

# Tema 8. Implementación del *front-end*. HTML5

DISEÑO Y DESARROLLO DE VIDEOJUEGOS WEB

Dra. Carolina García Vázquez.

[carolina.garcia@esne.es](mailto:carolina.garcia@esne.es)

# Tabla de contenidos

- 1. ESTRUCTURA BÁSICA**
- 2. SINTAXIS DE TEXTOS**
- 3. INSERCIÓN DE MULTIMEDIA**
- 4. NAVEGACIÓN: ENLACES Y ANCLAS**
- 5. OTROS ELEMENTOS: LISTAS, TABLAS, FORMULARIOS E ICONOS**
- 6. LA WEB SEMÁNTICA**



# 1. Estructura básica

Tema 8. Implementación del  
*front-end*. HTML5

# Introducción

📖 World Wide Web Consortium (2016). *HTML & CSS*.  
<https://www.w3.org/standards/webdesign/htmlcss>



- HTML es el lenguaje para describir la **estructura** y los **contenidos** de las páginas web.
- Nos permite añadir a nuestra página textos, multimedia (imágenes, vídeos y audios), elementos de navegación y formularios.
- Le damos la estructura según lo que representen los elementos:
  - Encabezados o títulos.
  - Cabecera o pie de la página.
  - Menú de navegación.
  - Etc.
- HTML no es un lenguaje de programación, es un lenguaje de marcado.



*La **presentación** de nuestra página, es decir, los estilos, formatos y maquetación, los definiremos en el CSS.*

# Sintaxis de HTML



- HTML (*Hypertext Markup Language*) es un lenguaje de marcas, etiquetas o elementos.
- Las etiquetas que pueden tener contenido deberán cerrarse.
- Habrá etiquetas que no haya que cerrar.
- Si definimos atributos, lo haremos en la etiqueta de apertura. Los atributos tienen unos nombres predefinidos y su valor debe ir entre comillas.

`<meta charset="utf-8">`

Elemento sin cierre  
y con atributo

`<body>`

`</body>`

Etiqueta de  
apertura y de cierre

`<!-- Comentario -->`

Se pueden añadir comentarios.  
No se presentan en la página,  
son para ayuda al desarrollador

# Estructura básica de un documento HTML



- Un **documento HTML** siempre tendrá la misma estructura:

```
<!DOCTYPE html>
```

Definimos el tipo de documento, en este caso, HTML5

```
<html lang="es">
```

**Todo** irá dentro de la etiqueta HTML. Le añadiremos el atributo del idioma

```
<head>  
  <!-- Metadatos -->  
</head>
```

En la etiqueta **<head>** añadiremos los **metadatos**, es decir, la información para el navegador que no forma parte del contenido que se le presentará al usuario

```
<body>  
  <!-- Contenido -->  
</body>
```

En la etiqueta **<body>** añadiremos el **contenido** de nuestra página, esto es, lo que le vamos a presentar al usuario

```
</html>
```

# Estructura básica de un documento HTML



- **Metadatos:** consisten en información que caracteriza datos, describen el contenido, calidad, condiciones, historia, disponibilidad y otras características de los datos.
- En HTML los metadatos que añadiremos serán:
  - Definimos el título de la página que aparecerá en la pestaña del navegador.
  - Definimos la codificación para que presente tildes y 'ñ' en castellano.
  - Definimos las hojas de estilo (CSS) asociadas a nuestro documento a través de su vínculo. También los vínculos a otras hojas de estilo, por ejemplo, a Font Awesome o a tipografías de Google Fonts.
  - Definimos las librerías JavaScript o añadimos código JavaScript.
  - Definimos el *viewport* para la adaptación a otros tamaños de pantalla.

# Estructura básica de un documento HTML



- Ejemplo de metadatos:

