

Software y estándares para la Web

P0. ENTORNO Y PAUTAS DE TRABAJO

Contenido

1	Objetivos	2
2	Preparación del puesto de trabajo.....	2
2.1	Editores	2
2.2	Navegadores.....	5
2.3	XAMPP	7
2.4	Cliente SSH-SFTP – Bitvise	10
2.5	Github Pages	10
3	Primer sitio web	12
3.1	Archivo index.html	12
4	Validadores.....	13
4.1	HTML https://validator.w3.org/	13
4.2	CSS https://jigsaw.w3.org/css-validator/	15
5	Accesibilidad Web	19
5.1	Herramienta TAW - http://www.tawdis.net/	19
5.2	Herramienta Wave - http://wave.webaim.org/	20
5.3	Herramienta aChecker - https://achecker.ca	22
6	Test de adaptabilidad (<i>responsive</i>)	23
6.1	Google Mobile-friendly – https://search.google.com/test/mobile-friendly	23
6.2	Screenfly - http://quirktools.com/screenfly/	24
7	Pautas de trabajo	25

NOTA: Para evitar caracteres extraños al utilizar “copiar y pegar”, se proporciona el archivo “PO.Recursos.zip” con los códigos identificados con la etiqueta en cada apartado de este guion.

XXXX

En la carpeta “images” se proporcionan los iconos de HTML5 y CSS para añadir en los sitios web validados que cumplan los estándares

1 Objetivos

En esta práctica se va a realizar la configuración del entorno de trabajo del estudiante para el desarrollo de la asignatura.

Además, 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.**

2 Preparación del puesto de trabajo

Las herramientas que vamos a utilizar en la asignatura son:

- Editores: Para editar el código de los estándares y lenguajes vamos a utilizar fundamentalmente tres editores, Visual Studio Code, Brackets y NotePad++
- 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.
- GitHub Pages: repositorio de sitios web donde se desplegarán los sitios web para utilizar las herramientas de validación.

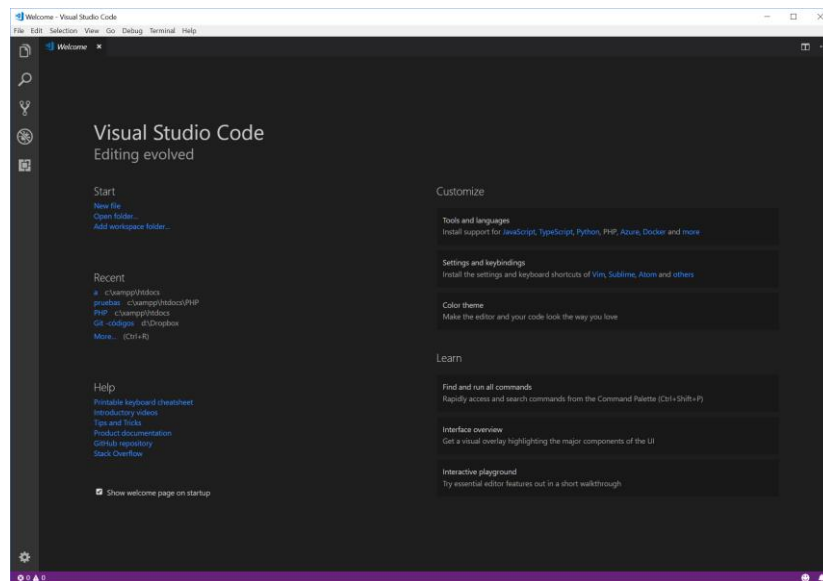
2.1 Editores

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.

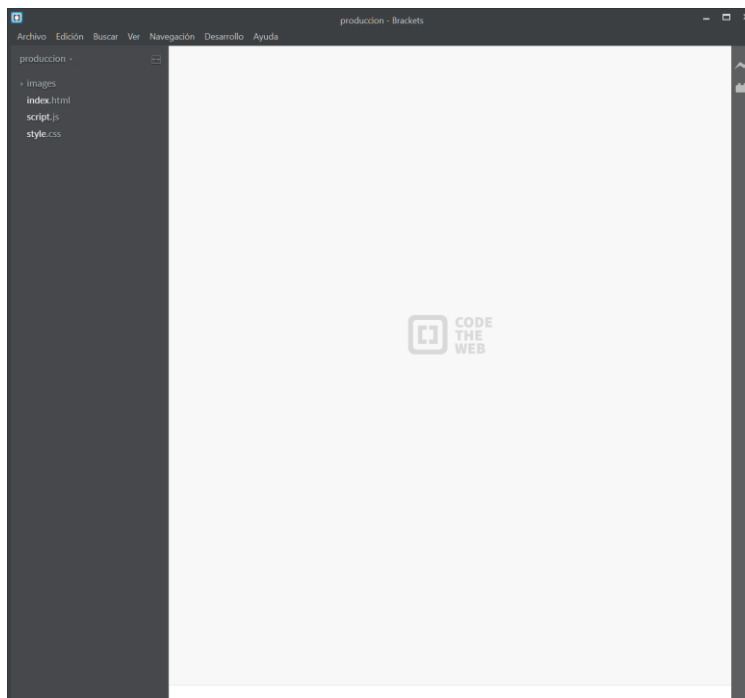
Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código.



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.

Brackets utiliza codificación en UTF-8.

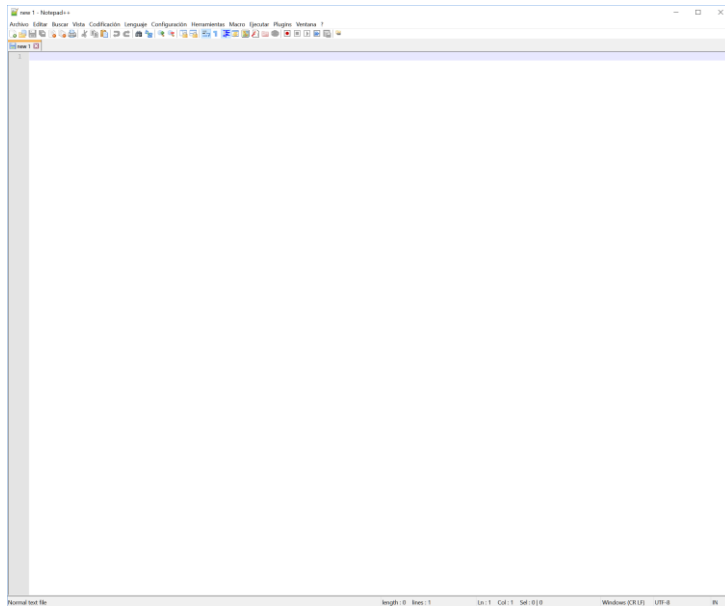




2.1.3 Editor Notepad++

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.

