




# Diseño de Interfaces Web

DIW - Tarea 08

Contenidos web interactivos

Asier García Aristu

21/05/2021



**Proceso:**

Partiendo de la imagen proporcionada para la tarea, primero hago una **estructura** rápida del formulario que hay que crear:

```
<body>
  <form>
    <fieldset>
      <input type="text" />
      <input type="text" />
      <input type="text" />
      <input type="text" />
      <input type="file" />
    </fieldset>
    <fieldset>
      <textarea></textarea>
      <select>
        <option></option> x5
      </select>
      <select multiple>
        <option></option> x7
      </select>
      <input type="radio" /> x3
    </fieldset>
    <input type="checkbox" />
    <input type="submit" />
  </form>
</body>
```

**Inscripción a las X Jornadas Matemáticas**

Los campos señalados con \* són obligatorios

**Datos personales**

Nombre \*
Apellidos \*

Correo electrónico \*
Teléfono \*

Fotografía para la acreditación

Examinar... No se ha seleccionado ningún archivo.

**Perfil de participante**

¿Por qué está interesado en participar? \*

Actividad profesional \*

Selecione una activid...

¿En qué talleres está interesado? \*

Para seleccionar más de una opción, mantened pulsada la tecla Control (Windows) / Command (Mac) mientras hacéis clic a las opciones

Selecione uno o más
La encriptación a lo la
Seguridad informática
Redes sociales y teoría
Matemáticas y azar
Problemas matemáticos
La predicción del tiempo
Fractales en el aula

Selecione una de las tres opciones siguientes: \*

☐ Asistiré los días 4 y 5 de mayo
☐ Asistiré solo el día 4 de mayo
☐ Asistiré solo el día 5 de mayo

☐ He leído y acepto la política de privacidad y el aviso legal \*

ENVIAR

Con la estructura ya creada, le añado la **declaración** del documento para un archivo **XHTML Strict**, y le añado también la información necesaria en el elemento **<head>**, que de momento es la codificación, el título y el **enlace al archivo css**:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8;" />
  <title>DIW Tarea 08</title>
  <link rel="stylesheet" type="text/css" href="css/style.css"/>
</head>
```

En el primer **fieldset**, para los cuatro primeros **input** utilizo en **tipo text**, y a cada uno le añado previamente un elemento **label** para identificar cada campo:

```
<label for="nombre">Nombre<span class="asterisco">*</span></label>
<input type="text" name="nombre" id="nombre" />
```

Este es el **ejemplo del primer input**, el del nombre, y los otros tres son similares. En label utilizamos el atributo **for** para indicar que es la etiqueta del input nombre.

El **asterisco**, como indica el enunciado que tiene que ir en otro color, lo he metido dentro de un elemento **<span>** con la clase **asterisco**, que en el **css** indico que aparezca de color rojo.

Al tener que hacer un documento XHTML, **no puedo utilizar ciertas mejoras que tiene HTML5** para los formularios, como por ejemplo en este caso utilizar dos tipos diferentes de input, el tipo **email** y el tipo **tel**.

Además, para validar el formulario, tampoco podemos utilizar otros avances como por ejemplo el atributo **required** para los campos obligatorios, el atributo **pattern** para indicar un patrón por ejemplo en el input del teléfono, o el atributo **placeholder** para indicar alguna indicación en los input.

A continuación se pueden ver las diferencias que acabo de comentar de los campos de email y teléfono de un tipo de documento a otro:

#### HTML5

```
<label for="email">Correo electrónico<span class="asterisco">*</span></label>
<input type="email" name="email" id="email" required>

<label for="telefono">Teléfono<span class="asterisco">*</span></label>
<input type="tel" name="telefono" id="telefono" placeholder="666666666"
pattern="[0-9]{9}" required>
```

#### XHTML

```
<label for="email">Correo electrónico<span class="asterisco">*</span></label>
<input type="text" name="email" id="email" />

<label for="telefono">Teléfono<span class="asterisco">*</span></label>
<input type="text" name="telefono" id="telefono" />
```

Por lo tanto, como tenemos que utilizar un documento XHTML, para las **validaciones** he creado un archivo **javascript** que se llama **validar.js**, que lo he enlazado en el elemento **head** a través de un elemento **<script>**.