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <title>Título de la página</title>    <!-- Título de la página en la pestaña del navegador -->
    <meta charset="utf-8">                <!-- Codificación para el castellano -->
    <link href="https://fonts.googleapis.com/css?family=Philosopher" rel="stylesheet"> <!-- Tipografía GF -->
    <link href="css/styles.css" rel="stylesheet" type="text/css"> <!-- Enlace a mi hoja de estilos -->
    <script src="js/script.js"></script>  <!-- Enlace a mi código JavaScript -->
    <meta name="viewport" content="width=device-width, initial-scale=1.0"> <!-- Para el responsive -->
  </head>
  <body>
    <!-- Contenido -->
  </body>
</html>
```

## 2. Sintaxis de textos

Tema 8. Implementación del  
*front-end*. HTML5

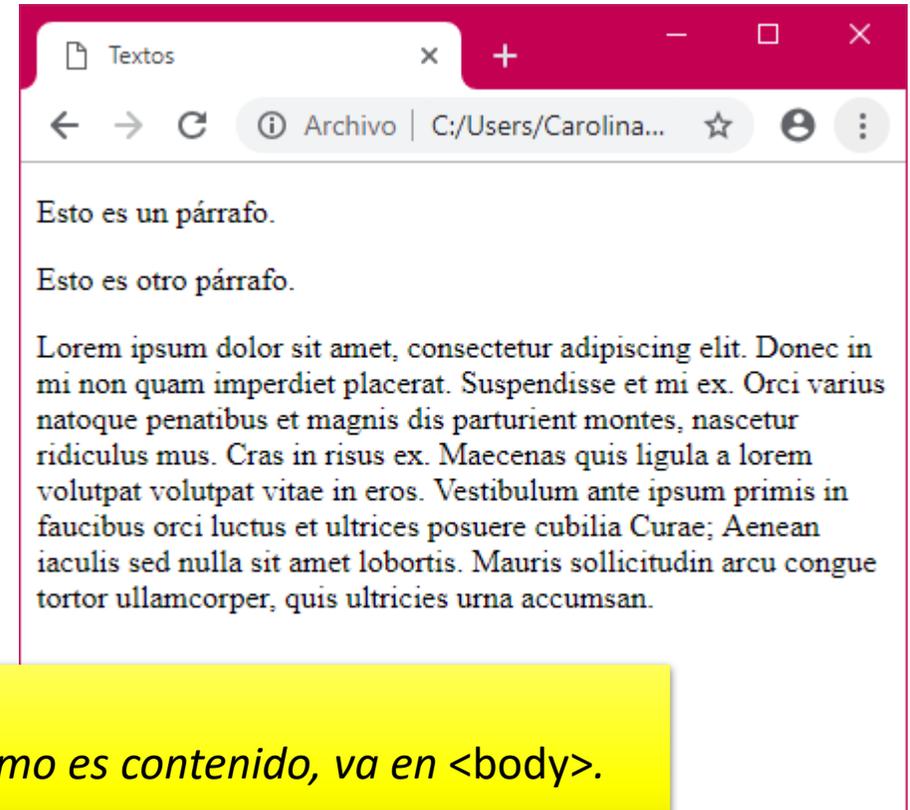
# Sintaxis de textos. Párrafos



`<p>Esto es un párrafo.</p>`

`<p>Esto es otro párrafo.</p>`

`<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec in mi non quam imperdiet placerat. Suspendisse et mi ex. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Cras in risus ex. Maecenas quis ligula a lorem volutpat volutpat vitae in eros. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aenean iaculis sed nulla sit amet lobortis. Mauris sollicitudin arcu congue tortor ullamcorper, quis ultricies urna accumsan.</p>`

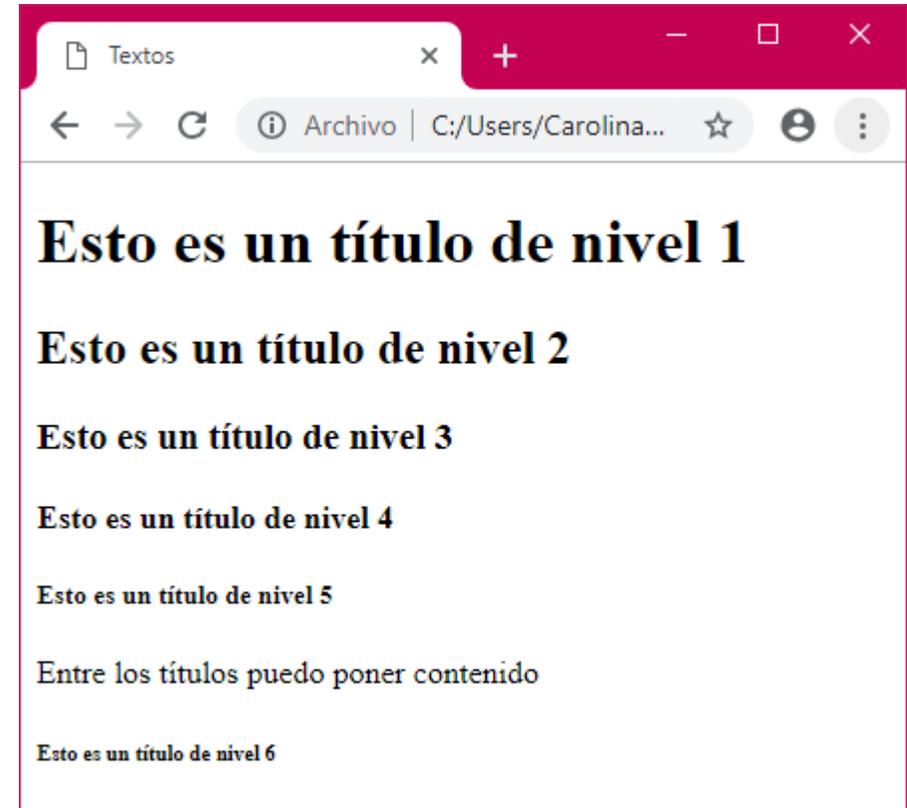


# Sintaxis de textos. Párrafos



`<h1>`Esto es un título de nivel 1`</h1>`  
`<h2>`Esto es un título de nivel 2`</h2>`  
`<h3>`Esto es un título de nivel 3`</h3>`  
`<h4>`Esto es un título de nivel 4`</h4>`  
`<h5>`Esto es un título de nivel 5`</h5>`  
`<p>`Entre los títulos puedo poner contenido`</p>`  
`<h6>`Esto es un título de nivel 6`</h6>`

*Como es contenido, va en `<body>`.*

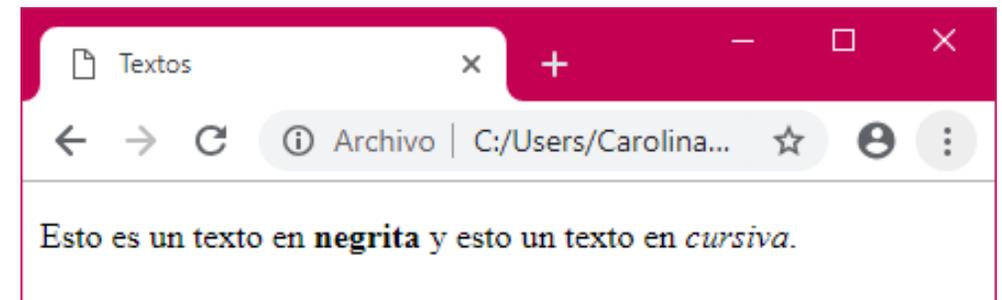


# Sintaxis de texto. Formatos básicos



- Los estilos siempre los definiremos en CSS, pero hay un par de excepciones:
  - `<strong>`: negrita.
  - `<em>`: cursiva.
- Por defecto, `<strong>` es negrita y `<em>` es cursiva, pero se adapta a la configuración que tenga el usuario en su navegador web. Además, podremos personalizar cómo se presentan en CSS.

`<p>`Esto es un texto en `<strong>`negrita`</strong>`  
y esto un texto en `<em>`cursiva`</em>`.`</p>`



# 3. Inserción de multimedia

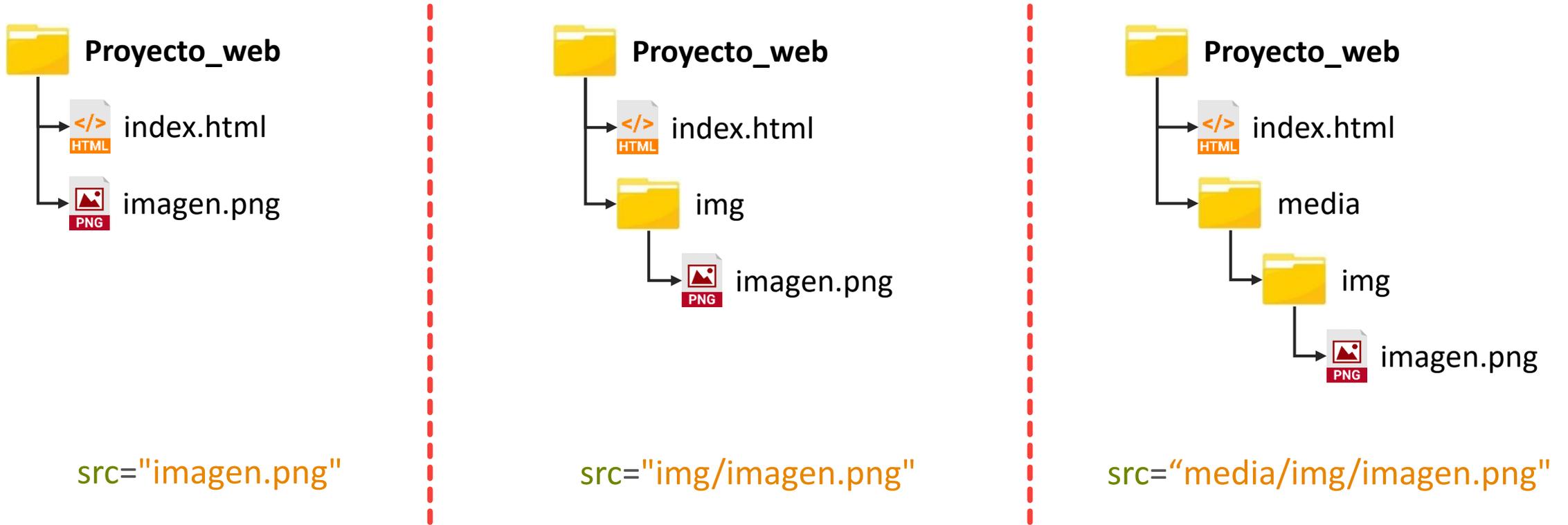
Tema 8. Implementación del  
*front-end*. HTML5

# Inserción de multimedia. Definición de rutas



- Cuando nuestro proyecto web disponga de elementos multimedia, crearemos una estructura de carpetas. Por ejemplo:
  - Una carpeta *media* que contenga todos los elementos multimedia.
  - Una carpeta *media* que contenga una subcarpeta *imagenes* para las imágenes y otra *video* para los vídeos.
  - Una carpeta *img* para las imágenes y una carpeta *audio* para los audios.
- Existen una serie de etiquetas para añadir multimedia, cada una para insertar imágenes, audio o vídeo.
- Estas etiquetas tiene un **atributo src** cuyo valor será la ruta relativa al archivo de imagen, audio o título.

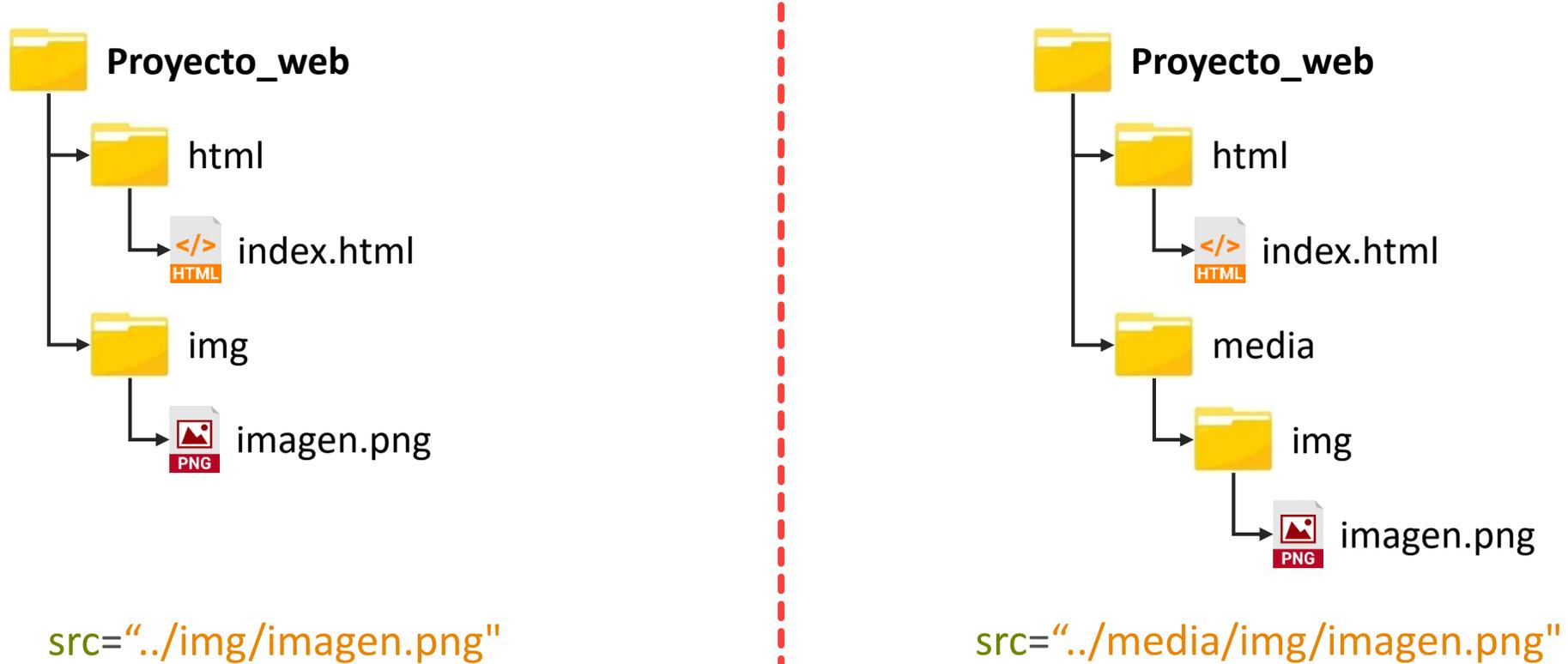
# Inserción de multimedia. Definición de rutas



*Los nombres de archivos multimedia no deben contener espacios ni caracteres extraños.*

*Esta filosofía para las rutas vale para cualquier archivo que queramos enlazar (imágenes, vídeos, hoja de estilos, etc.).*

# Inserción de multimedia. Definición de rutas



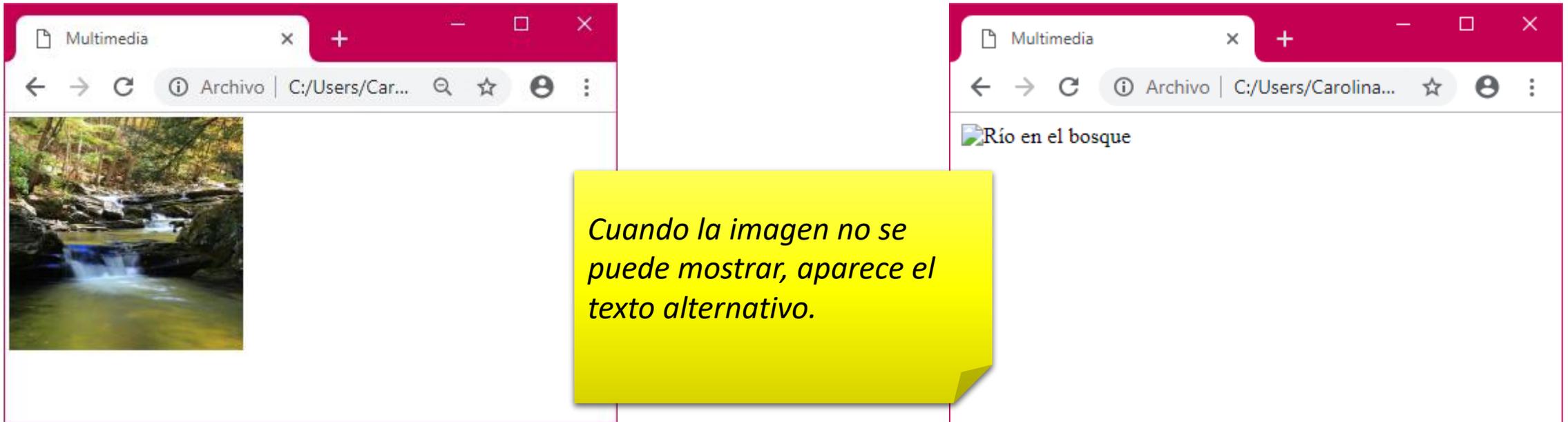
*Para subir un nivel en la estructura de carpetas utilizamos ../*

# Inserción de multimedia. Imágenes



- Se utiliza la etiqueta `<img>` que no tiene cierre.
- Esta etiqueta tiene los siguientes atributos principales:
  - `src`: ruta a la imagen. OBLIGATORIO.
  - `alt`: texto alternativo. Se muestra en algunos navegadores cuando se pasa el ratón por encima. Es lo que se lee con un lector de pantalla (utilizado por personas con problemas de visión). OBLIGATORIO.
  - `longdesc`: acceso a una página distinta con una descripción más larga. Se utiliza en gráficos complejos para complementar al contenido de `alt`.
- El ancho y el alto de la imagen lo especificaremos en la hoja de estilos.

# Inserción de multimedia. Imágenes