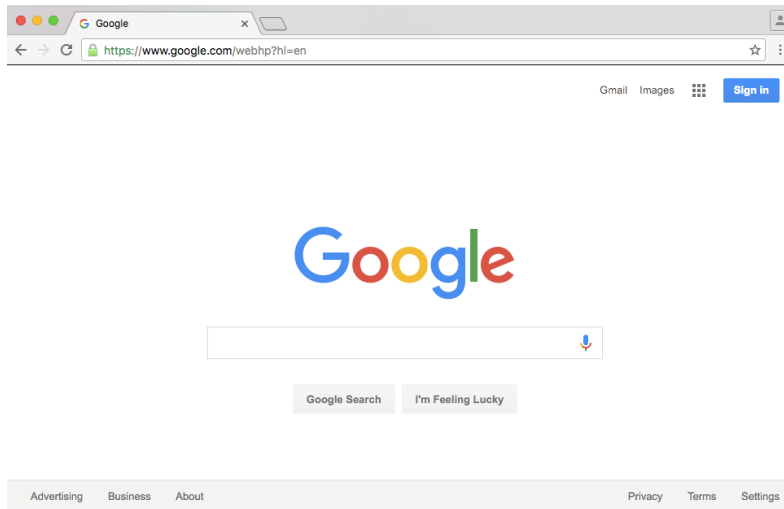


2.2 Navegadores

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

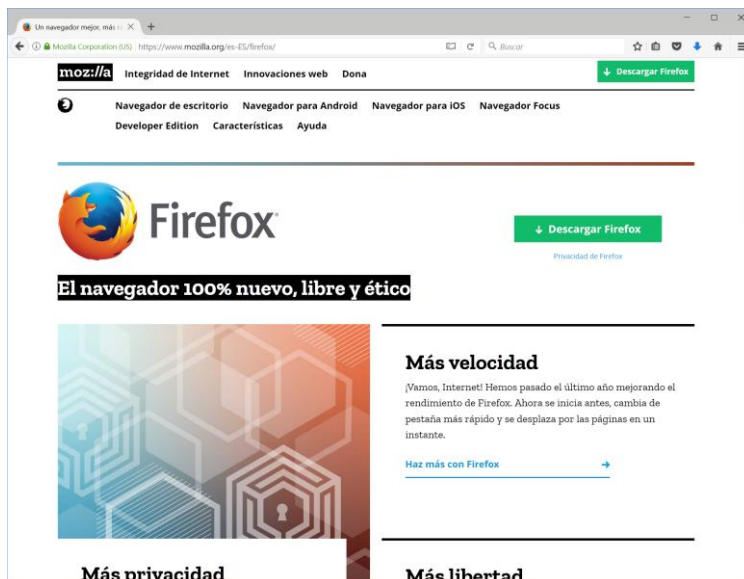
2.2.1 Chrome

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

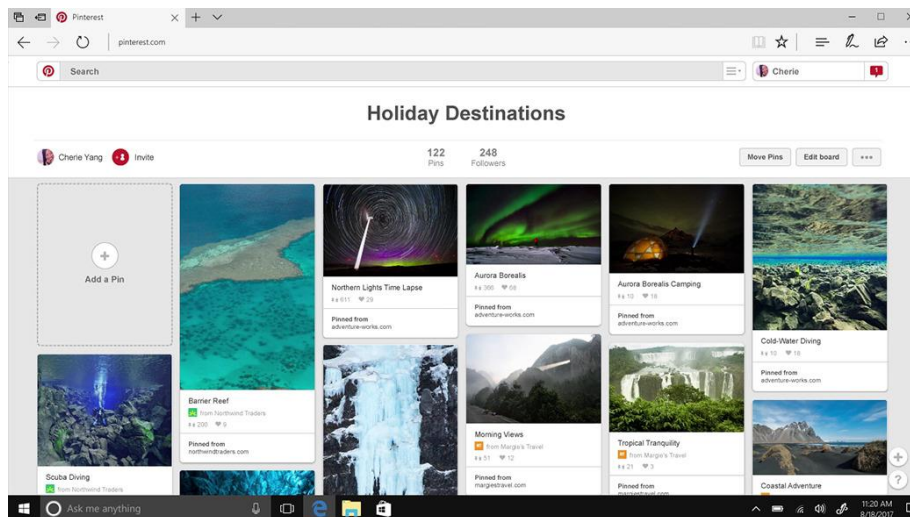
Navegador web desarrollado por la Fundación Mozilla, fecha de lanzamiento 23 de septiembre de 2002. Versiones para Windows, macOS, Linux, Android y iOS. Desarrollado en C++, XUL, XBL, JavaScript, C y Rust. Motor de renderizado Gecko



Microsoft Edge



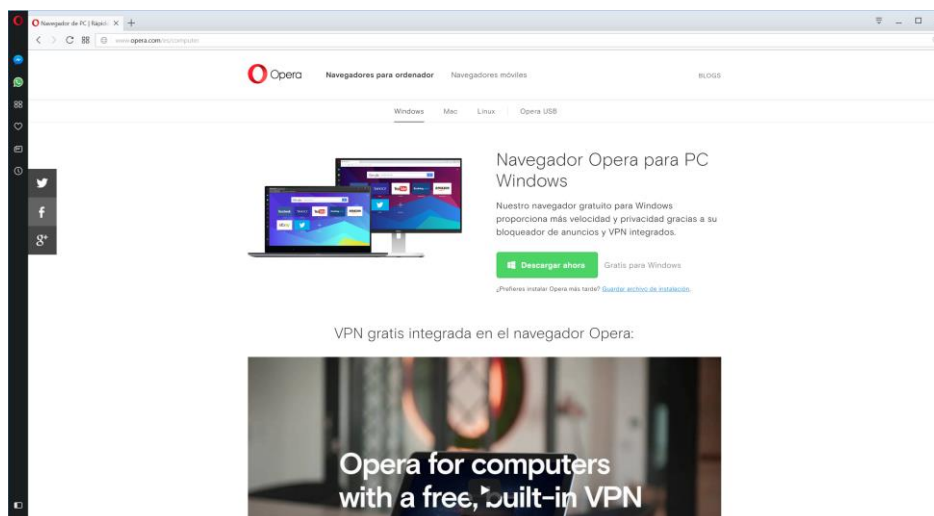
Navegador web desarrollado por Microsoft, fecha de lanzamiento 29 de julio de 2015. Versiones para Windows, macOS, Android y iOS. Desarrollado en C++. Motor de renderizado Blink. Basado en Chromium desde el 15 de enero de 2020.



2.2.3 Opera



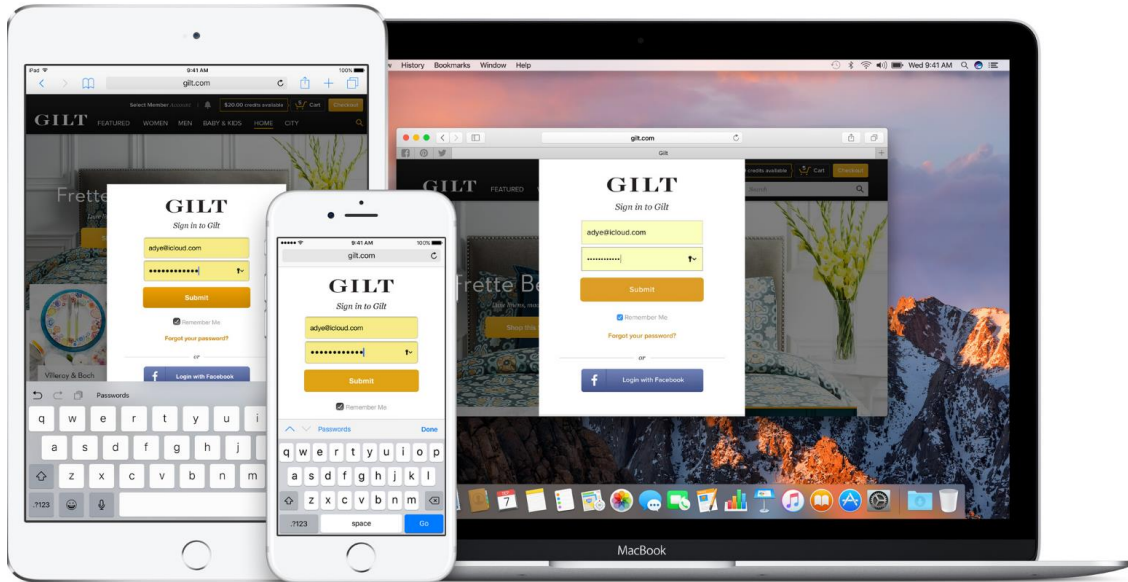
Navegador web desarrollado por Opera Software, fecha del lanzamiento Inicial 10 de abril de 1995. Versiones para Windows, macOS, Linux, FreeBSD, Solaris, Android y iOS. Desarrollado en C++. Motor de renderizado Blink.





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

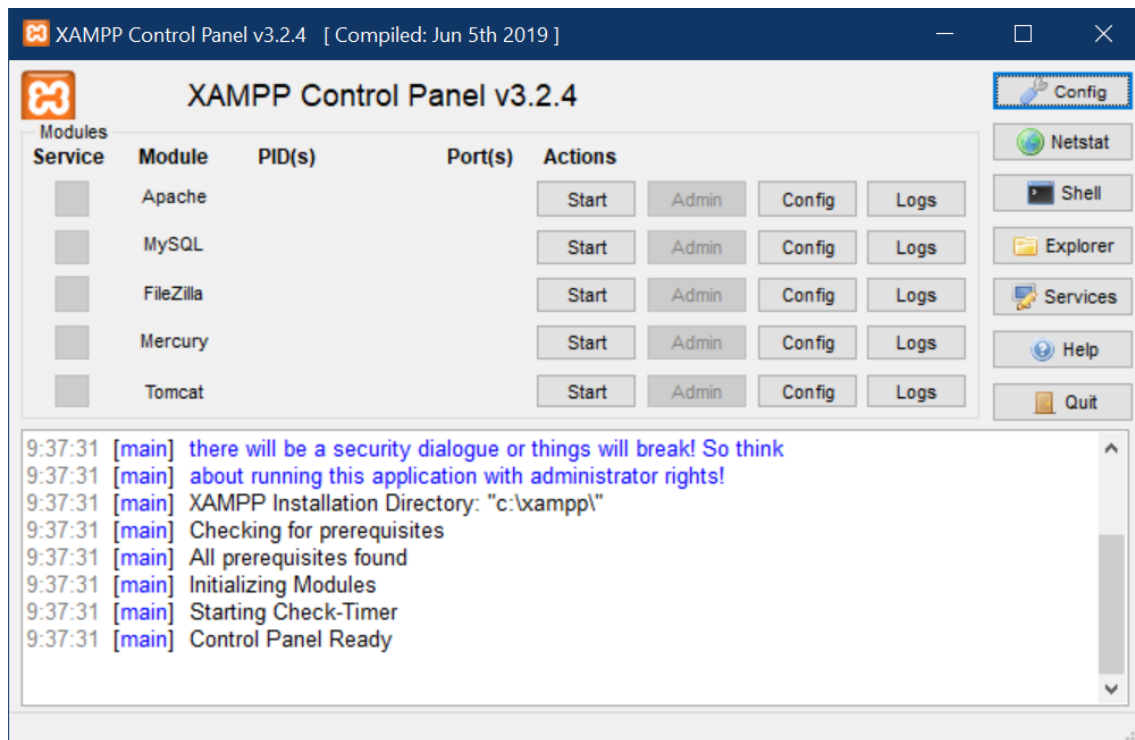
En la asignatura vamos a utilizar XAMPP para tener un **Apache y MySQL** en el Bloque de Computación en el Servidor

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.

