



## **ELABORAR UN SITIO WEB Y COMPROBAR SU ACCESIBILIDAD.**

**Autor:**

Cristo Manuel Rodríguez Rodríguez

## Contenido

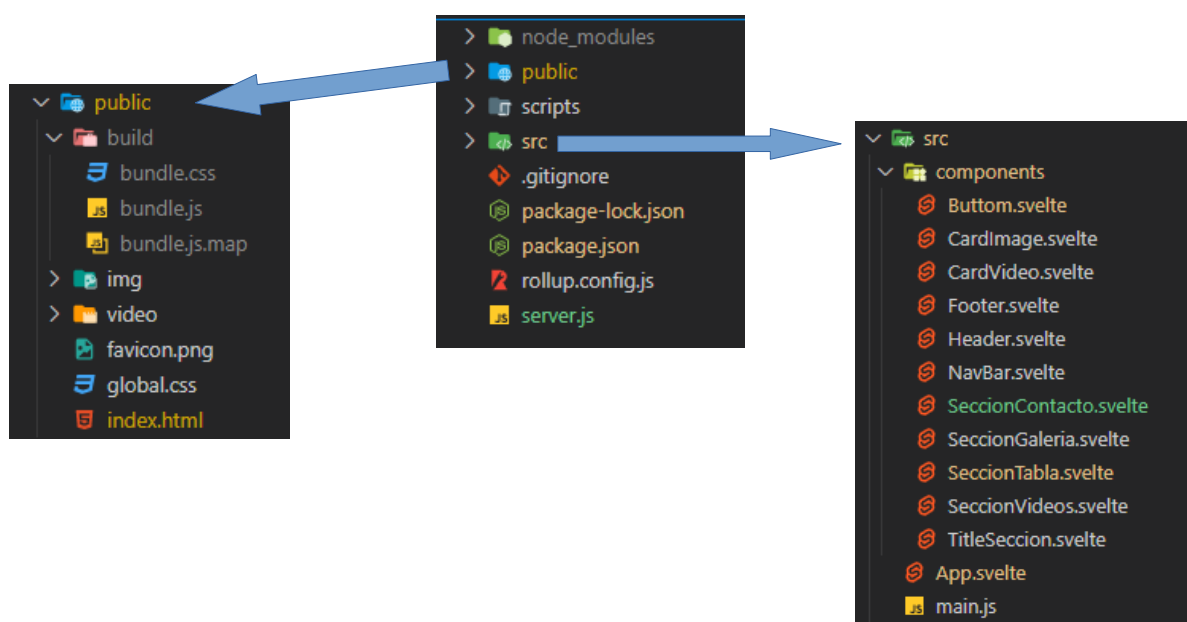
RESUELVE.....	3
SOLUCIÓN.....	4
CONCLUSIONES.....	13
BIBLIOGRAFÍA.....	13

**RESUELVE**

1. Elaborar un sitio web que al menos tenga una página principal donde aparezca un logo, con un menú de navegación, y un cuerpo de la página donde se incluyan imágenes, videos, tablas, etc.
2. Evalúa la página con TAW.
3. Evalúa la página con JAWS.
4. ¿Cuáles son las dificultades encontradas?

## SOLUCIÓN

Como estamos finalizando el curso he decidido crear la página con un framework, pero he escogido uno donde se puede ver claramente el código HTML, CSS y Javascript para la comprensión del trabajo, montando las partes del documento con componentes como si fuera un puzzle. Ese framework es **Svelte**. Veamos los archivos del proyecto:



Vemos en la imagen del centro el directorio principal donde destacamos:

- **Directorio raíz:**
  - **server.js:** Archivo de Express.js que se ejecutará en el servidor. Será el que devuelva la página al usuario.
- **Carpeta src:**
  - **Carpeta componentes:** Cada una de las partes del documento HTML junto a los estilos de éstos y funcionalidad javascript en el mismo archivo.
  - **App.svelte:** Es el archivo que monta todo el documento HTML que va dentro de la etiqueta body con los componentes de la carpeta anterior.
  - **Main.js:** Encargado de compilar *App.svelte* en los archivos HTML, CSS y Javascript que estarán ubicados en la carpeta *public*. Pero creo que será más fácil leerlo desde los archivos *.svelte*.
- **Carpeta public:**
- Todos los archivos HTML, CSS y Javascript de nuestro sitio junto a imágenes y videos utilizados.

