

Mejorando el contenido y estructura del sitio web

Introducción

¡Hola!  Te damos la bienvenida a un nuevo encuentro.

En clases anteriores hemos abordado los conceptos básicos de HTML, CSS, JavaScript y en el último encuentro has aplicado lo aprendido para crear una barra de navegación adaptable para tu sitio web.

En la clase de hoy nos enfocaremos en cómo mejorar el contenido y su organización, la forma adecuada de crear y vincular páginas, agregar medios y aplicar buenas prácticas de accesibilidad que capten la atención de los usuarios.


Al final habrás mejorado el sitio web sobre el que vienes trabajando.

¡Comencemos! 

Actividad 1

Vamos a iniciar con una actividad que tiene como objetivo crear las páginas adicionales que irán enlazadas en la barra de navegación que creamos en la clase pasada.

¡Manos a la obra!

 *Antes de comenzar recuerda crear una nueva carpeta con el nombre de la sección “Sección 8 – Mejorando el contenido y estructura del sitio web” y hacer una copia del documento “.html” sobre el que venías trabajando.*

1. Crear dos nuevos archivos HTML en la misma carpeta “mi-primer-pagina-web.html” y nombrar los archivos “services.html” y “contact.html” (puedes personalizarlos de acuerdo a las páginas que desees crear pero en este ejercicio usaremos esos nombres para identificarlas).

2. En el archivo "services.html" copiar el contenido de "mi-primera-pagina-web.html" como punto de partida. Reemplazar el contenido dentro del elemento `<div class="container">` con el siguiente código:

```
<h1>Nuestros servicios</h1>
<p>Descubre aquí los servicios que ofrecemos.</p>
<ul>
  <li>Servicio 1</li>
  <li>Servicio 2</li>
  <li>Servicio 3</li>
</ul>
```

Personaliza el contenido del código de manera que sea funcional a tu sitio. Por otra parte, en esta página, puedes borrar las etiquetas `<button>` y `<form>` junto con el contenido de las mismas.

3. En el archivo "contact.html" también copiar el contenido de "mi-primera-pagina-web.html" como punto de partida. Reemplazar el contenido dentro del elemento `<div class="container">` con el siguiente código:

```
<h1>Contacto</h1>
<p>¡Contáctanos para cualquier consulta o solicitud de información!</p>
<form>
  <!-- Agrega aquí los campos del formulario -->
</form>
```


A la etiqueta `<form>` vamos a reemplazarla con la que habíamos creado previamente y eliminaremos la etiqueta de `<button>` junto con su contenido.

También puedes eliminar la etiqueta `<form>` de "mi-primera-pagina-web.html" al quedar agregada en la página de contacto.

4. Actualizar los enlaces de navegación del elemento `<nav>` en "mi-primera-pagina-web.html" cambiando los atributos "href" para que incluyan las nuevas páginas. Por ejemplo, para "servicios", reemplazaremos el "#" por "services.html".

5. Actualizar los enlaces de la barra de navegación tanto en "services.html" como en "contact.html". También ten presente agregar el enlace para el inicio ("home").

6. Guardar todos los archivos ("mi-primera-pagina-web.html", "services.html" y "contact.html") y actualizar tu página web en el navegador. Deberías ver los enlaces actualizados en la barra de navegación.

 *Haz clic en los enlaces de Servicios ("services"), Contacto ("contact") e Inicio ("home") para verificar que las páginas se carguen correctamente.*

¡Primera actividad del día terminada! Has logrado crear nuevas páginas y las has vinculado correctamente en la barra de navegación de tu sitio web.

Continuemos...

Actividad 2

Ahora seguiremos con una nueva actividad que tiene como objetivo continuar ampliando el sitio web y agregar contenido visual.

¡Manos a la obra!

1. Crear un nuevo archivo HTML llamado *"about.html"* en la misma carpeta que *"mi-primera-pagina-web.html"*. Este nuevo archivo será tu página *"Acerca de"*.
2. Copiar el contenido de *"services.html"* en *"about.html"* como punto de partida. Reemplazar el contenido dentro del elemento `<div class="container">` con el siguiente código:


```
<h1>Acerca de</h1>
<p>¡Bienvenido a nuestra página "Acerca de"! Aquí, aprenderás más sobre
nuestra empresa y los servicios que ofrecemos.</p>
```

Personaliza el párrafo con información acorde al sitio que estás creando. También puedes agregar más párrafos de ser necesario.

3. Actualizar los enlaces de la barra de navegación de todos los archivos (*"mi-primera-pagina-web.html"*, *"services.html"*, *"contact.html"* y *"about.html"*) para que apunten a las páginas correctas.