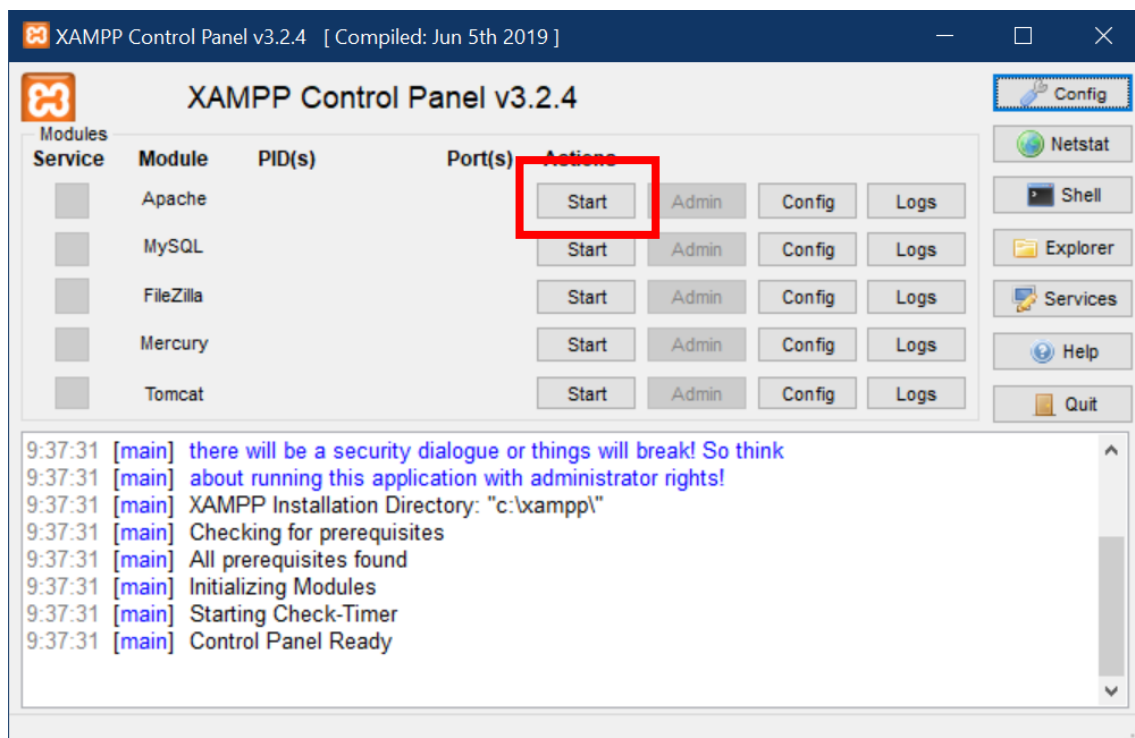
Sitio oficial de descarga de XAMPP: <https://www.apachefriends.org/es/index.html>

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



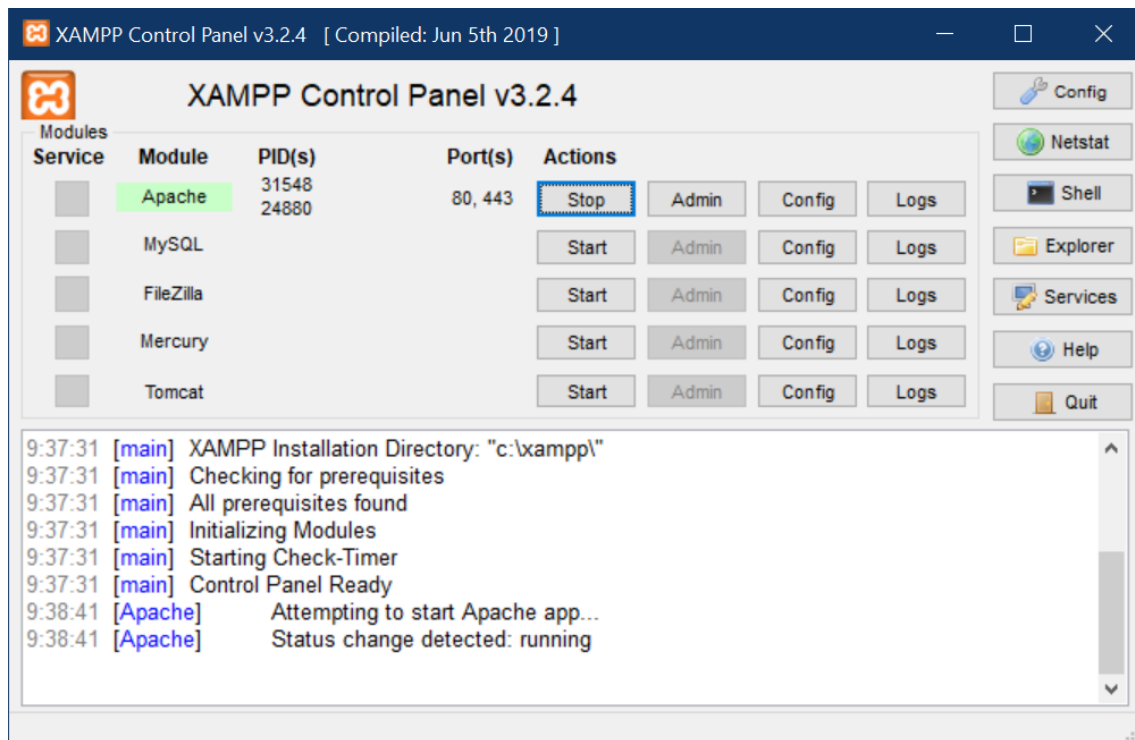
Activar Apache

Pulsamos Start para activar el Apache.



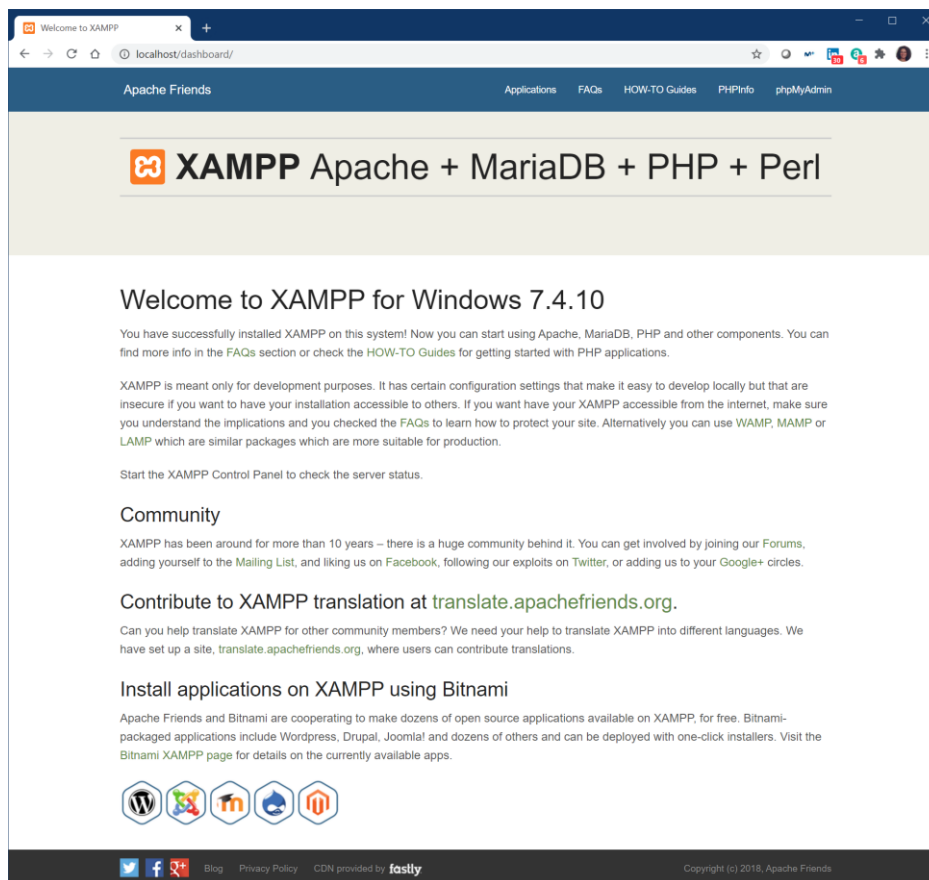
Atención: el Firewall del sistema mostrará un aviso. Debemos permitir el acceso a los puertos por defecto de Apache.