```

```

# Inserción de multimedia. Audio

- Etiqueta `<audio>` (aparece en HTML5). Esta etiqueta sí tiene cierre.
- Se le pueden definir los siguientes atributos:
  - `src`: ruta del archivo a reproducir. Puede ser local o remota. OBLIGATORIO.
  - `type`: tipo de archivo a reproducir. OBLIGATORIO.
  - `autoplay`: atributo sin valor. Inicia la reproducción del audio cuando se carga la página.
  - `controls`: atributo sin valor. Permite mostrar controles para el audio.
  - `loop`: atributo sin valor. Reproduce el audio en bucle.

```
<audio controls src="AudioTest.ogg" type="audio/ogg"
autoplay></audio>
```



# Inserción de multimedia. Múltiples versiones de audio



- Anidado en `<audio>` se añaden tantos `<source>` como audio alternativos tengamos para reproducir. En el caso de que no se pueda reproducir el primero, se intentará reproducir el segundo.

```
<audio controls>  
  <source src="audio.ogg" type="audio/ogg"/>  
  <source src="audio.mp3" type="audio/mp3"/>  
</audio>
```

- En `<source>` se definen siempre los atributos `src` y `type`. El resto de atributos (`controls`, `loop`, etc.) se definirán en `<audio>`.
- Dentro de `<audio>` se puede añadir texto que solo se muestra si el navegador no puede reproducir el audio.

# Inserción de multimedia. Tipos de audio soportados



NAVEGADOR	type=audio/mp3	type=audio/ogg	type=audio/wav
<b>Internet Explorer</b>	Sí	No	No
<b>Chrome</b>	Sí	Sí	Sí
<b>Firefox</b>	Sí	Sí	Sí
<b>Safari</b>	Sí	No	Sí
<b>Opera</b>	Sí	Sí	Sí



# Inserción de multimedia. Vídeo



- Etiqueta **<video>** (aparece en HTML5). Esta etiqueta sí tiene cierre.
- Se le pueden definir los siguientes atributos:
  - **src**: ruta del archivo a reproducir. Puede ser local o remota. OBLIGATORIO.
  - **type**: tipo de archivo a reproducir. OBLIGATORIO.
  - **autoplay**: atributo sin valor. Inicia la reproducción del vídeo cuando se carga la página. Solo funciona si está silenciado (muted).
  - **controls**: atributo sin valor. Permite mostrar controles para el vídeo.
  - **loop**: atributo sin valor. Reproduce el vídeo en bucle.
  - **muted**: atributo sin valor. Reproduce el vídeo sin sonido.

