

GUÍA DE ESTUDIO

INTERFACES HUMANO COMPUTADOR

PRÁCTICA ACCESIBILIDAD WEB



- Refactorizar HTML para ser mas accessible
- Utilizar CSS para aumentar la accesibilidad de un sitio web
- Extender la accesibilidad usando JavaScript

Facultad: Ciencias Informáticas

Carrera: Ingeniería en Tecnologías de la Información.

Asignatura: Interfaces Humano Computador

Número de horas: 144 horas

Créditos: 3

Periodo académico: Cuarto / SEMESTRE i

Docente: Ing. Elsa Vera Burgos, Mg.

| | | |
|---|--|--|
|  Uleam UNIVERSIDAD LAICA ELOY ALFARO DE MANABÍ | NOMBRE DEL DOCUMENTO: ELABORACIÓN DE GUÍA DE ESTUDIOS PARA EDUCACIÓN EN LÍNEA | CÓDIGO: PAA-03-F-018 |
| | PROCEDIMIENTO: ELABORACIÓN, MEJORAMIENTO Y SEGUIMIENTO DEL SÍLABO | REVISIÓN: 1 Página 2 de 23 |

RÚBRICA DEL TRABAJO:

Documento duplicado o con plagio total o parcial del trabajo tendrá una calificación de 0.

Rúbrica de evaluación

| | Insuficiente 0% | Necesita mejorar 25% | Aceptable 50% | Bueno 75% | Excelente 100% |
|------------------|----------------------------|---|---|---|--|
| Elementos 25% | El trabajo presenta Plagio | Cada trabajo tiene menos del 70% de los elementos solicitados | Cada trabajo tiene por lo menos el 70% de los elementos solicitados | Cada trabajo tiene por lo menos el 80% de los elementos solicitados | Cada trabajo tiene más del 90% de los elementos solicitados |
| Contenido 35% | El trabajo presenta Plagio | Se intentaron menos del 70% de los ejercicios | Se intentaron por lo menos el 70% de los ejercicios | Se intentaron por lo menos el 80% de los ejercicios | Se intentaron más del 90% de las ejercicios |
| Exactitud 25% | El trabajo presenta Plagio | Se resolvieron correctamente menos el 70% | Se resolvieron correctamente por lo menos el 70% | Se resolvieron correctamente por lo menos el 85% | Se resolvieron correctamente por lo menos el 90% |
| Presentación 15% | El trabajo presenta Plagio | El trabajo esta muy descuidado | El trabajo: no esta claro ni ordenado, ni de fácil revisión y lectura | El trabajo: no esta claro ó no es ordenado, ó no es de fácil revisión y lectura | El trabajo: es claro ordenado, y de fácil revisión y lectura |

| | | |
|---|--|--------------------------------------|
|  Uleam UNIVERSIDAD LAICA ELOY ALFARO DE MANABÍ | NOMBRE DEL DOCUMENTO: ELABORACIÓN DE GUÍA DE ESTUDIOS PARA EDUCACIÓN EN LÍNEA | CÓDIGO: PAA-03-F-018 |
| | PROCEDIMIENTO: ELABORACIÓN, MEJORAMIENTO Y SEGUIMIENTO DEL SÍLABO | REVISIÓN: 1 Página 3 de 23 |

Tabla de contenido

| | | |
|-------------|---|-----------|
| 1. | <i>¿Qué es el HTML semántico y por qué es importante?</i> | 4 |
| 2. | <i>Estructura de la app</i> | 4 |
| 2.1. | Usando HTML semántico en el header de nuestro sitio | 4 |
| 2.1.1. | Añadir el atributo Header con la etiqueta <header> | 4 |
| 2.1.2. | Añadir la etiqueta con el atributo alt="" | 5 |
| 2.1.3. | Añadimos las etiquetas <nav>, y | 5 |
| 2.2. | Usando HTML semántico en el contenido principal | 5 |
| 2.2.1. | Añadimos la etiquetas <main> | 5 |
| 2.2.2. | Añadimos la etiqueta <section> | 5 |
| 2.2.3. | Añadimos la etiqueta <p> | 6 |
| 2.2.4. | Añadimos la etiqueta <h2> | 6 |
| 2.2.5. | Añadimos las etiquetas de listas e item (Redes Sociales) | 6 |
| 2.2.6. | Añadimos las etiquetas de listas e item (Carrusel) | 6 |
| 2.3. | Usando HTML semántico en footer de nuestro sitio | 7 |
| 2.3.1. | Cambiar la etiqueta <div> por <footer> | 7 |
| 2.3.2. | Cambiar la etiqueta <div> por <h2> | 8 |
| 2.3.3. | Cambiar la etiqueta <div> por <form> | 8 |
| 2.4. | ARIA - Accessible Rich Internet Applications | 8 |
| 2.4.1. | ARIA Roles | 8 |
| 2.4.2. | ARIA Propiedades | 9 |
| 2.4.3. | ARIA Estados | 9 |
| 2.5. | CSS para aumentar la accesibilidad de un sitio web | 10 |
| 2.5.1. | Contraste de Color | 10 |
| 2.5.2. | Íconos | 11 |
| 2.5.3. | Estilos de foco y hover | 12 |
| 2.6. | Extender la accesibilidad usando JavaScript | 14 |
| 2.6.1. | Manejando el foco del teclado - botones | 14 |
| 2.6.2. | Manejando el foco del teclado - carrusel | 14 |
| 2.6.3. | Coordinando el foco del teclado con el foco de lectores de pantallas | 16 |
| 2.6.4. | Manejando modales | 17 |
| 2.6.5. | Mejorando la accesibilidad con teclado de nuestro modal | 18 |
| 2.6.6. | Ayudando a nuestros usuarios a interactuar correctamente con el sitio | 20 |
| 2.6.7. | Validad formularios | 20 |
| 2.6.8. | Manejando cambios dinámicos | 22 |

| | | |
|---|--|--------------------------------------|
|  Uleam <small>UNIVERSIDAD LAICA ELOY ALFARO DE MANABÍ</small> | NOMBRE DEL DOCUMENTO: ELABORACIÓN DE GUÍA DE ESTUDIOS PARA EDUCACIÓN EN LÍNEA | CÓDIGO: PAA-03-F-018 |
| | PROCEDIMIENTO: ELABORACIÓN, MEJORAMIENTO Y SEGUIMIENTO DEL SÍLABO | REVISIÓN: 1 Página 4 de 23 |

1. ¿Qué es el HTML semántico y por qué es importante?

