

Software y estándares para la Web

P0.PAUTAS DE TRABAJO

Contenido

1	Objetivos	2
2	Entorno de trabajo para la asignatura	2
2.1	Editores	2
2.2	Navegadores.....	4
2.3	XAMPP	6
2.4	Cliente SSH-SFTP – Bitvise	7
3	Validador de Lenguajes de marcas https://validator.w3.org/ y validador para HTML5 https://validator.w3.org/nu	8
4	Validador de CSS https://jigsaw.w3.org/css-validator/	9
5	Herramientas de adaptabilidad web.....	12
5.1	Google Mobile-friendly – https://search.google.com/test/mobile-friendly	12
5.2	Screenfly - https://bluetree.ai/screenfly/	13
6	Herramientas de Accesibilidad Web	15
6.1	Herramienta TAW - http://www.tawdis.net/	15
6.2	Herramienta Wave - http://wave.webaim.org/	17
7	Pautas de trabajo generales en la asignatura	18
8	Creación del primer sitio web simple.....	18
8.1	Ejercicio 1. Crear el documento index.html mínimo válido	18
8.2	Guía para resolver el ejercicio 1.....	19
8.3	Ejercicio 2. Crear una hoja de estilo estilo.css	20
8.4	Guía para resolver el ejercicio 2.....	20
8.5	Ejercicio 3. Aplicar la hoja de estilo al documento HTML5	21
8.6	Guía para resolver el ejercicio 3.....	22
8.7	Ejercicio 6. Incorporar los iconos de cumplimiento de los estándares HTML y CSS ...	22
8.8	Guía para resolver el ejercicio 6.....	22
8.9	Ejercicio 4. Comprobar la adaptabilidad del sitio web.....	23
8.10	Guía para resolver el ejercicio 4.....	23
8.11	Ejercicio 5. Comprobar la accesibilidad del sitio web	23
8.12	Guía para resolver el ejercicio 5.....	23
9	ANEXO: Github Pages.....	24
9.1	Creación del repositorio SEW en Github.....	24
9.2	Cargar archivos.....	26
9.3	Crear Pages.....	28

NOTA: Descargar el archivo “PO.Recursos.zip” con los códigos e imágenes. Los códigos se proporcionan para evitar los problemas de caracteres extraños al copiar y pegar desde el PDF.

1 Objetivos

En esta práctica se establecen las pautas y normas que deben **seguir todas las prácticas de la asignatura en cuanto al cumplimiento de los estándares y a la comprobación de la accesibilidad y usabilidad.**

Se realiza la configuración del entorno de trabajo del estudiante para el desarrollo de la asignatura

2 Entorno de trabajo para la asignatura

Las herramientas que vamos a utilizar en la asignatura son:

- Editores de texto: Para editar el código de los estándares y lenguajes vamos a utilizar fundamentalmente el editor Visual Studio Code. Alternativas NotePad++ y Brackets (sin mantenimiento de Adobe)
- Navegadores: Para visualizar los sitios web. Se utilizarán los navegadores más comerciales en la actualidad, Chrome, Firefox, Microsoft Edge y Opera. Para sistemas macOS también Safari.
- Cliente SSH-SFTP: para conectarse a otras máquinas, Bitvise en Windows. Alternativas para Mac OS <https://apple.stackexchange.com/questions/25661/whats-a-good-graphical-sftp-utility-for-os-x>
- Entorno XAMPP: para simular un servidor APACHE con PHP y MySQL.

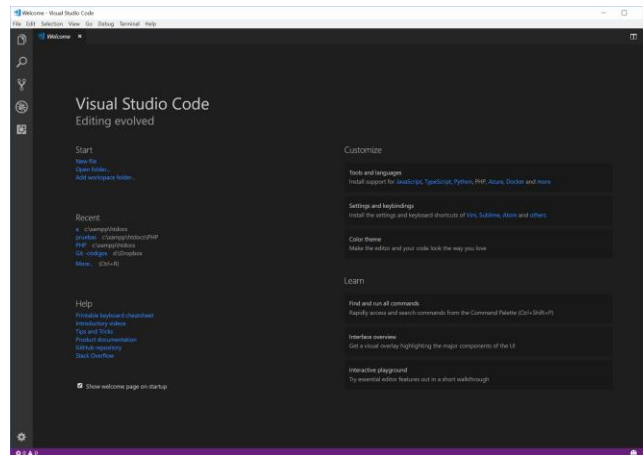
2.1 Editores

2.1.1 Visual Studio Code



Visual Studio Code (<https://code.visualstudio.com/>) es un editor de código fuente desarrollado por Microsoft para Windows, Linux y Mac OS.

Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código.

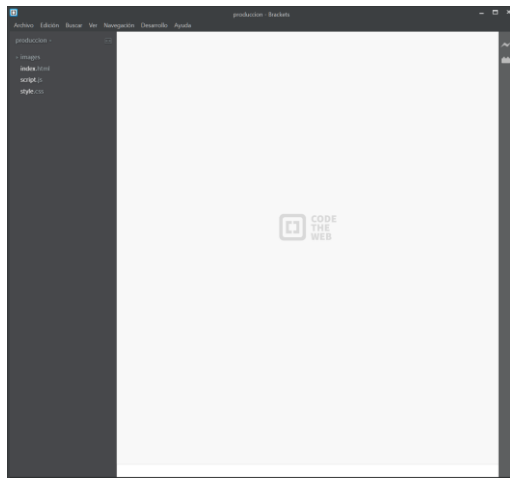


2.1.2 Brackets



Para editar el código de los estándares y lenguajes vamos a utilizar el editor Brackets (<http://brackets.io/>), es un editor open-source escrito en HTML, CSS y JavaScript creado por Adobe Systems bajo licencia MIT License y mantenido a través de GitHub. Tiene versiones para Mac, Windows y Linux. Actualmente sin mantenimiento de Adobe.

Brackets utiliza codificación en UTF-8.

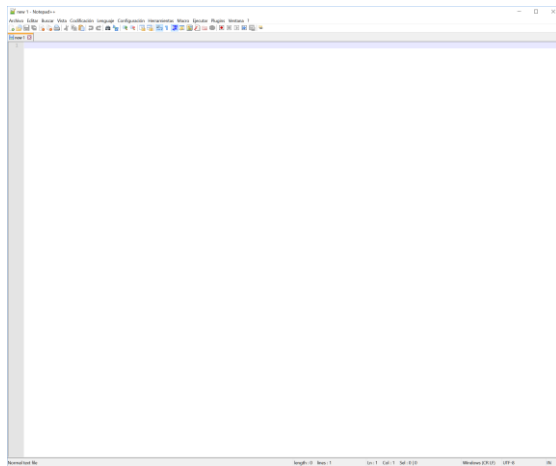


2.1.3 Editor Notepad++



Editor de código distribuido como software libre, hospedado inicialmente en SourceForge.net y desde 2015 en GitHub.

NotePad++ utiliza diversas codificaciones de los archivos, en el caso de esta asignatura se debe seleccionar codificación UTF8 sin BOM.

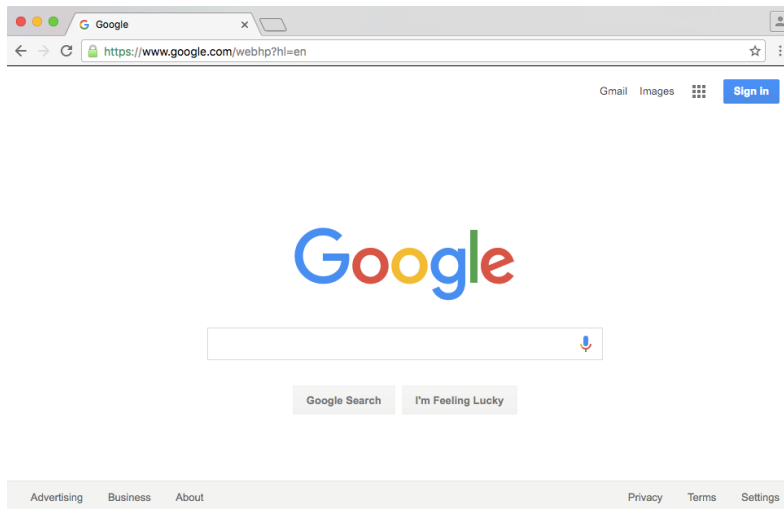


2.2 Navegadores

Vamos a utilizar varios navegadores en la asignatura, siempre deberemos comprobar la visualización de los sitios web en todos ellos.

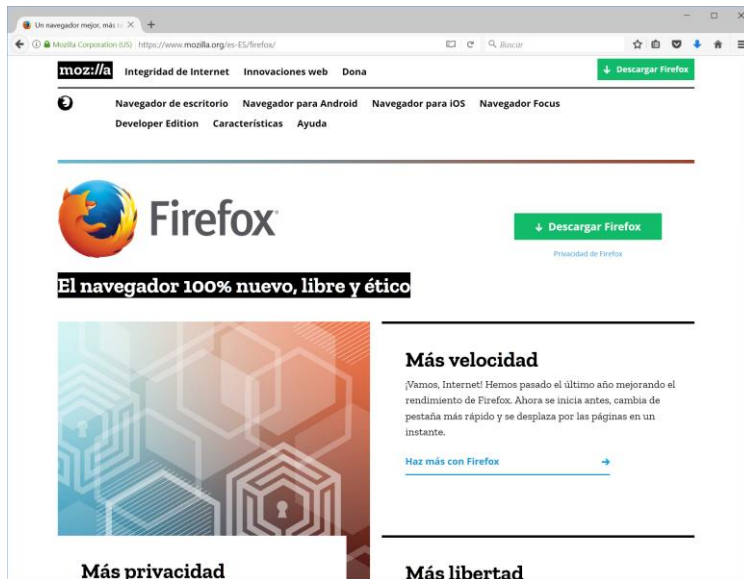
2.2.1 Chrome

Navegador web desarrollado por Google, fecha del lanzamiento Inicial 2 de septiembre de 2008. Versiones para Windows, macOS, Linux, Android y iOS. Desarrollado en C++, Python y JavaScript. Motor de renderizado Blink. Basado en Chromium.



