la ruta relativa. En este ejemplo la ruta relativa es directamente el nombre del documento ya que ambos se encuentran en el mismo lugar, pero si este se encuentra en otra ubicación, debemos expresar correctamente la ruta.

</ENLACES>

<IMÁGENES>

Continuamos ahora con las imágenes. Una vez que conocemos las rutas relativas, incluir imágenes te va a resultar muy sencillo. Para incluir una imagen, usamos la etiqueta **<img>** y dentro usamos el atributo **src** para indicar la ruta donde se encuentra dicha imagen. Como la etiqueta no contendrá nada en su interior (su contenido será el propio recurso cargado, podemos cerrar la etiqueta en la misma línea:

```

```

A dicha etiqueta podemos agregarle también un atributo alt que explique qué contiene la imagen. Esto es muy útil para los robots de búsqueda y el posicionamiento SEO de una página:

```

```

Además, y si lo deseamos, podemos especificar cual será la anchura **width** y altura **height** de esta con sendos atributos, si bien cuando entremos en las secciones de diseño ahondaremos más en profundidad en este aspecto:

```

```



## Hora de practicar

Crea un nuevo proyecto y coloca diferentes imágenes en rutas distintas.  
Prueba a crear un documento HTML que las muestre todas.

</IMÁGENES>

<VÍDEOS>

Una vez vistas las imágenes, es hora de dar cabida a la inclusión de vídeos en nuestras construcciones web.

La etiqueta de vídeo, así como la etiqueta de audio, forman parte de la nueva ampliación del estándar de HTML, y permanecen como etiquetas válidas desde la versión 5 (HTML5), por lo que son una de las novedades más interesantes incorporadas en la versión que permanece vigente a fecha de hoy. Al igual que con las imágenes, necesitaremos valernos de un atributo **src** mediante el cuál indicaremos la ubicación donde se encuentra el archivo.

```
<video src="mivideo.mp4"> Vídeo no soportado... </video>
```

Como puedes ver, la etiqueta vídeo funciona de forma similar a la etiqueta de imagen. Además, los atributos que podemos aplicar a una imagen, tales como la altura **height** o anchura **width** también nos serán válidos para la etiqueta de vídeo.

Si el navegador que utiliza el usuario es antiguo y no soporta vídeo, podemos colocar un mensaje que se mostrará para él dentro de la propia etiqueta, tal y como podemos ver en el ejemplo.

Sigamos profundizando en su uso. Uno de los puntos a tener en cuenta cuando colocamos vídeos en nuestra web, son los formatos de vídeo. Si bien el formato .mp4 es el más extendido y usado, existen otro tipo de formatos como el .ogg y el .webm que también son útiles. Si disponemos de nuestro vídeo en diferentes formatos, podemos especificar en la etiqueta de vídeo la ruta a todos ellos para que el navegador utilice la versión más favorable. Esto lo hacemos de la siguiente manera:

```
<video>
  <source src="mivideo.mp4" type="video/mp4">
  <source src="mivideo.ogg" type="video/ogg">
  <source src="mivideo.webm" type="video/webm">
</video>
```

Una vez hemos insertado el vídeo, es hora de analizar algunos atributos de interés. El primero de ellos es añadir un reproductor, es decir, añadir controles al vídeo para que el usuario pueda pausarlo, reanudarlo, ajustar el volumen, etcétera. Para añadir el reproductor bastará con hacer uso del atributo



**controls** de la etiqueta. Si lo indicamos, el vídeo aparecerá dentro de un reproductor de vídeo predeterminado según el navegador:

```
<video controls>
  <source src="mivideo.mp4" type="video/mp4">
  <source src="mivideo.ogg" type="video/ogg">
  <source src="mivideo.webm" type="video/webm">
</video>
```

Otra de las características que podemos indicar es si queremos que el vídeo inicie su reproducción automáticamente. Para ello, lo marcamos con el atributo de **autoplay**:

```
<video controls autoplay>
  <source src="mivideo.mp4" type="video/mp4">
  <source src="mivideo.ogg" type="video/ogg">
  <source src="mivideo.webm" type="video/webm">
</video>
```