Creo ha sido interesante crearlo mediante este framework para este trabajo para comentar una de las cosas que nos ofrece de cara a la accesibilidad. Veamos un ejemplo:

```

8   <div class="container">
9     <img {src} {alt} />
10    <div class="info">
11      <h4>{title}</h4>
12      <p>{description}</p>
13    </div>
14  </div>

```



```

6   </> A11y: <img> element should have an alt attribute svelte(a11y-
7     missing-attribute)
8   <d Ver el problema Corrección Rápida (Ctrl+.)
9   <img {src} /> You, seconds ago • Uncommitted changes
10  <div class="info">
11    <h4>{title}</h4>
12    <p>{description}</p>
13  </div>
14  </div>

```

Vemos en las imágenes anteriores que si quitamos el atributo "alt" en una etiqueta imagen del documento nos avisa de que deberíamos añadirlo, controlando que no nos falte los componentes necesarios para la accesibilidad del sitio.

Existen otras extensiones en muchos de los editores e IDEs del mercado que nos ofrecen esa ayuda para no olvidarnos de lo necesario para que nuestro sitio sea lo más accesible posible y es algo que se deberíamos tener en cuenta a la hora de comenzar un proyecto.

Para facilitar la lectura si no se conoce este framework, comentar que simplemente es el documento **index.html de la carpeta public**:

```

tarea1 - index.html
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4    <meta charset='utf-8'>
5    <meta name='viewport' content='width=device-width,initial-scale=1'>
6
7    <title>Svelte app</title>
8
9    <link rel='icon' type='image/png' href='/favicon.png'>
10   <link rel='stylesheet' href='/global.css'>
11   <link rel='stylesheet' href='/build/bundle.css'>
12
13   <script defer src='/build/bundle.js'></script>
14 </head>
15
16 <body>
17 </body>
18 </html>

```

Añadiendo dentro de la etiqueta `body` de `index.html` el contenido de `App.svelte`:

```
tarea1 - App.svelte
11 <Header />
12 <main>
13   <TitleSeccion title="Sección de usuarios registrados" />
14   <SeccionTabla />
15   <p></p>
16   <TitleSeccion title="Sección de fotos de vehículos comentados" />
17   <SeccionGaleria />
18   <p></p>
19   <TitleSeccion title="Sección de vídeos compartidos" />
20   <SeccionVideos />
21   <p></p>
22   <TitleSeccion title="Formulario de registro" />
23   <SeccionContacto />
24 </main>
25 <Footer />
```

Y cada una de las etiquetas que son componentes y no propio de html en su respectivo archivo con el mismo nombre. Por ejemplo en `App.svelte` vemos una **etiqueta Header**, así que si vamos a `Header.svelte` podremos ver ese código html que se añadirá ahí:

```
tarea1 - Header.svelte
5 <header class="hdr">
6   <h1>Soluciones CMRR</h1>
7   <div class="navBar">
8     <NavBar />
9   </div>
10 </header>
```

También se han añadido los atributos y el texto mediante javascript o desde la instancia del componente, por ejemplo en `CardImage`:

```
tarea1 - CardImage.svelte
8 <div class="container">
9   <img {src} {alt} />
10  <div class="info">
11    <h4>{title}</h4>
12    <p>{description}</p>
13  </div>
14 </div>
```

Vemos que tiene atributo `src` y `alt` que se añaden cuando se añade esa parte de código en la instancia del componente desde el componente padre. En la siguiente imagen vemos como se añaden mediante un bucle los datos de un array de objetos en el script:

```
tarea1 - SeccionGaleria.svelte
17   title: "Moto custom",
18   description: "Motocicleta custom de diseño personalizado",
19   src: "../img/moto/custom.png",
20   alt: "Moto custom americana roja de rueda trasera ancha con rubia al frente",
21 },
22 ];
23 </script>
24
25 <section>
26   {#each photos as { title, description, src, alt }}
27     <CardImage {src} {title} {description} {alt} />
28   {/each}
29 </section>
```

Lo que daría el siguiente resultado:



Con esto creo será más que suficiente para poder leer el código y comprender como se ha creado el sitio, ya que en los archivos resultantes de la compilación se dificulta un poco al estar todo añadido mediante javascript.