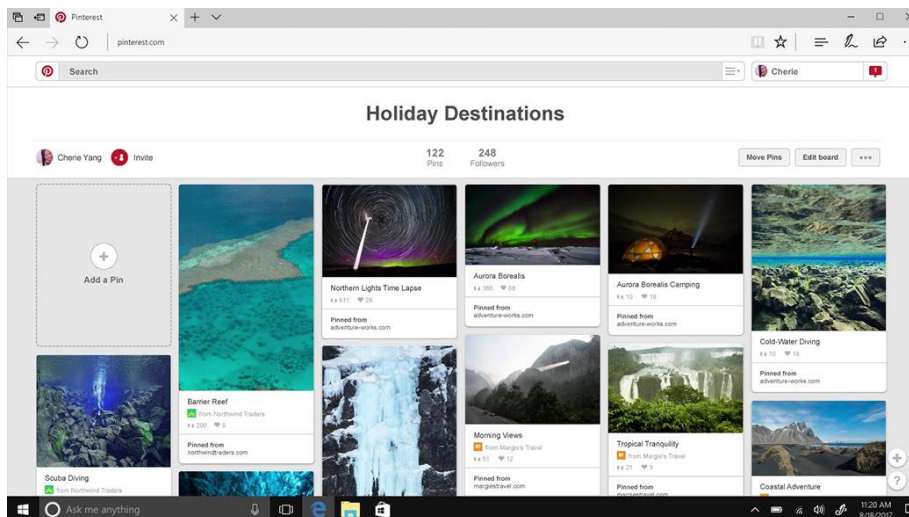
2.2.2 Firefox

Navegador web desarrollado por la Fundación Mozilla, fecha de lanzamiento 23 de septiembre de 2002. Versiones para Windows, macOS, Linux, Android y iOS. Desarrollado en C++, XUL, XBL, JavaScript, C y Rust. Motor de renderizado Gecko



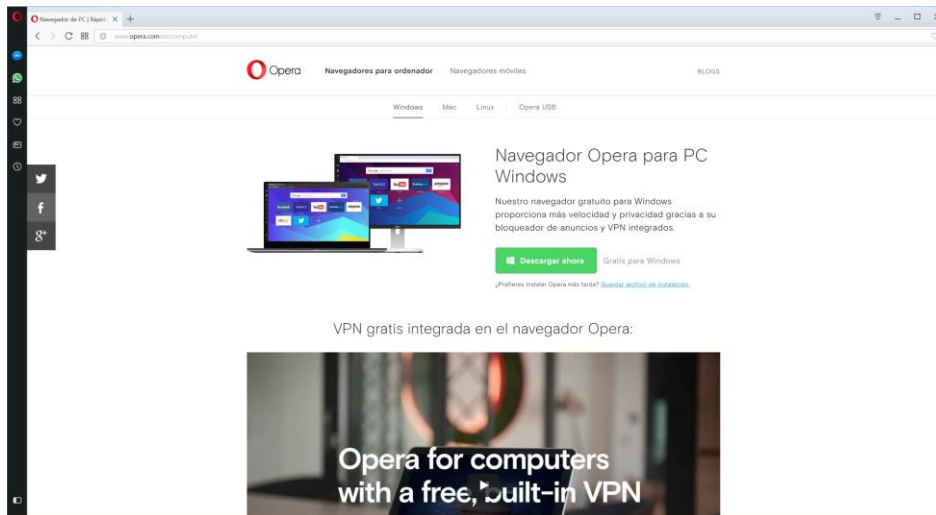
Microsoft Edge

Navegador web desarrollado por Microsoft, fecha de lanzamiento 29 de julio de 2015. Versiones para Windows, macOS, Android y iOS. Desarrollado en C++. Motor de renderizado Blink. Basado en Chromium desde el 15 de enero de 2020.



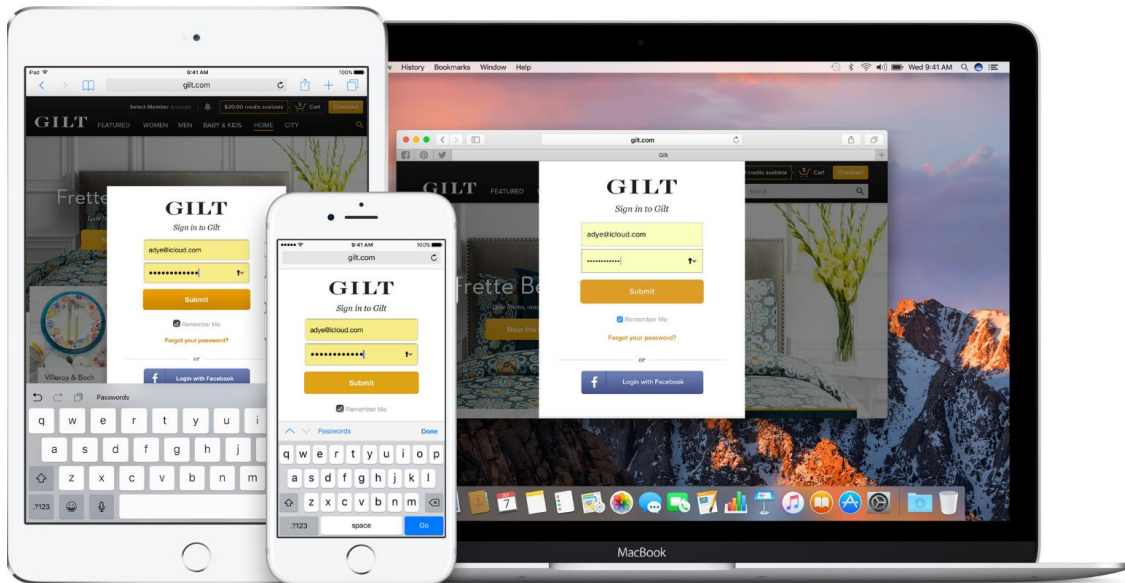
2.2.3 Opera

Navegador web desarrollado por Opera Software, fecha del lanzamiento Inicial 10 de abril de 1995. Versiones para Windows, macOS, Linux, FreeBSD, Solaris, Android y iOS. Desarrollado en C++. Motor de renderizado Blink.



2.2.4 Safari (iOS)

Navegador web desarrollado por Apple (código cerrado), fecha del lanzamiento inicial 7 de enero de 2003. Versiones para macOS y iOS (la versión para Windows hasta la versión 5.1.7). Desarrollado en C++, Objective-C y Swift. Motor de renderizado Webkit.



2.3 XAMPP

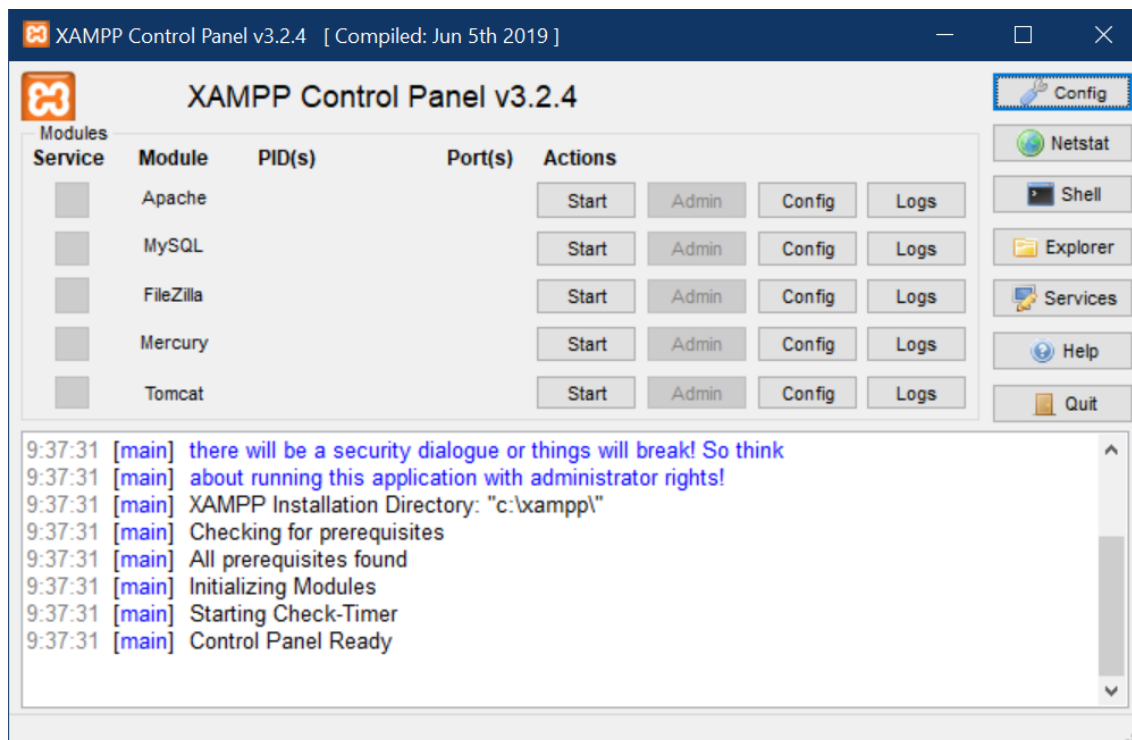
En la asignatura vamos a utilizar XAMPP para tener el servidor **Apache** y la base de datos **MySQL** en el Bloque de Computación en el Servidor

XAMPP es un servidor web de software libre que habilita un entorno con un servidor web Apache, el sistema de gestión de bases de datos MySQL y los interpretes de lenguajes de script PHP y Perl.

XAMPP es un acrónimo **X** (para cualquiera de los diferentes sistemas operativos), **A**pache, **M**ariaDB, **P**HP, **P**erl. A partir de la versión 5.6.15, XAMPP cambió la base de datos MySQL por MariaDB, un fork de MySQL con licencia GPL.

Sitio oficial de descarga de XAMPP: <https://www.apachefriends.org/es/index.html>

En el panel de control podremos activar o desactivar y administrar los diferentes componentes de XAMPP: Apache, MySQL, FileZilla, Mercury y Tomcat.



2.4 Cliente SSH-SFTP – Bitvise

Bitvise SSH es un cliente y servidor SSH y SFTP gratuito para entornos Windows. Permite establecer conexiones seguras:

- Bitvise SSH Client es el cliente, que se utiliza para establecer las conexiones remotas al protocolo SSH. Este cliente es el que se utiliza para establecer una conexión remota al terminal del servidor para controlarlo, para configurar y monitorizar el reenvío de puertos y para transferir archivos de forma segura a través de Secure-FTP. Este cliente es el que debe instalarse en la máquina desde la que vamos a conectarnos al servidor.
- Bitvise SSH Server es el servidor, que debe aceptar las conexiones del cliente. Este servidor está configurado para funcionar durante largos periodos de tiempo, a la espera de recibir una petición de conexión. El servidor debe instalarse en la máquina remota.

Alternativas para Mac OS <https://apple.stackexchange.com/questions/25661/whats-a-good-graphical-sftp-utility-for-os-x>

Se utilizará en los seminarios de Computación en la nube.

3 Validador de Lenguajes de marcas <https://validator.w3.org/> y validador para HTML5 <https://validator.w3.org/nu>

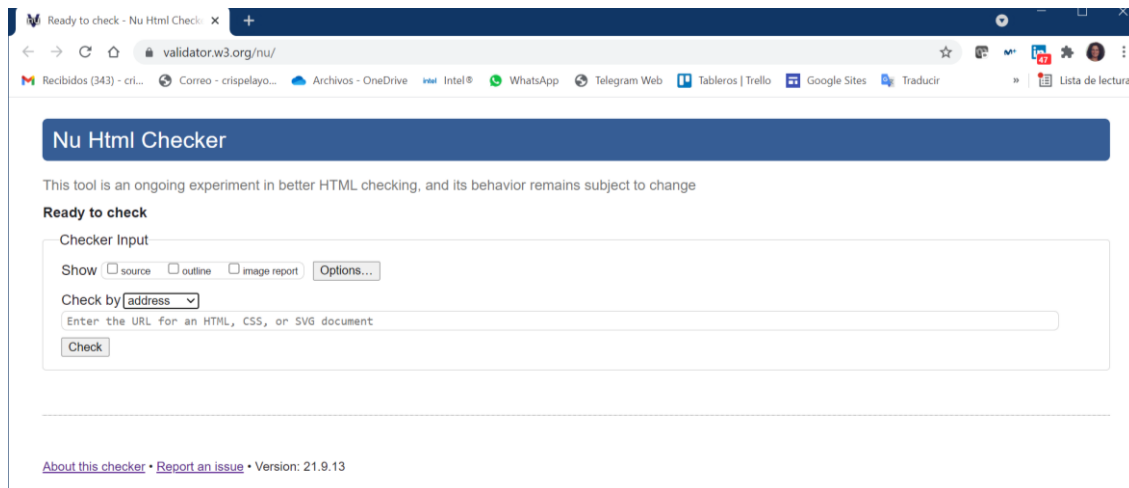
El contenido de los sitios web se escribe mediante un lenguaje de marcas HTML - HyperText Markup Language (puede ser estático o dinámico/generado).