Una vez arrancado el Apache se mostrará el nuevo estado de ejecución de Apache.



Una vez este activo Apache nos conectamos desde un navegador a la dirección “localhost” para comprobar que la ejecución es correcta.

Se muestra la pantalla de bienvenida de XAMPP



El directorio por defecto donde se deben ubicar los archivos del sitio web es: `\xampp\htdocs\`



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>

2.5 Github Pages

Para poder utilizar algunas de las herramientas de la comprobación de la accesibilidad y adaptabilidad se necesita desplegar el código en un repositorio accesible mediante URL. Se utilizará GitHub y su característica Pages para este propósito.

Antes de explicar cómo funciona Github Pages, es importante tener en cuenta los siguientes requisitos previos:

1. Tener instalado Git en la máquina en la que vamos a trabajar. Se pueden seguir las instrucciones disponibles [aquí](#).
2. 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.
3. Una vez completado el paso anterior, iniciar sesión en github.com con el usuario y contraseña previamente creados.

El proceso de preparación y subida de los archivos a GitHub para desplegarlos y tenerlos accesibles como si se tratara de un servidor web se compone de varios pasos.

2.5.1 Preparación de la estructura local

En un primer momento es necesario preparar el código para subirlo al repositorio de una manera ordenada, replicando con nuestra estructura de archivos los componentes clásicos de una página web (empezando por el fichero index.html). Todos estos ficheros deberán estar convenientemente almacenados en un directorio de trabajo dentro del equipo que se esté usando.

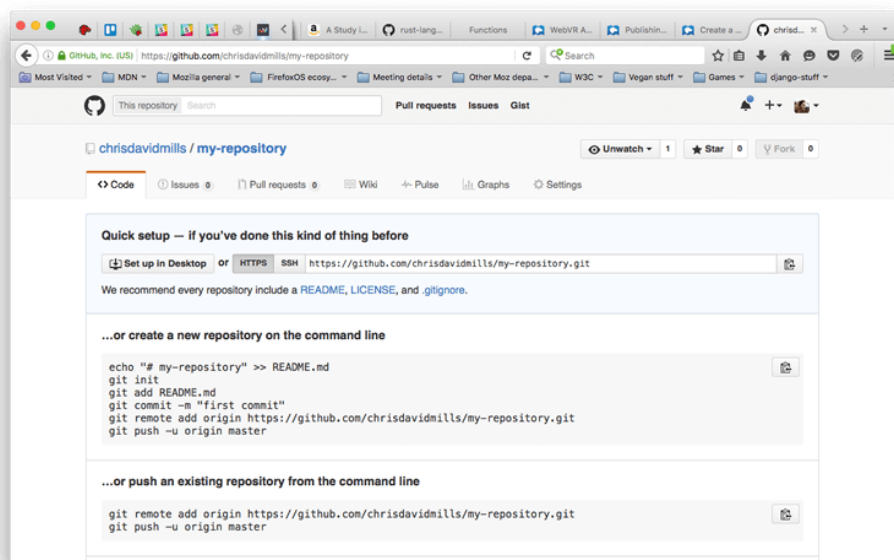
Se empieza por apuntar, usando la herramienta línea de comandos, al directorio de trabajo donde están almacenados los ficheros de la web; en este caso, se usará de ejemplo el directorio usuario/practica-sew. Para ello, una vez dentro de la línea de comandos se introduce el comando `cd usuario/practica-sew`.

Estando dentro del directorio anteriormente mencionado, se procederá a la inicialización del repositorio con el comando `git init`, ejecutado también desde línea de comandos.

2.5.2 Creación del repositorio 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 (+) disponible en la esquina superior derecha del interfaz. En la página relacionada con la creación del repositorio será necesario rellenar el nombre del repositorio (por ejemplo, “sew”) y una descripción del mismo.

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 el caso que nos ocupa, la forma de interacción que interesa es la segunda, que se encuentra bajo el epígrafe “... or push an existing repository from the command line”.



2.5.3 Subiendo los archivos a Github

Utilizando los comandos de la opción mencionada anteriormente, se procederá a subir los ficheros que componen la web de la práctica al repositorio que se acaba de crear en github.

En un primer instante se ejecutará, por línea de comandos y estando localizado en el directorio donde se encuentren los ficheros (usuario/practica-sew), el comando:

add origin https://github.com/uoXXXXXX/sew.git.

Una vez ejecutado el comando anterior, que vincula el directorio local de trabajo con el repositorio creado en github, es necesario añadir los ficheros de dicho directorio al repositorio utilizando los siguientes comandos (introducidos de uno en uno y pulsando enter cada vez que se introduce uno):

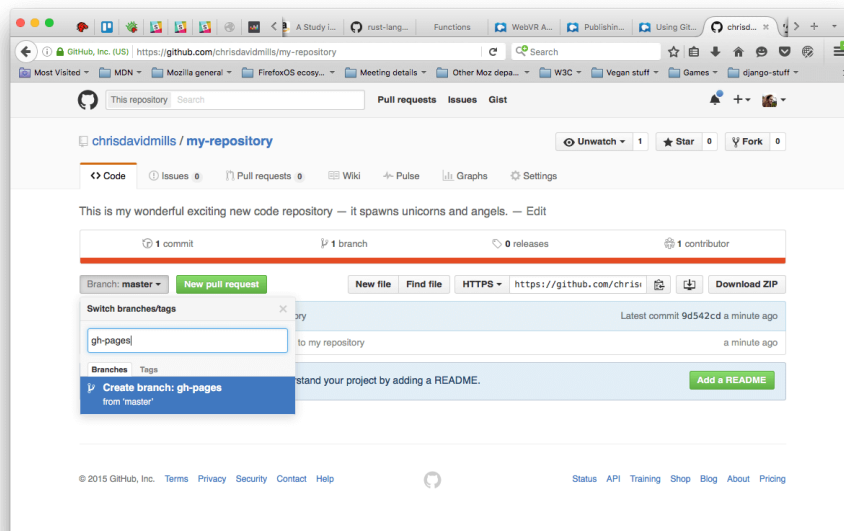
git add -all

git commit -m 'ficheros de la practica de sew'

Por último, y para hacer que los ficheros locales se suban a Github y estén disponibles para su acceso, será necesario introducir el comando:

git push -u origin master

Toda vez que los ficheros ya se han subido a github, es conveniente crear una nueva rama en el repositorio de nombre gh-pages. Para ello, en primer lugar, será necesario actualizar la ventana del navegador donde se muestra github para que se muestren los últimos cambios realizados. Con este refresco la página debería tener un contenido similar al que se muestra en la pantalla a continuación:



Pulsando sobre el control dropdown donde se lee “Branch: master” sale un desplegable con una caja de texto donde es necesario escribir el texto gh-pages y seleccionar la opción “Create branch: gh-pages”.

Una vez creada la nueva rama de nombre gh-pages será necesario entrar en la configuración del repositorio (opción Settings del menú) y marcar la opción “Branches” del menú lateral. Al acceder a dicha opción se muestran una serie de elementos de configuración entre los que se encuentra “Default branch”, que establece cual es la rama por defecto del repositorio; en el selector dropdown debajo de dicho texto será necesario seleccionar la rama gh-pages y pulsar el botón que permite almacenar la configuración.

A partir de este momento, los ficheros componen la web de la práctica y que se han subido al repositorio de github son accesibles como si de una web convencional se tratara. Así, accediendo a la URL correspondiente se visualizará en el navegador la web de la práctica compuesta por el alumno. Según los datos de este ejemplo, la url a introducir para visualizar la web sería: <https://uoXXXXXX.github.io/sew>.

3 Primer sitio web

3.1 Archivo index.html

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 busca este fichero y lo muestra.

