

## 2.2.2. Diseño responsivo y Flexbox



**LENGUAJES DE PROGRAMACIÓN NIVEL 5. UNIDAD 2**

DESARROLLO FRONTEND AVANZADO

## Contenido

Introducción al Diseño Responsivo y Flexbox .....	3
Importancia del Diseño Responsivo en el Desarrollo Web .....	4
Diseño responsivo y Flexbox.....	5
Diseño Responsivo con Media Queries.....	6
Conceptos Básicos y Sintaxis .....	7
Conceptos Básicos: .....	7
Ventajas de las Media Queries:.....	10
Flexbox: Conceptos Básicos .....	11
Elementos Fundamentales.....	11
Contenedores Flexibles .....	11
Elementos Flexibles:.....	11
Conceptos clave.....	11
Casos de Uso .....	12
Flexbox: Propiedades y Funcionalidades Avanzadas .....	13
Flex Grow, Flex Shrink y Flex Basis: .....	13
Flex Align y Flex Justify: .....	14
Ordenamiento de Elementos:.....	14
Ejemplos Prácticos: .....	14
Casos de Uso y Ejemplos de Implementación:.....	15
Diseño Responsivo con Flexbox y Media Queries.....	16
Integración de Flexbox y Media Queries:.....	16
Ejemplos Prácticos: .....	16
Casos de Uso y Ejemplos de Implementación:.....	17
Mejores Prácticas y Consideraciones .....	18
Conclusiones.....	20
Despedida y Recursos Adicionales .....	21
Recursos Adicionales .....	21

## Introducción al Diseño Responsivo y Flexbox



En el mundo actual de la web, donde la diversidad de dispositivos y pantallas es la norma, el diseño responsivo se ha vuelto esencial. La capacidad de adaptar una página web a diferentes tamaños de pantalla y dispositivos garantiza una experiencia de usuario óptima en cualquier contexto. En este contexto, Flexbox emerge como una herramienta poderosa y versátil para facilitar la creación de diseños responsivos de manera eficiente y efectiva.

El diseño responsivo se refiere a la práctica de crear interfaces web que se ajustan y se ven bien en una variedad de dispositivos, desde computadoras de escritorio hasta teléfonos móviles y tabletas. Esto se logra mediante el uso de técnicas como media queries y unidades relativas en CSS para adaptar el diseño y el contenido según el tamaño y la orientación de la pantalla del dispositivo del usuario.

Por otro lado, Flexbox es un modelo de diseño en CSS que facilita la creación de diseños flexibles y adaptables. Permite distribuir el espacio disponible entre los elementos de manera automática y proporciona un control preciso sobre la alineación, el orden y el tamaño de los elementos, lo que lo convierte en una herramienta ideal para implementar diseños responsivos de forma sencilla y eficiente.

En este módulo, exploraremos en detalle tanto los principios del diseño responsivo como las capacidades y técnicas de Flexbox para crear interfaces web adaptables y atractivas.

## Importancia del Diseño Responsivo en el desarrollo web

En la actualidad, la diversidad de dispositivos y pantallas utilizados para acceder a internet es asombrosa. Desde computadoras de escritorio con monitores de gran tamaño hasta teléfonos inteligentes con pantallas pequeñas, pasando por tabletas, televisores inteligentes y dispositivos portátiles, los usuarios acceden a la web desde una amplia variedad de plataformas y dispositivos. En este contexto, la importancia del diseño responsivo en el desarrollo web es fundamental por varias razones clave:

- **Experiencia del Usuario Consistente:** El diseño responsivo garantiza que los usuarios obtengan una experiencia consistente y óptima en cualquier dispositivo, independientemente de su tamaño de pantalla o resolución. Esto mejora la usabilidad y la accesibilidad del sitio web, lo que a su vez puede aumentar la retención de usuarios y la satisfacción del cliente.
- **Mejora del SEO:** Los motores de búsqueda, como Google, favorecen los sitios web que ofrecen una experiencia de usuario positiva en dispositivos móviles. El diseño responsivo es una práctica recomendada por Google y otros motores de búsqueda, lo que puede mejorar el posicionamiento en los resultados de búsqueda y aumentar la visibilidad del sitio.
- **Reducción del Costo y Complejidad del Desarrollo:** En lugar de crear múltiples versiones de un sitio web para diferentes dispositivos, el diseño responsivo permite desarrollar un único sitio web que se adapte de forma fluida a cualquier pantalla. Esto reduce el tiempo y los costos de desarrollo, así como la complejidad de mantener y actualizar múltiples versiones del sitio.
- **Adaptabilidad Futura:** Con el diseño responsivo, los sitios web están preparados para adaptarse a futuras innovaciones en dispositivos y tecnologías. A medida que surjan nuevos dispositivos con tamaños de pantalla diferentes, el sitio web seguirá siendo funcional y se verá bien sin necesidad de realizar cambios significativos en el código.
- **Alcance Global:** En un mundo cada vez más conectado, donde los usuarios acceden a internet desde todas partes del mundo y en una variedad de dispositivos, el diseño responsivo permite llegar a una audiencia más amplia y diversa, independientemente de su ubicación geográfica o del dispositivo que utilicen.