Todos los archivos se añadirán como siempre al final del trabajo para su visualización.

Ahora pasaremos a ver de que está compuesta nuestra página:



# Soluciones CMRR

MDN
Svelte
JSONPlaceholder

Sección de usuarios registrados

Nombre	Alias	Email	Website
Leanne Graham	Bret	Sincere@april.biz	hildegard.org
Ervin Howell	Antonette	Shanna@molissa.tv	anastasia.net
Clementine Bauch	Samantha	Nathan@yesenia.net	ramiro.info
Patricia Lebsack	Karianne	Jullianne.O'Conner@kory.org	kale.biz
Chelsey Dietrich	Kamren	Lucio_Hettinger@annie.ca	demarco.info
Mrs. Dennis Schulist	Leopoldo_Corkory	Karley_Dach@jasper.info	ola.org
Kurtis Weisnat	Diwyn.Skiles	Telly.Hoeger@billy.biz	elvis.io
Nicholas Runolfsdottir V	Maxime.Nierow	Sherwood@rosamond.me	jecynthe.com
Glenna Reichert	Dolphine	Chaim_McDermott@dana.io	conrad.com
Clementina DuBuque	Moriah.Stanton	Rey.Padberg@karina.biz	ambrose.net

Un total de 10 usuarios registrados


Lorem ipsum dolor sit amet, consectetur adipiscing elit. Consectetur enim qui eaque quidem quibusdam excepturi ad ut maxime ratione dolorum commodi repellat, dolores ipsam, voluptate alias ab dicta necessitatibus accusantium? Est cum, minima rem ullam consequuntur cupiditate aliquid, nobis ipsum debitis quibusdam odit saepe. Rerum doloribus vel a fugit enim, nisi nam, non natus provident dolorum distinctio excepturi tempore labore.

Vemos que comienza con un *header* que contiene un h1 de título y una barra de navegación donde hay una imagen de logo y unos enlaces a diferentes páginas.


Debajo en el *main* tenemos un h2 antes de cada *seccion* del documento, donde en la primera tenemos una tabla y un párrafo.

A continuación otro *h2* y otra *seccion* que contiene las *CardImages* vista anteriormente y nuevamente un párrafo:


**Sección de fotos de vehículos comentados**



**Harley Davison**  
Motocicleta Harley de la serie A



**Honda Cross**  
Motocicleta de Cross Honda vallón



**Moto custom**  
Motocicleta custom de diseño personalizado

Lorem ipsum dolor sit amet consectetur adipisicing elit. Minima, vero provident! Nobis officia sed in laudantium aperiam neque rerum ea sint a ullam accusantium illo debitis corrupti voluptatum, deserunt hic? Maiores hic esse magni cupiditate fuga incidunt earum voluptatem quisquam rerum. Velit excepturi doloribus, voluptatum quasi exercitationem fuga odio alias saepe quo. Ipsa quasi ipsam id justo eligendi non maxime! Quia inventore nisi sit accusamus corporis distinctio blanditiis at, dicta velit unde soluta magnam in nam, magni facilis, dolorem quidem illum! Corrupti in eos eveniet ea mollitia ullam consequatur neque?

Más abajo otro *h2* y una *seccion* un vídeo y el párrafo:

**Sección de vídeos compartidos**



Lorem ipsum dolor sit amet consectetur adipisicing elit. Mollitia maiores doloremque molestiae dolore natus dolor architecto rem odit nesciunt recusandae dolorem expedita, provident exercitationem quae, nam ut, nemo perspiciatis cupiditate! Pariatur velit minima harum vel laborum provident nulla, dolore ab omnis assumenda est quae nesciunt tempore sapiente itaque aut sint ea maiores cum! Fuga deleniti ipsam dolor harum voluptates iste. Molestiae temporibus cumque tempore voluptates inventore, voluptatibus et ex expedita quae deleniti distinctio nulla, mollitia harum minus consequatur optio eaque repellat tenetur laborum consectetur sint nesciunt sunt. Voluptatibus, omnis in. Ad sit quae voluptatem veniam, unde saepe dolores, consequatur pariatur sed adipisci eius ratione tenetur veritatis repellat similique! Minima esse eligendi amet quo cum tenetur assumenda aliquid iste adipisci voluptatum? Totam doloremque nemo aliquam suscipit quod rerum sed, doloribus necessitatibus porro assumenda! Nesciunt, similique! Ipsam assumenda nisi quae, asperiores debitis quidem. Sunt quae provident veniam ipsum molestiae, esse officis error?