Página web de HTML 5: <https://html.spec.whatwg.org/multipage/>

El W3C proporciona un servicio online y gratuito que nos permite comprobar el cumplimiento del estándar (la validez) de los documentos HTML. Si utilizamos HTML4. o XHTML 1.x el validador a utilizar es <https://validator.w3.org/> si utilizamos HTML5 el validador es <https://validator.w3.org/nu>

Pasos:

- Abrir en el navegador el validador de HTML5 <https://validator.w3.org/nu>



- Seleccionar la forma de validación:
 - Por URL: introducir la dirección web del sitio a validar
 - **Por carga de archivo: seleccionando el archivo local. En SEW al trabajar en modo local preferentemente utilizaremos esta opción.**
 - Copiando directamente el código
- Marcar la opción Ver fuente (para visualizar el código validado)
- Pulsar en “Check”

El resultado de la validación puede ser:

- **HTML Válido**

Nu Html Checker

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

Showing results for <http://156.35.94.1/~crispelayo/>

Checker Input

Show ☐ source ☐ outline ☐ image report

Check by

<http://156.35.94.1/~crispelayo/>

Document checking completed. No errors or warnings to show.

Used the HTML parser.

Total execution time 385 milliseconds.

- **HTML Inválido:** se muestran los errores que deben ser revisados y corregidos SIEMPRE. En los errores se muestra una descripción del error y donde se produce. Los errores se detallan en cascada, siempre se debe solucionar el primero (o segundo), validar y comprobar si se han solucionado el resto. Los errores suelen ser descriptivos, y en cualquier caso siempre señalan la línea donde se ha producido el error para su localización más fácil.

1. **Error** End tag for `body` seen, but there were unclosed elements.
[From line 13, column 1: to line 13, column 7](#)

```
a mundo! <</body><</html>
```
2. **Error** Unclosed element `h1`.
[From line 12, column 5: to line 12, column 8](#)

```
<h1>Hola
```

IMPORTANTE: Tener errores de validación del estándar HTML en cualquier archivo (escrito o generado) de cualquier práctica de la asignatura implica una calificación de 0 puntos en la práctica afectada.

4 Validador de CSS <https://jigsaw.w3.org/css-validator/>

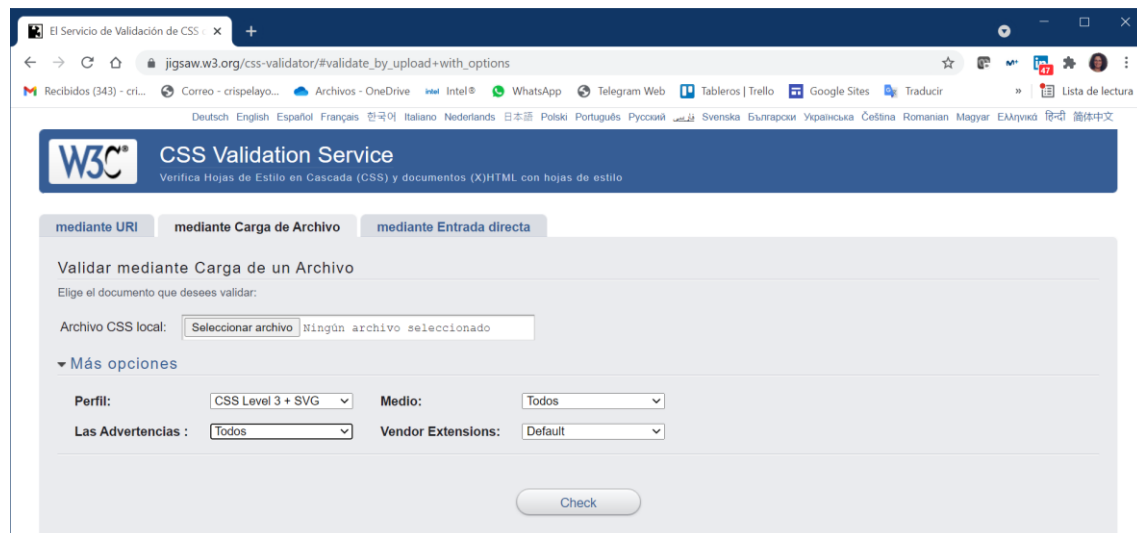
Las Hojas de Estilo en Cascada (Cascading Style Sheets), son el mecanismo simple que describe cómo se debe visualizar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos. Los archivos con extensión .CSS contienen las reglas de estilo a aplicar a un documento HTML.

Página Web de CSS: <http://www.w3.org/Style/CSS/>

El W3C proporciona un servicio online y gratuito que nos permite comprobar el cumplimiento del estándar (la validez) de las hojas de estilo CSS. Más información <https://jigsaw.w3.org/css-validator/>

Pasos:

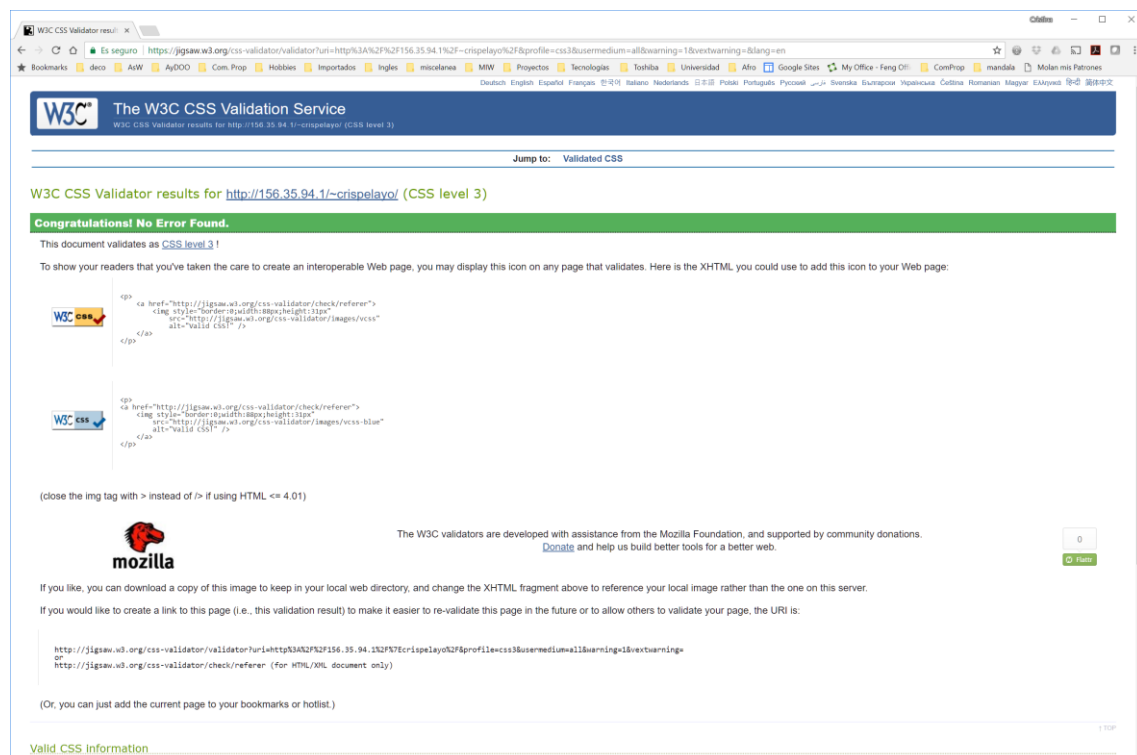
- Abrimos el validador <https://jigsaw.w3.org/css-validator>



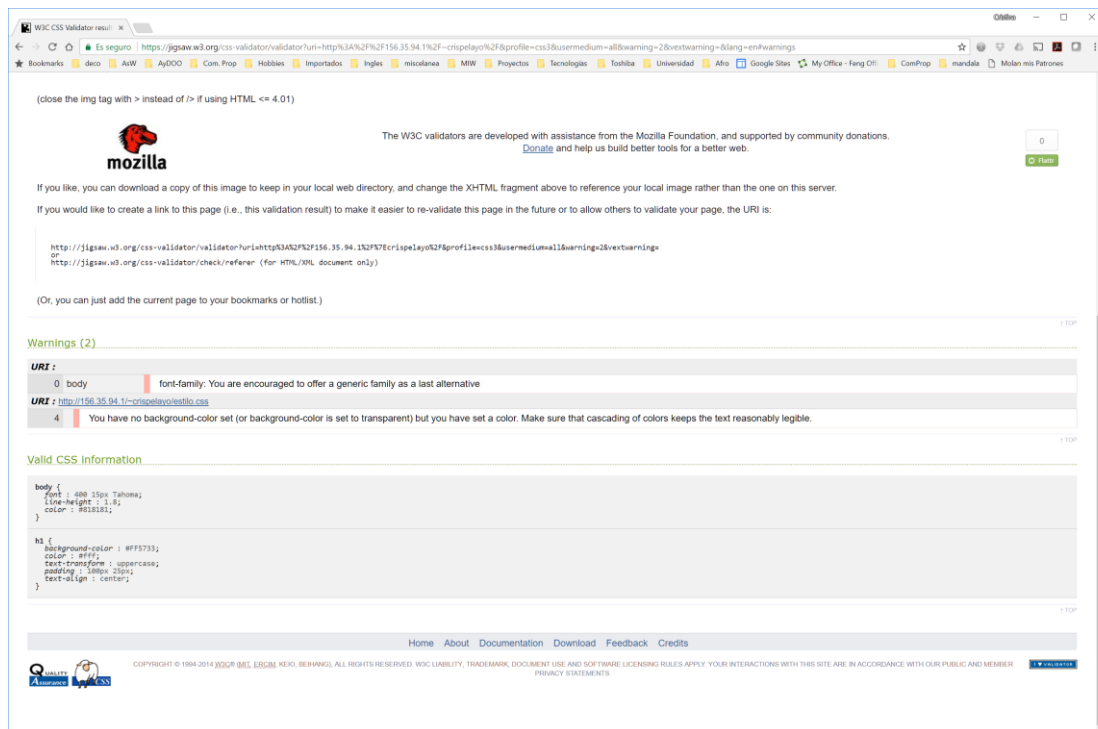
- Seleccionar la forma de validación:
 - Por URL introducir la dirección web del sitio a validar
 - **Por fichero: seleccionando el fichero local. En SEW al trabajar en modo local preferentemente utilizaremos esta opción.**
 - Copiando directamente el código
- **IMPORTANTE:** Seleccionar en “Más Opciones” la opción “TODAS” para que se muestre un listado completo de las advertencias (warnings) de la hoja de estilo.
- Pulsar en “Check”

El resultado de la validación puede ser:

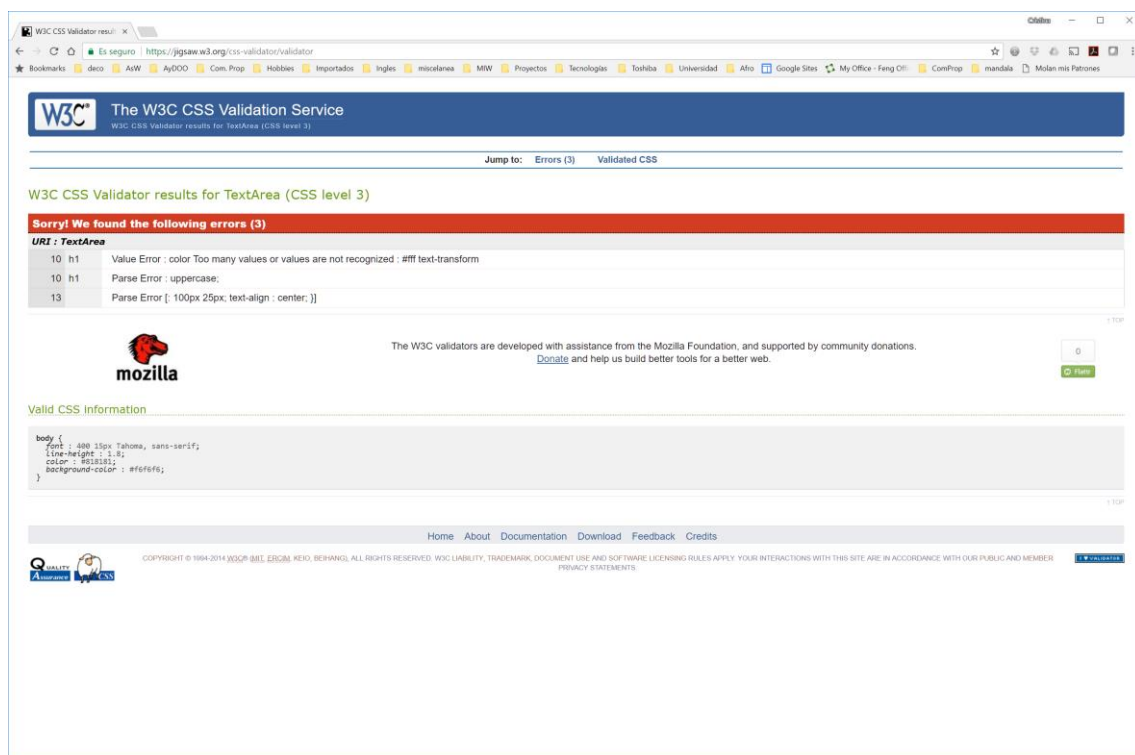
- **CSS válido**



- **CSS válido con warnings** que deben ser resueltos



- **CSS inválido:** se muestran los errores que deben ser revisados y corregidos SIEMPRE. En los errores se muestra una descripción del error y donde se produce. Los errores se detallan en cascada, siempre se debe solucionar el primero, validar y comprobar si se han solucionado el resto.