## Diseño Responsivo y Flexbox

Flexbox, o Flexible Box Layout, es un modelo de diseño en CSS que facilita la creación de diseños responsive al proporcionar un conjunto de propiedades que permiten controlar la distribución y el alineamiento de los elementos dentro de un contenedor de manera flexible y eficiente.

Algunas características clave de Flexbox que lo hacen especialmente útil para el diseño responsive son:

- **Distribución flexible del espacio:** Con Flexbox, los elementos pueden expandirse, contraerse y distribuirse automáticamente dentro de un contenedor en función del espacio disponible, lo que facilita la adaptación a diferentes tamaños de pantalla.
- **Alineación versátil:** Flexbox ofrece un control preciso sobre la alineación de los elementos tanto en el eje principal como en el eje secundario, lo que permite centrar, distribuir o alinear los elementos de manera uniforme, incluso cuando el tamaño de la pantalla cambia.
- **Ordenamiento dinámico:** Flexbox permite cambiar el orden visual de los elementos sin modificar el orden en el código HTML, lo que facilita la reorganización de la disposición de los elementos en dispositivos con pantallas más pequeñas o en diferentes contextos de visualización.
- **Facilidad de uso:** La sintaxis simple y declarativa de Flexbox lo hace fácil de entender y utilizar, incluso para principiantes en desarrollo web. Esto permite crear diseños responsive de manera más eficiente y con menos código.
- **Compatibilidad con navegadores:** Flexbox cuenta con un amplio soporte en los navegadores modernos y, gracias a su capacidad de degradación elegante, puede utilizarse de manera segura en proyectos que requieren compatibilidad con versiones antiguas de navegadores.

## Diseño Responsivo con Media Queries



Las media queries son una herramienta fundamental en el diseño responsivo, ya que permiten aplicar estilos específicos basados en las características del dispositivo en el que se está visualizando una página web, como el tamaño de la pantalla, la orientación, la densidad de píxeles, entre otros.

El papel de las media queries en el diseño responsivo es proporcionar una forma de adaptar el diseño y la apariencia de una página web a diferentes dispositivos y contextos de visualización. Esto se logra mediante la definición de reglas CSS condicionales que se aplican solo cuando se cumplen ciertas condiciones especificadas en la media query.

Por ejemplo, una media query puede establecer estilos diferentes para dispositivos con una anchura de pantalla menor a cierto tamaño, lo que permite ajustar el diseño para dispositivos móviles. Otra media query puede cambiar el diseño cuando el dispositivo se encuentra en modo apaisado en lugar de vertical, optimizando así la experiencia de usuario para diferentes orientaciones de pantalla.

## Conceptos básicos y sintaxis

Comprender los conceptos básicos y la sintaxis de las media queries es esencial para crear diseños adaptativos y mejorar la experiencia del usuario en diferentes dispositivos.

A continuación, se detallan los conceptos básicos y la sintaxis de las media queries:

### Conceptos básicos:

Las media queries en CSS3 permiten aplicar estilos de manera condicional según las características del medio en el que se visualiza la página web. Esto se logra a través de dos aspectos fundamentales: los tipos de medios y las características de medios.

- **Media Types (Tipos de Medios):**

Los media types definen los diferentes tipos de medios a los que se pueden aplicar las reglas de estilo.

Algunos tipos comunes de medios son:

- screen: Se refiere a pantallas de dispositivos, como monitores de computadora, tabletas y teléfonos inteligentes.
- print: Se utiliza para dispositivos de impresión, como impresoras.
- speech: Se aplica a dispositivos auditivos, como lectores de pantalla.