```
<script type="text/javascript" src="validar.js"></script>
```

Para terminar con los input del primer fieldset, queda el **tipo file**, al que le he añadido el **atributo accept** indicando que solo acepte archivos que sean imágenes:

```
<label for="foto">Fotografía para la acreditación</label>
<input type="file" id="foto" name="foto" accept="image/png,image/gif,image/jpeg" />
```

Y para cerrar del todo el fieldset, le he añadido el nombre mediante un **elemento <legend>** como primer hijo del elemento <fieldset>.

```
<legend>Datos personales</legend>
```

Para ajustar el diseño mediante css, he incluido **cada input en un elemento <div>**, y todos estos los he añadido a otro <div>.

En el segundo fieldset, he seguido los mismos pasos, por lo que tan solo comentaré las diferencias, que en este caso son los elementos del formulario.

Para el primer campo he utilizado el elemento **textarea**, en el que le he marcado un tamaño de 7 filas por 30 columnas. Este tamaño luego lo he variado, ya que el css le indicará que ocupe el ancho del fieldset.

Además, le he añadido también mediante css, con **resize:vertical**, la opción de que se pueda modificar el tamaño del textarea, pero solamente en vertical.

```
<label for="interes">¿Por qué está interesado en participar?<span
class="asterisco">*</span></label>
<textarea name="interes" id="interes" rows="7" cols="30" ></textarea>
```

Para la validación, le he añadido un límite **mínimo de longitud** al texto requerido en este campo.

He considerado oportuno que sean **50 caracteres** (sin tener en cuenta espacios).

```
<p>Longitud mínima 50 caracteres (<span id="intereslong">0</span>).</p>
```

He incluido también el **número de caracteres actuales** para que el usuario sepa en todo momento cuantos lleva:

- Si el número es **inferior a 50**, el texto aparecerá de color **rojo**, indicando que no es correcto.
- Si el número es **50 o mayor**, el texto cambiará a color **verde**.

Esto lo he hecho mediante javascript. He creado un evento cada vez que se pulse una tecla en el textarea, y este evento lo que hace es contar los caracteres (sin incluir espacios) que hay dentro del área, y modificar el atributo style del elemento <p> para modificar el color de la fuente.

El siguiente campo, he utilizado un **elemento <select>**, al que le he añadido las opciones seleccionables mediante un **elemento <option>** para cada opción.

```
<label for="profesion">Actividad profesional<span
class="asterisco">*</span></label>
<select name="profesion" id="profesion" >
  <option value="Estudiante_de_bachillerato">Estudiante de bachillerato</option>
  <option value="Estudiante_universitario">Estudiante universitario</option>
  <option value="Profesor_de_bachillerato">Profesor de bachillerato</option>
  <option value="Profesor_universitario">Profesor universitario</option>
  <option value="Otra_actividad_profesional">Otra actividad profesional</option>
</select>
```

Este elemento creará un **campo desplegable** con las opciones indicadas.

Los elementos <option> deben llevar un **atributo value**, que indica el valor de cada uno de los elementos. A diferencia de los campos anteriores, en los que el valor era el texto añadido por el usuario, en este caso el usuario no añade ningún texto, sino que selecciona una de las opciones establecidas.

El siguiente campo, es también un campo **select**, con la diferencia que en este caso se podrá seleccionar más de una opción, al contrario que el campo anterior en el que tan solo se puede seleccionar una.

Para conseguir esto le añado al elemento select el **atributo multiple="multiple"**.

```
<label for="talleres">¿En qué talleres está interesado?<span
class="asterisco">*</span></label>
<p>Para seleccionar más de una opción, mantened pulsada la tecla Control (Windows)
/ Command (Mac) mientras haceis click a las opciones.</p>
<select name="talleres" id="talleres" size="7" multiple="multiple" >
  <option value="La_encryptación_a_lo_largo_de_la_historia">La encriptación a lo
largo de la historia</option>
  <option value="Seguridad_informática">Seguridad informática</option>
  <option value="Redes_sociales_y_teoría_de_grafos">Redes sociales y teoría de
grafos</option>
  <option value="Matemáticas_y_azar">Matemáticas y azar</option>
  <option value="Problemas_matemáticos_del_milenio">Problemas matemáticos del
milenio</option>
  <option value="La_predicción_del_tiempo">La predicción del tiempo</option>
  <option value="Fractales_en_el_aula">Fractales en el aula</option>
</select>
```

Además le he ajustado el tamaño con **size="7"**, para que muestre todos los elementos seleccionables, que en este caso son 7.

Al no ser muchos elementos, he considerado esta opción como la más acertada, ya que serán visibles todas las opciones y no habrá necesidad de hacer scroll.

El siguiente campo, la asistencia, consta de **tres elementos input del tipo radio**. Estos, al tener el mismo nombre, permiten únicamente seleccionar uno de ellos, por lo que, si estando uno seleccionado se pulsa otro, la selección cambiará.

