

Alexandra Raluca, Savu

# Actividad Evaluable

Lenguaje de Marcas 1º DAM, 1º Evaluación

Alexandra Savu  
7-12-2023

# HTML

La primera actividad evaluable de la asignatura de lenguaje de marcas consiste en una página informativa de unas jornadas de informática con la siguiente estructura: index.html, programación.html e inscripción.html.

## Elementos Comunes

En todos los HTMLs empezamos con la etiqueta `<!DOCTYPE html>` que nos define el tipo de documento que vamos a crear. A continuación, definiremos el lenguaje del documento con una etiqueta de apertura `<html lang="es">` con la que también cerraremos el documento `</html>`.

### <HEAD>

Contiene metadatos, como el título de la página, enlaces a estilos, y otros elementos que no son visibles en el navegador.

En el `<head>` añadimos el `<meta charset="UTF-8">` para especificar la codificación de caracteres del documento. También añadiremos el título que tendrá la página en el navegador. Incluiremos el link al archivo de los estilos css con la etiqueta `<link rel="stylesheet" href="estilos.css">`, aunque hablaremos de esto en más profundidad en el apartado de css. Además, también incluiremos los links que nos ayudarán a añadir los iconos y las fuentes. Esto lo haremos con las etiquetas `<script>` (para los iconos) y un `<link rel="stylesheet">` para las fuentes.

También incluimos `<meta name="viewport" content="width=device-width, initial-scale=1.0">`: que configura la vista en dispositivos móviles. Además, el `<title class="titulo">Inicio</title>` establece el título de la página.

### <BODY>

En las tres páginas se repetirá el encabezado, el menú de navegación y el pie.

El encabezado de la página lo empezamos con la etiqueta `<div>` para señalar la apertura de un elemento en bloque. Dentro de este añadimos el logo de la página con la etiqueta `<img>` y especificaremos el origen de la foto con un `src=""`. A continuación, utilizaremos un `<h2>` para escribir el título de la página y un `<h5>` para el subtítulo. También incluiremos etiquetas `<a>` para los links a las redes sociales.

El menú de navegación lo cree utilizando una lista no ordenada con la ayuda de las etiquetas `<ul>` y `<li>` y las etiqueta `<a>` para vincular las páginas de navegación.

### <FOOTER>

El pie lo añadí con una etiqueta `<footer>` dentro de la cual añadimos una `<div>` seguido de una `<p>` con el texto del pie. Al footer le añadiré las características que lo hace un pie de página con la ayuda de css, que veremos más adelante.

## Índex

En la primera página, después del menú de navegación, tenemos un apartado llamado "II Jornadas de informática" con texto, título y un video. El título de este apartado lo empezamos con un `<h2>`, después de cual vinculamos el video con la ayuda de la etiqueta `<video>` y la atributo `src=""` con el link del video. También añadiremos los atributos `width=""` para regular el tamaño del video y `"controls"` para que el usuario pueda controlar el video. A continuación, añadimos el texto al lado del video con una etiqueta `<p>`. Todo esto englobado en un `<div>`.

El segundo apartado es para informarnos cuando y donde son las jornadas. Lo englobamos todo en un `div` y luego cada apartado en dos `divs` separados. En cada uno utilizamos un `<h3>` como títulos, `<p>` para el texto y `<a>` para los links de las redes sociales.

El último apartado consta de 6 imágenes con un pie de imagen. Estas se han colocado con la etiqueta `<img>` que consta de un `src=""` con el link de la imagen y un `alt=""` con la descripción de la imagen. La posición de las imágenes se ha conseguido con la ayuda de un atributo css `"float"`, del que hablaremos más adelante.

## Programación

En la página de programación primero tenemos una tabla con un título (`<h2>`). La tabla la empezamos con la etiqueta `<table>` dentro de la cual colocaremos el atributo de `<style>` con el cual definiremos el tipo de tabla, su posición y el ancho total. A continuación, definimos el ancho de las diferentes columnas con la ayuda de la etiqueta `<col>`. La tabla tiene varias partes, una con el título y otra con el cuerpo de la tabla. El título tiene una etiqueta `<thead>` y el body una `<tbody>`. Las filas las definimos con la etiqueta `<tr>` y las columnas con la etiqueta `<td>`. Así mismo la fusión de las columnas se define con el atributo `colspan=""` y el de las filas con el `rowspan=""`. También se han añadido dentro de algunas filas iconos con la ayuda de la etiqueta `<i>`.

Al final de la página hay un apartado con la programación del evento, esto se ha hecho con listas anidadas, ordenadas y no ordenadas (como ya hemos visto anteriormente). Las listas se han metido en dos `divs` que se han ordenado con la ayuda del `float` en `css`.

## Inscripción

En la última página tenemos el formulario que el usuario necesita rellenar para poder participar en las jornadas. Lo empezamos con un `<form>` seguido de un `action="" action_page.php` lo que hace que cuando el usuario pulse el botón enviar, se envíe los datos del formulario a un archivo llamado `"action_page.php"` (para procesar la entrada). Hay dos secciones del formulario, uno con los datos personales y otro con el perfil del usuario.

Empezamos primero con los nombres y los apellidos abriendo una etiqueta `<label>` (para el nombre del apartado a rellenar) y otra `<input>` para especificar el tipo de elemento. En el caso de los nombres, hemos utilizado un `<input>` de tipo "text", a los que le hemos añadido un atributo "pattern" para especificar que solo se permite de 2 a 128 letras y un "required" que especifica que el campo es obligatorio. Hacemos lo mismo con los campos del teléfono y el email, pero esta vez con un "type" de tipo "tel" y "email", que a su vez son un campo obligatorio. El último apartado de los datos personales es un elemento de subida de fichero, que en este caso se le pide al usuario que suba una foto de perfil. Lo conseguimos con un `<input>` de tipo "file".

