Conceptos de Accesibilidad



ÍNDICE

[Desarrollando para Accesibilidad Web 3](#_Toc95688548)

[Asociar una etiqueta con cada control de formulario 3](#_Toc95688549)

[Incluir texto alternativo para las imágenes 3](#_Toc95688550)

[Identificar el idioma de la página y los cambios de idioma 3](#_Toc95688551)

[Utilice el marcado para transmitir significado y estructura 3](#_Toc95688552)

[Ayudar a los usuarios a evitar y corregir errores 4](#_Toc95688553)

[Refleja el orden de lectura en el orden del código. 5](#_Toc95688554)

[Escribir código que se adapte a la tecnología del usuario 6](#_Toc95688555)

[Proporcionar significado para elementos interactivos no estándar. 6](#_Toc95688556)

[Asegúrese de que todos los elementos interactivos sean accesibles desde el teclado 7](#_Toc95688557)

[Evite CAPTCHA siempre que sea posible 8](#_Toc95688558)

[IMPLEMENTACIÓN EN ALGUNAS PÁGINAS QUE YA HEMOS CREADO 8](#_Toc95688559)

[PRUEBAS ACCESIBILIDAD CON WAVE 13](#_Toc95688560)

[HERRAMIENTAS DE ACCESIBILIDAD WEB 15](#_Toc95688561)

[HERRAMIENTAS ONLINE 15](#_Toc95688562)

[TAW 15](#_Toc95688563)

[HERA 15](#_Toc95688564)

[WEB ACCESIBILITY CHECKER 16](#_Toc95688565)

[CYNTHIA SAYS 17](#_Toc95688566)

[ACCESS COLORS 17](#_Toc95688567)

[WAVE 18](#_Toc95688568)

[TENON.IO 18](#_Toc95688569)

[PLUG INS 18](#_Toc95688570)

[ACCESIBILITY EVALUATION TOOLS (FIREFOX) 18](#_Toc95688571)

[ACCESIBILITY DEVELOPER TOOLS (CHROME) 19](#_Toc95688572)

[WEB ACCESIBILITY TOOLBAG (OPERA E INTERNET EXPLORER) 21](#_Toc95688573)

[LOCALES 21](#_Toc95688574)

[ACCESS SNIFF 21](#_Toc95688575)

[HTML CODE SNIFFER 22](#_Toc95688576)

[GRUNT CLI 23](#_Toc95688577)

[GRUNT ACCESIBILITY 24](#_Toc95688578)

[GRUNT HTML VALIDATOR 25](#_Toc95688579)

[¿Cuál es la mejor aplicación? 25](#_Toc95688580)

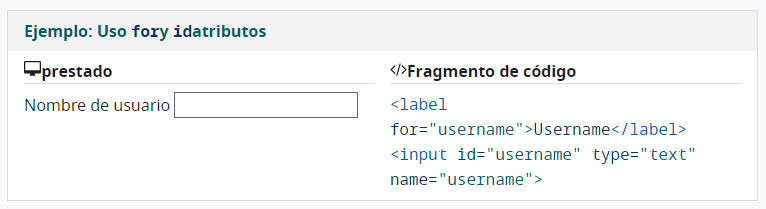
# Desarrollando para Accesibilidad Web

En este documento se presentarán ciertas consideraciones para tener en cuenta a la hora de desarrollar una web más accesible.

Estos consejos son buenas prácticas para ayudarlo a cumplir con los requisitos de las Pautas de accesibilidad al contenido web (WCAG).

## Asociar una etiqueta con cada control de formulario

Use un foratributo en el <label>elemento vinculado al idatributo del elemento de formulario, o use atributos WAI-ARIA . En situaciones específicas, puede ser aceptable ocultar <label>elementos visualmente, pero en la mayoría de los casos se necesitan etiquetas para ayudar a todos los lectores a comprender la entrada requerida.



## Incluir texto alternativo para las imágenes

Use texto alternativo vacío, alt=""para las imágenes, o inclúyelos en el CSS en su lugar. Las alternativas de texto suelen ser proporcionadas por los responsables del contenido escrito.