Vamos a realizar la creación de este archivo:

- Abrir el editor de texto
- Crear un nuevo archivo y copiar el siguiente código

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

<body>
  <!-- Datos con el contenido que aparece en el navegador -->
  <h1>¡Hola mundo!</h1>
</body>
</html>
```

Html

- Guardar el archivo con nombre `index.html`

4 Validadores

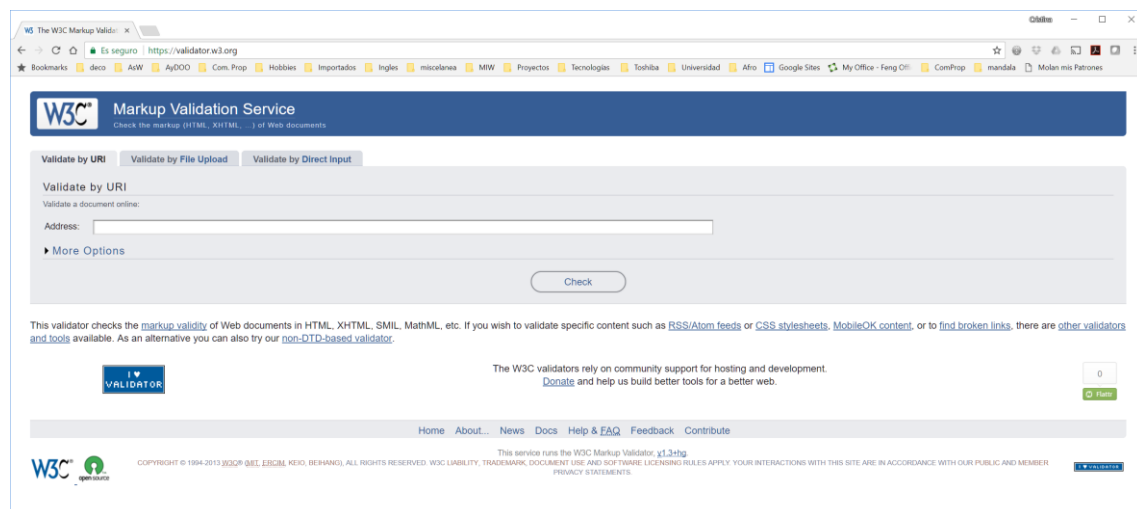
4.1 HTML <https://validator.w3.org/>

Los documentos HTML se inician con elementos que indican a los navegadores cómo deben interpretarlos y a los servicios de validación cómo deben probarlos.

El W3C proporciona un servicio online y gratuito que nos permite comprobar la validez de los documentos HTML. Más información https://www.w3.org/wiki/Validating_your_HTML

Pasos:

- Abrir en el navegador la página <http://validator.w3.org/>



- Seleccionar la forma de validación:

- Por URL: introducir la dirección de GitHub Pages
<https://uoXXXXXX.github.io/sew>
- Por fichero: seleccionando el fichero local
- Copiando directamente el código (DESACONSEJADO)
- Pulsar en “Check” que corresponda

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

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

Para forzar un error vamos a eliminar la etiqueta de cierre `</h1>` del encabezado de la página. Pasamos el validador y el resultado será:

Nu Html Checker

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

Showing results for contents of text-input area

Checker Input

Show ☒ source ☐ outline ☐ image report

Check by

```
<!DOCTYPE HTML>
<html lang="es">
<head>
  <!-- Datos que describen el documento -->
  <meta charset="UTF-8">
  <title>Comenzando</title>
</head>
<body>
  <!-- Datos con el contenidos que aparece en el navegador -->
  <h1>Hola mundo!
</body>
</html>
```

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

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](#)

```
r --><h1>Hola
```

Los errores suelen ser descriptivos, y en cualquier caso siempre señalan la línea donde se ha producido el error para su localización más fácil.

Una vez que el documento es HTML Válido podemos añadir a nuestro documento el icono de conformidad HTML5.

```
<a href="https://validator.w3.org/check?uri=referer">  
    </a>
```

iconoHTML



La carpeta “images” debe estar ubicada en el mismo sitio que el archivo index.html

4.2 CSS <https://jigsaw.w3.org/css-validator/>

Las Hojas de Estilo en Cascada (Cascading Style Sheets), son el mecanismo simple que describe cómo se va a mostrar 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. Página Web de CSS: <http://www.w3.org/Style/CSS/>

Vamos a crear una hoja de estilo muy básica para comprobar el funcionamiento del validador.

- Crear en el editor de texto un archivo denominado “estilo.css”
- Añadir el siguiente código:

```
body {  
    font: 400 15px Tahoma; /*Familia sans-serif*/  
    line-height: 1.8;  
    color: #818181;  
}  
h1 {  
    background-color: #FF5733;  
    color: #fff;  
    text-transform: uppercase;  
    padding: 100px 25px;  
    text-align: center;  
}
```

Css

- Editar el fichero “index.html”
- Añadir el enlace a la hoja de estilo, en la cabecera detrás de <title>:

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

enlaceCSS

- Probar en los navegadores el nuevo aspecto de la página

Vamos a realizar la validación de la hoja de estilo que hemos creado.

Pasos:

- 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, there is a 'I ♥ VALIDATOR' button and a note about community support with a 'Donate' link.

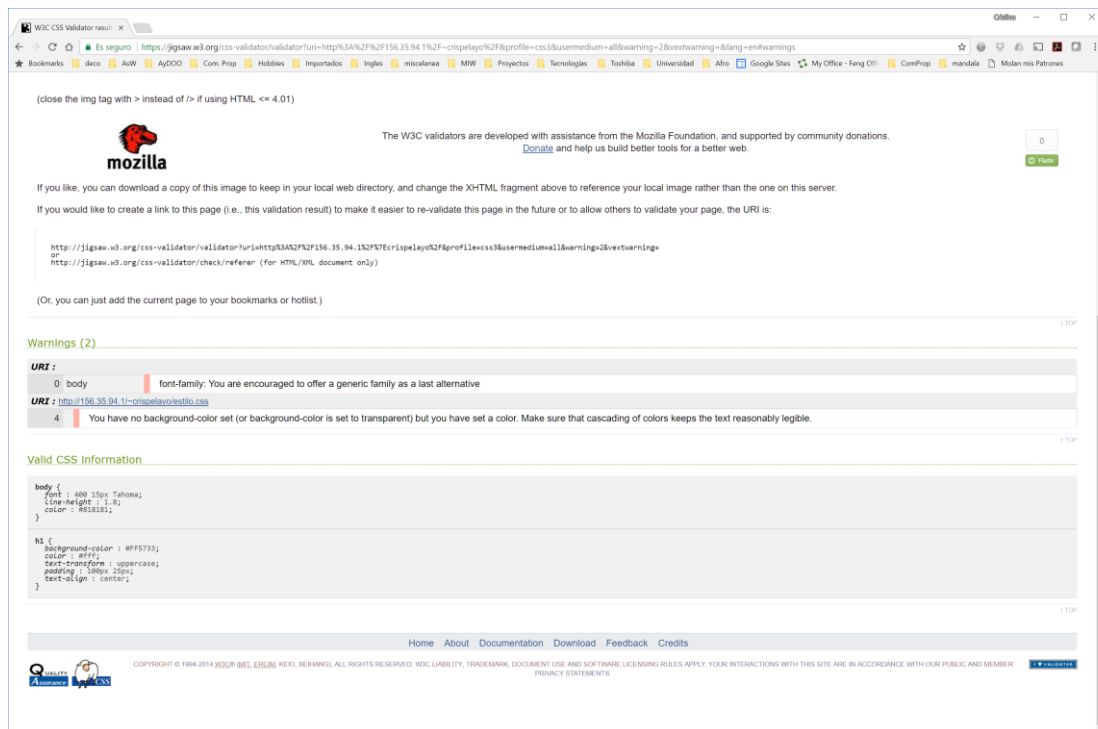
- Seleccionar la forma de validación:
 - Por URL introducir la dirección de GitHub Pages <https://uoXXXXXX.github.io/sew>
 - Por fichero: seleccionando el fichero local
 - Copiando directamente el código (DESACONSEJADO)
- Pulsar en “Check”

El resultado de la validación puede ser:

- CSS válido

The W3C CSS Validator results page shows 'W3C CSS Validator results for http://156.35.94.1/~crispelavoi/ (CSS level 3)'. A green banner says 'Congratulations! No Error Found.' Below this, it says 'This document validates as CSS level 3!'. There are two icons: a 'Valid CSS' icon and a 'Valid CSS' icon. The page also includes a 'Valid CSS Information' section at the bottom.

- CSS válido con warnings que deben ser resueltos. Cambiamos en el validador en “More Options” y la opción “Warnings = All” para que nos proporcione un listado completo de los warnings de la hoja de estilo. Volvemos a validar y comprobamos que tenemos 2 warnings.



Para solucionar los warnings debemos añadir la familia genérica de la fuente definida para h1 y el background-color dentro de body.

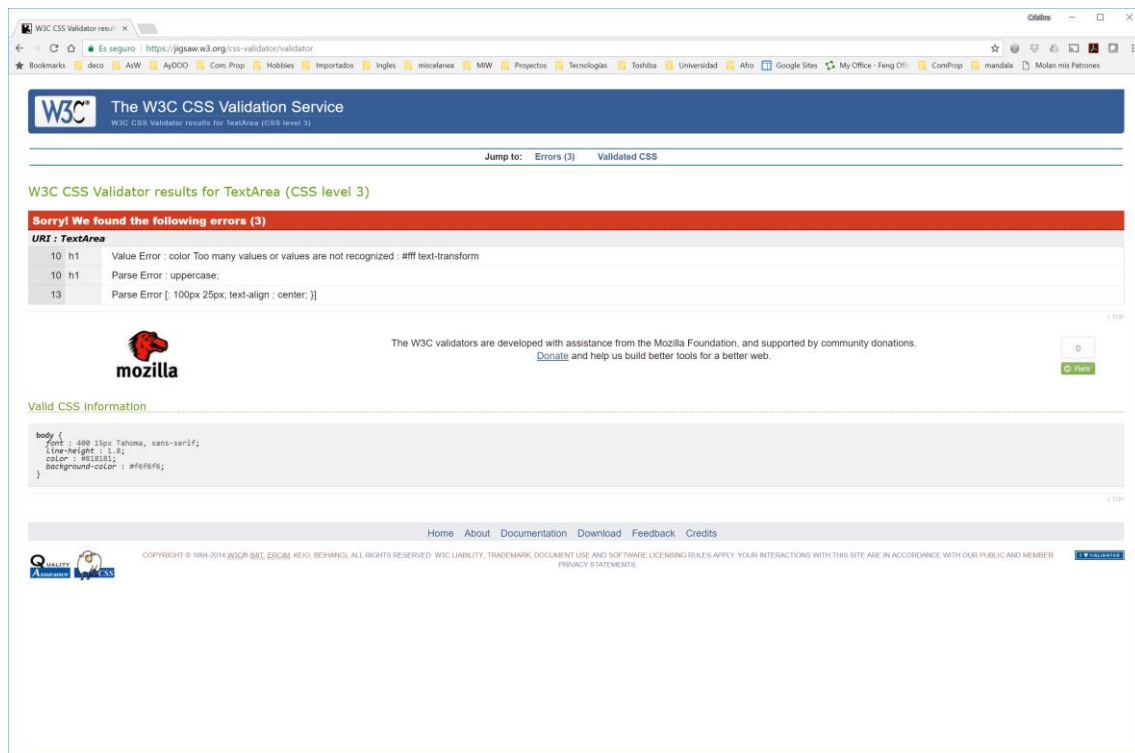
```
body {  
    font: 400 15px Tahoma, sans-serif; /*Familia sans-serif*/  
    line-height: 1.8;  
    color: #818181;  
    background-color: #f6f6f6;  
}  
h1 {  
    background-color: #FF5733;  
    color: #fff;  
    text-transform: uppercase;  
    padding: 100px 25px;  
    text-align: center;  
}
```

- CSS inválido

Para forzar el error eliminamos el “,” de la regla color del encabezado h1