El HTML semántico no solo nos ayuda con aspectos de presentación pero le agregan significado al contenido, cuando se dice que se agrega significado se habla de agregar significado al navegador, el usuario no va a saber si se esta utilizando HTML semántico o no, pero el navegador si porque el HTML semántico le agrega significado a los elementos, le dice al navegador que interacción tienen y algunos aspectos de presentación, por ejemplo:

| Si HTML Semántico | Con HTML Semántico |
|---|--|
| <pre><div></div> <div><div></div></div> <div></div></pre> | <pre><header></header> <article></article> <footer><footer></pre> |

Si miramos el lado izquierdo nos damos cuenta que el sitio esta construido completamente con `<div>`, del lado del usuario esta bien, como desarrolladores podemos aplicar estilos a estos `<div>` para que se vean con los elementos que queremos. Podemos crear botones, podemos crear input, lo que queramos con CSS, por otro lado, podemos agregar las interacciones con javaScript le podemos decir que eventos escuchar que hacer si el usuario hace clic, que hacer si el usuario imprime una tecla, pero el navegador no va a saber nada de esto. En cambio cuando usamos HTML semántico, le damos estas pistas al navegador y nos ahorraremos mucho trabajo, cuando se usa HTML semántico se usan una variedad de elementos que son creados de diferentes significados y que el navegador le implementa la presentación y algunos detalles de las interacciones. Si miramos el lado derecho, podemos ver que tenemos elementos como header, article, footer, esto tambien les ayuda al navegador a incluir en el árbol de accesibilidad, el árbol de accesibilidad es lo que usan los lectores de pantalla para mapear las interacciones y leer el contenido a los usuarios.

2. Estructura de la app

2.1. Usando HTML semántico en el header de nuestro sitio

Vamos a partir reestructurando el sitio partiendo del body, vamos a realizar las siguientes modificaciones al código:

2.1.1. Añadir el atributo Header con la etiqueta `<header>`.

Lo primero que vamos hacer es agregar la etiqueta `<header>`, la etiqueta de header indica que es la parte principal del sitio.

| | | |
|---|--|--------------------------------------|
|  Uleam UNIVERSIDAD LAICA ELOY ALFARO DE MANABÍ | NOMBRE DEL DOCUMENTO: ELABORACIÓN DE GUÍA DE ESTUDIOS PARA EDUCACIÓN EN LÍNEA | CÓDIGO: PAA-03-F-018 |
| | PROCEDIMIENTO: ELABORACIÓN, MEJORAMIENTO Y SEGUIMIENTO DEL SÍLABO | REVISIÓN: 1 Página 5 de 23 |

2.1.2. Añadir la etiqueta con el atributo alt="".

Luego miramos lo que tenemos como imagen y nos damos cuenta que esa imagen la tenemos en un `<div>`, tambien nos damos cuenta que la imagen ha sido añadida aplicando estilos CSS. Lo que vamos a usar en lugar de ello vamos a añadir una etiqueta ``

2.1.3. Añadimos las etiquetas `<nav>`, `` y ``.

Añadimos la navegación y los elementos de línea para los elementos del menú.

Nota: Tener en cuenta el nombre del atributo class para cada una de las etiquetas para que se aplique el estilo CSS establecido.

```
h1 = "header"
h2 = "subtitle"
ul = "navigation"
img = "header-img"
```



Figura 1: Resultado de Añadir HTML semántico del proyecto

2.2. Usando HTML semántico en el contenido principal

Vamos a seguir restructurando el sitio con las siguientes actividades:

2.2.1. Añadimos la etiquetas `<main>`.

Añadimos esa etiqueta para indicar que vamos a comenzar a trabajar con el contenido principal de nuestro proyecto. Y colocamos esta etiqueta conteniendo los elementos:

```
<div class="intro">...
</div>

<div class="carousel">...
</div>
```

2.2.2. Añadimos la etiqueta `<section>`

Añadimos esa etiqueta para indicar que vamos a separar en secciones los componentes que conforman esta parte de nuestro proyecto.

| | | |
|---|--|--------------------------------------|
|  Uleam UNIVERSIDAD LAICA ELOY ALFARO DE MANABÍ | NOMBRE DEL DOCUMENTO: ELABORACIÓN DE GUÍA DE ESTUDIOS PARA EDUCACIÓN EN LÍNEA | CÓDIGO: PAA-03-F-018 |
| | PROCEDIMIENTO: ELABORACIÓN, MEJORAMIENTO Y SEGUIMIENTO DEL SÍLABO | REVISIÓN: 1 Página 6 de 23 |

```

<section>
  <div class="intro">...
  </div>
</section>

<section>
  <div class="carousel">...
  </div>
</section>

```

2.2.3. Añadimos la etiqueta <p>

Añadimos esa etiqueta para indicar que vamos a comenzar a trabajar con un texto largo y le pasamos el nombre de la clase de la etiqueta <div> a la etiqueta <p>.

2.2.4. Añadimos la etiqueta <h2>

Añadimos un título a cada una de las secciones para indicarle al usuario que visita nuestro sitio de que va a encontrar en cada una de las secciones.

```

<section>
  <h2 class="header"> Acerca de</h2>
  <div class="intro">...
  </div>
</section>

<section>
  <h2 class="header"> Mis Proyectos</h2>
  <div class="carousel">...
  </div>
</section>

```

2.2.5. Añadimos las etiquetas de listas e item (Redes Sociales).

Si miramos el código tenemos las redes sociales pero están implementados con etiquetas <div>, lo que vamos a hacer es cambiar las etiquetas <div> por las etiquetas de listas . Y cada div dentro de esta lista con una etiqueta de item . A la etiqueta de imagen le colocamos el atributo alt=""". Eliminamos los <div> restantes.

```

<ul class="social-media">
  <li>
    <a href="https://twitter.com/" target="_blank">
      
    </a>
  </li>
  <!-- Añadir las otras redes sociales -->
</ul>

```

2.2.6. Añadimos las etiquetas de listas e item (Carrusel).

| | | |
|---|--|--------------------------------------|
|  Uleam UNIVERSIDAD LAICA ELOY ALFARO DE MANABÍ | NOMBRE DEL DOCUMENTO: ELABORACIÓN DE GUÍA DE ESTUDIOS PARA EDUCACIÓN EN LÍNEA | CÓDIGO: PAA-03-F-018 |
| | PROCEDIMIENTO: ELABORACIÓN, MEJORAMIENTO Y SEGUIMIENTO DEL SÍLABO | REVISIÓN: 1 Página 7 de 23 |

Si miramos el código tenemos la parte de carrusel vemos que también esta implementado con etiquetas `<div>`, lo que vamos a hacer es cambiar las etiquetas `<div>` por las etiquetas de listas ``. Y cada div dentro de esta lista con una etiqueta de item ``. Añadimos un botón, una imagen y a la etiqueta de imagen le colocamos el atributo `alt="”"`. **Debemos tener 5 proyectos en total.**

```
<li><button></button></li>
<!-- Completar hasta tener los 5 proyectos -->
```



2.3. Usando HTML semántico en footer de nuestro sitio

Si miramos el código tenemos una parte de sección de contacto que esta relacionado con la imagen que se muestra a continuación.