## Identificar el idioma de la página y los cambios de idioma

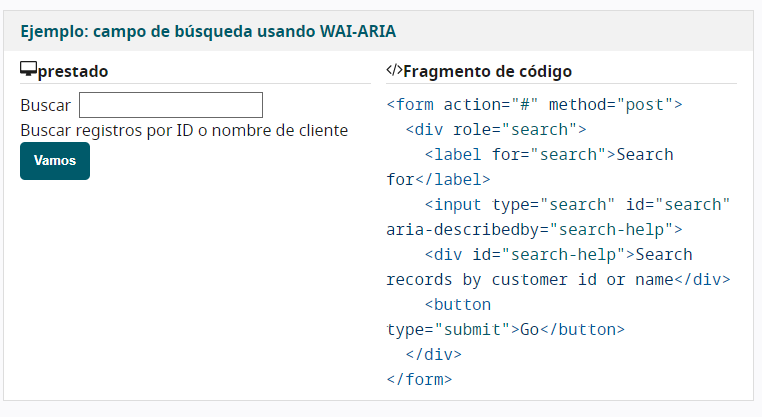
Indicar el idioma principal de cada página mediante el langatributo en la htmletiqueta, por ejemplo <html lang="en">.

## Utilice el marcado para transmitir significado y estructura

Use el marcado apropiado para encabezados, listas, tablas, etc. HTML5 proporciona elementos adicionales, como <nav>y <aside>, para estructurar mejor su contenido.

 Utilizar role="search"para identificar la funcionalidad de búsqueda.





## Ayudar a los usuarios a evitar y corregir errores

Proporcione instrucciones claras, mensajes de error y notificaciones para ayudar a los usuarios a completar formularios en su sitio. Cuando ocurre un error:

* Ayudar a los usuarios a encontrar dónde está el problema
* Proporcionar explicaciones específicas y comprensibles.
* Sugerir correcciones



## Refleja el orden de lectura en el orden del código.

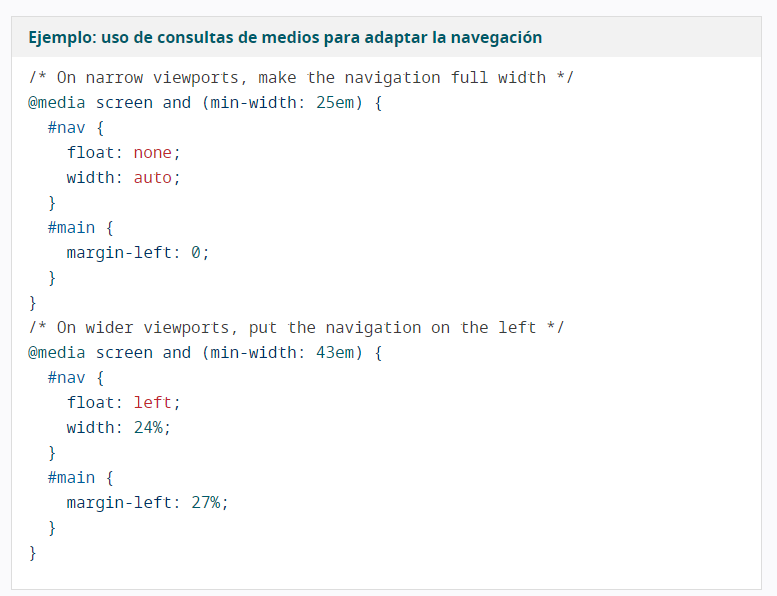
El orden en el que aparezcan los elementos en el código deben coincidir con el orden cronológico de la información que aparece.

Una forma en la que podemos verificarlo es eliminar el estilo CSS y revisar que el orden del contenido tenga sentido.



## Escribir código que se adapte a la tecnología del usuario

Utilice un diseño receptivo para adaptar la pantalla a diferentes estados de zoom y tamaños de ventana gráfica. Cuando el tamaño de fuente aumenta al menos en un 200 %, evite el desplazamiento horizontal y evite cualquier recorte de contenido. Utilice la mejora progresiva para ayudar a garantizar que la funcionalidad y el contenido principal estén disponibles independientemente de la tecnología que se utilice.



## Proporcionar significado para elementos interactivos no estándar.

Utilice WAI-ARIA para proporcionar información sobre la función y el estado de los widgets personalizados, como acordeones y botones personalizados.

Por ejemplo, role="navigation"y aria-expanded="true". Se requiere código adicional para implementar el comportamiento de dichos widgets, como expandir y contraer contenido o cómo responde el widget a los eventos del teclado.