Y finalizamos con un *h2*, un formulario y de pie de página el *footer*:

**Formulario de registro**

**REGISTRATE**

**Nombre:**

**Apellidos:**

**Email:**

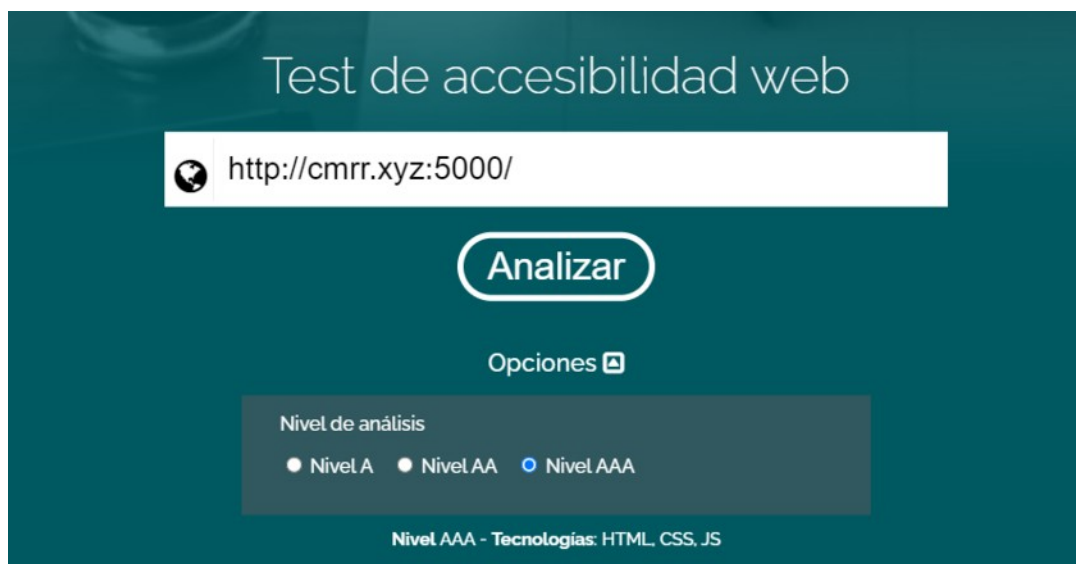
  

**Enviar**

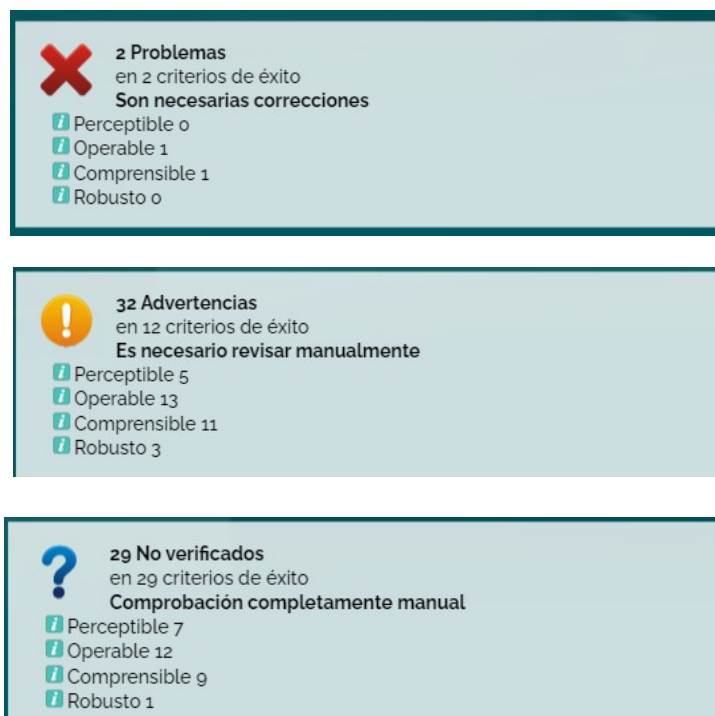
Cristo Manuel Rodríguez Rodríguez ©



Destacar que hemos dicho *h2* y posteriormente la *sección*, algo que nos va a recordar en el resultado del análisis de accesibilidad de nuestro sitio. Empecemos con el análisis con **TAW**. Iremos al sitio web y veremos un cuadro de texto donde añadiremos nuestra página y como opciones seleccionaremos el nivel más elevado de accesibilidad:



Nos da un informe muy grande y detallado junto al siguiente resumen de problemas o mejoras necesarias para la accesibilidad de nuestro sitio web.