En el segundo apartado del perfil se pide escribir por qué uno quiere atender a las jornadas. Empezamos con un `<input>` de tipo "text" y un "required" porque es un apartado obligatorio. También le pusimos una altura de 100px (con un atributo `height=""`) ya que es un apartado en el que se puede escribir mucho texto si se desea.

El siguiente apartado es de tipo `<radio>` para escoger el tipo de profesion del usuario, seguido de un apartado tipo `<checkbox>` para escoger los talleres a los que se atenderá.

Como último apartado, tenemos uno de tipo 'select' para escoger las fechas deseadas y otro para aceptar la política de privacidad de tipo "checkbox" y obligatorio.

Por último, tenemos el botón de enviar `<input>` de tipo "submit", al cual mejoraremos el aspecto con css.

## CSS

Para personalizar nuestra página web hemos utilizado un archivo css que hemos vinculado con nuestras páginas a través de `<link rel="stylesheet" href="estilos.css">`.

Para poder personalizar nuestros elementos html tenemos que utilizar diferentes técnicas para vincularlos con sus respectivos atributos css. Para hacer eso, hemos utilizado selectores de elemento html (ej. `h1`, `div`, `img...`), selectores de clase (ej. `.encabezado`, `título...`) y también selectores de id (`#jornadas`, `#periodismo`).

Empezamos nuestras reglas, por ejemplo, con un selector de clase `".título"` seguido de `"{"`, dentro de cual pondremos nuestra propiedad (ej. `color`) seguido del valor asociado a esta propiedad (ej. `rojo`) y cerrando con un `";"` y finalmente un `"}"`. También podemos mezclar selectores por orden de importancia, desde un selector de html pasando por un selector de clase y acabando con el más importante, un selector de id.

## Colores

Para dar un color específico a un elemento podemos utilizar varias propiedades, como por ejemplo "color:", "background-color:" (para el fondo) o "Font-color:" (para las fuentes). Al igual que "border-color" para definir el color de los bordes de cualquier elemento. Seguido pondremos el color, utilizando simplemente el nombre de un color (red), hexadecimal (#A9A9A9) o modelo RGB (rgb (241, 121, 121))

## Fuentes

Para poder declarar una fuente para los textos de nuestra página tenemos que primero definir una familia de fuentes en nuestro archivo html como ya hemos comentado anteriormente en el apartado de HEAD. Posteriormente, en nuestro archivo utilizaremos la propiedad "Font-family:" seguido de los nombres de las fuentes. También utilizaremos "Font-size" para definir el tamaño del texto o también "text-align:" para definir hacia donde irá orientado el texto.

## Bloques

Para dar características a los bloques o divs, hemos utilizado las siguientes propiedades: width (ancho), padding o margin (para darle características al borde o al espaciado interno). También se puede especificar en qué lado del bloque se puede dar dichas características (padding-top, margin-bottom).

Otra característica que he utilizado ha sido el "display" para decidir si un elemento es tratado como un bloque o un inline; y también el "position" para definir la posición final del bloque. Y como no, el "width" y "height" para definir el tamaño.

## Float

Ya que los bloques son elementos en línea, tenemos que utilizar la propiedad "float" para poder colocar dos elementos uno al lado de otro. Si queremos colocar un elemento a la derecha tenemos que hacer un "float: right" y si no queremos que al lado esté otro elemento, haremos un "clear: left". Por último, si queremos que el contenedor/bloque no tenga espacios blancos libres, le podemos añadir un "overflow: hidden".

# Validación

### Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

**Showing results for index.html**

Checker Input

Show ☐ source ☐ outline ☐ image report

Check by   No file chosen

Uploaded files with .xhtml or .xht extensions are parsed using the XML parser.

**Document checking completed. No errors or warnings to show.**

Used the HTML parser.  
Total execution time 18 milliseconds.

[About this checker](#) • [Report an issue](#) • Version: 23.11.28

### Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

**Showing results for programación.html**

Checker Input

Show ☐ source ☐ outline ☐ image report

Check by   No file chosen

Uploaded files with .xhtml or .xht extensions are parsed using the XML parser.

**Document checking completed. No errors or warnings to show.**

Used the HTML parser.  
Total execution time 14 milliseconds.

[About this checker](#) • [Report an issue](#) • Version: 23.11.28

### Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

**Showing results for inscripción.html**

Checker Input

Show ☐ source ☐ outline ☐ image report


Check by   No file chosen

Uploaded files with .xhtml or .xht extensions are parsed using the XML parser.

**Document checking completed. No errors or warnings to show.**

Used the HTML parser.  
Total execution time 19 milliseconds.

[About this checker](#) • [Report an issue](#) • Version: 23.11.28



## The W3C CSS Validation Service

W3C CSS Validator results for estilos.css (CSS level 3 + SVG)


Jump to: [Validated CSS](#)

W3C CSS Validator results for estilos.css (CSS level 3 + SVG)


**Congratulations! No Error Found.**

This document validates as [CSS level 3 + SVG](#) !

To show your readers that you've taken the care to create an interoperable Web page, you may display this icon on any page that validates. Here is the XHTML you could use to add this icon to your Web page:




```
<p>
<a href="http://jigsaw.w3.org/css-validator/check/referer">
  
</a>
</p>
```



```
<p>
<a href="http://jigsaw.w3.org/css-validator/check/referer">
  
</a>
</p>
```

(close the img tag with > instead of /> if using HTML <= 4.01)



Interested in "developing" your developer skills? In W3C's hands-on Professional Certificate Program, learn how to code the right way by creating Web sites and apps that use the latest Web standards. [Find out more!](#)