## Asegúrese de que todos los elementos interactivos sean accesibles desde el teclado

Piense en el acceso al teclado, especialmente cuando desarrolle elementos interactivos, como menús, información sobre el mouseover, acordeones plegables o reproductores multimedia.

Úselo tabindex="0"para agregar un elemento que normalmente no recibe el foco, como <div>o <span>, en el orden de navegación cuando se usa para la interacción. Use secuencias de comandos para capturar y responder a los eventos del teclado.



## Evite CAPTCHA siempre que sea posible

Existen otros medios para verificar que la entrada del usuario fue generada por un ser humano que son más fáciles de usar, como la detección automática o las interacciones de la interfaz.

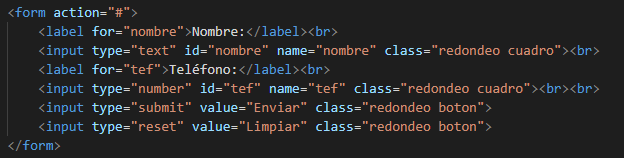
Si realmente necesita incluir CAPTCHA, debe seguir las siguientes pautas para que sea lo más accesible posible:

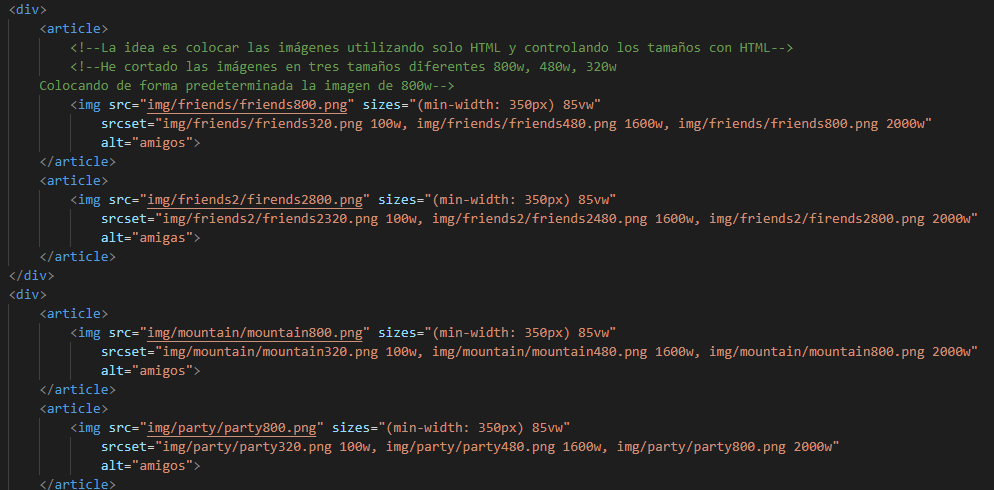
* Proporcionando más de dos formas de resolver los CAPTCHA
* Proporcionar acceso a un representante humano que puede omitir CAPTCHA
* No requiere CAPTCHA para usuarios autorizados.

# IMPLEMENTACIÓN EN ALGUNAS PÁGINAS QUE YA HEMOS CREADO

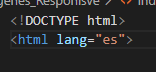
El primer ejemplo lo desempeñamos en un formulario que creamos específicamente para probar la accesibilidad.

Vemos como está asociado a etiquetas <label>.

Para mejorar la accesibilidad en las imágenes lo que tenemos que hacer es incluir un texto alternativo que especifique que hay en ellas.