Agregar, además, el atributo *"role"* al elemento `<nav>` en todas las páginas para mejorar la accesibilidad de nuestro sitio web:

```
<nav class="navbar" role="navigation">
```

 *Este atributo proporciona información adicional a las tecnologías de asistencia como lectores de pantalla para ayudar a los usuarios en condición de discapacidad a navegar y comprender mejor el contenido de tu sitio web.*

4. Ahora vamos a agregar imágenes a tu página *"Acerca de"*. Para ello, debes descargar tres imágenes que desees usar y guardarlas en la misma carpeta que tus archivos HTML.

En "about.html" agregar tres etiquetas `` dentro del elemento `<div class="container">`, debajo de la etiqueta `<p>`:

```





```

Reemplazar "elnombredetuimagen(...).jpg" con el nombre real de cada una de tus imágenes y proporcionarles una descripción relevante en el atributo "alt".

5. Alinear horizontalmente y centrar las tres imágenes utilizando Flexbox.

Los elementos deberían quedar alineados de esta manera:



💡 Necesitarás agregar un nuevo elemento contenedor ("div") a las imágenes para luego poder aplicarles los estilos CSS correspondientes.

6. Guardar todos los archivos ("mi-primera-pagina-web.html", "services.html", "contact.html" y "about.html") y actualizar la página en el navegador. Deberías ver los enlaces de navegación actualizados y la nueva imagen en la página "Acerca de".

👉 [Ver video](#)

¡Segunda actividad finalizada! 🎉

Desafío del día

¡Llegamos al desafío del día! El mismo consiste en mejorar la estructura de la página de inicio de tu sitio web además de agregar buenas prácticas de accesibilidad. También le incorporaremos un elemento audiovisual para enriquecerla con más contenido.

¡Empecemos!

1. En el archivo HTML *"mi-primer-pagina-web.html"*, reorganizar y mejorar la estructura del contenido principal dentro del elemento `<div class="container">` con el siguiente código.

```
<div class="container">
  <h1>Bienvenidos a mi primer Website</h1>
  <section>
    <h2>Presentación</h2>
    <p>Soy [Tu nombre], ¡y esta es mi primera página web!</p>
    <p>Estoy aprendiendo desarrollo web para crear sitios web visualmente
atrapantes y funcionales.</p>
  </section>
  <section>
    <h2>Lo que se viene...</h2>
    <p>
      A lo largo de este curso, estaré aprendiendo HTML, CSS y JavaScript. Estos
lenguajes me ayudarán a crear una base sólida en el desarrollo web front-
end.
    </p>
  </section>
</div>
```

2. Agregar un *elemento audiovisual* después del div con la clase *"container"*. Puedes buscar un video de Youtube o usar uno propio. Asegúrate de tener el archivo de video en el mismo directorio que tus archivos HTML.

- **Para un video de Youtube:** Podrás hacer clic en el botón de "Compartir" y luego seleccionar la opción de "Incorporar" para obtener el código iframe que deberás sumar a tu archivo HTML.

Ejemplo:

```
<iframe width="560" height="315"
src="https://www.youtube.com/embed/qWcwFMAqpME" title="YouTube video
player" frameborder="0" allow="accelerometer; autoplay; clipboard-write;
encrypted-media; gyroscope; picture-in-picture; web-share" allowfullscreen>
</iframe>
```

- **Para un video propio:**

```
<video src="nombre-del-video.mp4" controls></video>
```

Reemplaza "nombre-del-video.mp4" con el nombre y extensión real del archivo de video que deseas agregar. El atributo "controls" permitirá que se muestren los controles de reproducción del video en el navegador.

3. Ahora mejoraremos la accesibilidad del sitio web agregando el atributo "label" al botón que tenemos agregado. En el archivo "mi-primer-a-pagina-web.html", actualizar el código del botón existente de la siguiente manera:

```
<button onClick="showAlert()" aria-label="Dispara una alerta de saludo">¡Haz clic en mí!</button>
```


El atributo "aria-label" proporciona una etiqueta alternativa para el botón que es leída por las tecnologías de asistencia, como lectores de pantalla, para mejorar la accesibilidad.

4. Incluir las etiquetas <header>, <main> y <footer> dentro del <body>. Esto mejorará la estructura semántica del documento HTML y agregará información adicional para las tecnologías de asistencia.

```
<body>
  <header class="container-nav">
    <nav class="navbar">
      <!-- Contenido de la barra de navegación -->
    </nav>
  </header>

  <main>
    <!-- Contenido principal -->
  </main>

  <footer>
    <!-- Contenido del pie de página -->
  </footer>
</body>
```

 *Dentro de la etiqueta <header> puedes colocar la barra de navegación o cualquier otro contenido relacionado con la cabecera de la página, y en el elemento <footer>, coloca el contenido del pie de página como información de contacto o enlaces adicionales.*

5. Sumar contenido en la sección del *footer* (aunque sea un párrafo) y colocar allí dentro el botón que habíamos creado en una de las clases anteriores. *El footer deberá tener un alto mínimo de 200px.*

6. Alinear los elementos tanto del `<main>` como del `<footer>` utilizando la propiedad de *Flexbox*, logrando que su contenido quede centrado como en este ejemplo:

Lo que se viene...