```
<video controls src="video1.mp4" type="video/mp4" loop/>
```

# Inserción de multimedia. Múltiples versiones de vídeo



- Anidado en `<video>` se añaden tantos `<source>` como vídeos alternativos tengamos para reproducir. En el caso de que no se pueda reproducir el primero, se intentará reproducir el segundo.

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

- En `<source>` se definen siempre los atributos `src` y `type`. El resto de atributos (`controls`, `loop`, etc.) se definirán en `<video>`.
- Dentro de `<video>` se puede añadir texto que solo se muestra si el navegador no puede reproducir el vídeo.

# Inserción de vídeo. Tipos soportados



NAVEGADOR	type=video/mp4	type=video/webm	type=video/ogg
<b>Internet Explorer</b>	Sí	No	No
<b>Chrome</b>	Sí	Sí	Sí
<b>Firefox</b>	Sí	Sí	Sí
<b>Safari</b>	Sí	No	No
<b>Opera</b>	Sí	Sí	Sí



**MP4:** codificación de vídeo H.264 y codificación de audio AAC.

**WebM:** codificación de vídeo VP8 y codificación de audio Vorbis.

**Ogg:** codificación de vídeo Theora y codificación de audio Vorbis.

- Se utiliza la etiqueta `<iframe>`, que nos sirve para embeber otro tipo de contenido en nuestro documento.
- Se le pueden definir los siguientes atributos:
  - `src`: ruta del archivo a reproducir. Puede ser local al servidor o remota.
  - `width`, `height`: ancho y alto del vídeo en píxeles, respectivamente.
- Las opciones para ocultar los controles (presentados por defecto) o para reproducir el vídeo en bucle se especifican en la URL del vídeo.
- Para embeber el vídeo y que nos salgan las opciones por defecto, seleccionamos en YouTube *Compartir* → *Insertar*.

# Inserción de multimedia. Vídeos de YouTube



The screenshot shows a web browser window displaying a YouTube video. The video player is titled "Tour Virtual de ESNE - Escuela Universitaria de Diseño, Innovación...". The video content shows a building with a banner that says "We Love Your Talent". The "Embed Video" dialog box is open, showing the following code snippet:

```
<iframe width="560" height="315"
src="https://www.youtube.com/embed/NujQ7InCX1U" frameborder="0" allow="autoplay;
encrypted-media" allowfullscreen>
</iframe>
```

Below the code, there are options for video playback:

- Empezar en 0:00
- OPCIONES DE INSERCIÓN**
- Mostrar sugerencias de vídeos cuando finalice la reproducción.
- Mostrar los controles del reproductor.

At the bottom of the dialog, there are additional options:

- Mostrar el título del vídeo y las acciones del reproductor.
- Activar el modo de mejora de la privacidad.

# 4. Navegación. Enlaces y anclas

Tema 8. Implementación del  
*front-end*. HTML5

# Navegación. Enlaces



- Se utiliza la etiqueta `<a>`. Dentro del elemento pondremos el texto, imagen o vídeo que abrirá el vínculo.
- Tiene tres atributos principales:
  - `href`: ruta del enlace. OBLIGATORIO.
  - `target`: forma de abrir el enlace. El valor `_blank` lo abre en una nueva ventana.
  - `title`: información adicional.

```
<a href="http://www.esne.es/" target="_blank" title="Se abre en una nueva ventana">ESNE</a>
```

```
<a href="http://www.esne.es/"></a>
```

*Normalmente cuando tenemos un enlace que se abre en una ventana nueva lo indicamos en title.*

# Navegación. Anclas



- Las anclas son enlaces dentro del mismo documento HTML.
- Antes de crear el vínculo al elemento al que queremos ir con el enlace, hay que definir el atributo **id** en dicho elemento. Cualquier elemento incluido en `<body>` puede tener atributo `id`.
- El `id` tiene un valor único, no puede haber dos elementos en el mismo documento HTML con el mismo valor de `id`.

```
<h1 id="inicio">Introducción a HTML5</h1>
```

- Posteriormente, hacemos referencia al ese `id` en el atributo `href` del elemento `<a>` incluyendo antes el símbolo **#**:

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

# 5. Otros elementos

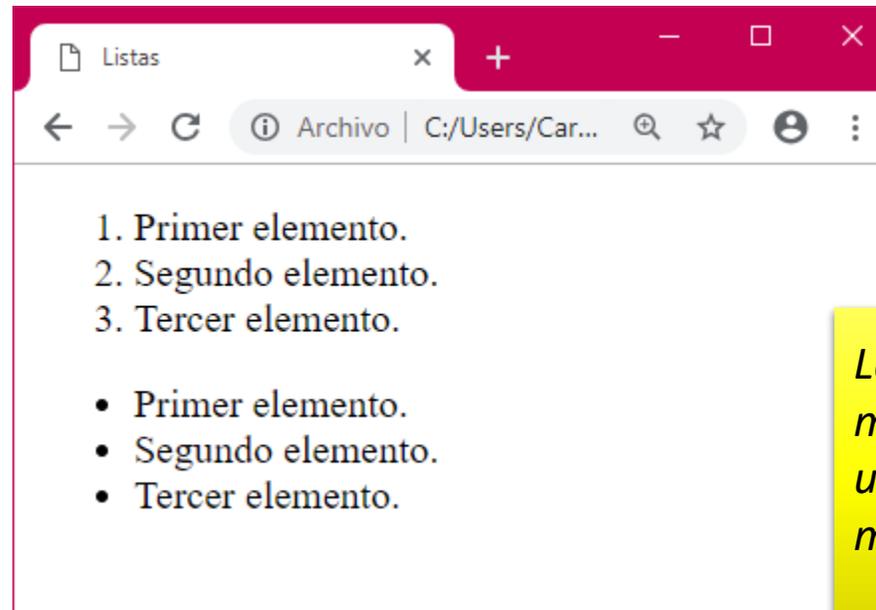
Tema 8. Implementación del  
*front-end*. HTML5

# Otros elementos. Listas

- Tienen un elemento raíz que indica si la lista es ordenada (**<ol>**) o desordenada (**<ul>**). Dentro **solo** puede haber elementos **<li>** (elementos de lista).

```
<ol>
  <li>Primer elemento.</li>
  <li>Segundo elemento.</li>
  <li>Tercer elemento.</li>
</ol>
```

```
<ul>
  <li>Primer elemento.</li>
  <li>Segundo elemento.</li>
  <li>Tercer elemento.</li>
</ul>
```



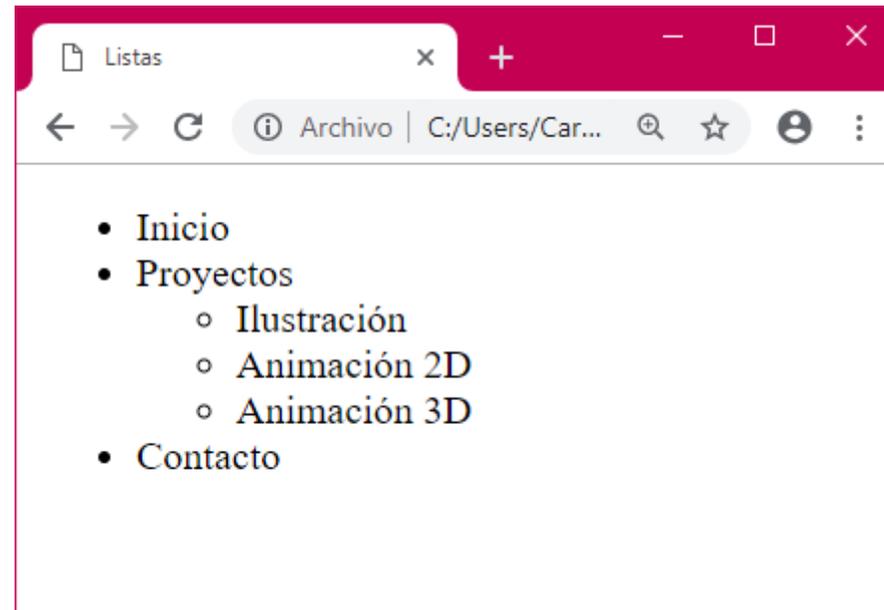
*Las listas desordenadas son muy importantes porque las utilizaremos para hacer los menús.*

# Otros elementos. Listas de varios niveles



- Para listas con varios niveles, añadiremos otra lista dentro de uno de los elementos `<li>`.