Categoría	Ícono	Cantidad	Criterios de éxito	Acción	Detalles
Problemas	❌	2	en 2 criterios de éxito	Son necesarias correcciones	<ul style="list-style-type: none"><li>Perceptible 0</li><li>Operable 1</li><li>Comprensible 1</li><li>Robusto 0</li></ul>
Advertencias	⚠️	32	en 12 criterios de éxito	Es necesario revisar manualmente	<ul style="list-style-type: none"><li>Perceptible 5</li><li>Operable 13</li><li>Comprensible 11</li><li>Robusto 3</li></ul>
No verificados	❓	29	en 29 criterios de éxito	Comprobación completamente manual	<ul style="list-style-type: none"><li>Perceptible 7</li><li>Operable 12</li><li>Comprensible 9</li><li>Robusto 1</li></ul>

Vemos que el resumen se compone de 3 apartados, **Problemas**, **Advertencias** y **No verificados**.

**No verificados** pueden ser por componentes del documento que no tenemos o no ha encontrado, por lo que no puede darnos un resultado, pero nos ofrece información de lo que debería haber si lo añadimos. Un ejemplo en nuestro resultado es con grabaciones de audio, las cuales no hemos añadido al documento, pero nos ofrece las posibles mejoras de accesibilidad:

1.2-Medios basados en el tiempo		
1.2.1 - Sólo audio y solo video (grabaciones)	A	na
1.2.2 - Subtítulos (pregrabados)	A	na
1.2.3 - Audiodescripción o Medio Alternativo (Pregrabado)	A	na
1.2.4 - Subtítulos (en directo)	AA	na
1.2.5 - Descripción auditiva (Pregrabada)	AA	na
1.2.6 - Lenguaje de signos	AAA	na
1.2.7 - Audiodescripción Extendida (Pregrabada)	AAA	na
1.2.8 - Alternativa textual completa	AAA	na
1.2.9 - Sólo audio (en directo)	AAA	na

**Advertencias** son posibles mejoras de accesibilidad que deberemos revisar nosotros mismos al no estar bien claras en el examen. Por ejemplo en nuestro caso tenemos algunas como las siguientes:

2.4.2 - Páginas tituladas	A	
Las páginas web tienen títulos que describen su tema o propósito.		
2.4.6 - Encabezados y etiquetas	AA	
Los encabezados y etiquetas describen el tema o propósito.		
3.3.1 - Identificación de errores	A	
Si se detecta automáticamente un error en la introducción de datos, el elemento erróneo es identificado y el error se describe al usuario mediante texto.		

Si nos fijamos en nuestro código vemos que tiene razón, el título de la página no describe de que trata el sitio, los encabezados no describen el tema claramente y no controlamos los datos introducidos en el formulario, simplemente avisamos al usuario de que ha sido enviado los datos con una ventana emergente, por lo que son advertencias a tener en cuenta para mejorar la accesibilidad y que no hemos tenido en cuenta a la hora de crear el sitio web.

**Problemas** es lo que más debemos tener en cuenta, por ejemplo en nuestro sitio han salido 2 que son los siguientes:

2.4.10 - Encabezados de sección	AAA	
Se usan encabezados de sección para organizar el contenido. Nota 1: "Encabezados" se usa en sentido general e incluye los títulos y otras formas de agregar encabezados a las distintos tipos de contenido. Nota 2: Este criterio de éxito se refiere al contenido propiamente dicho[1], no a los componentes de la interfaz de usuario. Los componentes de la interfaz de usuario se tratan en el Criterio de Cumplimiento 4.1.2.		
3.1.1 - Idioma de la página	A	
El idioma humano predeterminado de cada página web puede ser determinado de forma programática.		