IMPORTANTE: Tener errores de validación o warnings del estándar CSS en cualquier archivo (escrito o generado) de cualquier práctica de la asignatura implica una calificación de 0 puntos en la práctica afectada.

5 Herramientas de adaptabilidad web

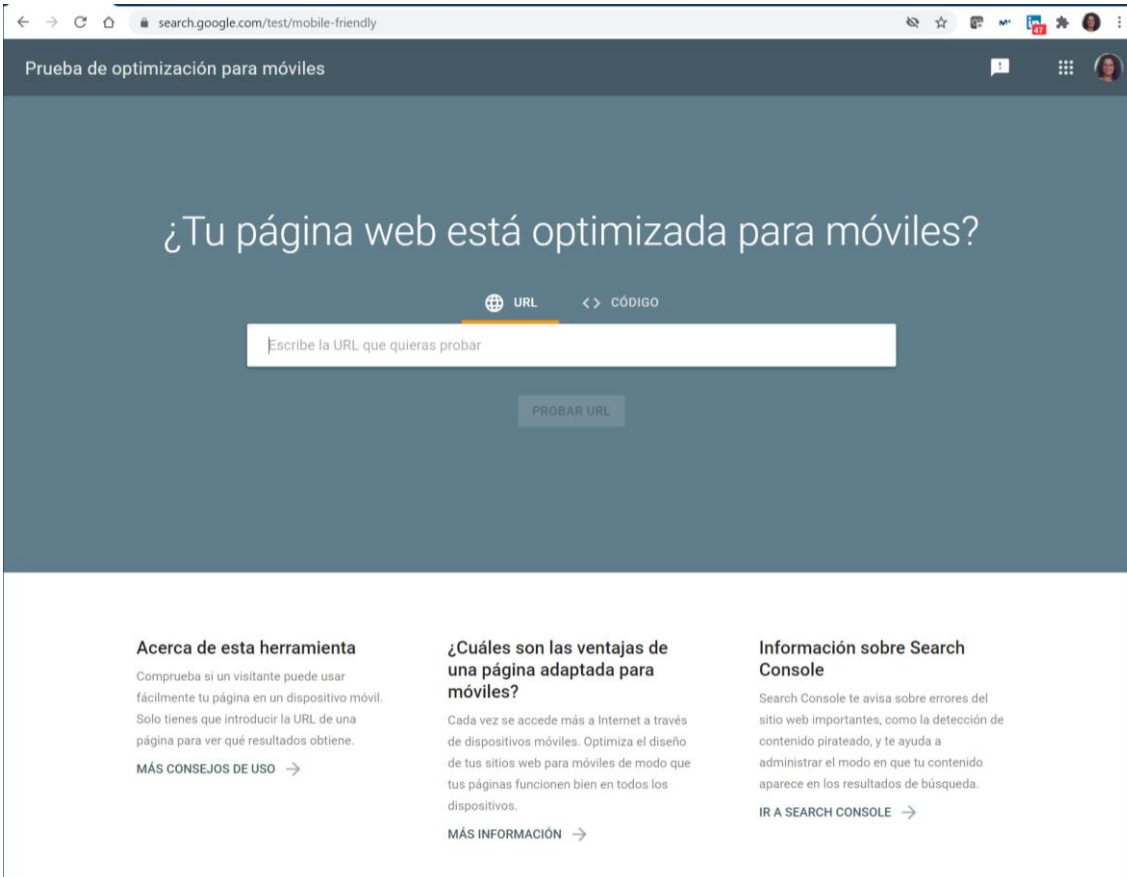
Con la popularización del uso de los dispositivos móviles debemos garantizar la adaptabilidad de los sitios web a diferentes tamaños de pantalla. Para ellos vamos a utilizar dos herramientas que permiten comprobar si el diseño es adaptable y como se visualizará en diferentes soportes.

NOTA: Para poder utilizar algunas de las herramientas de la comprobación de la accesibilidad se necesita desplegar el código en un repositorio accesible mediante URL (ver ANEXO GitHub Pages)

5.1 Google Mobile-friendly – <https://search.google.com/test/mobile-friendly>

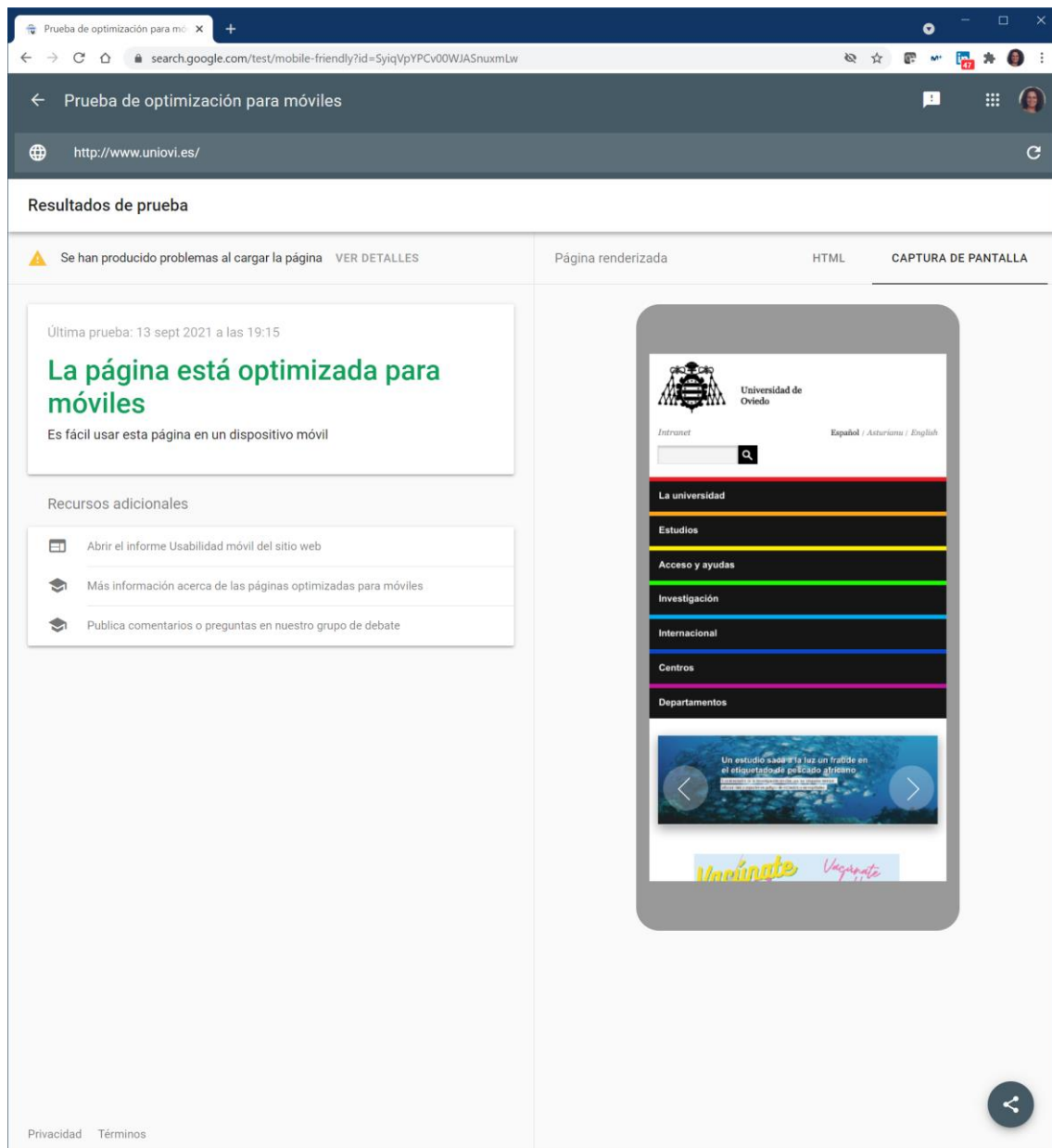
Informa si el web está adaptado para su visualización en dispositivos móviles. Es necesario desplegarlo en una URL pública para que la revisión sea de todos los documentos que componen el sitio web.

Los resultados de la prueba incluyen una captura de pantalla de cómo Google ve la página en un dispositivo móvil, así como una lista de los problemas de usabilidad móvil encontrados.



The screenshot shows the Google Mobile-Friendly Test interface. At the top, the browser address bar displays "search.google.com/test/mobile-friendly". The page title is "Prueba de optimización para móviles". The main heading asks "¿Tu página web está optimizada para móviles?". Below this, there are two tabs: "URL" (selected) and "CÓDIGO". A text input field prompts "Escribe la URL que quieras probar". A "PROBAR URL" button is positioned below the input field. The footer contains three sections: "Acerca de esta herramienta" (describing the tool's purpose), "¿Cuáles son las ventajas de una página adaptada para móviles?" (listing benefits), and "Información sobre Search Console" (providing links to learn more and access the console).