```
<ul>
  <li>Inicio</li>
  <li>Proyectos
    <ul>
      <li>Ilustración</li>
      <li>Animación 2D</li>
      <li>Animación 3D</li>
    </ul>
  </li>
  <li>Contacto</li>
</ul>
```



# Otros elementos. Tablas



- Para las tablas se utiliza el elemento `<table>`, en el que anidamos el contenido.
- Cada fila la comenzamos con el elemento `<tr>`, en el que anidamos sus celdas.
- Las celdas se definen de dos formas:
  - `<th>` si es una celda de encabezado.
  - `<td>` para el resto de celdas.
- Para definir el título de la tabla utilizamos el elemento `<caption>`.
- También existen los elementos `<thead>` y `<tbody>` para identificar el encabezado y el contenido, respectivamente.

# Otros elementos. Tablas



```
<table>
  <caption>Título de la tabla</caption>
  <tr>
    <th>Título</th>
    <th>Tipo</th>
  </tr>
  <tr>
    <td>Diseño Multimedia y Gráfico</td>
    <td>Grado</td>
  </tr>
  <tr>
    <td>Diseño y desarrollo de videojuegos</td>
    <td>Grado</td>
  </tr>
</table>
```

The screenshot shows a web browser window with a single tab titled 'Tablas'. The address bar shows 'Archivo | C:/Users/Carolina...'. The main content area displays a table with the following structure:

<b>Título</b>	<b>Tipo</b>
Diseño Multimedia y Gráfico	Grado
Diseño y desarrollo de videojuegos	Grado

# Otros elementos. Formularios

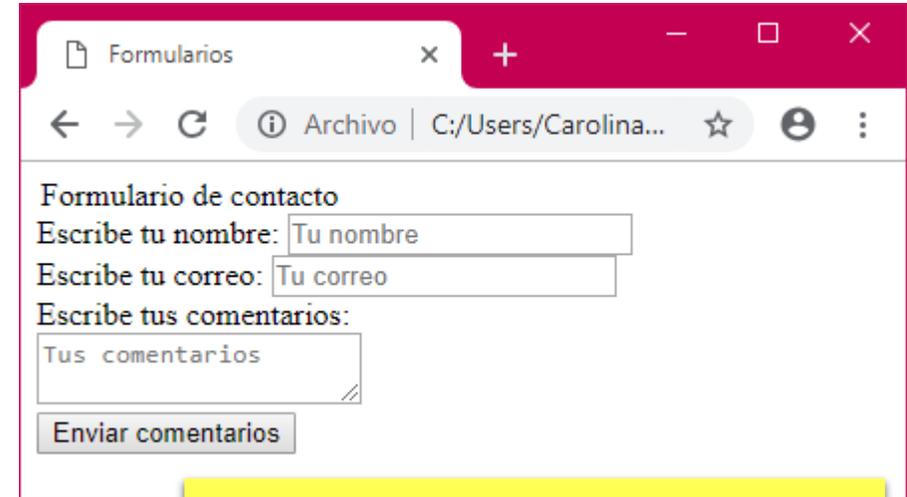


- Los formularios se definen con la etiqueta **<form>**, en la que se anidarán los campos que queremos que existan en el formulario.
- Tiene dos atributos principales (de momento no los vamos a utilizar):
  - `action`, donde le indicamos la acción a realizar cuando se envíe el formulario.
  - `method`, donde definimos el método de envío (tradicionalmente, `get` o `post`).
- A través de **<legend>** le damos título al formulario.
- Para los campos en los que escribirá el usuario:
  - **<label>**: nombre al campo. El atributo `for` debe coincidir con el `id` del `<input>`.
  - **<input>**: campo a rellenar. Definiremos el tipo en el atributo `type` y le daremos un `id` que coincidirá con el `for` del `<label>`. No tiene cierre.

# Otros elementos. Formularios



```
<form>
  <legend>Formulario de contacto</legend>
  <label for="nombre">Escribe tu nombre: </label>
  <input type="text" id="nombre" placeholder="Tu nombre">
  <br>
  <label for="correo">Escribe tu correo: </label>
  <input type="email" id="correo" placeholder="Tu correo">
  <br>
  <label for="comentarios">Escribe tus comentarios:</label>
  <br>
  <textarea id="comentarios" placeholder="Tus comentarios"></textarea>
  <br>
  <input type="submit" value="Enviar comentarios">
</form>
```



*Con placeholder definimos lo que queremos que aparezca en el campo de texto.*

*Textarea es un campo de texto grande.*

*El botón de enviar el formulario también es un elemento <input>.*

# Otros elementos. Iconos de Font Awesome



- *Framework* de iconos vectoriales que se tratan como si fueran fuentes:
  - Se añaden como si fueran estilos CSS.
  - Se tratan como fuentes, por lo que podemos cambiar su formato a través de CSS (color, tamaño).
- Ventajas:
  - Es un *framework* abierto, por lo que no hay que pagar licencias.
  - No es necesario diseñar nuevos iconos de distintos colores.
  - Como son vectoriales, se pueden reescalar sin perder resolución.
  - Disminuye la carga de imágenes de la página.

# Otros elementos. Iconos de Font Awesome



- En <head> importamos la librería de iconos.

```
<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.1/css/all.css"
integrity="sha384-fnmOCqbTIWIlj8LyTjo7mOUSTjsKC4pOpQbqyi7RrhN7udi9RwhKkMHpvLbHG9Sr"
crossorigin="anonymous">
```

- Se añaden a través del elemento <i> (a veces también con <span>) dándole como clase el icono que queremos utilizar.