2.3.1. Cambiar la etiqueta `<div>` por `<footer>`.

| | | |
|--|--|--------------------------------------|
|  Uleam <small>UNIVERSIDAD LAICA EOY ALFARO DE MANABÍ</small> | NOMBRE DEL DOCUMENTO: ELABORACIÓN DE GUÍA DE ESTUDIOS PARA EDUCACIÓN EN LÍNEA | CÓDIGO: PAA-03-F-018 |
| | PROCEDIMIENTO: ELABORACIÓN, MEJORAMIENTO Y SEGUIMIENTO DEL SÍLABO | REVISIÓN: 1 Página 8 de 23 |

El footer comunica que es la última parte del sitio, donde normalmente colocamos las redes sociales o un formulario de contactos, es la última información con la que queremos que nuestro usuario se quede.

2.3.2. Cambiar la etiqueta <div> por <h2>.

Colocamos la etiqueta <h2> a esta sección para indicarle al usuario que visita nuestro sitio de que va a encontrar en cada una de las secciones.

```
<h2 class="header">Contacto</h2>
```

2.3.3. Cambiar la etiqueta <div> por <form>.

Colocamos la etiqueta <form> a esta sección para indicarle al navegador que estamos trabajando con un formulario. Y esto se vuelve muy importante para los lectores de pantalla, porque los lectores de pantalla tienen una manera diferente de interactuar con los formularios, cuando se usa esa etiqueta le estamos diciendo que lo que sigue es un formulario y saben como navegar dentro de eso

```
<form class="form-container">
```

2.4. ARIA - Accessible Rich Internet Applications

A veces el HTML semánticos no nos alcanza especialmente en aplicaciones que tienen interacciones dinámicas o interacciones que nosotros mismos desarrollamos usando JavaScript. En estos momentos podemos usar ARIA.

ARIA es un conjunto de atributos para accesibilidad que pueden añadirse a cualquier etiqueta, pero especialmente adaptado a HTML. ARIA fue creado por la W3C a través de la iniciativa que ya se había mencionado la WAI ellos son los responsables de la especificación de ARIA, y la crearon para crear cambios especiales en nuestras aplicaciones. ARIA tiene tres atributos que podemos añadir a HTML, tienen los roles, propiedades y los estados.

2.4.1. ARIA Roles.

Los roles definen el tipo general del objeto, puede ser un artículo, una alerta o un deslizador, los roles le comunican al navegador cuales son las interacciones de debería esperar y cómo se va a usar este objeto dentro de nuestra página o aplicación. Existe una advertencia los roles solo se usan en situaciones muy específicas, es mejor que dependas del HTML semántico para comunicar los roles a agregarle roles a cada elemento o a algunos elementos y dejar otros elementos atrás, es una especificación muy difícil de usar y hay que saber cuales son los momentos apropiados.

Los roles comunican diferentes interacciones a navegadores y a lectores de pantalla como se navega en esa sección, los navegadores y especialmente los lectores de

| | | |
|---|--|--------------------------------------|
|  Uleam UNIVERSIDAD LAICA ELOY ALFARO DE MANABÍ | NOMBRE DEL DOCUMENTO: ELABORACIÓN DE GUÍA DE ESTUDIOS PARA EDUCACIÓN EN LÍNEA | CÓDIGO: PAA-03-F-018 |
| | PROCEDIMIENTO: ELABORACIÓN, MEJORAMIENTO Y SEGUIMIENTO DEL SÍLABO | REVISIÓN: 1 Página 9 de 23 |

pantalla navegan diferentes roles en diferentes estilos. Por ejemplo un articulo, un articulo un lector sabe no debe esperar mucha interacción del usuario, mientras que en un formulario que es mucha interacción con el usuario el lector de pantalla tiene diferentes métodos el lector de pantalla tiene muchos métodos para esperar esa interacción y para manejarla cuando pasa. Ten en cuenta que los roles se usan muy muy ligeramente.

2.4.2. ARIA Propiedades.

Las propiedades ARIA si las vamos a usar más seguidos, estas le comunican al navegador diferentes propiedades. La propiedades ARIA comunican atributos que son esenciales para el comportamiento o significado de un elemento pero que se suelen comunicar visualmente, eso que significa, normalmente vemos patrones de diseños donde tomamos por sentado mucho comunicado visual, pueden ser iconos puede ser una interacción que solemos saber que se hace, si tenemos un input en un formulario sabemos que vamos a llenar ese formulario y si estamos acostumbrados a llenar formulario sabemos toda esta comunicación que se hace a través de los medios visuales lo pierde cualquier persona que usa un lector de pantalla, entonces es importante usar ARIA para agregar ese significado y para asegurarnos que nuestro diseño se ve igual para todos.

Al aplicarlo en el código, tenemos dos lugares donde estamos comunicando visualmente que no estamos comunicando con texto, por ejemplo las flechas del carrusel si una persona las puede ver sabe e interpreta que este carrusel esas flechas sirven para mover de lado a lado, pero si tu no tienes una pantalla o si se tiene una discapacidad visual esas personas no van a saber para que sirven esas flechas porque como se encuentan en div, el lector de pantalla no va a saber interpretarlas. Para esto vamos a usar una propiedad de ARIA que se llama ARIA LABEL que podemos usar para agregarle un string que describe cualquier elemento que describe esa propiedad

```
<div class="arrow arrow-left" aria-label="backward button"></div>
<div class="arrow arrow-right" aria-label="forward button"></div>
```

En otro lugar donde Podemos añadir ARIA LABEL es en le formulario, en la la parte del input.

```
Nombre<input type="text" aria-label="nombre" /> <br />
Correo<input type="text" aria-label="correo" /> <br />
```