```
color: #fff
```

Pasamos el validador y el resultado será:



Los errores se detallan en cascada en los navegadores, siempre se debe solucionar el primero, validar y comprobar si se han solucionado el resto.

Una vez que el documento es CSS Válido podemos añadir a nuestro documento el icono de conformidad CSS.

iconoCSS

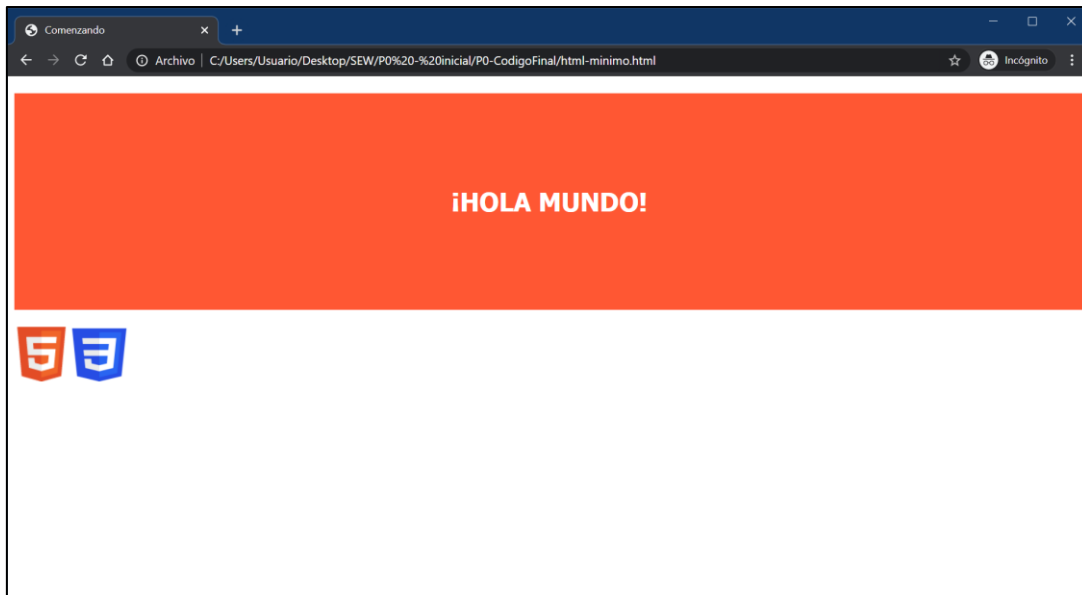
<footer>

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