Estos tipos de medios permiten definir estilos específicos que se aplicarán solo cuando el contenido se muestre en el medio correspondiente.

- **Media Features (Características de Medios):**

Las características de medios son propiedades del dispositivo que se pueden utilizar para definir condiciones en las media queries.

Algunas características comunes incluyen:

- width: Representa el ancho de la ventana del navegador o la pantalla del dispositivo.
- height: Indica la altura de la ventana del navegador o la pantalla del dispositivo.

- orientation: Define la orientación del dispositivo (horizontal o vertical).
- resolution: Especifica la resolución de la pantalla en puntos por pulgada (DPI).
- aspect-ratio: Representa la relación entre el ancho y la altura de la pantalla.
- color: Indica la profundidad de color del dispositivo.

Estas características de medios se utilizan en las media queries para aplicar estilos específicos en función de las propiedades del dispositivo en el que se está visualizando el contenido. Por ejemplo, se pueden definir estilos diferentes para dispositivos con una resolución alta o baja, o para pantallas en orientación horizontal o vertical.

### Sintaxis básica:

Una media query se inicia con el prefijo @media seguido de una lista de condiciones de medios entre paréntesis y un conjunto de estilos CSS entre llaves.

Las condiciones de medios pueden incluir una o más características de medios y sus respectivos valores, separados por operadores lógicos como and, or y not.

Las condiciones de medios pueden ser simples, como max-width o min-width, o compuestas, como (min-width: 768px) and (orientation: landscape).



La sintaxis básica de una media query consta por tanto de dos partes principales: la regla @media y las condiciones de medios.

- Regla @media: Esta regla indica que se aplicarán ciertos estilos CSS solo si se cumplen ciertas condiciones de medios. La sintaxis es la siguiente:

```
@media tipo-de-medio y (condiciones) {
  /* Estilos CSS específicos */
}
```

1

- Condiciones de Medios: Las condiciones de medios definen las características del dispositivo que deben cumplirse para que se apliquen los estilos CSS dentro de la media query. Algunos ejemplos comunes de condiciones de medios incluyen width, height, orientation, min-width, max-width, entre otros.

```
@media screen and (min-width: 768px) and (max-width: 1024px) {
  /* Estilos aplicados cuando el ancho de la pantalla está entre 768px y 1024px */
}
```

2

<sup>1</sup> Accesibilidad

```
“media tipo-de-medio y (condiciones) {
  /* Estilos CSS específico:
}”
```

<sup>2</sup> Accesibilidad

```
@media screen and (min-width: 768px) and (max-width: 1024px) {
  /* Estilos aplicados cuando en ancho de la pantalla está entre 768px y 1024px
}”
```

## Ventajas de las Media Queries:

**Flexibilidad:** Permiten crear diseños adaptables que se ajustan automáticamente a diferentes dispositivos y tamaños de pantalla.

**Mejora la experiencia del usuario:** Garantizan una experiencia de usuario consistente y óptima en una amplia gama de dispositivos.

**Optimización para dispositivos móviles:** Facilitan la creación de sitios web optimizados para dispositivos móviles, lo que es crucial en la era actual de la navegación móvil.

## Flexbox: Conceptos básicos



Flexbox, o Flexible Box Layout, es un modelo de diseño en CSS que permite crear diseños flexibles y adaptables dentro de un contenedor.

### Elementos fundamentales

Los conceptos fundamentales de Flexbox incluyen los contenedores y elementos flexibles.

#### Contenedores Flexibles:

Un contenedor flexible, o flex container, es un elemento HTML que contiene uno o varios elementos hijos y se configura con la propiedad `display: flex;`.

Este contenedor establece un contexto de diseño flexible para sus elementos hijos.

#### Elementos Flexibles:

Los elementos flexibles, o flex items, son los hijos directos de un contenedor flexible. Estos elementos se distribuyen automáticamente dentro del contenedor flexible según las reglas de Flexbox.

### Conceptos clave

- **Flex Direction (Dirección Flexible):** Define la dirección en la que los elementos flexibles se distribuyen dentro del contenedor flexible. Puede ser `row` (horizontal), `column` (vertical), `row-reverse` o `column-reverse`.