```
<p>Seleccione una de las tres opciones siguientes:<span
class="asterisco">*</span></p>

<input type="radio" id="d4y5" name="asistencia" value="Días_4_y_5_de_mayo" />
<label for="d4y5">Asistiré los días 4 y 5 de mayo</label>

<input type="radio" id="d4" name="asistencia" value="Día_4_de_mayo" />
<label for="d4">Asistiré solo el día 4 de mayo</label>

<input type="radio" id="d5" name="asistencia" value="Día_5_de_mayo" />
<label for="d5">Asistiré solo el día 5 de mayo</label>
```

Ya fuera de los fieldset quedan dos elementos del formulario.

Primero tenemos un **checkbox** en el que el usuario indicará que ha leído la política de privacidad y el aviso legal.

```
<input type="checkbox" id="legal" name="legal" value="legal" />
<label for="legal">He leído y acepto la política de privacidad y el aviso
legal
    <span class="asterisco">*</span>
</label>
```

Por último está el botón de **submit**, que no tiene mucho misterio, al ser de este tipo, enviará el formulario.

```
<input type="submit" id="submit" value="Enviar" />
```

Y el formulario se enviará a donde se haya indicado en el **atributo action** del elemento form, que en este caso he creado una página de confirmación que no hace nada, pero lo suyo sería que se enviaran a una base de datos.:

```
<form id="formulario" method="post" action="confirmacion.html">
```

Los datos he decidido que se envíen con el **method post**, ya que ciertos datos, como por ejemplo el email o el teléfono, es mejor mandarlos de forma segura, y el método get deja a la vista en la URL los datos.

Con todos estos elementos estaría **completado el formulario**.

Al igual que con los elementos del primer fieldset, estos también los he incluido cada uno dentro de un <div> dándole una clase para así poder crear el estilo más fácilmente.

## Validaciones y envío

Como he comentado antes, **mediante javascript** he añadido validaciones para el formulario, que son las siguientes:

- **Nombre:** no puede estar vacío, ni solo con espacios.
- **Apellidos:** no puede estar vacío, ni solo con espacios.
- **Email:** - No puede estar vacío, ni solo con espacios.  
- Tiene que tener un formato correcto de email.
- **Teléfono:** - No puede estar vacío, ni solo con espacios.  
- No puede contener letras, solamente números.  
- Tiene que tener 9 dígitos
- **Interés en participar:** tiene que contener como mínimo 50 caracteres sin contar espacios.
- **Talleres:** mínimo hay que seleccionar un taller.
- **Asistencia:** hay que seleccionar una de las opciones.
- **Aviso legal y política de privacidad:** hay que aceptarla.

Estas validaciones se van haciendo una a una **en ese orden**, y si una de ellas falla se deja de validar y se informa al usuario de dónde está el error. Además de ello se hace **focus** en el elemento para que sea más claro.

**Si se valida correctamente**, sin cambiar de página, se muestra un resumen de los datos del usuario para que los repase antes de confirmar:

**Confirma que tus datos son correctos:**

- **Nombre:** Asier
- **Apellidos:** García Aristu
- **Correo electrónico:** asier@email.com
- **Teléfono:** 666666666
- **Interés en participar:** Me parecen unos talleres muy interesantes que pueden aportar mucho a mi formación.
- **Actividad profesional:** Otra actividad profesional
- **Talleres que te interesan:** Seguridad informática, Matemáticas y azar, Fractales en el aula.
- **Asistencia:** Días 4 y 5 de mayo

**VOLVER****ENVIAR**

Si hay algún fallo, con el **botón volver** se vuelve a mostrar el formulario para poder modificar cualquier dato. Si se pulsa **enviar**, el formulario se envía y lleva a la página en la que se confirma al usuario que el formulario se ha enviado correctamente.

## Interacción

El elemento de interacción más visible de todos es el **hover o focus-within sobre los fieldset**. Al situar el ratón sobre uno de los fieldset o que un elemento del interior del fieldset tenga el foco, el fondo cambia de color para que el usuario se ubique más fácilmente.

Al hacer **hover sobre los campos email, teléfono y foto** del primer fieldset, aparecen ciertas **indicaciones** que he añadido, ya que podrían ser los campos que puedan requerir este tipo de aclaraciones.

En todos los campos del formulario, al estar el foco en ellos, cambian el fondo de blanco a gris claro, para que así el usuario sepa dónde está ubicado el cursor.

Teléfono\*

Introduce un número de teléfono válido en formato de 9 dígitos.

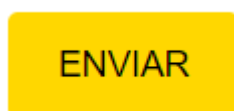
En el segundo fieldset, el **textarea** exige una **longitud mínima** de 50 caracteres. Para que el usuario sepa esto he añadido un **texto explicativo**:

Longitud mínima 50 caracteres (0).

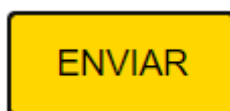
Longitud mínima 50 caracteres (59).

Como se puede ver en las imágenes, entre paréntesis está el número de caracteres actual, que se va modificando según el usuario escribe. Cuando llega a 50 caracteres el texto cambia de color rojo a verde, para indicar así que la longitud ya es suficiente.

El **botón de enviar** también cambia al ubicar el ratón sobre él o al tener el foco si se utiliza la navegación por teclado:



normal



:focus



:hover

## Accesibilidad

He añadido una **ayuda para la navegación accesible**, que se muestra y se oculta pulsando la **tecla esc**. También se muestra haciendo clic en el texto que se ve en la imagen inferior y se puede cerrar pulsando la X.

Como se puede ver también en la imagen, cada campo tiene asociado un acceso mediante las teclas **ctrl + número**. Esto aparece y desaparece junto con la ayuda de navegación por teclado, además de algún texto explicativo para la navegación por teclado en los campos select.

Ocultar ayuda de navegación accesible. (tecla esc)

Apellidos\*

ctrl + 2

Teléfono\*

ctrl + 4

**Navegación por teclado:** X

- Siguiente: tab
- Anterior: shift+tab
- Navegar por selecciones: arriba/abajo