```
<i class="fa fa-bars"></i> Menú hamburguesa      ≡ Menú hamburguesa
```

- Con CSS, modificamos su estilo.
- Ruta para importar: <https://fontawesome.com/start>
- Ruta para ver los iconos: <https://fontawesome.com/icons>

# 6. La web semántica

Tema 8. Implementación del  
*front-end*. HTML5

# Creación de contenedores



- Se hacen con el elemento **<div>**. Define cajas en el documento, permitiendo agrupar los elementos visuales en bloques que se formatearán con CSS.
- Solo tiene dos atributos que no son obligatorios:
  - **id**: identifica cada <div>. Los id son únicos en cada página.
  - **class**: asigna un formato definido en CSS para un <div>. Varios elementos de la misma página pueden compartir un mismo class si van a presentar el mismo formato.
- Serán muy prácticos para después maquetar fácilmente los elementos visuales de nuestra página.
- Los elementos semánticos funcionan como <div>.

## ¿QUÉ ES?

Web extendida que se centra en el significado de las búsquedas del usuario, no solo en las palabras.

## ¿PARA QUÉ SIRVE?

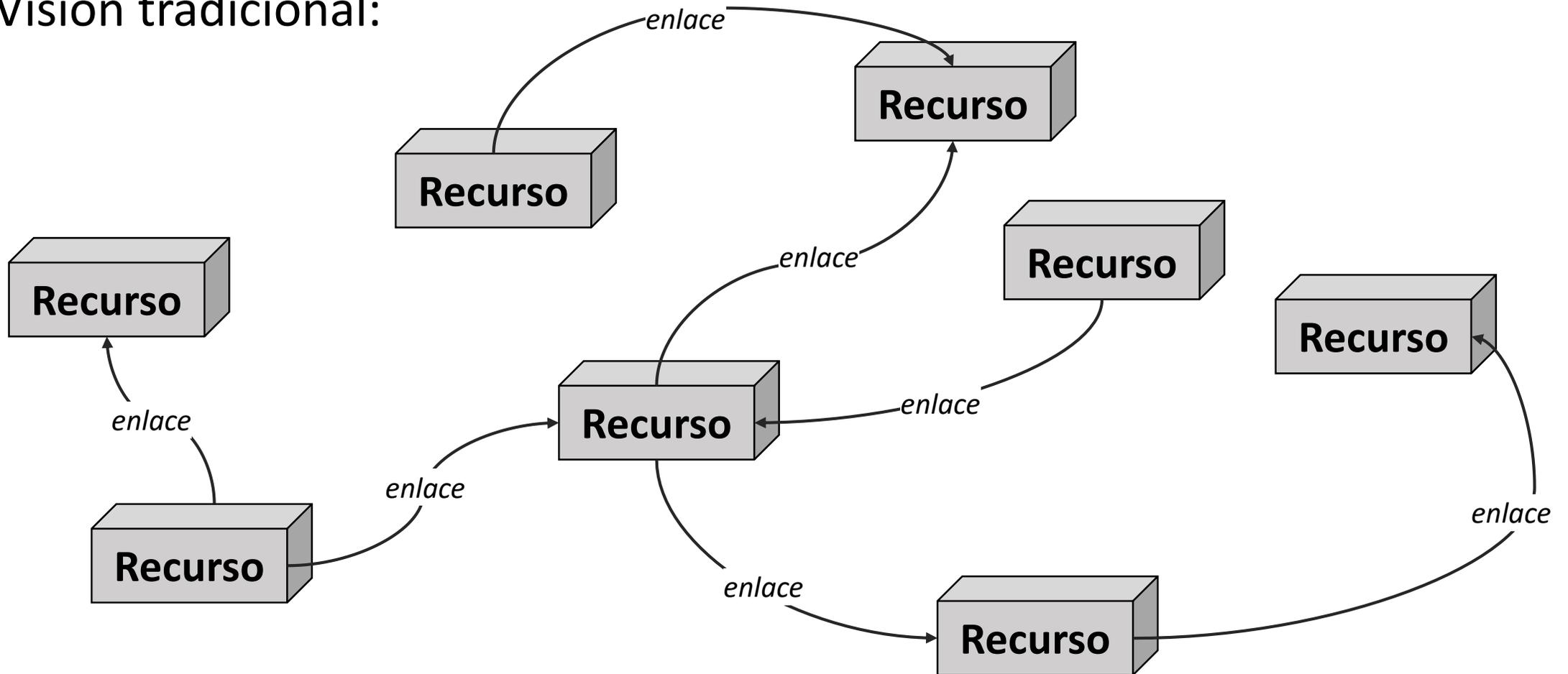
Gracias a la semántica web, el software puede procesar el contenido, razonar con él, combinarlo y realizar deducciones lógicas.