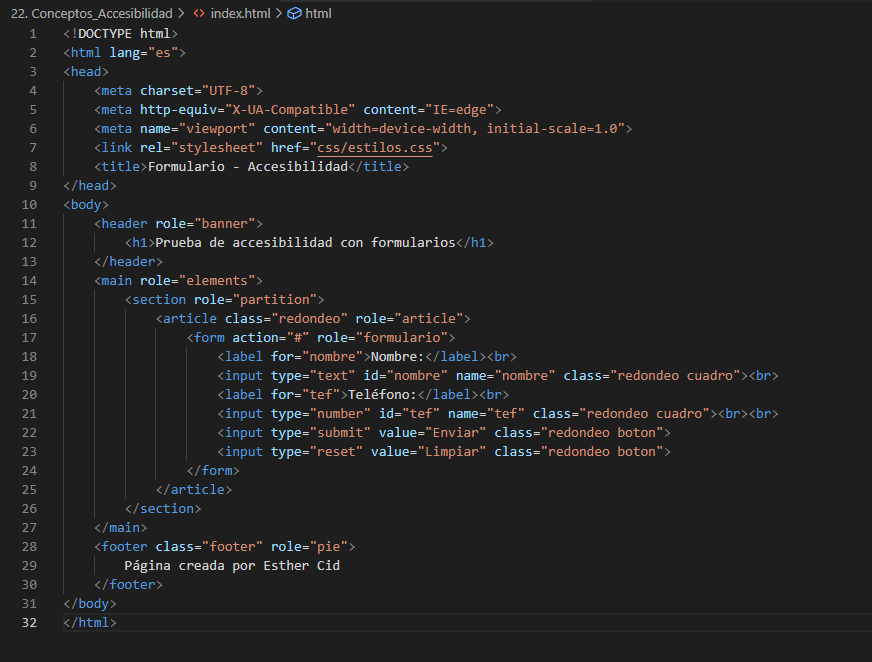


Debemos especificar el idioma en el que se encuentra la página que vamos a desarrollar.



Debemos utilizar las etiquetas semánticas que nos proporciona HTML5 para que quede clara la parte de la página en la que se encuentra cada elemento.

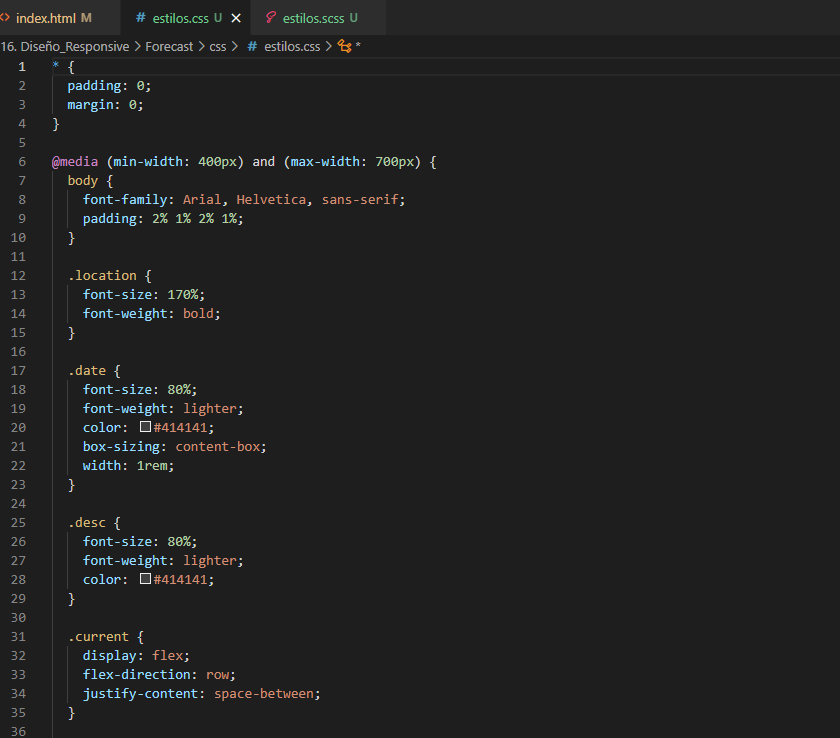
Podemos utilizar el atributo “role” para ayudar a la compresión de la utilización de cada etiqueta.



Los formularios pueden tener un patrón que limite el envío de datos que no se ajusten a él. Otra de las cosas que tenemos que hacer es ayudar al usuario señalándole donde se encuentra el error.



El código debe adaptar el contenido en función de la pantalla en la que se vea, de esta forma lo podemos hacer más accesible en caso de que se acceda mediante un dispositivo con la pantalla más pequeña como una Tablet o un teléfono móvil.





También podemos utilizar una serie de etiquetas que son capaces de, utilizando una serie de widgets personalizados, indicar si ese elemento transmite el estado actual del componente o si, por ejemplo, el botón funciona como una casilla de verificación.



# PRUEBAS ACCESIBILIDAD CON WAVE

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Escala de tiempo

Descripción generada automáticamente con confianza media

# HERRAMIENTAS DE ACCESIBILIDAD WEB

## HERRAMIENTAS ONLINE

### TAW

Lo que tenemos que hacer para utilizarlo es introducir la url de la página que queremos analizar en el buscador.