2.4.3. ARIA Estados

| | | |
|---|--|---------------------------------------|
|  Uleam UNIVERSIDAD LAICA ELOY ALFARO DE MANABÍ | NOMBRE DEL DOCUMENTO: ELABORACIÓN DE GUÍA DE ESTUDIOS PARA EDUCACIÓN EN LÍNEA | CÓDIGO: PAA-03-F-018 |
| | PROCEDIMIENTO: ELABORACIÓN, MEJORAMIENTO Y SEGUIMIENTO DEL SÍLABO | REVISIÓN: 1 Página 10 de 23 |

Hay muchas veces que terminamos comunicando cambios o significados con cosas visuales y en la pantalla. Y lo que pasa en esos casos es que las personas que navegan con lectores de pantalla no van a ver esa interacción, lo importante de la accesibilidad es que le estamos dando una experiencia igual o lo más igual posible a personas que están usando tecnología asistiva. No es justo ofrecerles otra experiencia completa simplemente porque no tiene la capacidad de usar las máquinas como nosotros lo usamos. Los estados de ARIA comunican estados y cambio de estados en elementos que suelen comunicar visualmente, esto que significa, digamos que tenemos un checkbox, nosotros sabemos que visualmente si el checkbox campo esta check sabemos que esa opción está seleccionada, pero alguien con el lector de pantalla puede no ver eso tenemos que comunicar los diferentes estados que tenemos en las aplicaciones, para ofrecerle una experiencia parecida o igual a una persona usando lector de pantalla.

Al usarlo en el sitio hay algo muy curioso que pasa con el carrusel, nosotros visualmente podemos ver que hay tres proyectos enfocados, pero cuando se hace uso del lector de pantalla, este no se da cuenta que solo hay tres elementos mostrados visualmente, entonces lee todos los elementos que están en el HTML de la página incluyendo los que tenemos escondidos, vamos a usar un estado que se llama ARIA-HIDDEN para esconder estos elementos para que el lector de pantalla tenga la misma experiencia que tiene una persona que puede visualizar estos elementos.

```
<li aria-hidden="true"><button></button></li>
<li aria-hidden="true"><button></button></li>
```

2.5. CSS para aumentar la accesibilidad de un sitio web

2.5.1. Contraste de Color

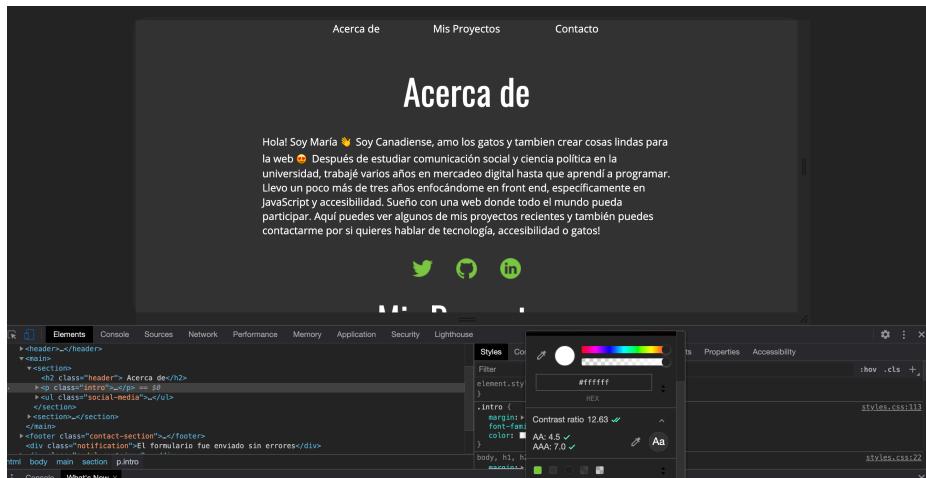
Los contrastes de colores se refieren a la diferencia entre el color del fondo y color del texto que vemos. Si el contraste de color es muy bajo, no se puede ver el texto correctamente.

Entonces vamos a pasar al código y comenzamos a cambiar estilo, y comenzamos con cambiar el estilo para la etiqueta `<p class="intro">`.

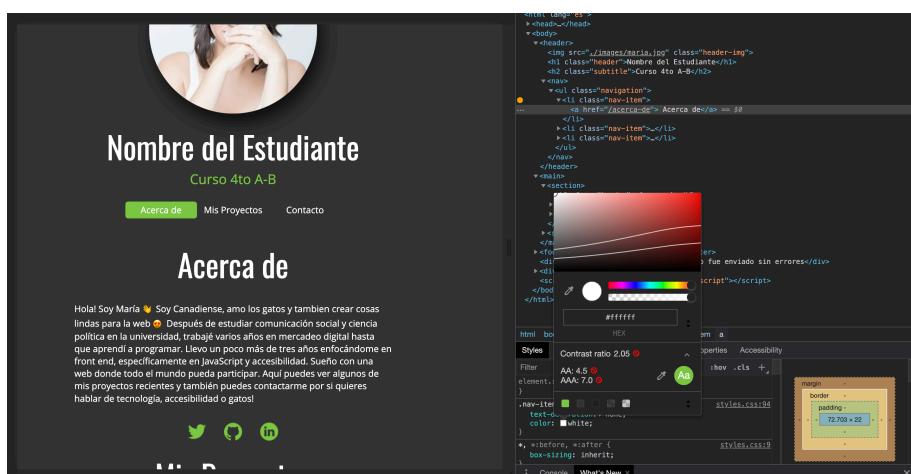
```
.intro {
  margin: 24px 196px 0;
  font-family: var(--secondary-font);
  color: var(--blanco);
}
```

| | | |
|---|--|---------------------------------------|
|  Uleam UNIVERSIDAD LAICA ELOY ALFARO DE MANABÍ | NOMBRE DEL DOCUMENTO: ELABORACIÓN DE GUÍA DE ESTUDIOS PARA EDUCACIÓN EN LÍNEA | CÓDIGO: PAA-03-F-018 |
| | PROCEDIMIENTO: ELABORACIÓN, MEJORAMIENTO Y SEGUIMIENTO DEL SÍLABO | REVISIÓN: 1 Página 11 de 23 |

Si analizamos el código nos damos cuenta que cumplimos con dos de los niveles de WCAG el AA y el AAA.