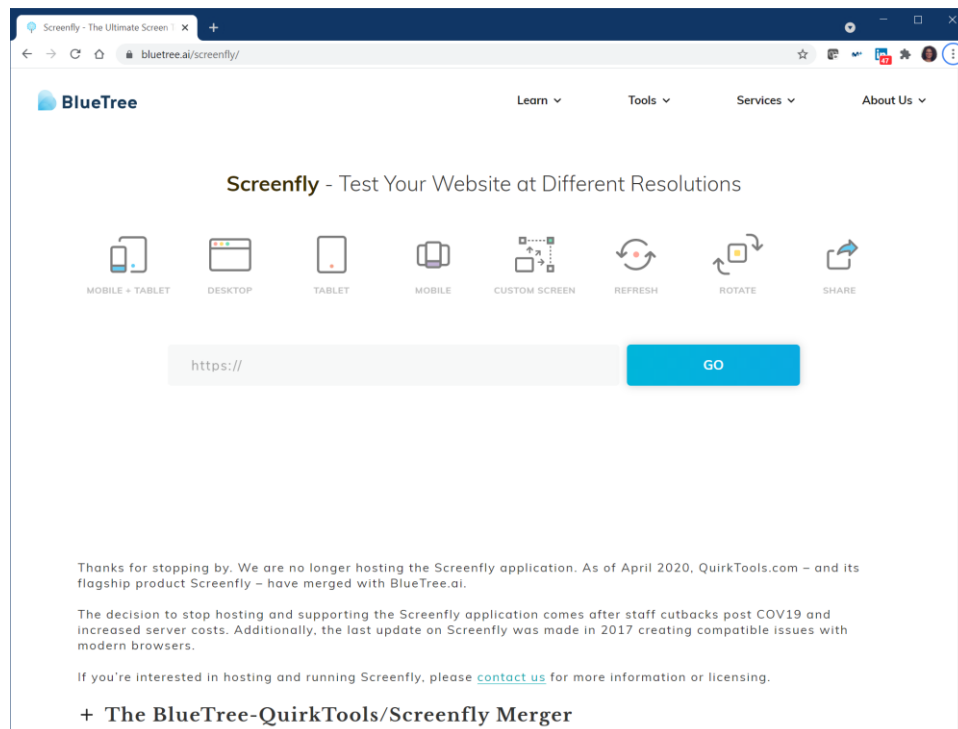
El resultado del análisis de la página web de la Universidad de Oviedo se muestra a continuación:



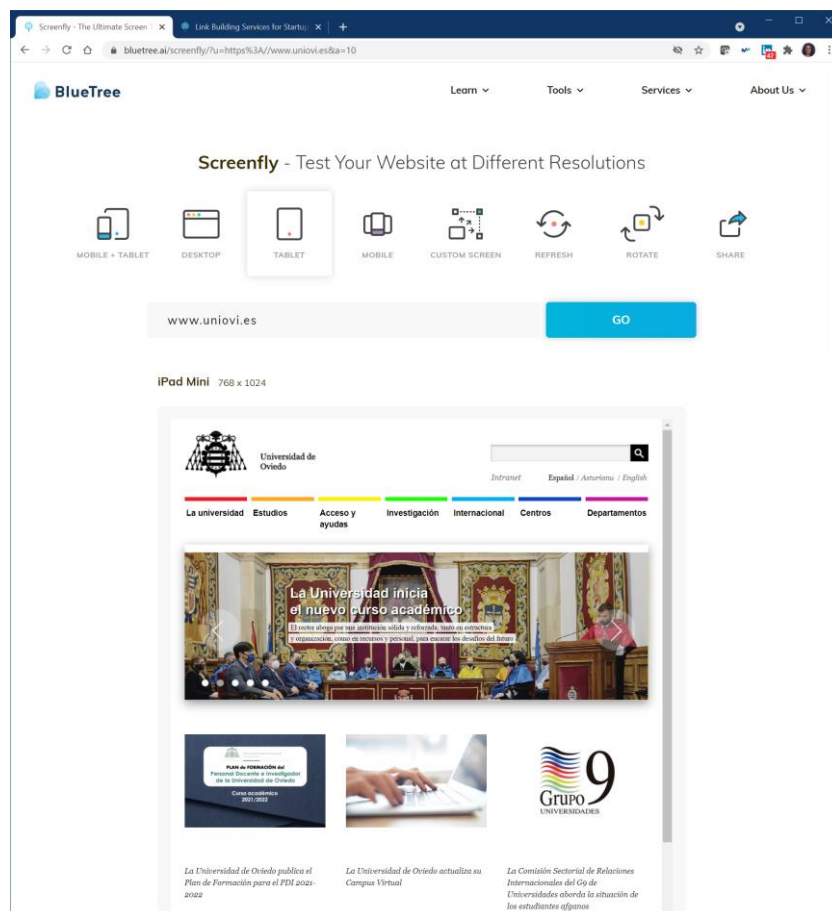
En detalles se muestran los problemas (en el ejemplo de carga) que se hayan producido.

5.2 Screenfly - <https://bluetree.ai/screenfly/>

Esta herramienta permite simular la visualización del Sitio web en multitud de soportes de diferentes tipos y marcas.



Es recomendable simular la visualización para varios dispositivos para garantizar la visualización óptima.



6 Herramientas de Accesibilidad Web

La Accesibilidad Web trata del acceso universal a la Web, independientemente del tipo de hardware, software, infraestructura de red, idioma, cultura, localización geográfica y capacidades de los usuarios.

El W3C desarrolla la “Iniciativa de Accesibilidad Web”, conocida como WAI (Web Accessibility Initiative) su objetivo es facilitar el acceso de las personas con discapacidad, desarrollando pautas de accesibilidad, mejorando las herramientas para la evaluación y reparación de accesibilidad Web.

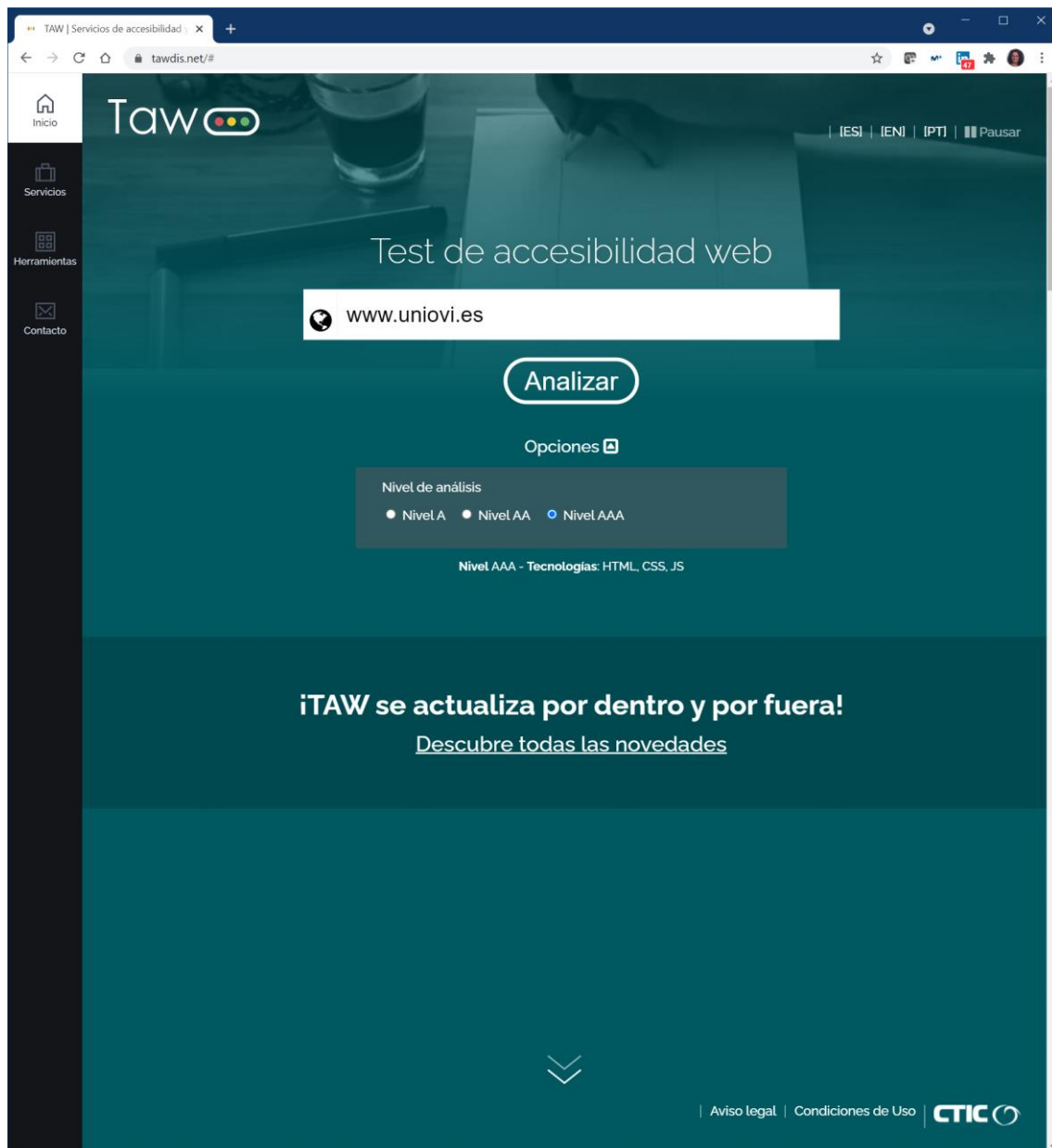
Las herramientas que vamos a utilizar en la asignatura proporcionan un análisis automático de la accesibilidad en el nivel AAA que siempre debería completarse con un análisis manual de las pautas de accesibilidad.

NOTA: Para poder utilizar algunas de las herramientas de la comprobación de la accesibilidad se necesita desplegar el código en un repositorio accesible mediante URL (ver ANEXO GitHub Pages)

6.1 Herramienta TAW - <http://www.tawdis.net/>

TAW es una familia de herramienta para el análisis de la accesibilidad de sitios web, alcanzando de una forma integral y global a todos los elementos y páginas que lo componen. Esta familia la componen diversas herramientas, desde los más conocidos analizadores de páginas a los sistemas de monitorización o una herramienta para la realización de observatorios. El nexo común a todos son los motores de análisis, que han ido creciendo en funcionalidades desde el de accesibilidad a los de calidad web, SEO o movilidad.

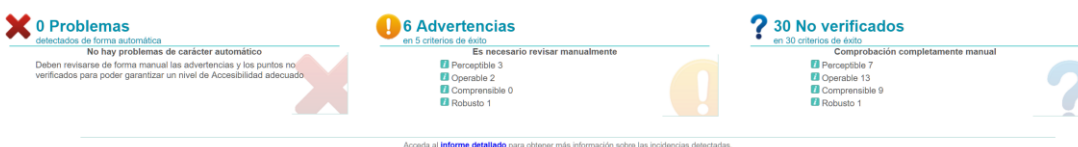
El analizador de páginas es:



Permite seleccionar el nivel de Accesibilidad (INDICAR SIEMPRE AAA) a evaluar y las tecnologías empleadas. Como resultado del análisis nos proporciona un informe resumido y otro completo sobre la accesibilidad de los sitios web.

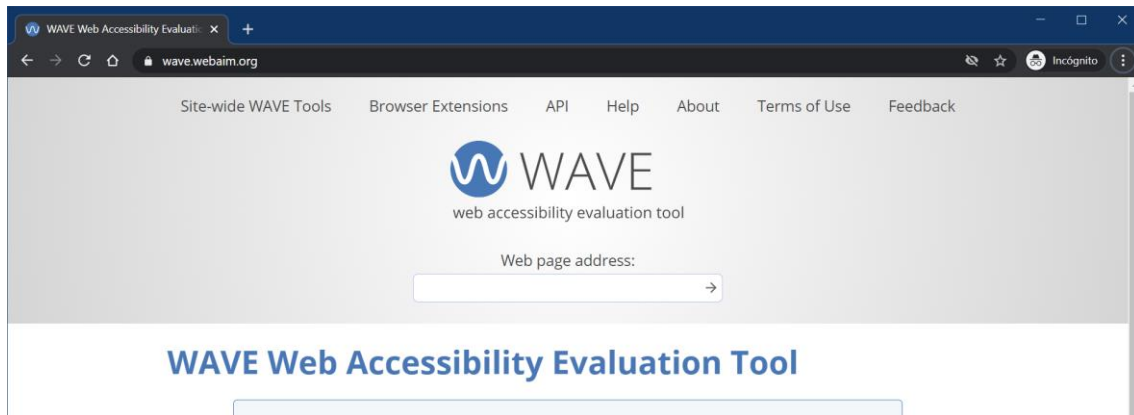


En general el resultado tiene que ser 0 problemas en el análisis automático.