## ¿CÓMO FUNCIONA?

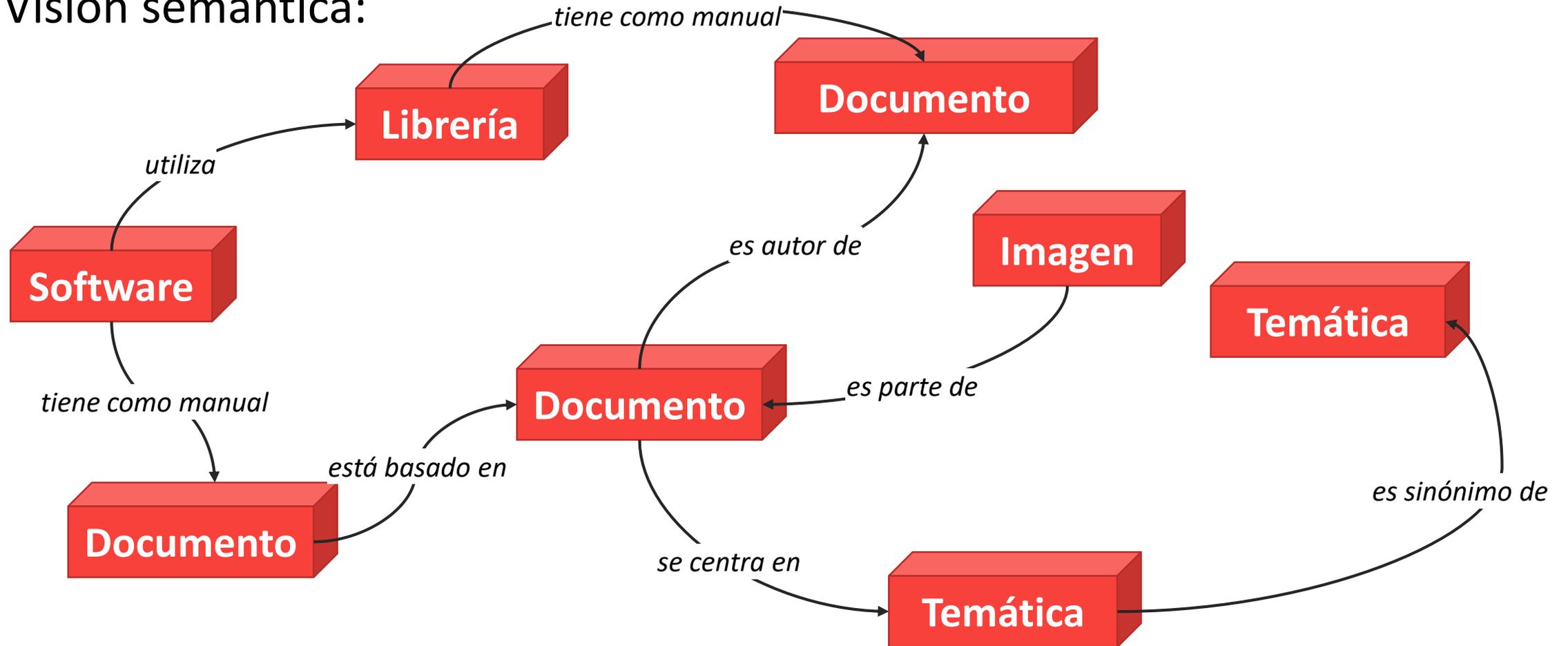
A través de la creación de una base de conocimiento según las búsquedas de otros usuarios.

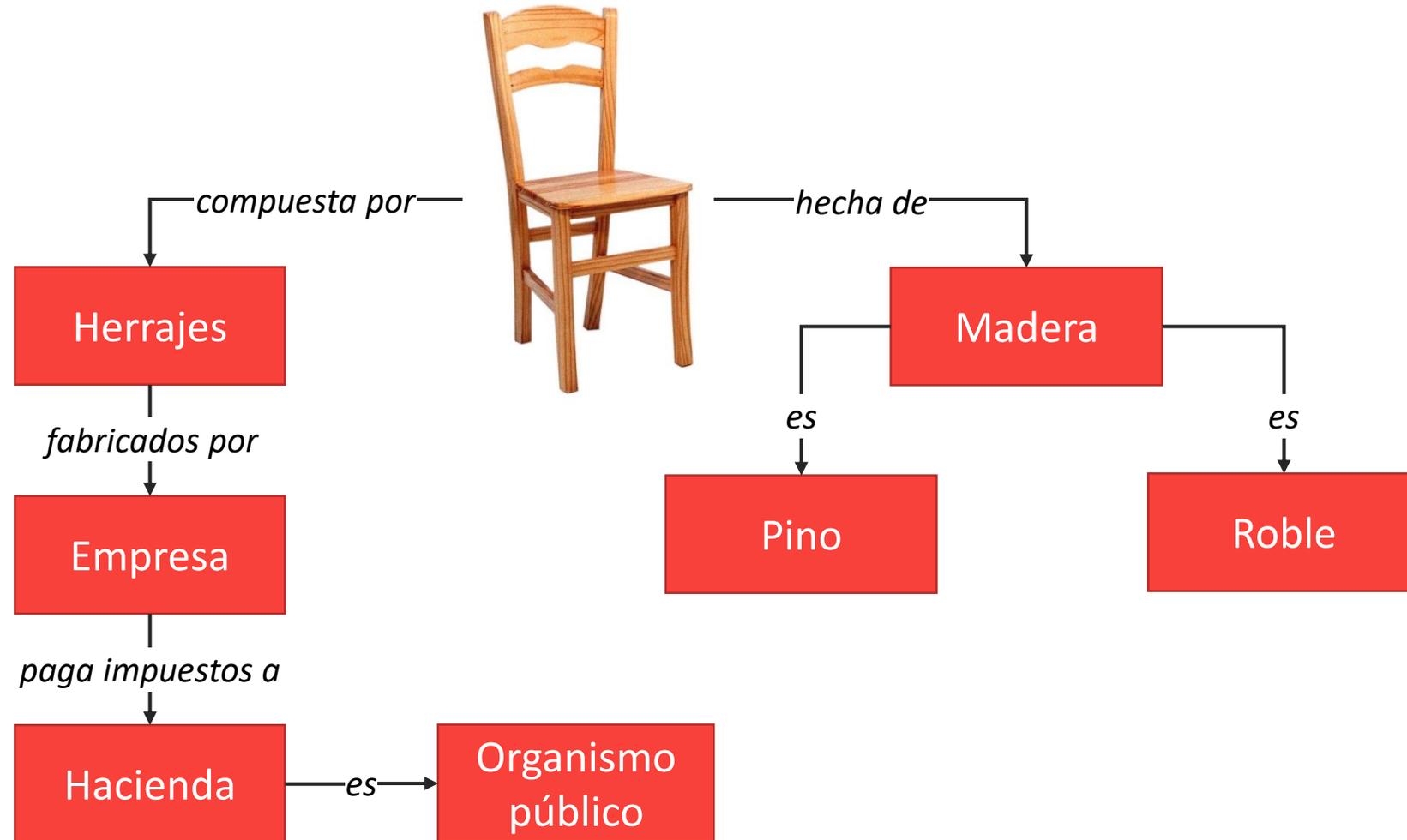


- Visión tradicional:



- **Visión semántica:**





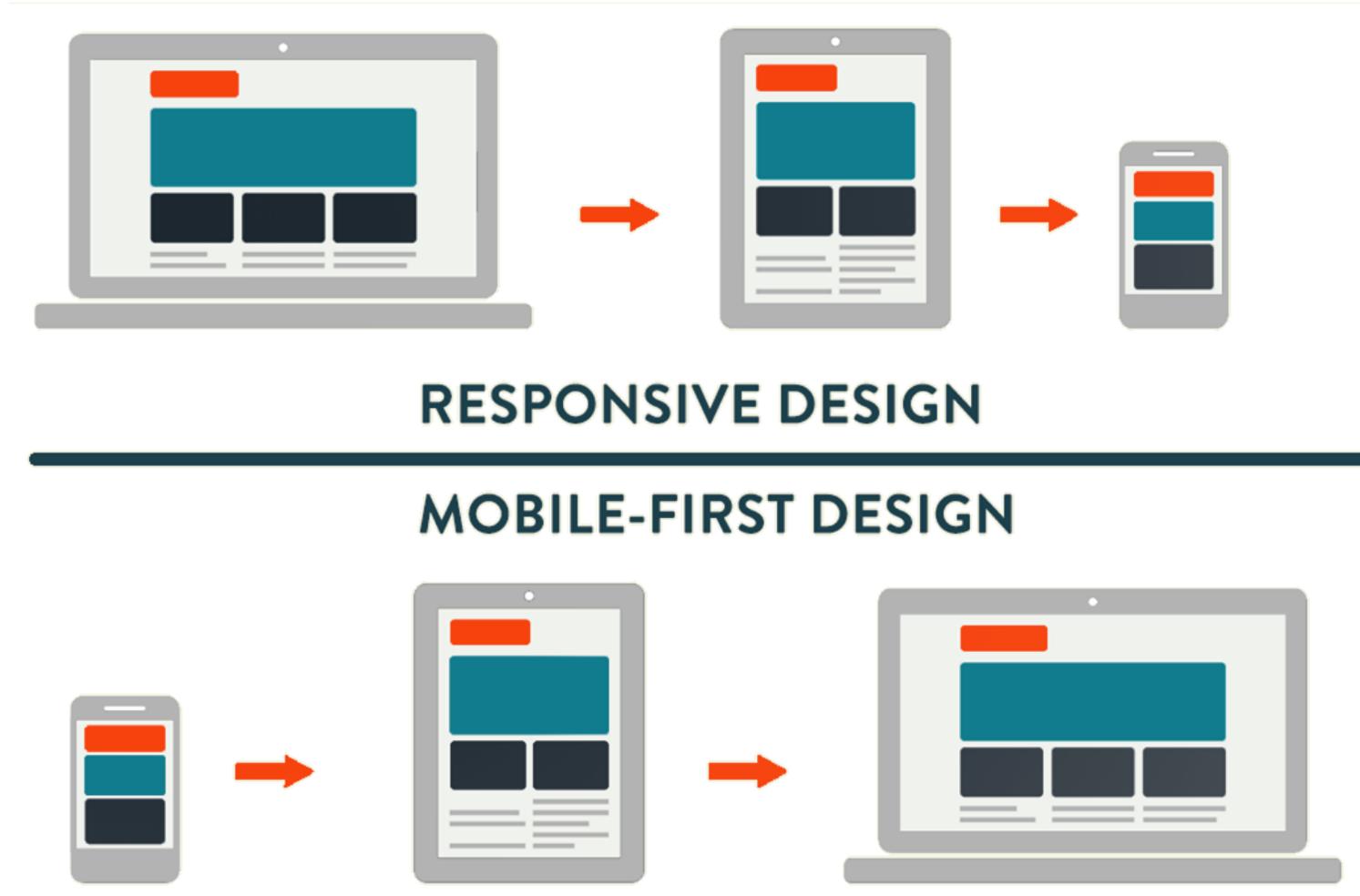


# Nuevos criterios de diseño

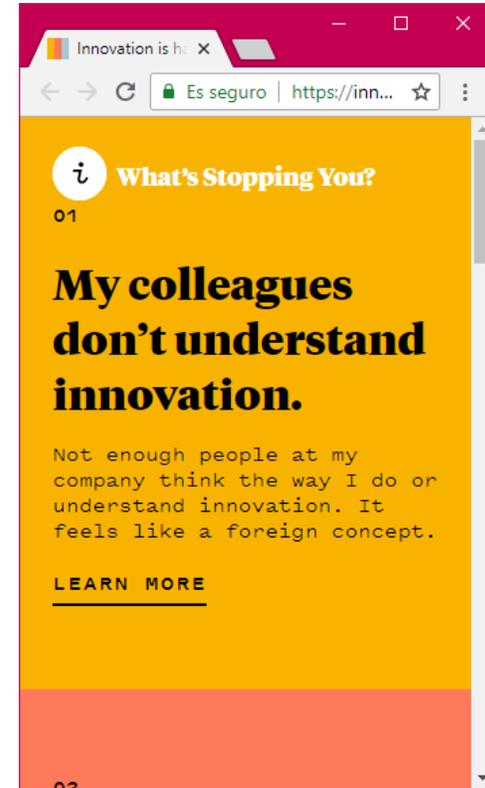
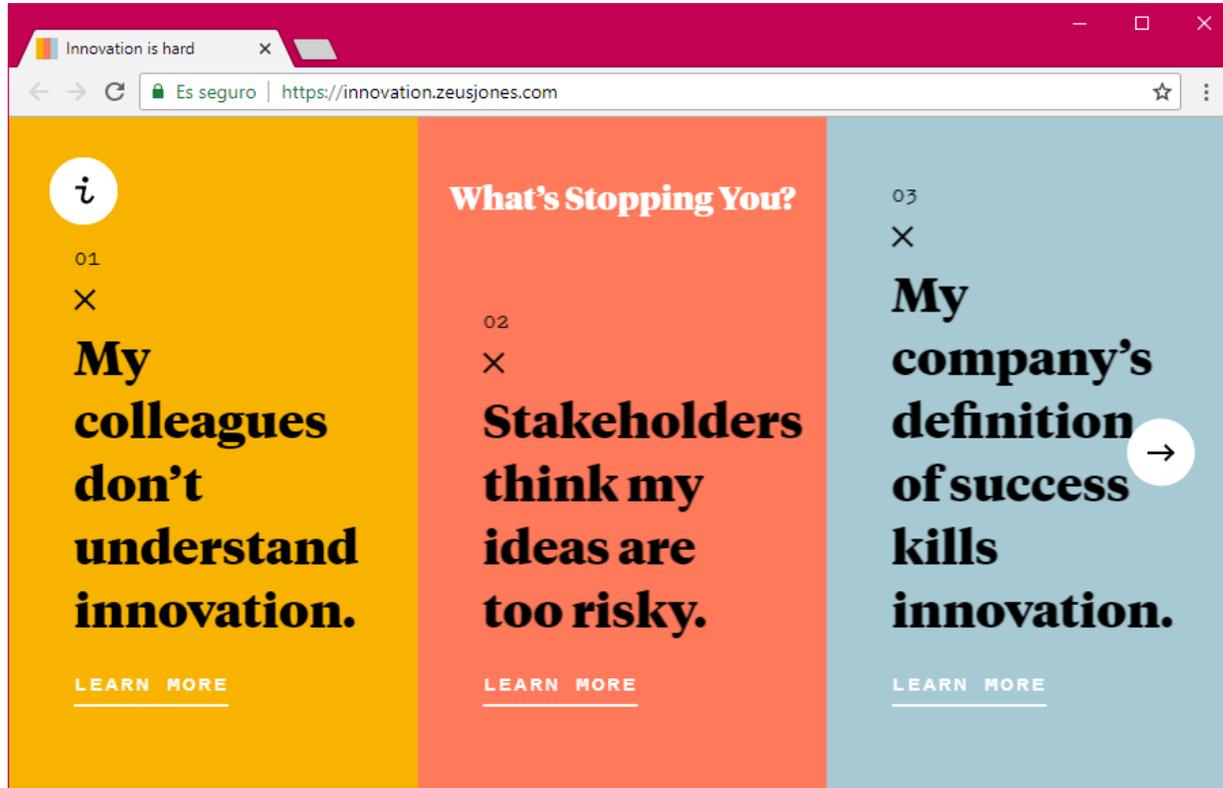


- *Responsive design*: utilizar HTML y CSS para cambiar automáticamente el tamaño, ocultar, reducir o ampliar un sitio web para que se vea bien en todos los dispositivos (PC, tabletas y móviles).
- *Flat design*: diseño minimalista enfocado a la funcionalidad y a la transmisión clara del mensaje.
- *Mobile first*: se centra en la realización del diseño inicial para móvil. De esta forma, es más fácil centrarse en los elementos, las acciones y la información más importantes de un sitio web.

# Responsive design vs. Mobile first

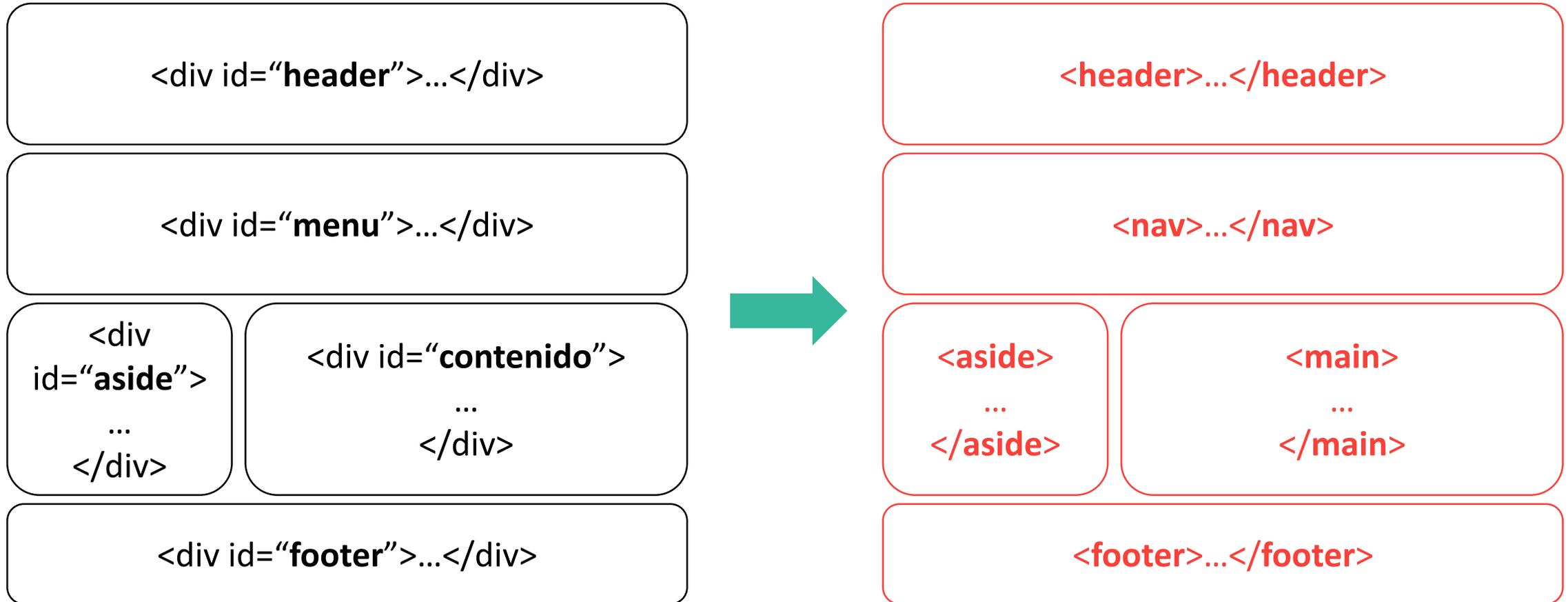


# Flat design



<https://innovation.zeusjones.com/>

# Nuevas funcionalidades, avances tecnológicos y demanda



# Elementos semánticos más utilizados

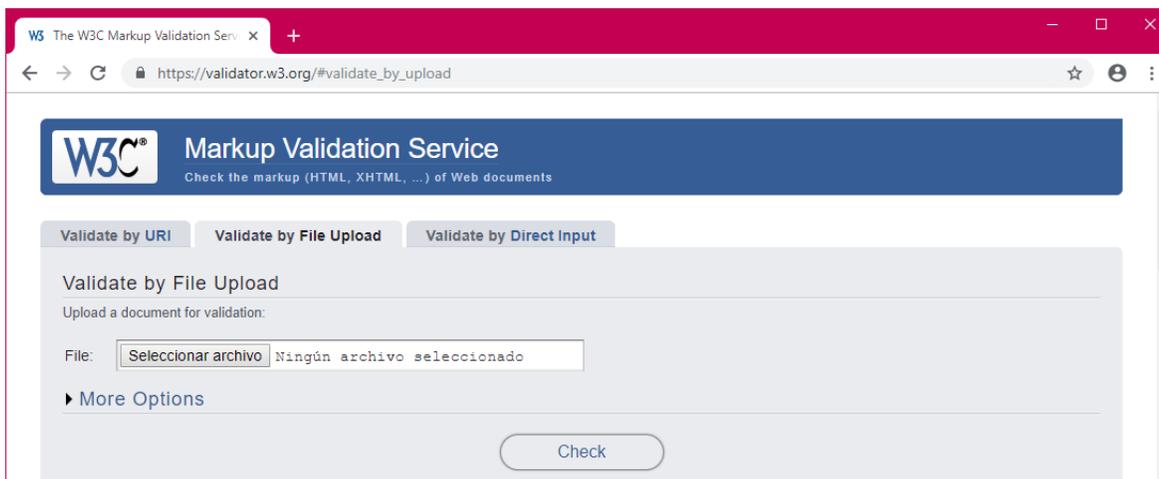