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

Cuando tengamos la url le damos a Analizar.

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

Nos saca la siguiente información que podemos enviarla a nuestro correo.

### HERA

Intentamos realizar la comprobación utilizando los recursos online que nos facilita.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Para poder utilizarlo y necesitamos descargar el complemento para Firefox que aparece en el siguiente enlace.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

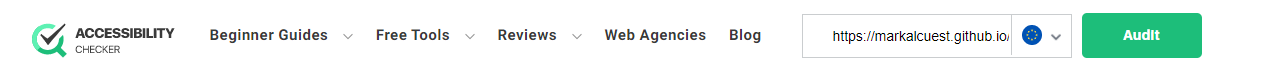
Cuando intentamos añadirlo como complemento a Firefox nos da un error.

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

### WEB ACCESIBILITY CHECKER

Dentro de esta página nos aparece en el navegador un espacio para colocar la página que queremos analizar y un botón para analizarla.



Interfaz de usuario gráfica, Aplicación, Sitio web

Descripción generada automáticamente

### CYNTHIA SAYS

En agosto de 2021 el equipo que llevaba a cabo el proyecto CynthiaSays.com decidió acabar con la vida útil del mismo, por lo que, desde esa fecha no se puede acceder a la página para realizar las comprobaciones de accesibilidad,.

### ACCESS COLORS

Esta página funciona de una manera un poco distinta al resto, lo que tenemos que añadir son, el color que hemos elegido como color primario y el color que hemos elegido como color secundario. la página funciona con tres claves, AA, la cual significa que no hemos pasado el test de color y que debemos aumentar nuestro ratio para que los colores sean más accesibles, Passes AA, es cuando el ratio de contraste es mucho más accesible.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

### WAVE

En la parte superior del documento ya tenemos un ejemplo de cómo funciona Wave.

### TENON.IO

Podemos acceder a un consultoría de accesibilidad gratuito presionando el botón verde que tenemos a la derecha.



Nos hace rellenar un formulario en el que tenemos que especificar nuestro nombre, nuestros apellidos, un número de teléfono y la página web que deseamos que consulte.

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

## PLUG INS

### ACCESIBILITY EVALUATION TOOLS (FIREFOX)

No nos aparece ningún plugin con este nombre, pero nos aparece este otro.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Lo que nos aparece una vez tenemos instalado este plugin es el siguiente elemento en la barra de navegación de Firefox.

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

En este cuadro de advertencia nos señala que tenemos que ir al apartado de DevTools.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

De esa forma ya tenemos iniciada la aplicación.

### ACCESIBILITY DEVELOPER TOOLS (CHROME)

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

Lo podemos instalar en la tienda web de Chrome.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Comprobamos que lo tenemos instalado dentro de todas las extensiones.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Una vez lo tenemos instalado nos aparecerá la pestaña dentro de las extensiones instaladas para Google Chrome, dentro podemos modificar los datos que queremos que analice la aplicación.

### WEB ACCESIBILITY TOOLBAG (OPERA E INTERNET EXPLORER)

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Nos la podemos instalar desde esta página web.

Texto

Descripción generada automáticamente

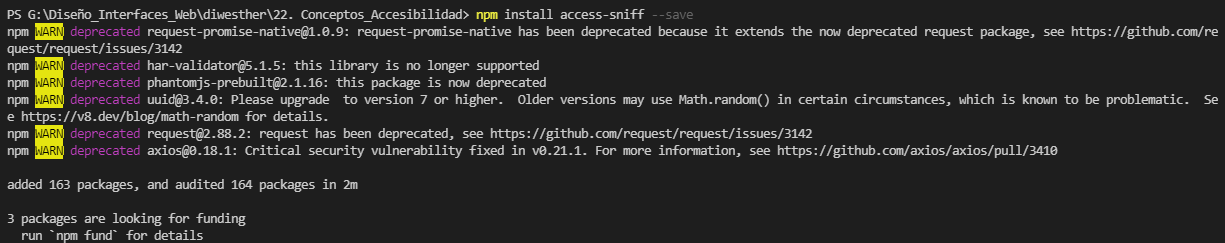
No nos permite instalar la aplicación por un problema que hay en él XML.