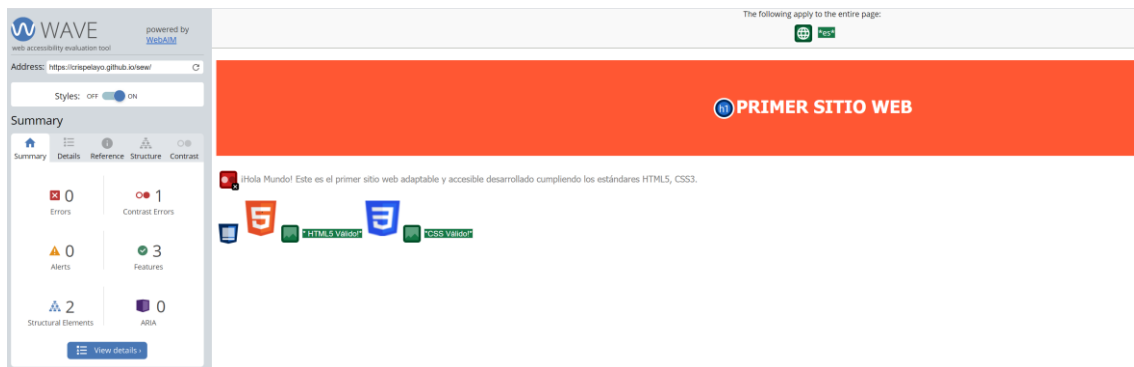


6.2 Herramienta Wave - <http://wave.webaim.org/>

WAVE es una herramienta que ayuda a los desarrolladores web a hacer su contenido web más accesible, porque ayuda a evaluar la accesibilidad del contenido web.



Proporciona un informe “visual” de los problemas de accesibilidad detectados.



En general debemos garantizar que el número de errores en el análisis automático de AAA sea 0. Y se debe realizar la comprobación de los errores de contraste y resolverlos.

7 Pautas de trabajo generales en la asignatura

En la asignatura se va a trabajar en “**modo local**” por tanto será responsabilidad de cada estudiante preservar la seguridad de su código para evitar el acceso indebido al mismo, sobre todo en las máquinas del laboratorio.

Los pasos generales en las prácticas de la asignatura son:

- Crear una carpeta en local para ubicar los archivos de las prácticas
- Utilizar el editor de texto para escribir el código HTML, CSS, XML, JavaScript o PHP
- Realizar la validación de todos los archivos con código HTML (véase el apartado **Validador de HTML**)
 - o **Siempre se debe realizar la comprobación de cumplimiento del estándar HTML de todos los documentos**
 - o **IMPORTANTE:** Se debe validar tanto el código HTML estático como el código HTML dinámico generado a través de computación en el cliente, computación en el servidor o con XML.
- Realizar la validación de todo el código CSS (véase el apartado **Validador de CSS**)
 - o **Siempre se debe realizar la comprobación de cumplimiento del estándar CSS de todas las hojas de estilo**
 - o **IMPORTANTE:** Se debe validar tanto el código CSS estático como el posible código dinámico generado a través de computación en el cliente, computación en el servidor o con XML.
- Comprobar la adaptabilidad de todos los documentos que componen el sitio desarrollado [*]
- Comprobar la accesibilidad web para el nivel AAA de todos los documentos que componen el sitio desarrollado [*]

[*] **IMPORTANTE:** Se debe desplegar los archivos en GitHub Pages para poder utilizar algunas herramientas de comprobación de la adaptabilidad y la accesibilidad web. (Ver ANEXO: Github Pages)

8 Creación del primer sitio web simple

8.1 Ejercicio 1. Crear el documento index.html mínimo válido

El archivo index.html es el punto de entrada de la página web. Cuando en un navegador Web se escribe la dirección de una página web (p.ej www.uniovi.es) el servidor busca dentro del directorio el archivo index.html y lo muestra. Si no existe el archivo index.html muestra un error.

Se debe crear un documento index.html que contiene el documento mínimo válido HTML5 con el contenido de un sitio web dentro del directorio de trabajo local.

```
<!DOCTYPE HTML>
<html lang="es">
<head>
  <!-- Datos que describen el documento -->
  <meta charset="UTF-8" />
  <title>SEW.P0</title>
```

Html

```
</head>

<body>
  <!-- Datos con el contenido que aparece en el navegador -->
  <h1> Primer sitio web</h1>
  <p> ¡Hola Mundo! Este es el primer sitio web adaptable y accesible
desarrollado cumpliendo los estándares HTML5, CSS3. </p>
</body>
</html>
```

8.2 Guía para resolver el ejercicio 1

PASO 1. Crear el directorio local SEW

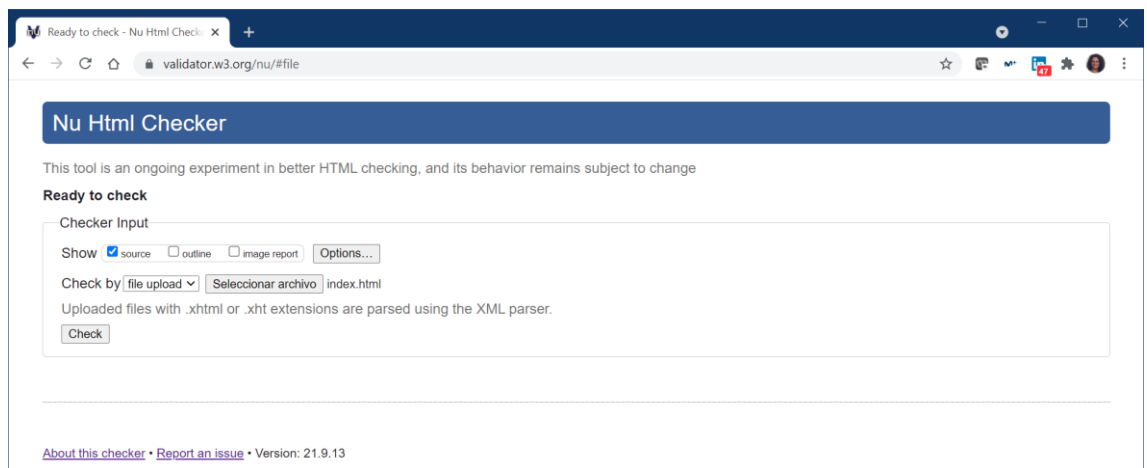
Crea un directorio local donde vas a ubicar los archivos de la asignatura.

PASO 2. Crear el archivo index.html

- Abrir el editor de texto
- Copiar el código del recurso HTML
- Guardar el archivo con nombre `index.html` dentro del directorio local creado
- Abrir el archivo `index.html` en un navegador para ver su aspecto

PASO 3. Realizar la validación del cumplimiento de HTML5 de index.html

- Abrir en el navegador el validador de HTML5 <https://validator.w3.org/nu>
- Seleccionar la forma de validación:
 - Por fichero (File upload): seleccionando el fichero local “index.html”
 - Marcar la opción de Mostrar fuente (Show source)



- Pulsar en “Check” (validar)

El resultado de la validación debe ser HTML Válido

- HTML Válido

Nu Html Checker

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

Showing results for <http://156.35.94.1/~crispelajo/>

Checker Input

Show ☐ source ☐ outline ☐ image report

Check by

<http://156.35.94.1/~crispelajo/>

Document checking completed. No errors or warnings to show.

Used the HTML parser.

Total execution time 385 milliseconds.

8.3 Ejercicio 2. Crear una hoja de estilo estilo.css

Se debe crear una hoja de estilo básica válida y sin advertencias con el contenido proporcionado en el archivo de recursos CSS.

```
/* Especificidad: 001*/
body {
    font-family: Tahoma, sans-serif;
    color: #818181;
    background-color: #fff;
}
/* Especificidad: 001*/
h1 {
    color: #fff;
    background-color: #FF5733;
    padding: 2em;
    text-transform: uppercase;
    text-align: center;
}
```

Css

8.4 Guía para resolver el ejercicio 2

PASO 1. Crear el archivo estilo.css

- Abrir el editor de texto
- Copiar el código del recurso CSS
- Guardar el archivo con nombre `estilo.css` dentro del directorio local creado

PASO 2. Realizar la validación del cumplimiento de CSS de estilo.css

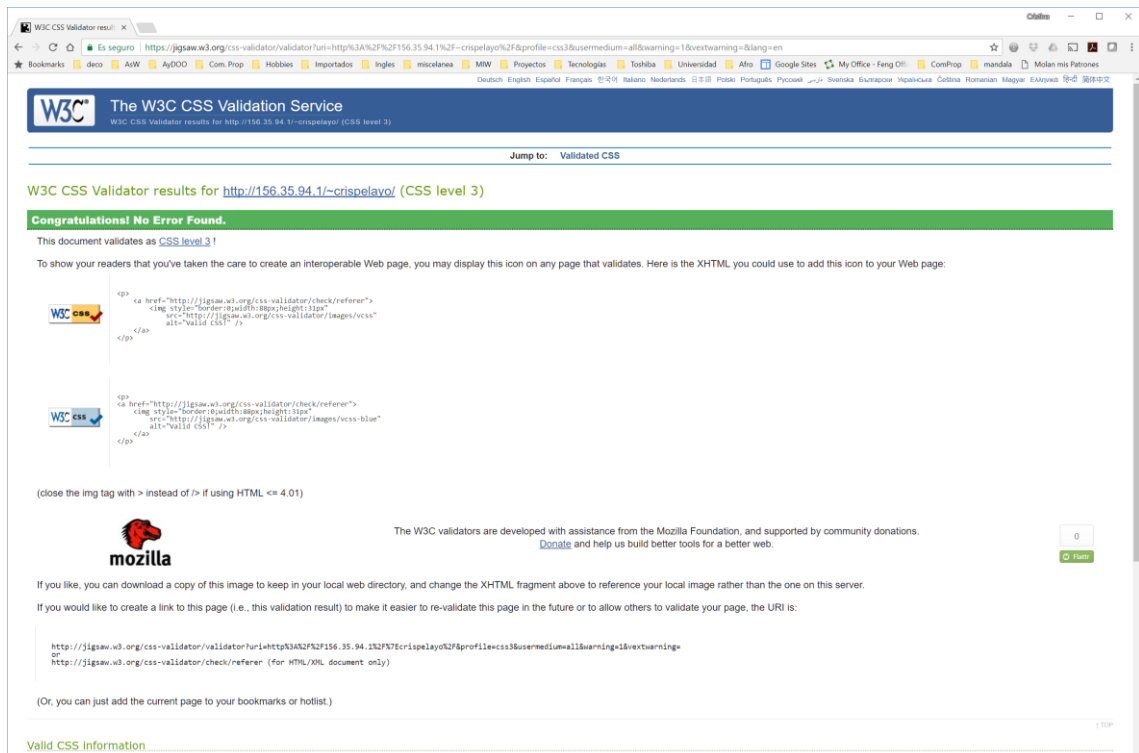
- Abrimos el validador <https://jigsaw.w3.org/css-validator>

The W3C CSS Validation Service interface shows the 'Validate by URI' tab. The 'Address' field is empty. Under 'More Options', 'Profile' is set to 'CSS level 3', 'Medium' is 'All', 'Warnings' is 'Normal report', and 'Vendor Extensions' is 'Default'. A 'Check' button is at the bottom right. Below the form is a small 'I VALIDATOR' logo and a note about community support with a 'Donate' link.

- Seleccionar la forma de validación:
 - Por fichero: seleccionando el fichero local
 - “More Options” y la opción “Warnings = All” para que nos proporcione un listado completo de los warnings de la hoja de estilo.
- Pulsar en “Check”

El resultado de la validación puede ser:

- CSS válido



8.5 Ejercicio 3. Aplicar la hoja de estilo al documento HTML5

Aplicar los estilos definidos en el Ejercicio2 al documento HTML5 del ejercicio1. Comprobar el cumplimiento del estándar HTML5 y visualizar el aspecto del documento en el navegador.