Por último, vamos a ver el atributo **loop**. Con él, indicamos que el vídeo debe reproducirse en bucle, es decir, que cuando finalice debe volver a comenzar de forma automática:

```
<video controls autoplay loop>
  <source src="mivideo.mp4" type="video/mp4">
  <source src="mivideo.ogg" type="video/ogg">
  <source src="mivideo.webm" type="video/webm">
</video>
```

</VÍDEOS>

<AUDIO>

Veamos ahora la etiqueta de audio. Al igual que ocurre con la etiqueta de vídeo, **<audio>** fue introducida en la ampliación del estándar en su versión 5, y su funcionamiento es muy similar al de la etiqueta vídeo:

```
<audio src="miaudio.mp3">
  Tu navegador no soporta la reproducción de audio.
</audio>
```

Al igual que ocurre con el vídeo, también podemos incluir el audio en diferentes formatos, siendo los más usados **.mp3**, **.ogg**, **.wav**, **.aac** y **.opus**:

```
<audio>
  <source src="miaudio.mp3" type="audio/mp3">
  <source src="miaudio.wav" type="audio/wav">
</audio>
```





Por último, también podemos añadirle los atributos de **autoplay**, **loop** y **controls**:

```
<audio controls autoplay loop>  
  <source src="miaudio.mp3" type="audio/mp3">  
</audio>
```

</AUDIO>

<IFRAME>

En última instancia, y dentro de las etiquetas multimedia, vamos a ver el **<iframe>**. El **iframe** es una etiqueta que debe usarse con cuidado, y que empieza a quedar en desuso. Sin embargo, aún sigue siendo útil en ciertos escenarios. El **iframe** nos permite incrustar un html dentro de otro. Por ejemplo, incrustar otra página web dentro de la nuestra en una sección determinada.

Esto puede acarrear problemas de seguridad, por lo que sólo usaremos **iframes** con sitios confiables, por ejemplo: *YouTube*, *iVoox* o *Spotify*.

A día de hoy, el principal uso de los **iframe** es incluir vídeos de *YouTube* o *Vimeo*, o audios de *Spotify* o *iVoox* dentro de tu página web. Y es que habrás observado que, cuando accedemos a YouTube por ejemplo, dentro de las opciones de compartir cuando vemos un vídeo, una de ellas nos ofrece un enlace, y otra un código **iframe** para que podamos pegar el contenido en nuestra página.

Al igual que ocurre con las etiquetas de **audio** y **video**, podemos aplicar una anchura **width** y altura **height** a nuestro **iframe**.

Ejemplo de iframe de YouTube:

```
<iframe width="560" height="315" src="https://www.youtube.com/embed/GLlbmirxla0" title="YouTube video player"  
frameborder="0" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture"  
allowfullscreen></iframe>
```

En este ejemplo, podemos observar que el **iframe** tiene unas dimensiones de 560 píxeles de ancho y 315 de alto. Nos aparece también la ruta donde se encuentra el vídeo y el título del mismo.

Por último, especifica que el **iframe** no tendrá borde y que se permitirá ampliar a pantalla completa. El resto de atributos son propios de la plataforma *YouTube*, especificados dentro de **allow**.

¡Y eso es todo dentro de este bloque! Recuerda practicar todo lo posible hasta familiarizarte con lo visto antes de continuar. Como habrás podido observar, y aún a falta de diseño, nuestra web ahora comienza a *"hacer cosas"*. Sigamos dando pasos hacia delante.

Si quieres repasar en vídeo todo este bloque, o te has quedado atascado, puedes ver todo el proceso en el canal YouTube de GOGODEV.



GOGODEV / @JJRuizEmpresa

**Lo importante no es el final. Es disfrutar del camino, pues como desarrollador web, siempre estarás caminando.**

</IFRAME>

<https://youtube.com/c/gogodev>