A lo largo de este curso, estaré aprendiendo HTML, CSS y JavaScript. Estos lenguajes me ayudarán a crear una base sólida en el desarrollo web front-end.



Este es el footer de mi página

[¡Haz clic en mí!](#)

7. Agregar más estilos a tu archivo *"style.css"* para personalizar el inicio de tu sitio web y hacerlo más atractivo visualmente.

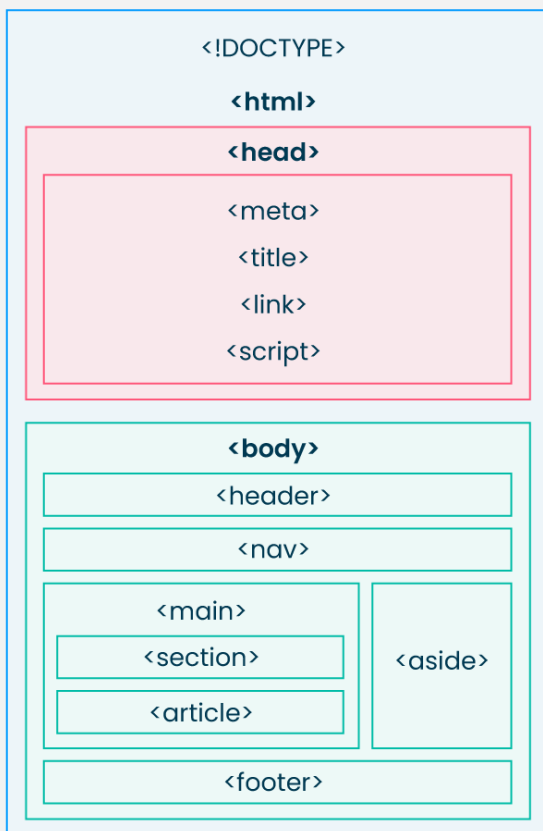
Al seguir estos pasos habrás mejorado la estructura de tu página de inicio, agregado un elemento audiovisual y aplicado prácticas de accesibilidad importantes. Recuerda siempre probar tu página en diferentes dispositivos y navegadores para garantizar una experiencia óptima para todos los usuarios.

¡Desafío terminado! 🎉

Etiquetas semánticas

En el desafío anterior incluimos algunas **etiquetas semánticas** en el HTML que **son elementos que proporcionan información adicional sobre la estructura y el significado del contenido dentro de un documento web**

Estructura página web



A diferencia de las etiquetas no semánticas, las etiquetas semánticas tienen un propósito y significado específicos.

A continuación, te recordamos la utilidad de algunas de estas etiquetas:

- **<header>**: Se utiliza para incluir contenido introductorio o de navegación en la parte superior de una página como el logo del sitio, el título de la página, el menú de navegación y otros elementos relacionados con su identidad.
- **<nav>**: Se utiliza para definir una sección de navegación dentro de un sitio web. Suele contener enlaces a otras páginas o secciones del mismo sitio.
- **<main>**: Se utiliza para envolver el contenido principal de la página excluyendo encabezados, pies de página, barras laterales u otras secciones. Ayuda a los usuarios a enfocarse en la información más relevante. Es importante tener en cuenta que solo debe haber un elemento `<main>` por documento HTML.

- **<article>**: Se utiliza para identificar una sección independiente o autónoma de contenido dentro de un documento. Por ejemplo, puede ser utilizado para un artículo de noticias, una publicación de blog o un comentario.
- **<section>**: Se utiliza para agrupar contenido relacionado dentro de un documento. Puede ser utilizado para dividir el contenido en diferentes secciones temáticas o para organizar el contenido en una página.
- **<aside>**: Se utiliza para marcar el contenido relacionado pero no esencial para el flujo principal del documento. Puede contener información adicional como publicidad, enlaces relacionados, citas o elementos destacados.
- **<footer>**: Se utiliza para incluir información de contacto, enlaces relacionados, derechos de autor y otros elementos que suelen ubicarse en la parte inferior de una página. Proporciona una ubicación específica para la información de cierre.