8.6 Guía para resolver el ejercicio 3

PASO 1. Editar el archivo index.html del ejercicio1

- Editar el archivo index.html y añadir el enlace a la hoja de estilo, en la cabecera detrás de <title>:

```
<title>SEW.P0</title>
<link rel="stylesheet" type="text/css" href="estilo.css" />
```

enlaceCSS

- Guardar los cambios

PASO 2. Realizar la validación del cumplimiento de HTML5 de index.html

- Al realizar cualquier cambio en un documento HTML5 debe comprobarse el cumplimiento del estándar <https://validator.w3.org/nu>

PASO 3. Comprobar con los navegadores el aspecto del documento

- Abrir el archivo index.html en un navegador para ver su nuevo aspecto

8.7 Ejercicio 6. Incorporar los iconos de cumplimiento de los estándares HTML y CSS

Una vez se ha comprobado el cumplimiento de los estándares HTML y CSS incorporar al sitio web los iconos de cumplimiento de HTML y CSS

Crear una sección footer dentro del documento index.html que añada

8.8 Guía para resolver el ejercicio 6

PASO 1. Editar el archivo index.html del ejercicio1 modificado en el ejercicio3

- Copiar la carpeta images y su contenido a mismo directorio donde se ubique el sitio web de los ejercicios anteriores.
- Editar el archivo index.html y añadir una sección de tipo <footer> después del párrafo y antes de </body>

```
<footer>
  <img src= "images/HTML5.png" alt= "HTML5 válido!" />
  <img src= "images/CSS3.png" alt= "CSS3 válido!" />
</footer>
```

iconos

- Guardar los cambios

PASO 2. Realizar la validación del cumplimiento de HTML5 de index.html

- Al realizar cualquier cambio en un documento HTML5 debe comprobarse el cumplimiento del estándar <https://validator.w3.org/nu>

PASO 3. Comprobar con los navegadores el aspecto del documento

- Abrir el archivo index.html en un navegador para ver su nuevo aspecto

8.9 Ejercicio 4. Comprobar la adaptabilidad del sitio web

Se debe comprobar la adaptabilidad del sitio web desarrollado en los ejercicios anteriores y su visualización en al menos un teléfono móvil, una tableta y monitor.

8.10 Guía para resolver el ejercicio 4

PASO 1. Desplegar el sitio web desarrollado en GitHub Pages

Cargar los archivos index.html, estilo.css y la carpeta images en el repositorio sew de github Pages. Generar el commit. Hacer pública la página y comprobar su visualización en el navegador web.

PASO 2. Comprobar la adaptabilidad con la herramienta Google Mobile-friendly

Utilizar la URL del sitio web desplegado en Github para comprobar la adaptabilidad.

PASO 3. Comprobar la visualización del sitio web con la herramienta Screenfly

Utilizar la URL del sitio web desplegado en Github para comprobar la visualización del sitio web en al menos un formato móvil, un formato Tablet y un monitor.

RECUERDA. Quitar de acceso público el sitio web desarrollado en GitHub Pages para garantizar la privacidad de tu código

8.11 Ejercicio 5. Comprobar la accesibilidad del sitio web

Se debe comprobar la accesibilidad para el nivel AAA del sitio web desarrollado en los ejercicios anteriores.

8.12 Guía para resolver el ejercicio 5

PASO 1. Desplegar el sitio web desarrollado en GitHub Pages

Cargar los archivos index.html y estilo.css en el repositorio sew de github Pages. Generar el commit. Hacer pública la página y comprobar su visualización en el navegador web.

PASO 2. Comprobar la accesibilidad con la herramienta TAW

Utilizar la URL del sitio web desplegado en Github para comprobar la accesibilidad AAA con la herramienta TAW.

ATENCIÓN como solo existe el documento index.html que es el punto de entrada por defecto no es necesario indicar el nombre del archivo.

PASO 3. Comprobar la accesibilidad con la herramienta Wave

Utilizar la URL del sitio web desplegado en Github para comprobar la accesibilidad AAA con la herramienta Wave.

ATENCIÓN como solo existe el documento index.html que es el punto de entrada por defecto no es necesario indicar el nombre del archivo.

RECUERDA. Quitar de acceso público el sitio web desarrollado en GitHub Pages para garantizar la privacidad de tu código

9 ANEXO: Github Pages

Se va a utilizar GitHub Pages para poder desplegar los desarrollos en una dirección URL pública. Es importante tener en cuenta los siguientes requisitos previos:

1. Darse de alta en GitHub, si es que no se tiene cuenta ya. El registro se realiza desde el siguiente [enlace](#). Se recomienda usar el identificador uoXXXXXX del alumno como nombre de usuario en github.
2. Una vez completado el paso anterior, iniciar sesión en github.com con el usuario y contraseña previamente creados.

9.1 Creación del repositorio SEW en Github

Utilizando el navegador web y una vez que ya se ha iniciado sesión en GitHub tras haber pasado por los pasos anteriores de este proceso, se procederá a crear un nuevo repositorio utilizando la opción (NEW). En la página relacionada con la creación del repositorio será necesario rellenar el nombre del repositorio (por ejemplo, “sew”) y una descripción del mismo. Lo marcamos como “Private”

ATENCIÓN: Será responsabilidad de cada estudiante mantener seguro su código teniendo el repositorio como Privado para evitar accesos indebidos. O en su defecto como Público pero que este vacío y solo se carguen archivos cuando sea necesario.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?
[Import a repository.](#)


Repository template

Start your repository with a template repository's contents.

No template ▾

Owner *

Repository name *

 crispelayo ▾


 /


sew ✓

Great repository names are short and memorable. Need inspiration? How about **stunning-winner**?

Description (optional)

Repositorio para Software y Estándares

☐  **Public**
Anyone on the internet can see this repository. You choose who can commit.

☒  **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

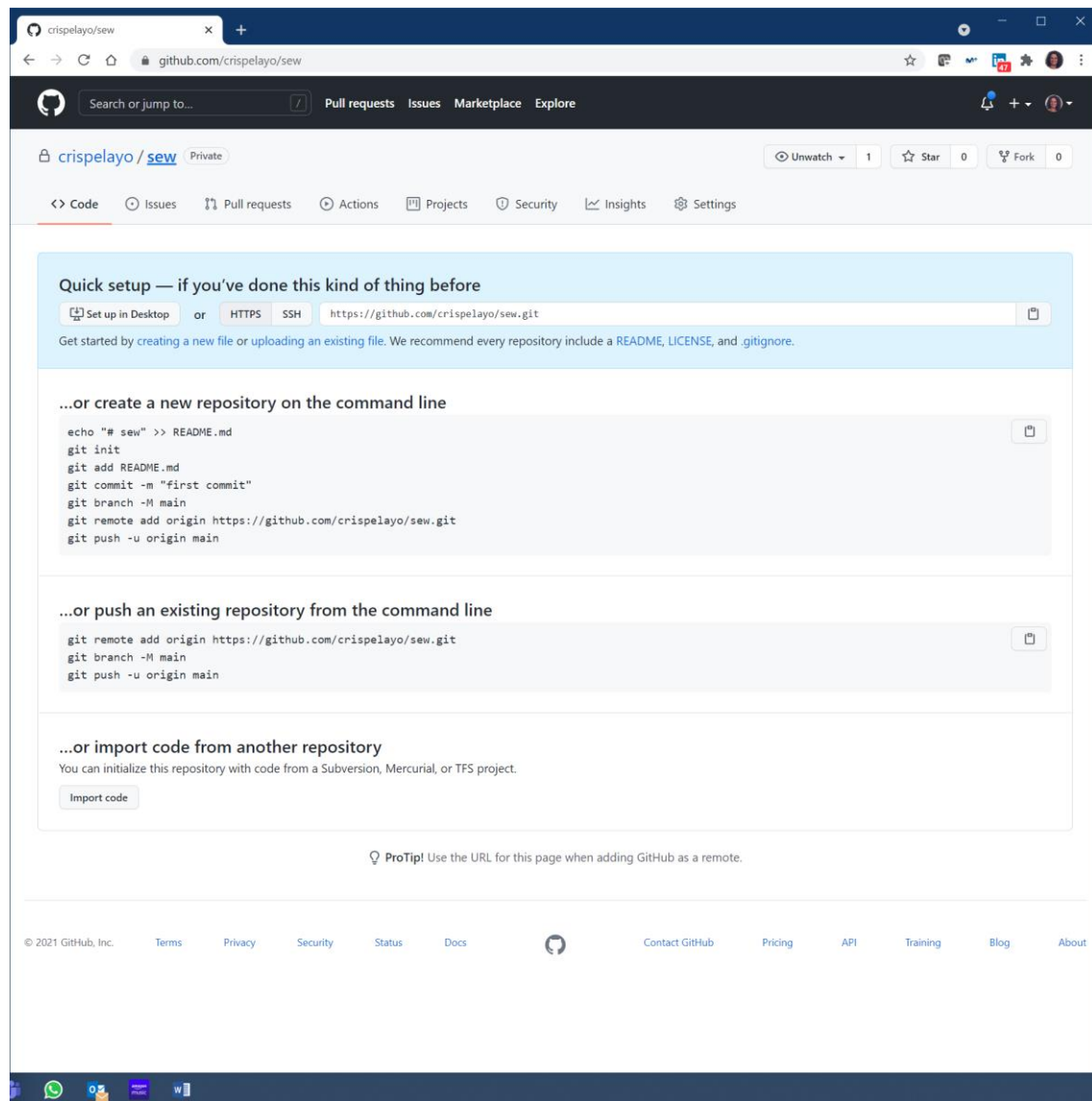
☐ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

☐ **Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)

☐ **Choose a license**
A license tells others what they can and can't do with your code. [Learn more.](#)

Create repository

Tras introducir los datos anteriormente indicados, GitHub procede con la creación del repositorio y se muestra al usuario una pantalla como la siguiente, donde se establecen las formas que se pueden utilizar para interactuar con el repositorio.

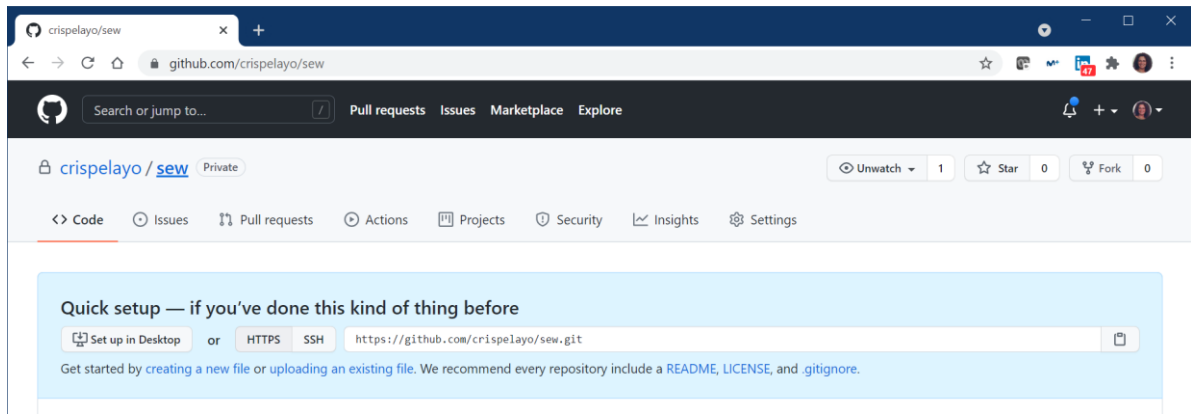


En esta asignatura solo utilizaremos GitHub Pages cargando los archivos vía web. Por ello utilizaremos la opción [uploading an existing file](#) para cargar los archivos que necesitemos desplegar para utilizar las herramientas de accesibilidad y usabilidad.