Ahora vamos a inspeccionar la navegación para agregarle un mejor contraste, entonces nos damos cuenta que el contraste que tiene no es suficiente para los niveles de WCAG:



Para mejora los niveles vamos a cambiar los estilos de la siguiente manera:

```
.nav-item:hover {
  background-color: var(--verde);
  cursor: pointer;
}
.nav-item:hover a {
  color: var(--gris-oscuro);
}
```

Y por último mejoramos los estilos del formulario de contactos:

```
.form-label{
  color: var(--verde)
```

2.5.2. Íconos

| | | |
|---|--|---------------------------------------|
|  Uleam <small>UNIVERSIDAD LAICA ELOY ALFARO DE MANABÍ</small> | NOMBRE DEL DOCUMENTO: ELABORACIÓN DE GUÍA DE ESTUDIOS PARA EDUCACIÓN EN LÍNEA | CÓDIGO: PAA-03-F-018 |
| | PROCEDIMIENTO: ELABORACIÓN, MEJORAMIENTO Y SEGUIMIENTO DEL SÍLABO | REVISIÓN: 1 Página 12 de 23 |

Hay algo muy curioso que pasa cuando usamos íconos y es que estamos muy acostumbrados a nuestra propia experiencia y pensamos que los íconos son universales y no siempre es así los íconos pueden ser confusos si no se los ha visto antes o si no estamos acostumbrados a esa aplicación que estamos usando. Vamos a aplicar estilos que mejoren la accesibilidad a los íconos.

```
.social-media {
  align-items: center;
  display: flex;
  justify-content: space-around;
  margin: 32px auto 0;
  /* width: 20%; */
}

.social-media li a {
  align-items: center;
  display: flex;
  color: var(--blanco);
  font-family: var(--secondary-font);
  text-decoration: none;
}

.social-link-text {
  padding-right: 8px;
  padding-left: 8px;
}
```

```
<li>
  <a href="https://github.com/" target="_blank">
    
    <p class="social-link-text">Github</p>
  </a>
</li>
```

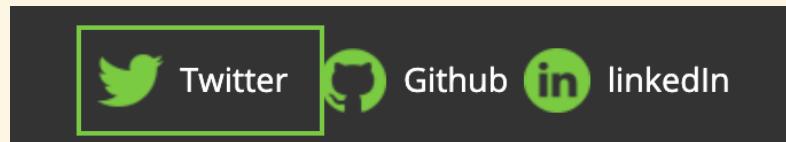


2.5.3. Estilos de foco y hover

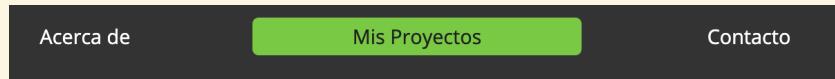
Los estilos del foco y hover nos dan una oportunidad de coordinar este elemento de nuestro sitio, cuando nuestros estilos de hover, osea los estilos para las personas que están usando un cursor son muy diferentes a las personas que están usando un teclado, los podemos coordinar para crear una experiencia igual para las dos personas, vamos a aplicarlo en el sitio.

| | | |
|--|--|---|
|  Uleam <small>UNIVERSIDAD LAICA EOY ALFARO DE MANABÍ</small> | NOMBRE DEL DOCUMENTO: ELABORACIÓN DE GUÍA DE ESTUDIOS PARA EDUCACIÓN EN LÍNEA | CÓDIGO: PAA-03-F-018 |
| | PROCEDIMIENTO: ELABORACIÓN, MEJORAMIENTO Y SEGUIMIENTO DEL SÍLABO | REVISIÓN: 1 Página 13 de 23 |

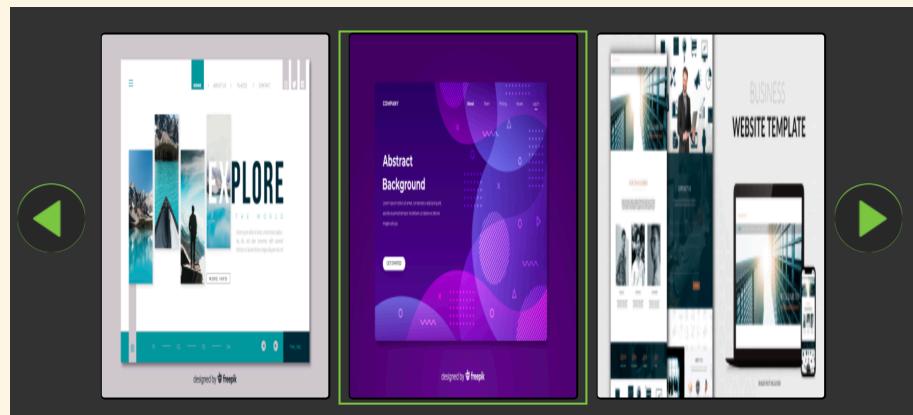
```
.nav-item:focus-within {
  background-color: var(--verde);
}
.nav-item a:focus {
  outline: none;
  color: var(--gris-oscuro);
}
```



```
.nav-item:focus-within {
  background-color: var(--verde);
}
.nav-item a:focus {
  outline: none;
  color: var(--gris-oscuro);
}
```



```
window {
  overflow: hidden;
  width: 815px;
  padding: 8px;
}
.project-container button {
  background: none;
  border: none;
  padding: 0;
}
.project-container button:focus-within {
  outline: 3px solid var(--verde);
```



| | | |
|---|--|---------------------------------------|
|  Uleam UNIVERSIDAD LAICA ELOY ALFARO DE MANABÍ | NOMBRE DEL DOCUMENTO: ELABORACIÓN DE GUÍA DE ESTUDIOS PARA EDUCACIÓN EN LÍNEA | CÓDIGO: PAA-03-F-018 |
| | PROCEDIMIENTO: ELABORACIÓN, MEJORAMIENTO Y SEGUIMIENTO DEL SÍLABO | REVISIÓN: 1 Página 14 de 23 |

```

input {
  width: 100%;
}

input:focus{
  outline: 3px solid var(--verde);
  padding: 8px;
}