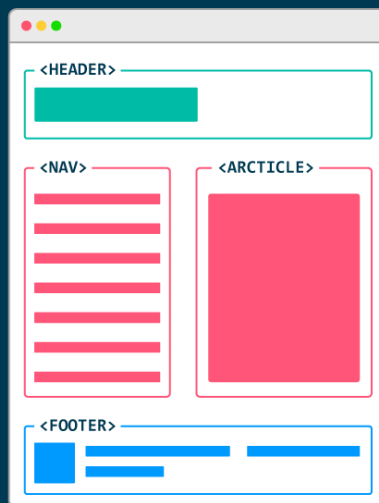
💡 **Etiquetas <div> y **: Aunque no son etiquetas semánticas en sí mismas, las etiquetas <div> y se utilizan comúnmente como contenedores genéricos para agrupar y estilizar elementos. Pueden ser útiles para organizar el contenido y aplicar estilos específicos pero se recomienda utilizar las etiquetas semánticas cuando sea posible.

En resumen, **el uso adecuado de todas estas etiquetas proporciona una estructura clara y semántica, facilitando la navegación y mejorando la accesibilidad del sitio web**. De igual forma, permiten a los motores de búsqueda y tecnologías de asistencia comprender mejor el contenido y presentarlo de manera adecuada a los usuarios.

Sin HTML semántico



Con HTML semántico



Otro recurso a tener en cuenta es el de los **comentarios** en el código HTML (`<!-- Comentario -->`). No afectan la visualización del contenido en el navegador pero **son útiles para agregar notas y aclaraciones dentro del código**. Puedes utilizar comentarios para explicar la estructura de tu página, hacer anotaciones o proporcionar instrucciones para otros desarrolladores.

Resolución del desafío

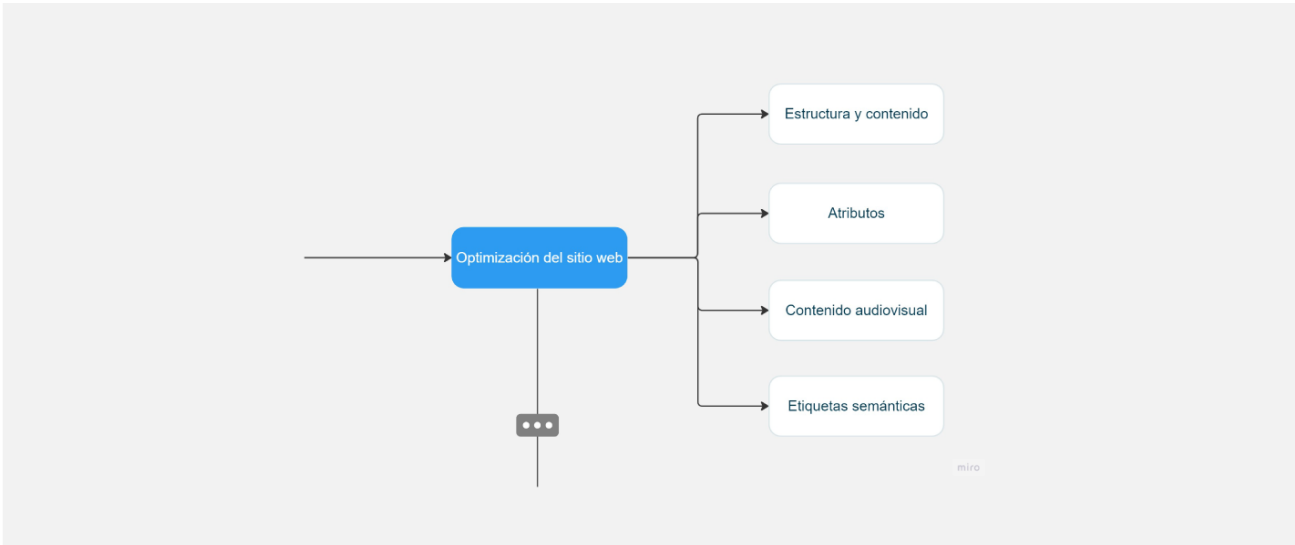
En el siguiente video te compartimos un paso a paso de cómo resolver el desafío de hoy:

 [Resolución de Ejercicio | Mejorando el contenido y estructura del sitio web |](#)

Egg

Mapa de conceptos vistos

¡Felicidades! 🎉 Has alcanzado el hito de este encuentro agregando más contenido, vinculando páginas, añadiendo medios e implementando las mejores prácticas de accesibilidad para hacer que tu sitio web sea lo que tus usuarios esperan.



¿Nos cuentas cómo te fue hoy?

 [Responder encuesta](#)

En la siguiente clase nos centraremos en el proyecto final y repasaremos los conceptos clave que hemos aprendido a lo largo del curso. Crearás un sitio web completo y adaptable desde cero, participarás en la revisión y retroalimentación entre pares y discutirás los próximos pasos y recursos para seguir aprendiendo.

¡Nos vemos pronto! 