ELEMENTO	DESCRIPCIÓN
<b>&lt;article&gt;</b>	Define un artículo en el documento. Suele contener título y párrafos.
<b>&lt;aside&gt;</b>	Define contenido relacionado con el principal, pero que no pertenece a él.
<b>&lt;figure&gt;, &lt;figcaption&gt;</b>	Define autocontenido (normalmente contiene una imagen) y su título.
<b>&lt;footer&gt;</b>	Define el pie de un document o sección.
<b>&lt;header&gt;</b>	Define la cabecera de un documento o sección.
<b>&lt;main&gt;</b>	Define el contenido principal del documento.
<b>&lt;nav&gt;</b>	Define enlaces de navegación.
<b>&lt;section&gt;</b>	Define una sección en un documento.

# Validación de HTML5



- Para comprobar que nuestros documentos HTML son correctos (la parte que se puede comprobar automáticamente) el W3C nos provee de un validador.
- El validador nos permite comprobar nuestro documento a través de URL (no lo vamos a poder utilizar para nuestros desarrollos), de subida de archivo o de entrada directa del contenido.



<https://validator.w3.org>

# Tema 8. Implementación del *front-end*. HTML5

DISEÑO Y DESARROLLO DE VIDEOJUEGOS WEB (AD)

Dra. Carolina García Vázquez.

[carolina.garcia@esne.es](mailto:carolina.garcia@esne.es)