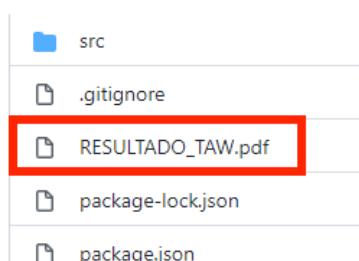
Vemos que son 2, **el primero es de los encabezados de sección**, como dijimos anteriormente se añadió los **h2 fuera de cada etiqueta sección**, por lo que cada sección del documento no tiene un título descriptivo de esa sección o parte del documento de cara al resultado del examen y al motor de búsqueda que examine nuestro sitio, pudiendo esto afectar a nuestro posicionamiento en Google a la hora de las búsquedas al no tener bien claro el contenido del sitio.

El otro problema es el idioma humano utilizado, en nuestro caso hemos hecho uso de *lorem*, que es una funcionalidad de la extensión *Emmet* para añadir texto de ejemplo en latín, pero nuestro documento dice que el idioma es en español como indica el atributo de la etiqueta *html*:

```
tarea1 - index.html
1 <!DOCTYPE html>
2 <html lang="es">
```

Mucha gente puede no darle importancia, ya que muchos lectores de pantalla pueden leer incluso así el texto, pero si ponemos un idioma que no va con el idioma de nuestro sitio esos lectores reproducirán el texto con un acento inadecuado y mucho contenido incorrectamente, complicando al usuario entender el contenido de nuestra página web.

El resultado del examen completo se añadirá en el repositorio de **Github** con todo el código del proyecto con el nombre **RESULTADO\_TAW.pdf**:



Ahora pasamos a probar la página con el lector de pantallas JAW, por lo que iremos a la página web y descargaremos la última versión:



Hay que destacar que JAW es de pago, pero disponemos de un tiempo de 40 minutos para probarlo, tiempo suficiente para ver el resultado de nuestra página.

No comentaré el proceso de instalación al ser muy simple, solo aceptar la licencia y siguiente, siguiente. Una vez instalado lo abrimos y veremos la ventana del programa, momento en el que el lector ya estará funcionando y leyendo la ventana que tengamos activa:



Una vez probamos la página web vemos los siguientes inconvenientes:

- **Mucho contenido es inaccesible**, solo se puede acceder con el botón de tabulación a los enlaces del documento, vídeo y formulario, por lo que para una persona que no puede ver el documento le resultaría imposible saber que contiene al saltarse el lector todo ese contenido y no leerlo. Incluso poniendo los párrafos en español no consigue llegar a leerlos.
- **Descripciones de los elementos inadecuada**. Tanto el título de la página como los enlaces no tienen una descripción adecuada, solo el nombre, cosa que no da suficiente información de lo que es o para que sirve. El vídeo no explica de que es, solo avisa de los controles y además como no tiene sonido pues no da a saber que hay ahí.
- **Confusión en el formulario de registro**. Está muy mal descrito el formulario, no dando la información clara de para que es y del botón de enviar, que sería mejor avisar de que se envía para registrar al usuario, no solo enviar.

Por lo que como conclusión después de usar el lector, veo que el sitio es nada accesible para gente con problemas de visión, ya que no se accede al contenido y al que puede no lo describe con claridad, ni siquiera el título da información clara de donde se está.

Añado como siempre el repositorio donde están todos los archivos del trabajo en GitHub y de la página web para probar:

Repositorio con el código y resultado de TAW en GitHub [🔗 Enlace](#)

Enlace al resultado de TAW (no sé si estará operativo aún el día de revisión) [🔗 Enlace](#)

Página web de este trabajo [🔗 Enlace](#)

### CONCLUSIONES

Este trabajo ha sido muy interesante y me ha dado a ver que sin darnos cuenta creamos sitios nada accesibles para personas con alguna deficiencia. Volveré a probar, como se ha solicitado en este trabajo, futuros proyectos para evitar estos inconvenientes a los usuarios que tengan algún problema de accesibilidad y así poder solucionarlo, ya que la web debería ser un recurso accesible para todos, además de que a la hora de crear un producto a una empresa, ésta le interesará que sea accesible al mayor público posible.

### BIBLIOGRAFÍA

Se ha consultado la unidad o lo aprendido durante el curso y utilizado los enlaces ofrecidos en el trabajo.