## LOCALES

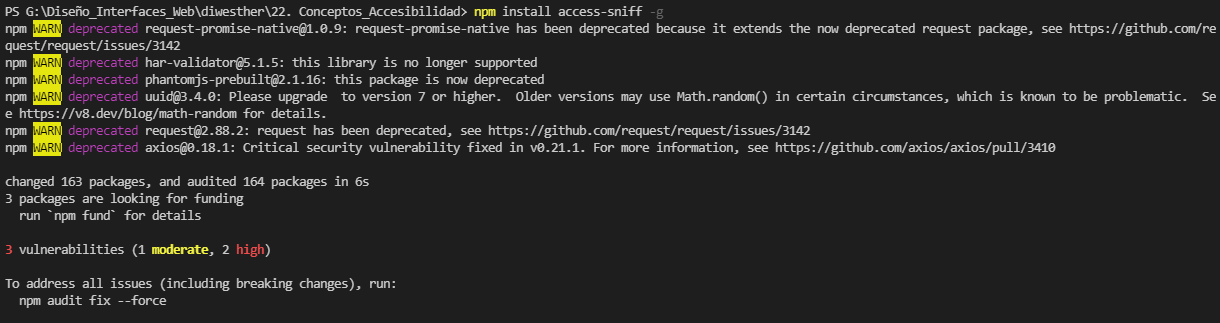
### ACCESS SNIFF

Se instala mediante un paquete de nodejs. “*npm install access-sniff –save*”.

Lo instalamos dentro del documento del que vayamos a comprobar su accesibilidad.



Después instalamos el acceso con el siguiente paquete: “npm install access-sniff -g”.

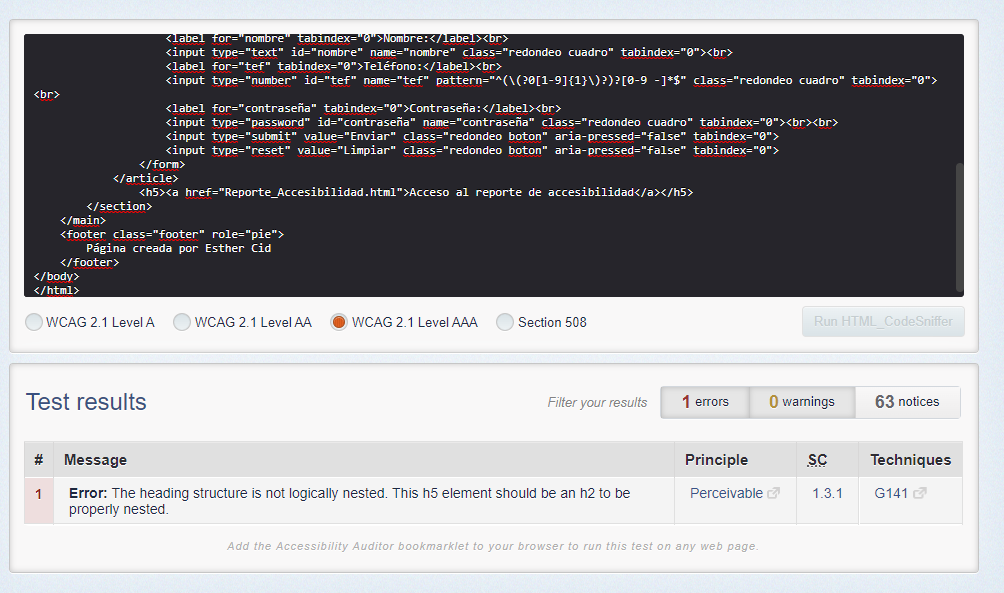


Después podemos comprobar la accesibilidad de un archivo local que tengamos en nuestro disco duro sin necesidad de desplegarlo de la siguiente manera.



### HTML CODE SNIFFER

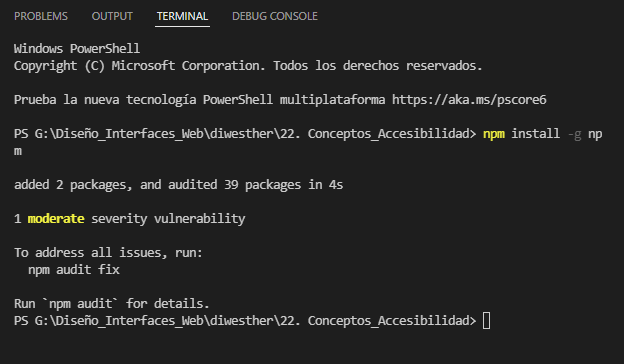
Podemos acceder a él a través de su GitHub, en el mismo GitHub nos permite usar la aplicación de forma OnLine, sin necesidad de descargarlo.