- Selección de campo: ctrl + número
- Escribir por voz el campo actual: F2
- Seleccionar: espacio
- Enviar formulario: intro



El **menú de ayuda** tiene una **posición fixed**, por lo que se mantendrá en la posición que esté aunque se haga scroll, por lo que siempre será visible. Además he hecho que **se pueda mover**, el cursor cambia al situarse sobre el menú para indicarlo. Se puede desplazar a un punto al que al usuario le moleste menos y seguirá permanentemente en la pantalla debido a la posición fixed. Si se cierra el menú y se vuelve a abrir, el menú habrá regresado a su posición inicial.

Si el usuario está habituado a utilizar la navegación por teclado, no necesitará prácticamente la ayuda, ya que a excepción del comando que he añadido de grabación por voz, o los comandos de acceso directo a los campos, lo demás vienen por defecto en el navegador.

Para usuarios que tengan problemas a la hora de escribir con el teclado, he añadido la opción de poder **rellenar los campos que requieran texto mediante el micrófono**.

Nombre\* ctrl + 1

Correo electrónico\* ctrl + 3

El icono del micrófono aparece en cada uno de los campos de inserción de texto, y al hacer **click** se **iniciará el reconocimiento** de voz para el campo correspondiente. En los input de tipo text, al ser textos muy cortos, lo que se recoja del reconocimiento de voz sustituirá al contenido previo.

En el campo de intereses, que es un textarea, cambia un poco el funcionamiento, ya que se pueden hacer varios reconocimientos de voz y las transcripciones se irán poniendo al final del texto existente. Si en el reconocimiento de voz se dice solo la **palabra "borrar"** el texto del textarea se borrará.

Para iniciar la grabación puede hacerse de dos maneras, la primera sería pulsando sobre el **icono del micrófono** y la segunda haciendo click en **F2** teniendo el cursor ubicado en el campo correspondiente. Cuando se inicie el reconocimiento, **el icono del micrófono** rojo tachado **cambiará** a un icono de un micrófono verde y aparecerá un círculo rojo parpadeante.

Nombre\* ctrl + 1

Cuando el micrófono deje de captar sonido, la grabación terminará y volverá a cambiar el icono por el del micrófono rojo tachado y desaparecerá el círculo rojo.

Firefox no soporta esto, por lo que **he hecho que solo aparezca en Chrome** (o navegadores que usen el motor de Chrome como por ejemplo Edge). En el resto de navegadores no aparecerá el icono del micrófono ni tampoco funcionará la tecla F2.

Todo se muestra correctamente en Firefox, pero está más completa y más ajustada para funcionar en Chrome.

## Adición a la web

Aquí he cometido un “fallo”, y es que al haber estado trabajando como si fuera una página nueva, al ser un proyecto tan pequeño no he utilizado sass ni bem como sí se había hecho en la web anterior. Por lo que en la parte de CSS que he añadido a la ya existente, no utilizo las ventajas de SASS ni el nombrado BEM.

He ajustado el SCSS para aprovechar ciertas ventajas de SASS, consiguiendo que quede más ordenado. Y para solucionar un pequeño problema, que era lo que tenía modificando el elemento body, he creado un div en el que he incluido todo al que le he dado el estilo que tenía antes el body. Creo que tras ajustar todo el estilo se mantiene igual, aunque puede que se me haya escapado algo en algún punto.

A parte de eso no he tenido ningún problema. Tan solo he tenido que añadir al documento el header, menú de navegación y footer. Para que valide como XHTML Strict he tenido que quitar de los links de las redes sociales ubicados en el footer, los atributos target que no me validarán.

Como en la web que habíamos hecho hasta ahora, en el menú de navegación había un elemento “preprocesadores” que no llevaba a ninguna parte, he decidido cambiarle el nombre por “Jornadas Matemáticas” y así completar la web.

El color púrpura no me convence como queda, pero como el enunciado dice que debe ser así, he intentado que el resto de colores (hovers y focus) se adapten lo mejor posible, y no quede un diseño muy “cantoso”. He jugado con colores morados más claros que vayan bien con el púrpura, y con grises, que a parte de ir bien con los morados, ya los había usado en la web, por lo que se consigue algo de uniformidad.

He añadido una imagen sobre el formulario, como sucede en las otras páginas de la web, y así conseguir más uniformidad. He querido hacerlo con el elemento <picture> como vimos en unidades anteriores, pero no me pasaba la validación de XHTML Strict.

Al formulario, para que se envíe correctamente en Netlify le he tenido que añadir tres atributos: **name**, **data-netlify** y **enctype**.