```



2.6. Extender la accesibilidad usando JavaScript

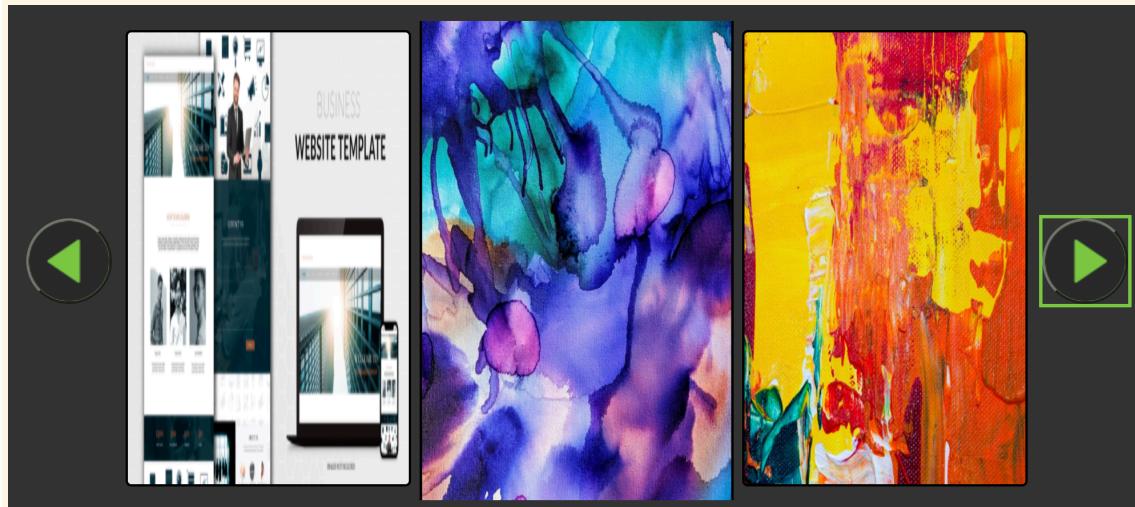
2.6.1. Manejando el foco del teclado - botones

```

<button class="arrow arrow-right" aria-label="Boton para ir hacia adelante"></button>

.arrow:focus{
  outline: 3px solid var(--verde);
}

```



2.6.2. Manejando el foco del teclado - carrusel

Vamos a ver como usando JavaScript para hacer que nuestro carrusel funcione para alguien que este usando el teclado o el mouse para eso vamos a manejar un atributo llamado Tabindex, este atributo indica si su elemento puede ser enfocado, y si participa en la navegación secuencial del teclado.

| | | |
|---|--|---|
|  Uleam <small>UNIVERSIDAD LAICA ELOY ALFARO DE MANABÍ</small> | NOMBRE DEL DOCUMENTO: ELABORACIÓN DE GUÍA DE ESTUDIOS PARA EDUCACIÓN EN LÍNEA | CÓDIGO: PAA-03-F-018 |
| | PROCEDIMIENTO: ELABORACIÓN, MEJORAMIENTO Y SEGUIMIENTO DEL SÍLABO | REVISIÓN: 1 Página 15 de 23 |

Valores de Tabindex:

- Valor negativo(-1): El elemento debe ser enfocado, pero no debe de ser accesible a través de la navegación.
- Valor positivo (>0): Debe poder ser enfocado y su orden relativo es definido por el valor del atributo.
- Valor de 0: Debe ser enfocado y ser accesible a través de la navegación secuencial del teclado, pero su orden relativo es definido por convención de la plataforma.

Vamos a realizar las modificaciones en el carrusel, primero modificamos el html de esta manera.

```
<li aria-hidden="true"><button class="project4" tabindex="-1"></button></li>

<li aria-hidden="true"><button class="project5" tabindex="-1"></button></li>
```

Modificados el archivo de index.js de la siguiente manera

```
/** Esta funcion se llama cuando la persona hace click en la fecha derecha del
carousel para navegar a la derecha */
function clickRight() {
  ...
  switch (newValue){
    case -270:
      document.querySelector('.project1').setAttribute("tabindex", "-1");
      document.querySelector('.project4').removeAttribute("tabindex");
      break;
    case -540:
      document.querySelector('.project2').setAttribute("tabindex", "-1");
      document.querySelector('.project5').removeAttribute("tabindex");
      break;
    default:
      break;
  }
}

/** Esta funcion se llama cuando la persona hace click en la fecha izquierda
del carousel para navegar a la izquierda */
function clickLeft() {
  ...
  switch (newValue){
    case -270:
```

| | | |
|---|--|---------------------------------------|
|  Uleam UNIVERSIDAD LAICA ELOY ALFARO DE MANABÍ | NOMBRE DEL DOCUMENTO: ELABORACIÓN DE GUÍA DE ESTUDIOS PARA EDUCACIÓN EN LÍNEA | CÓDIGO: PAA-03-F-018 |
| | PROCEDIMIENTO: ELABORACIÓN, MEJORAMIENTO Y SEGUIMIENTO DEL SÍLABO | REVISIÓN: 1 Página 16 de 23 |

```

document.querySelector('.project5').setAttribute("tabindex", "-1");
document.querySelector('.project2').removeAttribute("tabindex");
break;
case 0:
  document.querySelector('.project4').setAttribute("tabindex", "-1");
  document.querySelector('.project1').removeAttribute("tabindex");
  break;
default:
  break;
}
}
}

```

2.6.3. Coordinando el foco del teclado con el foco de lectores de pantallas

Vamos a realizar las modificaciones en el carrusel, primero modificamos el html de esta manera.

```

<li class="project1-container"><button class="project1"></button></li>

<li class="project2-container"><button class="project2"></button></li>

<li class="project3-container"><button class="project3"></button></li>

<li class="project4-container" aria-hidden="true"><button class="project4"
  tabindex="-1"></button></li>

<li class="project5-container" aria-hidden="true"><button class="project5"
  tabindex="-1"></button></li>

```

Modificamos el archivo de index.js de la siguiente manera

```

/** Esta función se llama cuando la persona hace click en la fecha derecha del
carousel para navegar a la derecha */
function clickRight() {
  ...
  switch (newValue){
    case -270:
      document.querySelector('.project1').setAttribute("tabindex", "-1");

```

| | | |
|---|--|---------------------------------------|
|  Uleam UNIVERSIDAD LAICA ELOY ALFARO DE MANABÍ | NOMBRE DEL DOCUMENTO: ELABORACIÓN DE GUÍA DE ESTUDIOS PARA EDUCACIÓN EN LÍNEA | CÓDIGO: PAA-03-F-018 |
| | PROCEDIMIENTO: ELABORACIÓN, MEJORAMIENTO Y SEGUIMIENTO DEL SÍLABO | REVISIÓN: 1 Página 17 de 23 |

```

    document.querySelector('.project1-container').setAttribute("aria-
hidden", true);
    document.querySelector('.project4').removeAttribute("tabindex");
    document.querySelector('.project4-container').removeAttribute("aria-
hidden")
    break;
  case -540:
    document.querySelector('.project2').setAttribute("tabindex", "-1");
    document.querySelector('.project2-container').setAttribute("aria-
hidden", "true");
    document.querySelector('.project5').removeAttribute("tabindex");
    document.querySelector('.project5-container').removeAttribute("aria-
hidden");
    break;
  default:
    break;
}
}

/** Esta funcion se llama cuando la persona hace click en la fecha izquierda
del carousel para navegar a la izquierda */
function clickLeft() {

  ...
  switch (newValue){
    case -270:
      document.querySelector('.project5').setAttribute("tabindex", "-1");
      document.querySelector('.project5-container').setAttribute("aria-
hidden", "true");
      document.querySelector('.project2').removeAttribute("tabindex");
      document.querySelector('.project2-container').removeAttribute("aria-
hidden");
      break;
    case 0:
      document.querySelector('.project4').setAttribute("tabindex", "-1");
      document.querySelector('.project4-container').setAttribute("aria-
hidden", "true");
      document.querySelector('.project1').removeAttribute("tabindex");
      document.querySelector('.project1-container').removeAttribute("aria-
hidden");
      break;
    default:
      break;
  }
}

