

Creando una barra de navegación responsive

Introducción

¡Te damos la bienvenida a una nueva clase! 🙌

En encuentros anteriores, hemos abordado los conceptos básicos de **HTML**, **CSS** y **JavaScript**.

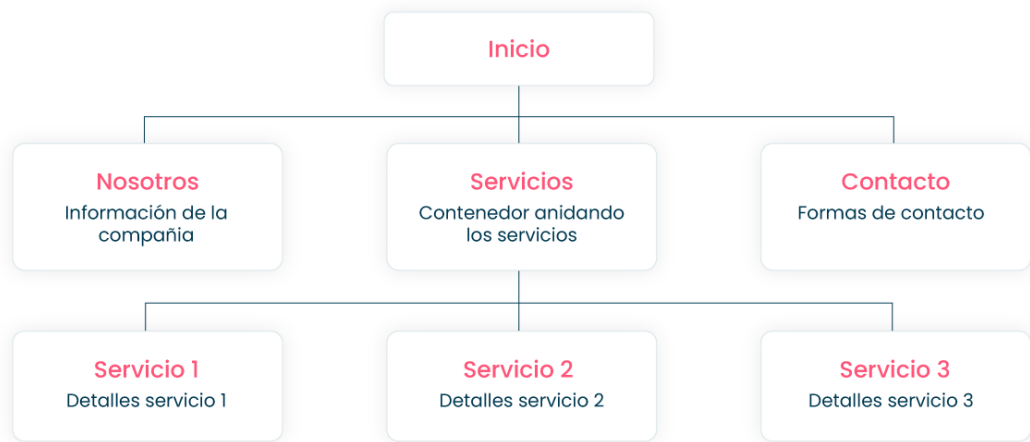
En la clase de hoy construiremos una barra de navegación adaptable (“responsive”) para tu sitio web. Comenzaremos con la planificación de su estructura, continuaremos con la creación del HTML y CSS para hacerla adaptable, y finalizaremos con la forma de aplicar JavaScript para brindarle interactividad.

Al final de este encuentro *tendrás una barra de navegación (o también conocida como menú de navegación) responsive para la página HTML que creaste hasta ahora a lo largo del curso.*




Estructura de la barra de navegación

Antes de comenzar a escribir código **es fundamental planificar la estructura de tu barra de navegación.**



Imaginemos que estamos armando la página web de un negocio que vende plantas, ¿cuáles serían las opciones que deberían estar a primera vista en su menú de navegación?

¡Discutámoslo un poco!

 Pongan en común sus ideas y tómense 10 minutos para definir al menos 5 opciones que deberían estar en la barra de navegación de una tienda de plantas. Piensen en las secciones o categorías que serían relevantes para los usuarios.

¿Coincidieron en todas?, ¿en algunas? No hay respuestas correctas o incorrectas pero sí hay buenas prácticas a tener en cuenta al momento de idear la estructura de nuestra barra de navegación.

Por ejemplo, por tratarse de una tienda de plantas, una de las opciones claves que como usuarios esperaríamos poder ver en el menú de navegación es la de “Plantas” o un término asociado que nos lleve a poder verlas, ¿no es cierto?.

Ahora, busquen en el buscador de “Google”: *tienda online de plantas*. Visiten al menos 5 tiendas y revisen sus barras de navegación.


 **Una persona de la mesa de trabajo puede compartir pantalla para ir viéndolas al mismo tiempo.**


Y ahora, analicen, ¿Coinciden con las opciones que habían definido inicialmente? ¿Encuentran elementos comunes entre las diferentes tiendas?.

Perfecto, ahora realicemos una actividad similar pero para una página web de recetas.

¡Manos a la obra!

1. Simular que están creando una página web de recetas y definir 5 opciones principales que deberían estar en la barra de navegación de esta página.
2. Comparar y analizar con el equipo si hay similitudes o diferencias en las opciones propuestas. *No hay respuestas correctas o incorrectas, pero es importante considerar la experiencia del usuario y la relevancia de las secciones en la barra de navegación.*
3. Realizar una búsqueda en el buscador de "Google" utilizando el término "página web de recetas". Visitar al menos 5 páginas de recetas analizando sus barras de navegación.
4. Como grupo, compartir y analizar las diferentes estructuras de las barras de navegación encontradas; ¿Coinciden con las opciones que se habían definido inicialmente? ¿Se encuentran elementos comunes entre las diferentes páginas de recetas?
5. Tomar nota de las ideas y estructuras que se consideren interesantes y relevantes para luego poder aplicarlas en la página web propia.