```
<form id="formulario"
  method="post"
  action="confirmacion.html"
  name="formTarea08"
  data-netlify="true"
  enctype="application/x-www-form-urlencoded"
>
```

Si no le añadía estos atributos, al hacer submit, me decía que la página no se ha encontrado.

Los atributos **name** y **data-netlify** no pasan tampoco la validación de XHTML Strict, pero los tenía que dejar para la funcionalidad del formulario al subirlo a netlify. Por lo que luego en la validación, los quitaré para pasarla y demostrar que el documento valida correctamente. Lo suyo sería cambiar de tipo de documento, pero como este es el exigido, hay que adaptarse.


## Validación

Tras estos ajustes (y alguno más pequeño que seguramente haya pasado por alto) y tras añadir el estilo (aunque lo del color púrpura que indica el enunciado no me termina de convencer), he pasado los documentos por el **validador de (X)HTML y de CSS del W3C**.

Con el archivo CSS no he tenido ningún problema y me lo valida correctamente.

Con el XHTML, como he comentado antes, validando el archivo tal cual está, me da los errores por los atributos **name** y **data-netlify** que contiene el elemento form.



Estos errores se pueden ver en la siguiente captura:


**Markup Validation Service**  
Check the markup (HTML, XHTML, ...) of Web documents

**Errors found while checking this document as XHTML 1.0 Strict!**

|                        |   |                          |
|------------------------|---|--------------------------|
| <b>Result:</b>         | 2 Errors  |                          |
| <b>File :</b>          | <input type="button" value="Seleccionar archivo"/> Ningún archivo seleccionado<br><small>Use the file selection box above if you wish to re-validate the uploaded file <i>Jornadas.html</i></small> |                          |
| <b>Encoding :</b>      | utf-8   | (detect automatically) ▼ |
| <b>Doctype :</b>       | XHTML 1.0 Strict  | (detect automatically) ▼ |
| <b>Root Element:</b>   | html  |                          |
| <b>Root Namespace:</b> | <a href="http://www.w3.org/1999/xhtml">http://www.w3.org/1999/xhtml</a>   |                          |

### Validation Output: 2 Errors


**Line 63, Column 73: there is no attribute "name"**