- **Justify Content (Justificar Contenido):** Controla la alineación de los elementos flexibles a lo largo del eje principal del contenedor flexible. Se puede utilizar para alinear los elementos a la izquierda, derecha, centro, o distribuirlos uniformemente a lo largo del contenedor.
- **Align Items (Alinear Elementos):** Controla la alineación de los elementos flexibles a lo largo del eje transversal del contenedor flexible. Se puede utilizar para alinear los elementos en la parte superior, inferior, centro, o distribuirlos uniformemente a lo largo del contenedor.
- **Flex Wrap (Envoltura Flexible):** Especifica si los elementos flexibles deben envolverse o no cuando no haya suficiente espacio en el contenedor flexible. Puede ser nowrap (no envolver), wrap (envolver en varias líneas) o wrap-reverse.
- **Flex Grow, Flex Shrink y Flex Basis (Crecimiento, Encogimiento y Base Flexible):** Propiedades que controlan cómo se distribuye el espacio disponible entre los elementos flexibles dentro del contenedor flexible.

## Casos de uso

A continuación, se presentan algunos ejemplos prácticos de cómo utilizar Flexbox para crear diseños flexibles y adaptables:

- **Barra de navegación horizontal:** Utilizar Flexbox para alinear los elementos de una barra de navegación horizontal y distribuirlos uniformemente a lo largo del contenedor.
- **Cuadrícula de tarjetas:** Crear una cuadrícula de tarjetas flexibles que se ajusten automáticamente al tamaño del contenedor y se envuelvan en varias líneas si es necesario.
- **Diseño de formulario:** Utilizar Flexbox para alinear y distribuir los campos de un formulario de manera uniforme, con una disposición flexible que se adapte al tamaño de la pantalla.

## Flexbox: Propiedades y funcionalidades avanzadas



En Flexbox, además de los conceptos básicos como la alineación y distribución de elementos, existen varias propiedades y funcionalidades avanzadas que permiten crear diseños más complejos y sofisticados.

A continuación, se describen algunas de estas propiedades y se proporcionan ejemplos prácticos de su uso:

### Flex Grow, Flex Shrink y Flex Basis:

Estas propiedades avanzadas permiten controlar cómo se distribuye el espacio disponible entre los elementos flexibles dentro del contenedor flexible.

- **flex-grow:** Determina la proporción en la que los elementos flexibles crecerán para llenar el contenedor.
- **flex-shrink:** Controla la capacidad de los elementos flexibles para encogerse si es necesario para ajustarse al contenedor.
- **flex-basis:** Establece el tamaño base de los elementos flexibles antes de que se distribuya el espacio adicional o se encogen según sea necesario.

## Flex Align y Flex Justify:

Estas propiedades avanzadas permiten controlar la alineación y la distribución de los elementos flexibles dentro del contenedor flexible en ambos ejes, tanto principal como transversal.

- **align-items y justify-content:** Se utilizan para alinear y distribuir los elementos a lo largo del eje transversal y principal, respectivamente.
- **align-self y justify-self:** Permiten controlar la alineación de elementos individuales dentro del contenedor flexible, anulando las propiedades aplicadas a todos los elementos.

## Ordenamiento de elementos:

La propiedad order permite cambiar el orden visual de los elementos flexibles dentro del contenedor flexible, sin modificar el orden del marcado HTML. Esto es útil para reorganizar visualmente los elementos en diferentes dispositivos o en respuesta a cambios en el diseño.

## Ejemplos prácticos:

Con estos elementos avanzados es posible crear las siguientes funcionalidades:

- **Galería de imágenes Flexible:** Utilizar flex-grow para permitir que las imágenes crezcan y llenen el espacio disponible de manera proporcional, manteniendo su relación de aspecto original.
- **Menús de navegación dinámicos:** Emplear flex-align y flex-justify para alinear y distribuir los elementos del menú de navegación horizontal o vertical de manera uniforme, incluso cuando el tamaño del contenedor cambia.
- **Diseños de cuadrícula complejos:** Utilizar una combinación de flex-basis, flex-grow y order para crear diseños de cuadrícula flexibles y adaptables que se ajusten automáticamente al contenido y al tamaño de la pantalla.

## Casos de uso y ejemplos de implementación:

- **Diseños de página complejos:** Como páginas de inicio o páginas de producto en las que se requiere una disposición compleja de elementos.
- **Aplicaciones web interactivas:** Donde se necesita una disposición dinámica de componentes que se adapten a las acciones del usuario.