```
</a>
```

</footer>

El resultado visual de la página realizada es:



5 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.

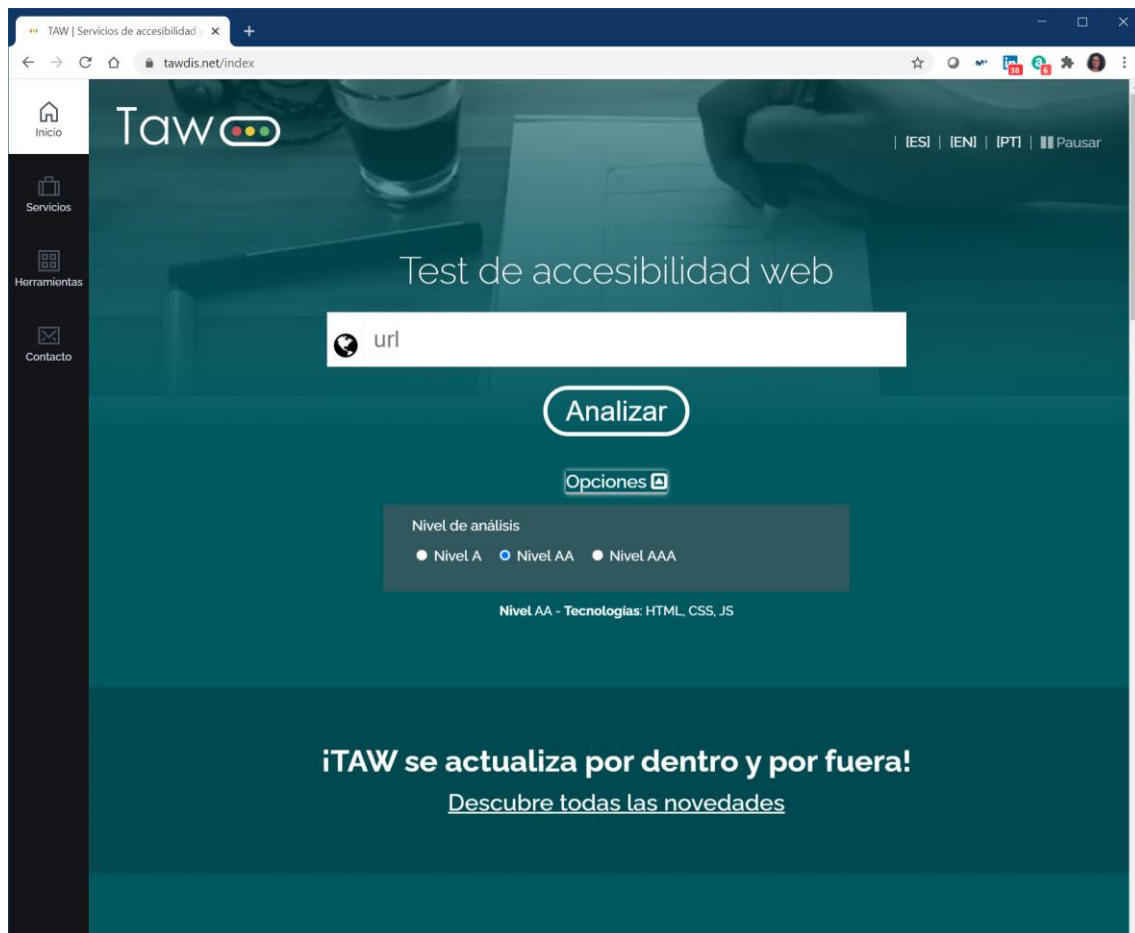
El W3C desarrolla la “Iniciativa de Accesibilidad Web”, conocida como WAI (Web Accessibility Initiative) su objetivo es facilitar el acceso de las personas con discapacidad, desarrollando pautas de accesibilidad, mejorando las herramientas para la evaluación y reparación de accesibilidad Web.

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

5.1 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.

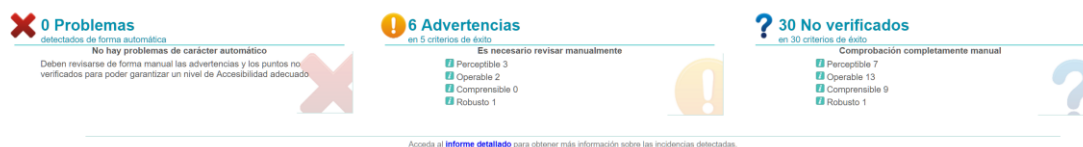
El analizador de páginas es:



Permite seleccionar el nivel de Accesibilidad 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.

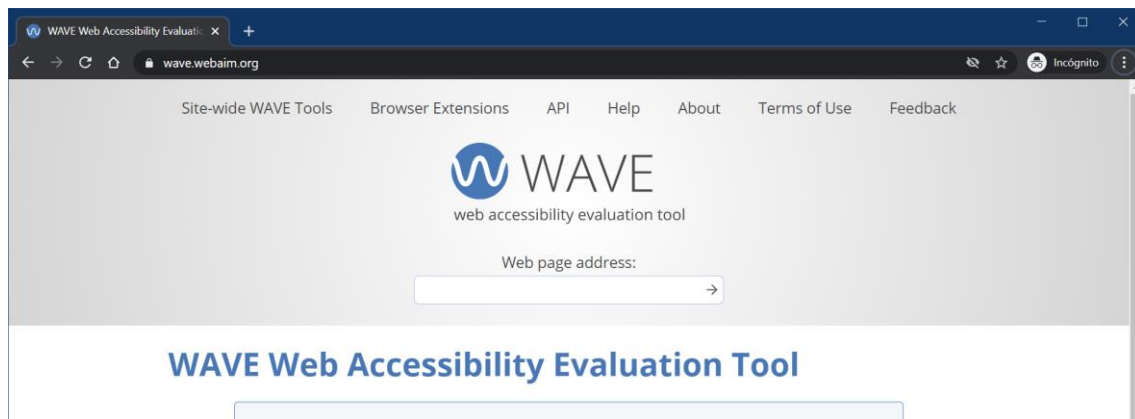


En general el resultado tiene que ser 0 problemas en el análisis automático.

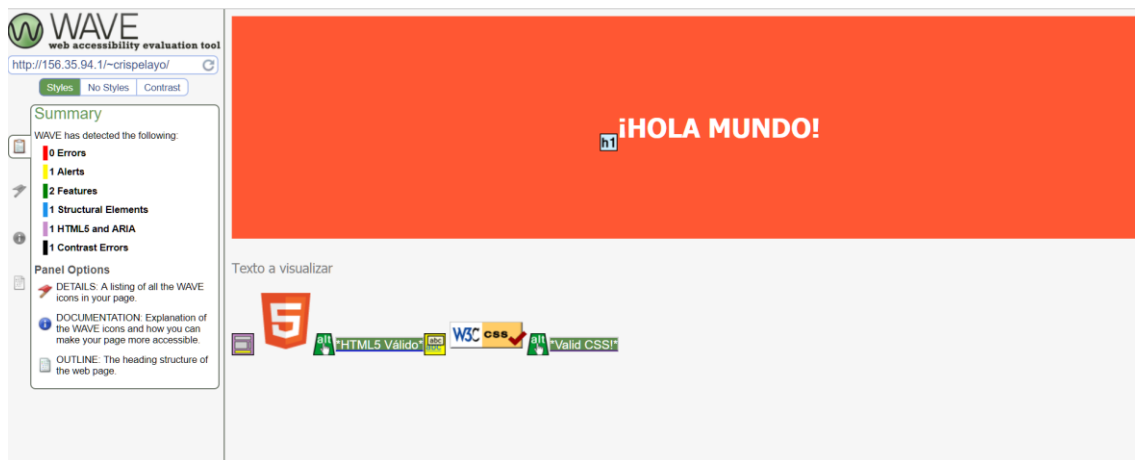


5.2 Herramienta Wave - <http://wave.webaim.org/>

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.



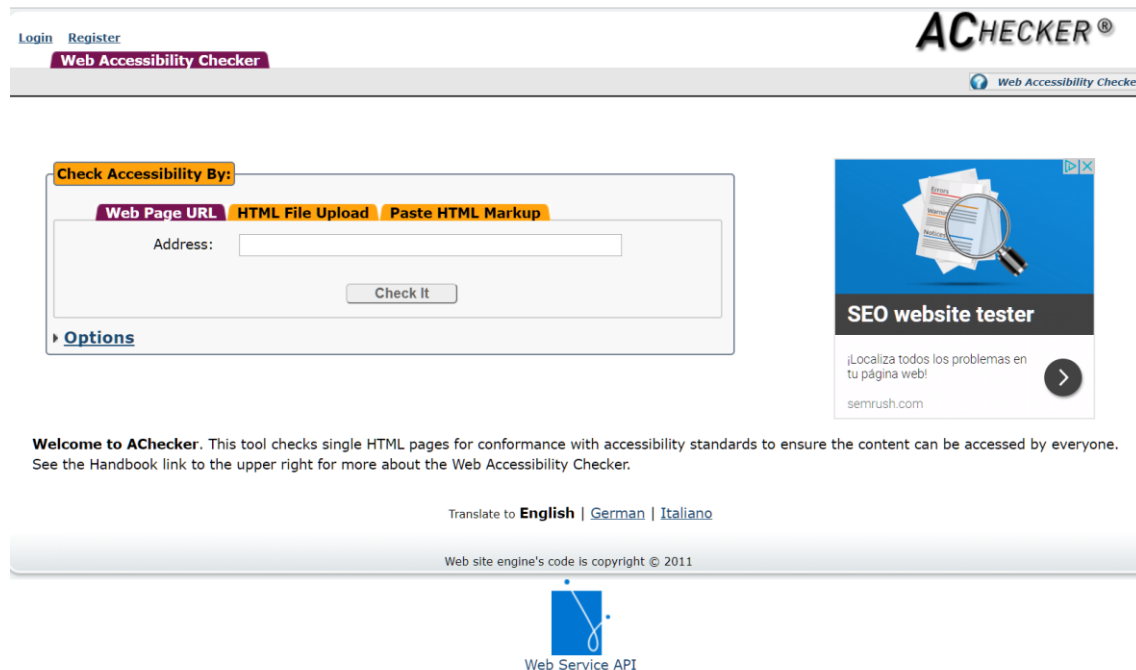
Proporciona un informe “visual” de los problemas de accesibilidad detectados.



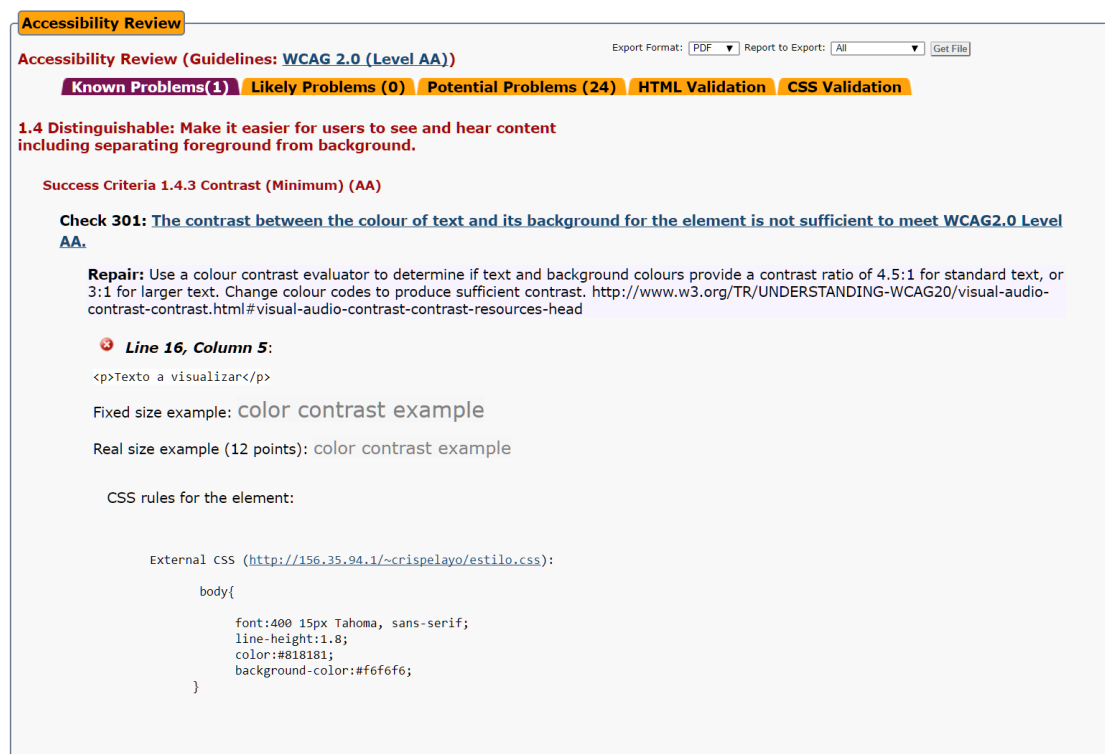
En general debemos garantizar que el número de errores sea 0.

5.3 Herramienta aChecker - <https://achecker.ca>

Esta herramienta evalúa la conformidad de páginas HTML con los estándares de accesibilidad y asegura que el contenido pueda ser accesible por cualquiera.