 *Un buen sistema para determinar si la estructura de una página web y su navegabilidad son buenas es mediante la **regla del uso de los 3 clics**. Si el usuario no puede llegar de un punto de la página a su objetivo final mediante el uso de 3 clics implica que la navegabilidad puede mejorar.*

Para conocer otras recomendaciones útiles como la anterior, te invitamos a leer el material complementario de  **Buenas prácticas para una barra de navegación responsive.**

Ahora que tenemos una mejor comprensión de las opciones y estructuras comunes en las barras de navegación ya podemos comenzar a implementarla en el código HTML y CSS.

HTML para la barra de navegación

Muy bien, ¿qué te parece si empezamos a darle forma a una barra de navegación?

Para eso, vamos a comenzar con una actividad:

¡Manos a la obra!

1. Abrir el editor de código de Visual Studio Code.
2. Crear un nuevo archivo HTML y guardarlo como *"index.html"*.
3. Crear una estructura básica de HTML. Asegúrate de incluir las etiquetas `<html>`, `<head>` y `<body>`.
4. Dentro de la etiqueta `<body>` agregar la etiqueta `<header>` y dentro de la misma una etiqueta `<nav>` para crear la barra de navegación.
5. Agregar una lista "desordenada" `` dentro de la etiqueta `<nav>`, .
6. Agregar un mínimo de 3 elementos dentro de lista ``. Cada uno de ellos representará un enlace en tu barra de navegación.
7. Crear las páginas correspondientes que se vincularán a los enlaces de la barra de navegación. Puedes hacerlo creando nuevos archivos HTML en el mismo directorio que tu archivo de *"index"*.
8. Agregar un enlace `<a>` dentro de cada elemento de lista ``, .
 - Usar el atributo *"href"* para especificar la dirección URL a la que el enlace debe dirigirse (por ejemplo: ``).
 - Agregar un texto descriptivo dentro de cada etiqueta `<a>` para indicar a dónde se dirige el enlace (por ejemplo: ` Inicio `).
9. Guardar el archivo HTML.
10. Abrir el archivo HTML en tu navegador web para ver la barra de navegación creada.

¡Fin de la actividad! ¿Pudiste lograrlo?

Utilizar HTML y CSS para crear una barra de navegación es ideal debido a su capacidad para estructurar y personalizarla de manera efectiva permitiendo mantener el diseño limpio, accesible y optimizado para el rendimiento.

En lo que respecta a la **estructura HTML de la barra de navegación**, en la actividad anterior vimos que se puede incluir en la etiqueta `<header>` (que no es la misma que la etiqueta `<head>`) que va dentro de la etiqueta `<body>` del documento HTML.

La etiqueta `<header>` es utilizada para definir el encabezado o cabecera de una sección o página web. Esta etiqueta se utiliza para incluir información como el

logotipo, la barra de navegación, un carrito de compras, entre otros elementos importantes del encabezado de un sitio web.

💡 *Es una buena práctica separar la barra de navegación del resto del contenido de la página teniendo en cuenta que a menudo es una sección fija que aparece en todas las páginas del sitio web.*

Luego, **para crear la barra de navegación utilizamos la etiqueta `<nav>`**. Dentro de esta etiqueta incluimos una lista “desordenada” `` que contenía los elementos de lista ``. En cada elemento de la lista incluimos un enlace `<a>` para que llevara a diferentes secciones del sitio web u otras páginas.

Te compartimos un ejemplo de cómo debería haber quedado la estructura del código solicitado en la actividad:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Barra de navegación</title>
  </head>
  <body>
    <header>
      <nav>
        <ul>
          <li><a href="index.html">Inicio</a></li>
          <li><a href="servicios.html">Servicios</a></li>
          <li><a href="nosotros.html">Sobre nosotros</a></li>
          <li><a href="contacto.html">Contacto</a></li>
        </ul>
      </nav>
    </header>
  </body>
</html>
```

Una vez armada la estructura de la barra de navegación ya podemos avanzar con la parte de CSS.

¡Sigamos!

CSS para la barra de navegación

Con CSS, vamos a poder aplicar estilos a la barra de navegación para darle el

aspecto deseado y mejorar su apariencia. Por ejemplo, se pueden cambiar la fuente, el tamaño, el color, tipografías y otros aspectos visuales.

Pongámoslo en práctica dándole estilos a la barra de navegación que acabaste de crear.

Nuevamente, **¡manos a la obra!**

1. Crear un nuevo archivo en el mismo directorio que tu archivo `"index.html"` y guárdalo como `"style.css"`.

2. Enlazar el archivo CSS a tu archivo HTML agregando la siguiente línea dentro de la etiqueta `<head>` de tu archivo HTML:

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

3. Agregar estilos a la etiqueta `<header>` para darle un color de fondo y centrar su contenido en la página. Puedes utilizar el siguiente código CSS:

```
header {  
  background-color: #333;  
  color: #fff;  
  text-align: center;  
  padding: 20px;  
}
```

Deberías ver que el fondo del header ahora es de color gris oscuro y que el texto está centrado en la página.

4. Agregar estilos a la etiqueta `<nav>` para darle un estilo de lista y quitarle los márgenes predeterminados. Puedes utilizar el siguiente código CSS:

```
nav ul {  
  list-style: none;  
  margin: 0;  
  padding: 0;  
}
```

Deberías ver que la lista ahora no tiene viñetas y que los márgenes han desaparecido.

5. Agregar estilos a los elementos de lista `` para hacer que se muestren en línea y se separen entre sí. Puedes utilizar el siguiente código CSS:

```
nav li {  
  display: inline-block;
```

```
margin-right: 20px;
}
```

Deberías ver que los elementos de lista ahora se muestran en línea y están separados entre sí.

6. Agregar estilos a los enlaces `<a>` para darles un color y un estilo al pasar el cursor por encima. Puedes utilizar el siguiente código CSS y personalizarlo:

```
nav a {
  color: #fff;
  text-decoration: none;
  padding: 5px;
}
nav a:hover {
  color: #1BF7DE;
}
```

Deberías ver que los enlaces ahora son de color blanco y que al pasar el cursor por encima cambian al color celeste.

7. Personalizar a gusto los valores de los diferentes códigos de CSS que agregamos para darle a la barra de navegación un estilo de tu preferencia.

8. Guardar el archivo CSS y actualizar la página del navegador web para ver los cambios.

¡Y listo! Finalizamos con la actividad.

Al combinar el código HTML y CSS proporcionado, construiste una barra de navegación básica y funcional que se puede personalizar y adaptar según las necesidades de un sitio web.

Ahora veamos cómo podemos lograr que se adapte a las pantallas de diferentes dispositivos.

Barra de navegación responsiva

Como ya vimos en la clase 3, el **diseño responsive o adaptable** permite crear sitios web que se ajusten de manera automática a diferentes tamaños de pantalla y dispositivos.

Ver video

Entre los beneficios del *diseño responsive* se encuentran:

- **Mejor experiencia de usuario** → Permite que el sitio web se ajuste al tamaño de pantalla del dispositivo lo que facilita la navegación y la lectura del contenido.
- **Velocidad de carga** → Mejora la velocidad de carga de la página especialmente en dispositivos móviles con conexiones de internet más lentas.
- **SEO** → Favorece los motores de búsqueda lo que puede aumentar la visibilidad en los resultados de búsqueda y generar más tráfico.
- **Mantenimiento** → Facilita el mantenimiento del sitio web ya que solo es necesario actualizar una única versión del sitio en lugar de múltiples versiones para diferentes dispositivos.

Por todo esto y más **es importante entonces tener una barra de navegación responsive que se adapte a diferentes tamaños de pantalla y dispositivos** garantizando una experiencia consistente y agradable para todos los usuarios.

¡Pongámoslo en práctica!

Para ello, vamos a aplicar el diseño responsive a la barra de navegación que creamos y personalizamos en la actividad del paso anterior:

1. Abrir el archivo HTML ("*index.html*") que creaste en la actividad anterior.
2. Agregar la etiqueta "*meta viewport*" dentro de la etiqueta `<head>`. Esta etiqueta define cómo el contenido debe ser renderizado en diferentes dispositivos.

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

3. Abrir el archivo CSS ("*style.css*") que creaste en la actividad anterior.
4. Agregar *media queries*. Por ejemplo, para cambiar el color de fondo del encabezado en pantallas pequeñas agregar lo siguiente al final del archivo CSS:


```
@media (max-width: 767px) {  
  header {  
    background-color: #333;  
  }  
}
```

Esto cambia el color de fondo del encabezado a #333 en pantallas con un ancho máximo de 767 píxeles.

5. Agregar reglas CSS específicas para diferentes tamaños de pantalla. Por ejemplo, para cambiar el tamaño de fuente del texto de navegación en pantallas pequeñas agregar lo siguiente al final del archivo CSS:

```
@media (max-width: 767px) {  
  nav {  
    font-size: 16px;  
  }  
}
```

Esto cambia el tamaño de fuente del texto de navegación a 16 píxeles en pantallas con un ancho máximo de 767 píxeles.

6. Agregar más reglas CSS para adaptar el contenido a diferentes tamaños de pantalla. Por ejemplo para ajustar el ancho del encabezado en pantallas grandes agregar lo siguiente al final del archivo CSS:

```
@media (min-width: 1200px) {  
  header {  
    width: 1200px;  
  }  
}
```

Esto ajusta el ancho del encabezado a 1200 píxeles en pantallas con un ancho mínimo de 1200 píxeles.

7. Guardar los cambios y revisar cómo se comporta la barra de navegación al ajustar el tamaño de la ventana del navegador en diferentes tamaños. Asegúrate de que todos los elementos se vean y funcionen correctamente.

¡Y listo! Te recomendamos seguir experimentando luego con diferentes estilos y diseños.

JavaScript para interactividad

JavaScript se puede utilizar para agregar interactividad a tu barra de navegación permitiendo abrir y cerrar un menú desplegable o alternar la visibilidad de la barra de navegación en pantallas más pequeñas.

Como hemos visto en la clase 5 y practicado a lo largo de la clase 6, JavaScript es un programa que permite implementar una gran variedad de cambios y personalizaciones en el código HTML de una página web. Por lo anterior, aplicarlo en la barra de navegación va a:

- **Facilitar la navegación:** Al agregar interactividad mejora la experiencia de usuario al facilitar el acceso a las diferentes secciones del sitio web.
- **Mejorar la apariencia:** A través del uso de efectos visuales y animaciones a los elementos.
- **Aumentar la accesibilidad:** La interactividad en la barra de navegación puede hacer que un sitio web sea más accesible para cualquier perfil de usuario al proporcionar una navegación más sencilla y funcional.

Habiendo quedado claro el por qué es beneficioso aplicarlo, pasemos a aplicarlo en el desafío que sigue..

Desafío del día

Muy bien, ¡llegó el momento de aplicar todo lo aprendido! El desafío de hoy consiste en crear una barra de navegación adaptable para la página HTML sobre la que viniste trabajando en los encuentros anteriores.

⚠ *Antes de comenzar recuerda crear una nueva carpeta con el nombre de la sección “Sección 7 - Creando una barra de navegación responsive” y hacer una copia del documento “.html” sobre el que venías trabajando.*

Ahora sí, ¡empecemos!

1. Abrir tu archivo “*mi-primer-pagina.html*” y agregar el siguiente código HTML para la barra de navegación dentro de la etiqueta `<body>`, antes del elemento “*.container*”:

```
<div class="container-nav">

  <nav class="navbar">

    <a href="#" class="brand">Mi página web</a>

    <ul class="nav-links">

      <li><a href="#">Home</a></li>

      <li><a href="#">About</a></li>

      <li><a href="#">Services</a></li>

      <li><a href="#">Contact</a></li>

    </ul>

  </nav>

  <button class="button-menu-toggle">Menu</button>

</div>
```

Este código crea una barra de navegación simple con un nombre de marca y cuatro enlaces de navegación.

2. Cambiar los nombres de “Home”, “About”, “Services” y “Contact” por los de tu preferencia.

3. Agregar el siguiente código CSS en tu archivo “script.js” para estilizar la barra de navegación:

```
.container-nav {
  display: flex;
  padding: 10px;
  height: 50px;
  background-color: black;
}

.navbar {
  display: flex;
```

```
justify-content: space-between;  
width: 100%;  
}
```

```
.brand {  
  display: flex;  
  align-items: center;  
}
```

```
li>a {  
  padding: 8px 24px;  
}
```



```
li {  
  list-style-type: none;  
}
```

```
a {  
  color: white;  
  text-decoration: none;  
}
```

```
.nav-links {  
  display: flex;  
  align-items: center;  
}
```

```
.button-menu-toggle {  
  display: none;  
}
```

Este código estiliza la barra de navegación con un fondo oscuro, texto blanco y un diseño horizontal para los enlaces.

 Te sugerimos investigar la página de  coolers.co y elegir una paleta de colores para aplicarla a tu página web y así darle más identidad.

4. Agregar el siguiente código CSS a tu archivo “styles.css” para hacer que la barra de navegación sea adaptable, utilizando una media query:

```
@media (max-width: 768px) {  
  .column {  
    flex: 100%;  
  }  
}
```

```
.nav-links {  
  display: none;  
}
```

```
.button-menu-toggle {  
  display: block;  
  background-color: white;  
  border: none;  
  border-radius: 50px;  
  padding: 0 24px;  
  font-family: monospace;  
  font-weight: bold;
```

```
font-size: 20px;
}

.nav-links.nav-links-responsive {
  position: absolute;
  display: flex;
  background-color: rgba(20,20,20,0.95);
  width: 100%;
  height: 100%;
  top: 70px;
  left: 0;
  flex-direction: column;
  padding-top: 200px;
  align-items: center;
  gap: 70px;
  font-size: 40px;
}
}
```

Este código oculta los enlaces de navegación y muestra el botón “Menú” en pantallas con un ancho de 768 píxeles o menos, y utiliza una clase extra para poder hacer que el botón menú despliegue los enlaces .

5. Agregar JavaScript para alternar la visibilidad de los enlaces de navegación cuando se haga clic en el botón “Menú”.


En tu archivo “script.js” agregar el siguiente código:

```
document.querySelector("button.button-menu-toggle")
  .addEventListener("click", function() {
    document.querySelector(".nav-links").
```

```
classList.toggle("nav-links-responsive"))
```

Este código agrega un detector de eventos al botón “Menú” que alterna la clase “nav-links-responsive” del elemento con la clase “.nav-links” cuando se hace clic.

6. Guardar todos tus archivos (“mi-primer-pagina.html”, “styles.css” y “script.js”) y actualizar tu página web en el navegador.

 Recuerda usar esta línea de código “<meta name=“viewport” content=“width=device-width, initial-scale=1”>” para poder ver mejor la pagina correctamente en dispositivos mobile.

Deberías ver la barra de navegación en la parte superior de la página. Intenta cambiar el tamaño de la ventana del navegador a un ancho menor y deberías ver que los enlaces de navegación desaparecen y aparece el botón “Menú”. Haz clic en el botón “Menú” y los enlaces de navegación deberían aparecer en un menú desplegable.

¡Desafío terminado! 

Resolución del desafío

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

 [Resolución de Ejercicio | Creando una barra de navegación responsive | Egg](#)

Valida tus conocimientos

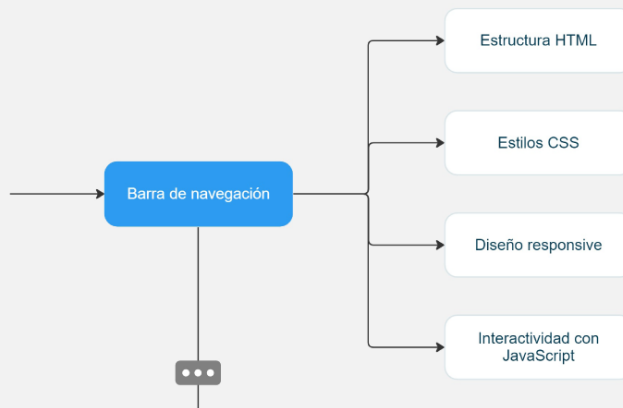
Te compartimos el enlace del **material complementario** para que puedas continuar profundizando sobre el siguiente tema en el momento que quieras:

- [Buenas prácticas para una barra de navegación responsive](#)

Además, te proponemos realizar la autoevaluación para poner a prueba los principales aprendizajes del día de hoy. ¿Vamos?

Mapa de conceptos vistos

¡Felicidades! 🎉 En el día de hoy has creado con éxito una barra de navegación adaptable para tu página HTML. Has aprendido cómo planificar la estructura de la barra de navegación, crear el HTML y CSS para ella, cómo hacerla adaptable y agregar JavaScript para lograr que sea atractiva e interesante.



En la siguiente clase, nos centraremos en mejorar el contenido y la estructura del sitio web. Cubriremos la organización del contenido, la creación y vinculación de páginas, el agregado de medios audiovisuales y la implementación de las mejores prácticas de accesibilidad.

Por último queremos entender cómo viene siendo tu experiencia. Por te pedimos que nos ayudes respondiendo la siguiente encuesta:

 [Responder encuesta](#)

¡Hasta la próxima! 🙌