```

2.6.4. Manejando modales

| | | |
|---|--|---------------------------------------|
|  Uleam UNIVERSIDAD LAICA ELOY ALFARO DE MANABÍ | NOMBRE DEL DOCUMENTO: ELABORACIÓN DE GUÍA DE ESTUDIOS PARA EDUCACIÓN EN LÍNEA | CÓDIGO: PAA-03-F-018 |
| | PROCEDIMIENTO: ELABORACIÓN, MEJORAMIENTO Y SEGUIMIENTO DEL SÍLABO | REVISIÓN: 1 Página 18 de 23 |

Los modales son un reto muy interesante para la accesibilidad porque pueden hacer cambios que estamos comunicando visualmente, ¿Cómo podemos hacer un modal mas accesible?

Vamos a colocar accesibilidad al modal de nuestro sitio:

- Añadir que mediante teclado se pueda acceder al modal.

```
document.querySelectorAll(".project").forEach(element => {
  element.addEventListener("click", e => openModal(e));
});
document.body.addEventListener("click", e => closeModal(e));
```

Para que realice la interacción con el Código HTML hay que modificar un poco el código.

- Modificamos el HTML del modal

```
<ul class="project-container">
  <li class="project1-container"><button class="project1 project"></button></li>
  <li class="project2-container"><button class="project2 project"></button></li>
  <li class="project3-container"><button class="project3 project"></button></li>
  <li class="project4-container" aria-hidden="true"><button class="project4 project" tabindex="-1"></button></li>
  <li class="project5-container" aria-hidden="true"><button class="project5 project" tabindex="-1"></button></li>
</ul>
```

2.6.5. Mejorando la accesibilidad con teclado de nuestro modal

- Modificamos los elementos de nuestro JavaScript

```
/** Esta función se llama cuando la persona hace click en cualquier proyecto del carousel */
function openModal(e) {
  document.querySelector(".modal-container").style.display = "flex";
  document.getElementById('modal-header').focus();
}
```

- Modificamos los elementos de nuestro HTML

| | | |
|---|--|---------------------------------------|
|  Uleam UNIVERSIDAD LAICA ELOY ALFARO DE MANABÍ | NOMBRE DEL DOCUMENTO: ELABORACIÓN DE GUÍA DE ESTUDIOS PARA EDUCACIÓN EN LÍNEA | CÓDIGO: PAA-03-F-018 |
| | PROCEDIMIENTO: ELABORACIÓN, MEJORAMIENTO Y SEGUIMIENTO DEL SÍLABO | REVISIÓN: 1 Página 19 de 23 |

```

<div class="modal-container">
  <div class="modal">
    <h2 class="header" id="modal-header" tabindex="0">Ejemplo de un
    proyecto mio</h2>
    
    <button class="modal-button">Cerrar</button>
  </div>
</div>

```

- Modificamos los elementos de nuestro CSS

```

.send-button,
.modal-button {
  align-items: center;
  background-color: var(--verde);
  border-radius: 5px;
  color: var(--gris-oscuro);
  font-family: var(--secondary-font);
  display: flex;
  height: 50px;
  justify-content: center;
  margin-bottom: 48px;
  width: 150px;
  margin-left: auto;
  font-size: 16px;
  cursor: pointer;
}

.send-button:focus,
.modal-button:focus {
  border: 5px solid var(--gris-oscuro);
  outline: 3px solid var(--verde)
}

#modal-header{
  outline: 3px solid var(--verde);
}

.modal {
  align-items: center;
  background-color: var(--gris-oscuro);
  border-radius: 5px;
  border: 2px solid #000;
  display: flex;
  flex-direction: column;
  height: 600px;
  justify-content: center;
  position: fixed;
  top: 5%;
}

```

| | | |
|---|--|---------------------------------------|
|  Uleam UNIVERSIDAD LAICA ELOY ALFARO DE MANABÍ | NOMBRE DEL DOCUMENTO: ELABORACIÓN DE GUÍA DE ESTUDIOS PARA EDUCACIÓN EN LÍNEA | CÓDIGO: PAA-03-F-018 |
| | PROCEDIMIENTO: ELABORACIÓN, MEJORAMIENTO Y SEGUIMIENTO DEL SÍLABO | REVISIÓN: 1 Página 20 de 23 |

```

width: 700px;
padding: 5px;
}

```

- Cerrando el modal con la tecla ESC.

Modificar añadiendo esta linea de esta parte del código

```

window.onload = () => {
  ...
  document.body.addEventListener("keyup", e => listenForEsc(e));
};

```

Añadimos esta función

```

/**Escucha por la clave esc para cerrar el modal */
Para saber cual es el codigo que debe escuchar lo verificamos con el sitio
keycode.info
function listenForEsc(e) {
  if (e.keyCode === 27){
    closeModal(e)
  }
}

```

2.6.6. Ayudando a nuestros usuarios a interactuar correctamente con el sitio

El tercer principio de WCAG nos dice que debemos ser comprensibles, tenemos que hacer todo el esfuerzo que podamos como creadores de productos web para asegurarnos que nuestros productos, nuestros sitios, nuestras aplicaciones todo lo que hacemos es fácil de entender, que sea predecible significa que estamos usando patrones que se usan en otras aplicaciones en otros contextos para hacer las mismas acciones, claro ejemplo es el modal, usamos la teclas ESC para cerrar el modal, esa es una interacción muy común en muchos productos y aplicaciones hay que acordarnos que nosotros no tenemos una experiencia universal el mundo funciona de varias maneras y hay tantas que usan la web de tantas maneras y tienen un rango variado de conocimiento por eso es mejor ser muy explícito en nuestras instrucciones especialmente cuando requerimos algo del usuario para ello tenemos que realizar validaciones de formularios.