Proporciona un informe organizado por tipo de problema donde se especifica el elemento y la línea que produce el problema, así como la explicación del mismo.



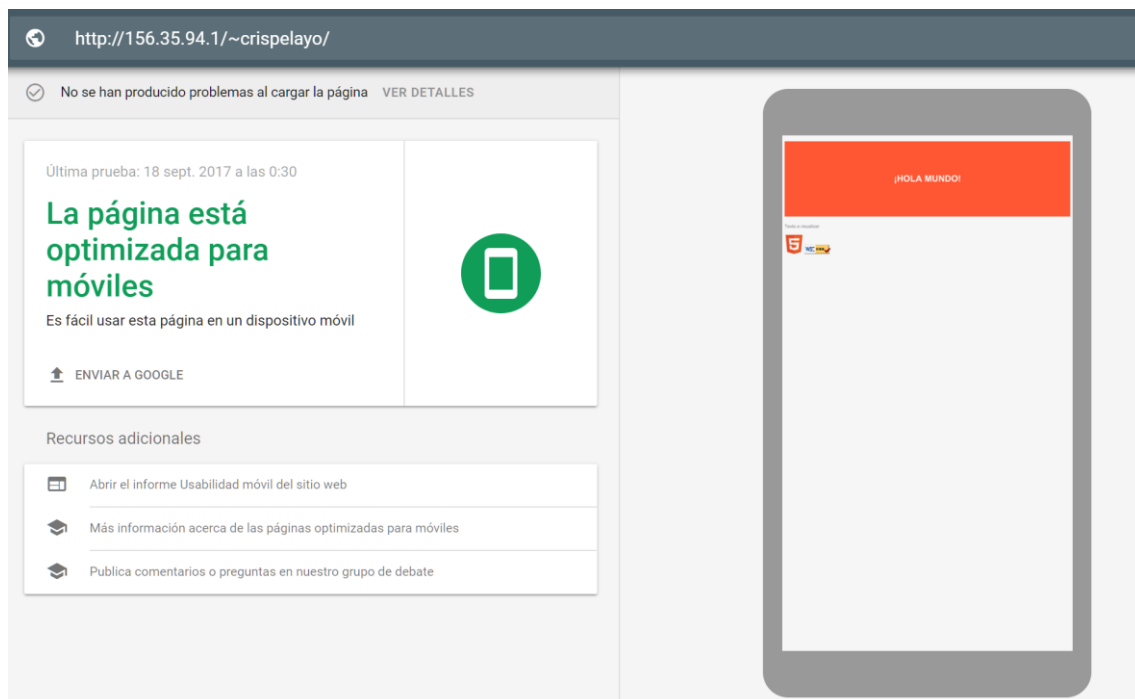
6 Test de adaptabilidad (*responsive*)

Con la popularización del uso de los dispositivos móviles debemos ser conscientes de cómo es la visualización de los sitios web en diferentes tamaños de pantalla. Para ellos vamos a utilizar dos herramientas que permiten comprobar si el diseño es *responsive* o no.

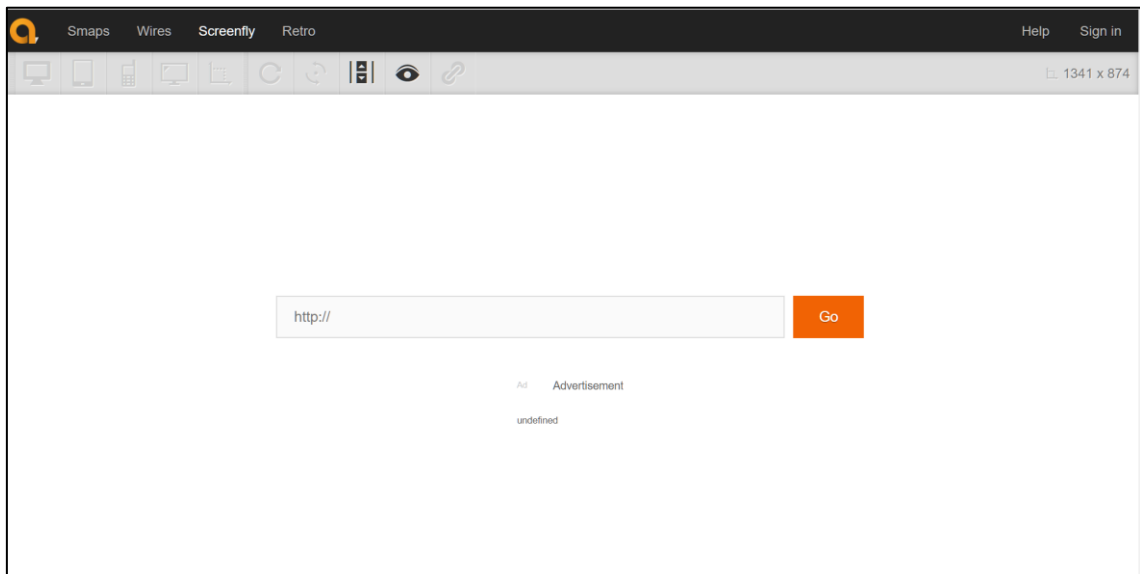
6.1 Google Mobile-friendly – <https://search.google.com/test/mobile-friendly>



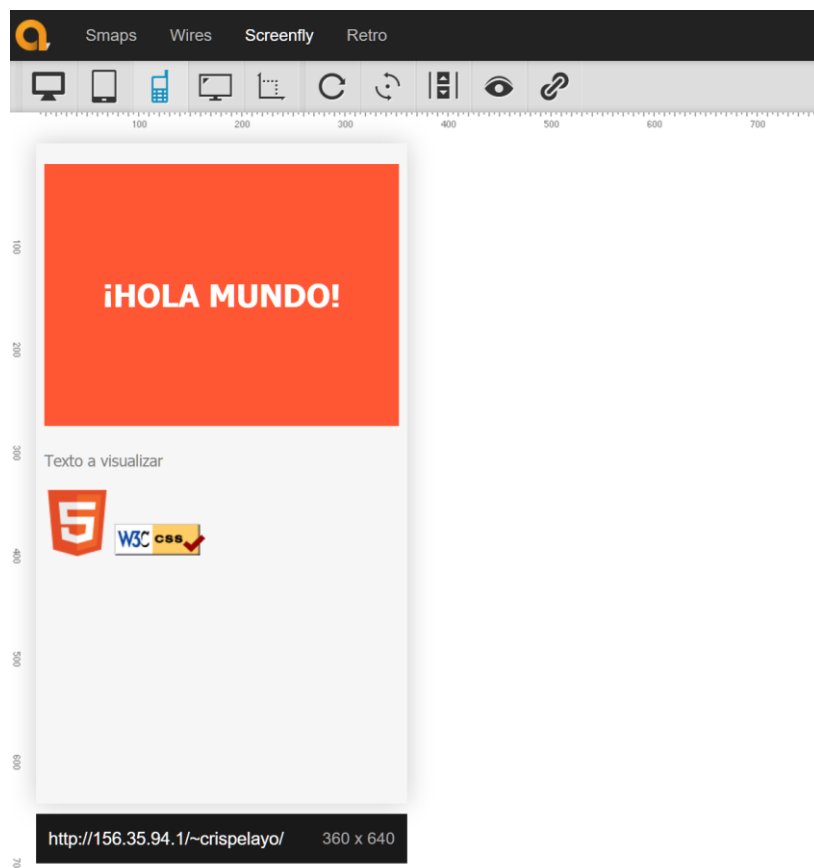
El resultado de la prueba:



6.2 Screenfly - <http://quirktools.com/screenfly/>



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



7 Pautas de trabajo

En todas las prácticas de la asignatura debemos siempre tener en cuenta el cumplimiento de los estándares, la accesibilidad y la adaptabilidad, por ello, recuerda que:

- Debes comprobar la validez del HTML cada vez que hagas modificaciones en los archivos HTML
- Debes comprobar la validez del CSS cada vez que hagas modificaciones en los archivos CSS
- Debes comprobar la accesibilidad y garantizar 0 errores en modo automático
- Debes comprobar la adaptabilidad

En la asignatura entregar archivos con errores de validación en HTML y CSS implicará una calificación de 0 puntos en la práctica implicada.