```
...post" action="confirmacion.html" name="formTarea08" data-netlify="true" enctype=...
```

You have used the attribute named above in your document, but the document type you are using does not support that attribute for this element. This error is often caused by incorrect use of the "Strict" document type with a document that uses frames (e.g. you must use the "Transitional" document type to get the "target" attribute), or by using vendor proprietary extensions such as "marginheight" (this is usually fixed by using CSS to achieve the desired effect instead).

This error may also result if the element itself is not supported in the document type you are using, as an undefined element will have no supported attributes; in this case, see the [element-undefined](#) error message for further information.

How to fix: check the spelling and case of the element and attribute, (Remember XHTML is all lower-case) and/or check that they are both allowed in the chosen document type, and/or use CSS instead of this attribute. If you received this error when using the <embed> element to incorporate flash media in a Web page, see the [FAQ item on valid flash](#).

---


**Line 63, Column 100: there is no attribute "data-netlify"**


```
...html" name="formTarea08" data-netlify="true" enctype="application/x-www-form-u...
```


You have used the attribute named above in your document, but the document type you are using does not support that attribute for this element. This error is often caused by incorrect use of the "Strict" document type with a document that uses frames (e.g. you must use the "Transitional" document type to get the "target" attribute), or by using vendor proprietary extensions such as "marginheight" (this is usually fixed by using CSS to achieve the desired effect instead).

This error may also result if the element itself is not supported in the document type you are using, as an undefined element will have no supported attributes; in this case, see the [element-undefined](#) error message for further information.

How to fix: check the spelling and case of the element and attribute, (Remember XHTML is all lower-case) and/or check that they are both allowed in the chosen document type, and/or use CSS instead of this attribute. If you received this error when using the <embed> element to incorporate flash media in a Web page, see the [FAQ item on valid flash](#).

Como esos atributos solamente los he incluido porque son necesarios para que el submit funcione en netlify, los he quitado para realizar la validación. De esta manera ambos documentos validan correctamente:

- Validación del **XHTML Strict** (sin):


**Markup Validation Service**  
 Check the markup (HTML, XHTML, ...) of Web documents

---

**Jump To:**    [Congratulations](#) · [Icons](#)

---

**This document was successfully checked as XHTML 1.0 Strict!**

|                        |  |  |  |
|------------------------|--|--|--|
| <b>Result:</b>         | Passed   |  |  |
| <b>File :</b>          | <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">             Seleccionar archivo           </div> <span>Ningún archivo seleccionado</span><br><small>Use the file selection box above if you wish to re-validate the uploaded file<br/>Tarea08DIW_XHTML.html</small> |  |  |
| <b>Encoding :</b>      | utf-8  | <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">             (detect automatically)           </div> |  |
| <b>Doctype :</b>       | XHTML 1.0 Strict   | <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">             (detect automatically)           </div> |  |
| <b>Root Element:</b>   | html   |  |  |
| <b>Root Namespace:</b> | <a href="http://www.w3.org/1999/xhtml">http://www.w3.org/1999/xhtml</a>  |  |  |

- Validación del **CSS**:


**El Servicio de Validación de CSS del W3C**  
 Resultados del Validador CSS del W3C para style.css (CSS versión 3 + SVG)

---

**Ir a:**    [Su Hoja de Estilo validada](#)

---

## Resultados del Validador CSS del W3C para style.css (CSS versión 3 + SVG)

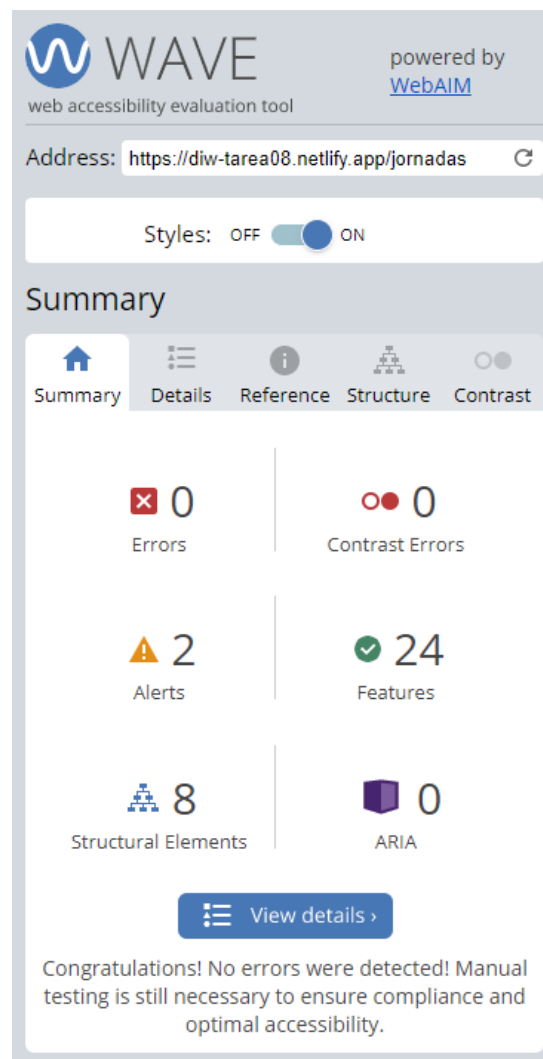
**¡Enhorabuena! No error encontrado.**

¡Este documento es [CSS versión 3 + SVG](#) válido!

Puede mostrar este icono en cualquier página que valide para que los usuarios vean que se ha preocupado por crear una página Web interoperable. A continuación se encuentra el XHTML que puede usar para añadir el icono a su página Web:

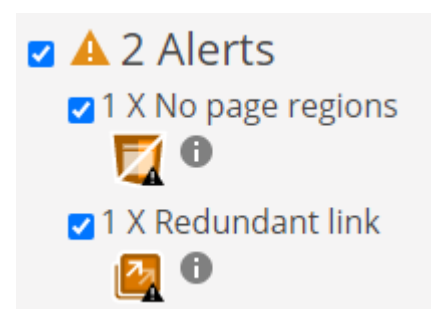
```
<img alt="W3C CSS Validation Service icon" data-bbox="292 858 319 870" />
```

Respecto al **validador de accesibilidad**, achecker dejó de dar servicio el 30 de abril de este año, por lo que he utilizado la otra opción indicada en la tarea, la **herramienta WAVE**. Tras desplegar el sitio con netlify, he pasado la URL de mi sitio por el validador, y cumple correctamente con los requisitos de accesibilidad:



Estas son las dos alertas que me indica:

- **“No page regions”**: El XHTML Strict no acepta el atributo **role**, por lo que me exigiría cambiar de tipo de documento. Si fuera un error y no una alerta me habría planteado el cambiar.
- **“Redundant link”**: El header es un link a la página index, al igual que el primer elemento del menú de navegación, que también lleva a la misma página.



**Links para la corrección**

- Repositorio en Github:

[https://github.com/Reisgaar/DIW\\_Tarea08](https://github.com/Reisgaar/DIW_Tarea08)

- Deployment en Netlify:

<https://diw-tarea08.netlify.app/jornadas>

- Análisis de accesibilidad en WAVE:

<https://wave.webaim.org/report#/https://diw-tarea08.netlify.app/jornadas>

- Análisis del documento W3C:

<https://validator.w3.org/check?uri=https%3A%2F%2Fdiw-tarea08.netlify.app%2Fjornadas&charset=%28detect+automatically%29&doctype=Inline&group=0>

- Análisis del CSS en W3C:

<https://jigsaw.w3.org/css-validator/validator?uri=https%3A%2F%2Fdiw-tarea08.netlify.app%2Fjornadas&profile=css3svg&usermedium=all&warning=1&vextwarning=&lang=es>