Podemos ver los errores que nos devuelve, nos dice que el “<h5>” que tenemos en el código, estructuralmente debería ser un “<h2>”.

Lo primero que nos indica su GitHub es que debemos actualizar node.js.

### GRUNT CLI



Texto

Descripción generada automáticamente

Instalamos el paquete y nos deberán aparecer una serie de archivos .json dentro de los que aparecen los módulos que acabamos de instalar.

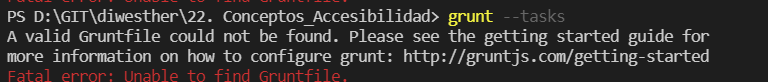
Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

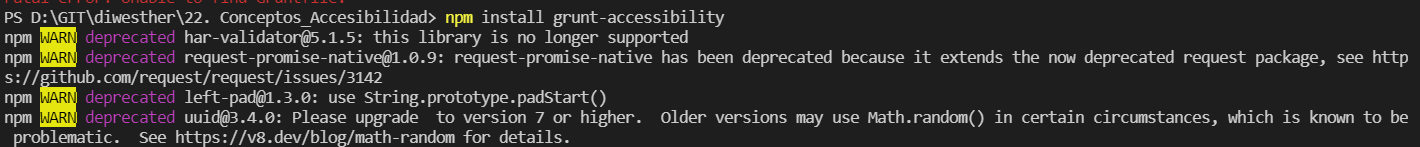
De esta forma podemos acceder a la ayuda que nos proporciona la aplicación.



el resto de la aplicación es completamente inaccesible da errores fatales.

### GRUNT ACCESIBILITY

Este es el módulo de accesibilidad de la anterior aplicación que hemos trabajado.



Le tenemos que añadir la siguiente línea de código al script llamado “grunt.js”.



Dentro del kitáb del desarrollador de esta aplicación nos pide que añadamos el siguiente código dentro de él “package-lock.json” pero en ningún momento especifica en qué parte, presuponemos, que en este código tenemos que poner la url que queremos analizar y la carpeta en la que se encuentra.

Imagen que contiene Forma

Descripción generada automáticamente

Volvemos abrir la ayuda que nos ofrece npm y no encontramos ningún tipo de acceso para ejecutar el comando que hemos añadido anteriormente ni esta parte del json que hemos añadido.

### GRUNT HTML VALIDATOR

Texto

Descripción generada automáticamente

Esta aplicación, no solo da bastantes errores a la hora de instalarla, sino que, no especifican qué paquete tenemos que integrar las siguientes funciones que aparecen a la hora de configurarlo.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

# ¿Cuál es la mejor aplicación?

Sin duda las mejores aplicaciones, o las que me han resultado más útiles son, la primera una de las extensiones que hemos tenido que instalar para poder hacer comprobaciones, en este caso es WAVE, es muy sencilla de utilizar, simplemente tenemos que instalar la extensión en nuestro navegador favorito y la pantalla se divide en dos, un apartado en el que podemos ver todos los fallos elementos que podríamos mejorar para hacer nuestra web más accesible y al otro lado de forma visual podemos ver cuáles son los fallos o los elementos que podríamos mejorar según esta aplicación.

La otra aplicación que más útil me ha resultado es la de RAW, esta aplicación se encuentra online y no tenemos que instalar nada en nuestro ordenador, el problema de esta aplicación es que es más difícil conocer cuáles son los fallos concretos que tenemos en nuestro código, puesto que no nos deja verlos.

El mayor problema que veo nada más aplicaciones es que necesitamos tener nuestra página alojada en un Hosting, y no podemos analizar páginas que tengamos dentro de nuestro ordenador.

Por otra parte, las aplicaciones que tenemos que instalar para poder utilizar dentro de nuestro ordenador y que además funcionan con elementos que todavía no hemos desplegado, tienen una documentación pobre de la que es prácticamente imposible sacar alguna utilidad de ellas.