2.6.7. Validad formularios

Vamos a realizar la validación del formulario, y se realizan este tipo de validaciones para decirle al usuario si cometió un error y para darle una oportunidad al usuario de corregirlo.

| | | |
|---|--|---------------------------------------|
|  Uleam UNIVERSIDAD LAICA ELOY ALFARO DE MANABÍ | NOMBRE DEL DOCUMENTO: ELABORACIÓN DE GUÍA DE ESTUDIOS PARA EDUCACIÓN EN LÍNEA | CÓDIGO: PAA-03-F-018 |
| | PROCEDIMIENTO: ELABORACIÓN, MEJORAMIENTO Y SEGUIMIENTO DEL SÍLABO | REVISIÓN: 1 Página 21 de 23 |

Modificamos en windows.onload() cambiando:

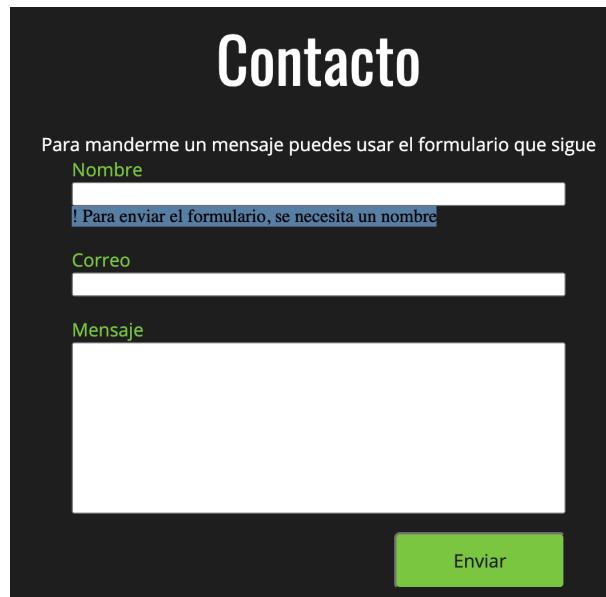
```
.addEventListener("click", showNotification));
```

por

```
.addEventListener("click", e => validateForm(e));
```

Para que antes de mostrar una notificación, valide primero el formulario, además añadimos esta porción de código, que es la función para validar el fomulario.

```
/* Validar el formulario antes de mostrar la notificación */
function validateForm(e) {
  e.preventDefault();
  const nameField = document.getElementById("name");
  if (nameField.value === "") {
    document.getElementById("name-error").innerHTML = "! Para enviar el formulario, se necesita un nombre";
  } else {
    showNotification();
  }
}
```



Una vez que el usuario ha enviado el mensaje vamos a poner en blanco los elementos para no causar confusión para el usuario cuando esté usando nuestro sitio para ello modificamos la función `showNotification()`, añadiendo.

```
/* Esta función se llama cuando la persona hace click en el botón de enviar del formulario de contacto */
function showNotification() {
  document.getElementById("name-error").innerHTML = "";
  document.querySelector('.form-container').reset();
  document.querySelector(".notification").style.display = "flex";
```

| | | |
|---|--|---------------------------------------|
|  Uleam <small>UNIVERSIDAD LAICA ELOY ALFARO DE MANABÍ</small> | NOMBRE DEL DOCUMENTO: ELABORACIÓN DE GUÍA DE ESTUDIOS PARA EDUCACIÓN EN LÍNEA | CÓDIGO: PAA-03-F-018 |
| | PROCEDIMIENTO: ELABORACIÓN, MEJORAMIENTO Y SEGUIMIENTO DEL SÍLABO | REVISIÓN: 1 Página 22 de 23 |

```

setTimeout(function() {
    document.querySelector(".notification").style.display = "none";
}, 3000);
}

```

2.6.8. Manejando cambios dinámicos

Hoy en día usamos muchos cambios dinámicos para comunicar si algún dato ha cambiado o si el usuario necesita ponerle algo nuevo, para manejar estos cambios dinámicos.

Lo primero que vamos a hacer es modificar la notificación para que esta sea anunciada por el lector de pantalla o por el teclado y para esto vamos a usar una propiedad de ARIA llamado ARIA-LIVE, y la manera que aria-live funciona es añadiendo dos atributos assertive o polite (se usa cuando no quieras interrumpir la lectura del lector de pantalla y se la puede usar para una notificación que no es muy importante o que no necesita la atención inmediata del usuario), en este caso queremos informarle al usuario de que la información fue enviada sin errores por eso usaremos assertive porque este elemento hace que interrumpa al lector de pantalla para que comunique al usuario algo importante que el formulario fue enviado. El assertive funciona leyendo si el elemento html dentro de este ha cambiado y lo anuncia.

```

<footer class="contact-section">
    <h2 class="header">Contacto</h2>
    <p class="intro">Para manderme un mensaje puedes usar el formulario que sigue</p>
    <form class="form-container">
        <label class="form-label">Nombre</label><input type="text" aria-label="nombre" id="name" />
        <span id="name-error" aria-live="assertive"></span>
        <br />
        <label class="form-label">Correo</label><input type="text" aria-label="correo" /> <br />
        <label class="message-input">Mensaje<input type="text" /> <br /></label>
        <button class="send-button">Enviar</button>
    </form>
</footer>

<div class="notification" aria-live="assertive"></div>

```

| | | |
|---|--|---------------------------------------|
|  Uleam <small>UNIVERSIDAD LAICA ELOY ALFARO DE MANABÍ</small> | NOMBRE DEL DOCUMENTO: ELABORACIÓN DE GUÍA DE ESTUDIOS PARA EDUCACIÓN EN LÍNEA | CÓDIGO: PAA-03-F-018 |
| | PROCEDIMIENTO: ELABORACIÓN, MEJORAMIENTO Y SEGUIMIENTO DEL SÍLABO | REVISIÓN: 1 Página 23 de 23 |

```
/** Esta función se llama cuando la persona hace click en el botón de enviar
   del formulario de contacto */
function showNotification() {
  document.getElementById("name-error").innerHTML = "";
  document.querySelector('.form-container').reset();
  document.querySelector(".notification").innerHTML= "El formulario fue
  enviado sin errores";
  document.querySelector(".notification").style.display = "flex";
  setTimeout(function() {
    document.querySelector(".notification").style.display = "none";
  }, 3000);
}
```