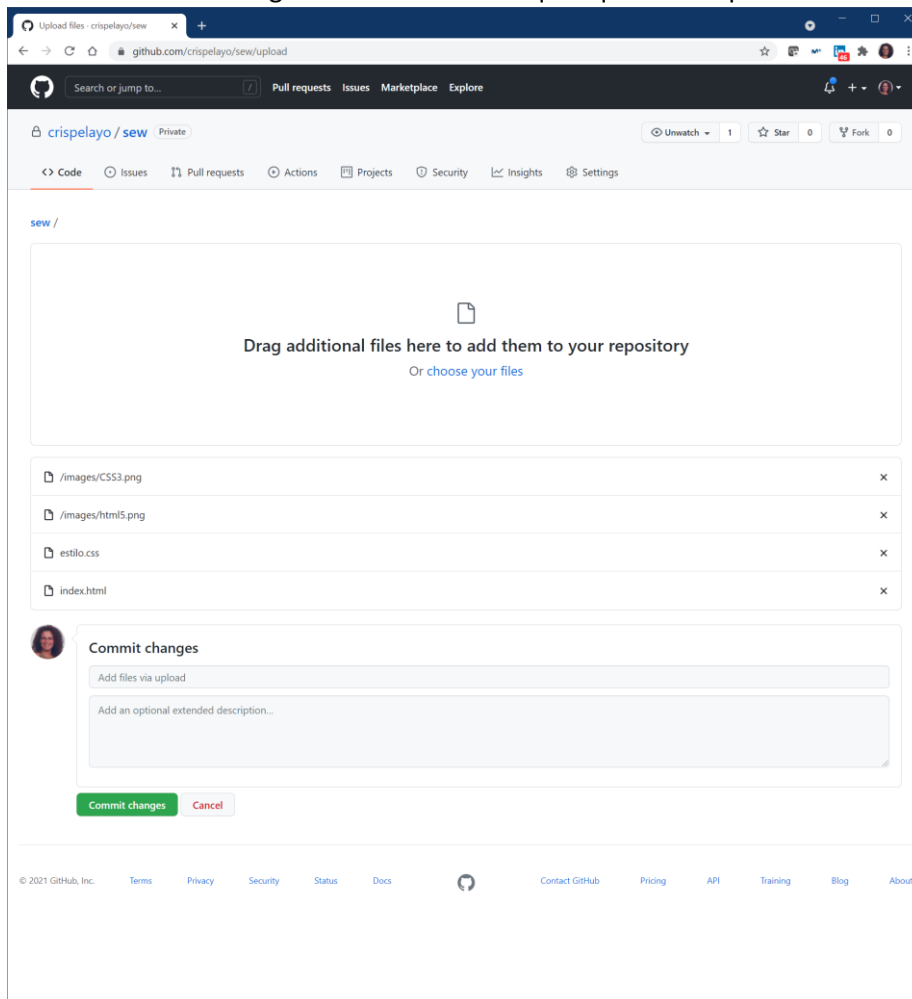
9.2 Cargar archivos

Cuando queramos comprobar la adaptabilidad y la accesibilidad de los sitios web desarrollados debemos subir al repositorio SEW el sitio web completo (código completo y los recursos multimedia).

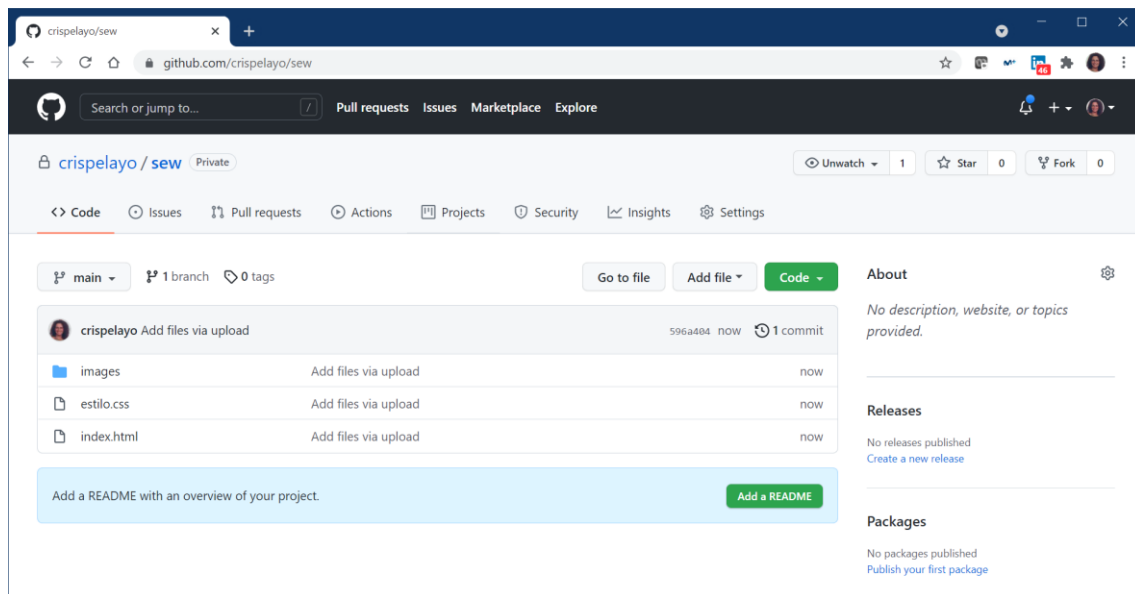
Seleccionamos la opción “uploading an existing file”



Y arrastramos o cargamos los archivos que queramos publicar mediante GitHub Pages



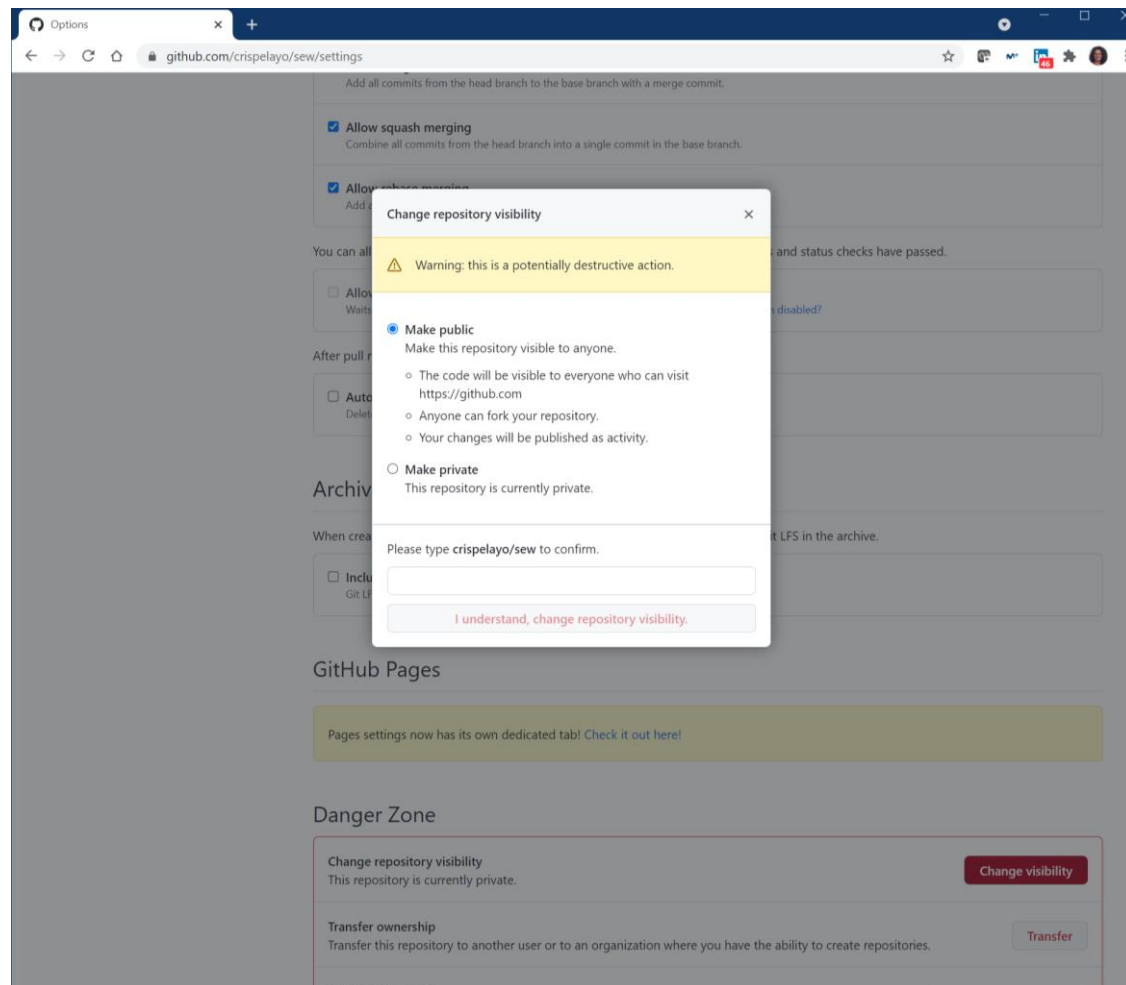
Pulsamos sobre el “Commit Changes” y se realizará la carga de los archivos que es accesible desde la opción <>Code del menú.



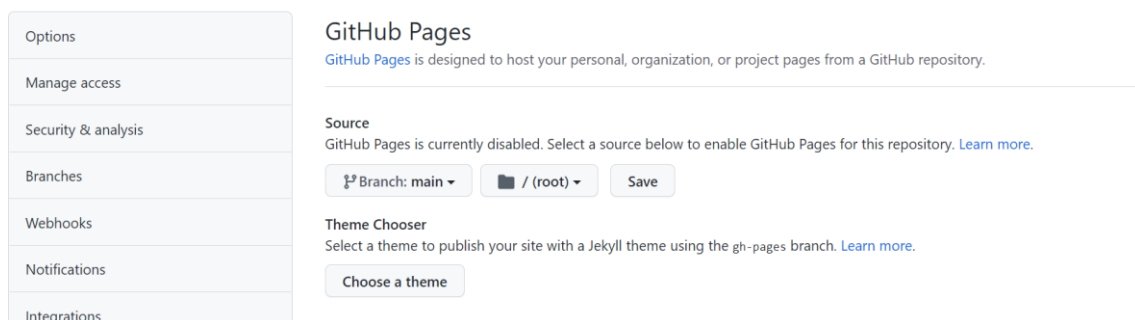
9.3 Crear Pages

A continuación debemos configurar en “Settings” las opciones para convertir el repositorio en un sitio web a través de Github Pages.

Si hemos creado el repositorio como “Privado” debemos hacerlo “Publico” en “Settings/Option/Danger Zone/ **Change repository visibility**”



A continuación en “Settings/Pages” hay que habilitar el GitHub Pages indicando el punto de entrada del repositorio que queremos utilizar.



Si todo es correcto aparece la información relativa a la dirección pública del GitHub Pages

GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

📘 Your site is ready to be published at <https://crispelajo.github.io/sew/>

Source

Your GitHub Pages site is currently being built from the `main` branch. [Learn more.](#)

📁 Branch: `main` ▾

📁 / (root) ▾

Save

Theme Chooser

Select a theme to publish your site with a Jekyll theme. [Learn more.](#)

Choose a theme

Custom domain

Custom domains allow you to serve your site from a domain other than `crispelajo.github.io`. [Learn more.](#)


Save

Remove

☒ **Enforce HTTPS**

— Required for your site because you are using the default domain (`crispelajo.github.io`)

HTTPS provides a layer of encryption that prevents others from snooping on or tampering with traffic to your site. When HTTPS is enforced, your site will only be served over HTTPS. [Learn more.](#)



Publish privately to people with read access to this repository
Try it risk-free for 14 days using a GitHub Enterprise organization, or [learn more about changing the visibility of your GitHub Pages site.](#)

×

En el navegador podemos poner la dirección del sitio de GitHub Pages y visualizar el contenido.

En la dirección **`https://NombreUsuarioGITHUB.github.io/sew/`** se accederá a los documentos que componen el sitio web, por defecto se busca el archivo `index.html` para la visualización.

Esta es la dirección URL raíz que debe utilizarse en las herramientas de Adaptabilidad y Accesibilidad.

IMPORTANTE: Una vez se realice la revisión de la adaptabilidad y la accesibilidad con el objeto de evitar el acceso indebido al código desarrollado se pueden eliminar los archivos del Repositorio público o convertir en Privado el repositorio.