## Diseño responsivo con Flexbox y Media Queries



La combinación de Flexbox y media queries ofrece una poderosa herramienta para crear diseños web responsivos y adaptables a diferentes dispositivos y tamaños de pantalla.

A continuación, se detalla cómo integrar estas dos técnicas y se proporcionan ejemplos prácticos:

### Integración de Flexbox y Media Queries:

La mejor manera de combinar estos dos elementos es:

- **Flexbox para la estructura:** Utiliza Flexbox para definir la estructura y disposición de los elementos en tu diseño. Flexbox permite crear diseños flexibles y dinámicos que se ajustan automáticamente al contenido y al tamaño del contenedor.
- **Media Queries para la adaptabilidad:** Emplea media queries para establecer reglas de estilo condicionales basadas en las características del dispositivo, como el ancho de la pantalla o la orientación. Esto te permite modificar el diseño y los estilos según el tamaño de la pantalla del dispositivo en el que se visualiza la página.

### Ejemplos prácticos:

- **Navegación Flexbox Responsiva:** Utiliza Flexbox para crear un menú de navegación flexible y adaptable que se ajuste automáticamente al tamaño de la pantalla. Emplea media queries para cambiar la



disposición del menú entre una barra de navegación horizontal y un menú desplegable en dispositivos móviles.

- **Diseño de cuadrícula Flexible:** Utiliza Flexbox para crear una cuadrícula de elementos flexibles que se adapten dinámicamente al espacio disponible en la pantalla. Usa media queries para ajustar el número de columnas o el tamaño de los elementos en función del ancho de la pantalla.

### Casos de uso y ejemplos de implementación:

- **Páginas de producto:** Diseña páginas de productos con diseños flexibles que muestren la información de manera óptima en dispositivos de diferentes tamaños, desde pantallas de escritorio hasta dispositivos móviles.
- **Blogs y sitios de noticias:** Crea diseños responsivos que se adapten al contenido variable, como artículos de diferentes longitudes y tipos de medios, garantizando una experiencia de lectura óptima en cualquier dispositivo.

La combinación de Flexbox y media queries permite crear diseños web adaptables y flexibles que ofrecen una experiencia de usuario consistente y optimizada en una amplia variedad de dispositivos y tamaños de pantalla. Experimenta con estas técnicas y ajusta tu diseño según las necesidades específicas de tu proyecto y tu audiencia.

## Mejores prácticas y consideraciones



El diseño responsivo con Flexbox ofrece una forma eficaz de crear interfaces web adaptables y flexibles. Aquí se presentan algunas mejores prácticas y consideraciones para optimizar tu diseño responsivo:

- **Planificación y diseño modular:**
  - Estructura Clara y Modular: Diseña tu sitio web con una estructura clara y modular, dividiendo el contenido en componentes reutilizables. Esto facilita la adaptación del diseño a diferentes dispositivos y tamaños de pantalla.
  - Wireframes y Prototipos: Antes de comenzar a codificar, crea wireframes y prototipos para visualizar cómo se verá tu diseño en diferentes dispositivos. Esto te ayudará a identificar posibles problemas y optimizar la experiencia del usuario.
- **Uso efectivo de Flexbox:**
  - Evitar Anidamiento Excesivo: Aunque Flexbox permite anidar contenedores flexibles, evita el anidamiento excesivo, ya que puede complicar el diseño y afectar el rendimiento.
  - Alineación y Distribución: Utiliza las propiedades de Flexbox como justify-content y align-items para alinear y distribuir elementos de manera uniforme en el diseño, manteniendo una apariencia ordenada y equilibrada.
- **Optimización de imágenes y contenido multimedia:**
  - Imágenes Responsivas: Utiliza imágenes responsivas con la etiqueta <img> y el atributo srcset para cargar diferentes

versiones de una imagen según el tamaño de la pantalla del dispositivo.

- Contenido Multimedia Adaptativo: Asegúrate de que el contenido multimedia, como videos incrustados, se adapte correctamente a diferentes tamaños de pantalla sin comprometer la experiencia del usuario.
- **Accesibilidad y usabilidad:**
  - Pruebas de Accesibilidad: Realiza pruebas de accesibilidad para garantizar que tu diseño sea accesible para todos los usuarios, incluidos aquellos con discapacidades visuales o de movilidad.
  - Tamaño de Táctil: Asegúrate de que los elementos interactivos, como botones y enlaces, tengan un tamaño suficiente para ser seleccionados fácilmente en dispositivos táctiles.
- **Rendimiento y carga rápida:**
  - Optimización del Código: Optimiza tu código CSS y JavaScript para reducir el tiempo de carga de la página, lo que mejora la experiencia del usuario y el posicionamiento en los motores de búsqueda.
  - Compresión de Imágenes: Utiliza herramientas de compresión de imágenes para reducir el tamaño de los archivos de imagen sin comprometer la calidad visual.
- **Pruebas en diferentes dispositivos:**
  - Pruebas Cruzadas: Realiza pruebas exhaustivas en una variedad de dispositivos y navegadores para asegurarte de que tu diseño se vea y funcione correctamente en todos ellos.
  - Herramientas de Depuración: Utiliza herramientas de desarrollo del navegador para identificar y corregir posibles problemas de diseño y rendimiento.

*Aplicar estas mejores prácticas y consideraciones te ayudará a crear diseños responsivos efectivos y accesibles que ofrezcan una experiencia óptima para los usuarios en cualquier dispositivo y pantalla. **¡Experimenta y adapta estas técnicas según las necesidades específicas de tu proyecto!***

## Conclusiones

En este módulo hemos explorado los fundamentos del diseño responsivo y Flexbox, dos técnicas fundamentales en el desarrollo web moderno. Aquí hay una recapitulación de los conceptos clave aprendidos:

- **Importancia del diseño responsivo:** Aprendimos sobre la importancia del diseño responsivo en un mundo donde los usuarios acceden a sitios web desde una amplia variedad de dispositivos y tamaños de pantalla. El diseño responsivo garantiza una experiencia de usuario consistente y optimizada en todos los dispositivos.
- **Flexbox como herramienta de diseño:** Exploramos Flexbox, una poderosa herramienta de diseño en CSS que permite crear diseños flexibles y adaptables de manera eficiente. Aprendimos sobre los conceptos fundamentales de Flexbox, como contenedores flexibles y elementos flexibles, y cómo utilizar sus propiedades para controlar la distribución y alineación de elementos en un diseño.
- **Integración de Media Queries:** Descubrimos cómo utilizar media queries para crear diseños responsivos que se adapten dinámicamente a diferentes dispositivos y tamaños de pantalla. Aprendimos sobre las condiciones de medios y características de medios, así como las mejores prácticas para definir reglas de medios efectivas.

*En resumen, el diseño responsivo y Flexbox son herramientas esenciales para crear sitios web modernos y adaptables. Al animar a los estudiantes a practicar y experimentar con estos conceptos en sus propios proyectos web, les proporcionamos las habilidades necesarias para diseñar interfaces web efectivas y atractivas que ofrezcan una experiencia de usuario excepcional en cualquier dispositivo. ¡Atrévete a explorar y desarrollar tu creatividad con diseños responsivos y Flexbox!*

## Despedida y Recursos Adicionales

**¡Enhorabuena por completar este módulo sobre diseño responsivo y Flexbox!** Espero que hayas encontrado útiles los conceptos y técnicas que hemos explorado aquí. Recuerda que el diseño web es un campo emocionante y en constante evolución, y dominar estas habilidades te ayudará a destacar como desarrollador frontend.

### Recursos Adicionales

Para seguir profundizando en el diseño responsivo y Flexbox, te recomiendo explorar los siguientes recursos:

- Tutoriales en línea: Busca tutoriales interactivos y cursos en plataformas como Codecademy, freeCodeCamp o Udemy. Estos recursos pueden ofrecerte una experiencia práctica y guiada para mejorar tus habilidades.
- Documentación Oficial: Consulta la documentación oficial de CSS en MDN Web Docs y la especificación de Flexbox en el sitio web de W3C. Estas fuentes proporcionan información detallada y precisa sobre las propiedades y características de CSS.
- Herramientas de Diseño Responsivo: Explora herramientas como BrowserStack, Responsinator o Am I Responsive para probar y depurar tus diseños en una variedad de dispositivos y tamaños de pantalla.

*Recuerda que la práctica constante es clave para mejorar en el diseño web. **¡Sigue explorando, experimentando y desarrollando tus habilidades para convertirte en un experto en diseño responsivo y Flexbox!***