

PROGRAMA DE ESTUDIOS

**DESARROLLO DE
SISTEMAS DE
INFORMACIÓN
FRONTEND ORIENTADO
A SERVICIOS**

Tema

DISEÑO RESPONSIVO PARTE II

DISEÑO RESPONSIVO PARTE II

1. Grid y Flexbox: Herramientas para el Diseño de Páginas Fluidas

CSS Grid y Flexbox son sistemas de diseño en CSS que facilitan la creación de layouts flexibles y adaptables. Ambos permiten organizar el contenido en filas y columnas, adaptándose a diferentes tamaños de pantalla y ofreciendo control sobre el espacio entre los elementos.

1.1 CSS Grid

Concepto: CSS Grid es un sistema bidimensional que permite colocar elementos en filas y columnas. Es ideal para estructuras más complejas, ya que facilita la creación de layouts con múltiples áreas de contenido organizadas en cuadrículas.

Características:

- **Bidimensional:** Permite organizar elementos tanto en filas como en columnas.
- **Control preciso:** Se puede definir el tamaño de las filas y columnas y distribuir el espacio entre los elementos.
- **Propiedades clave:** display: grid, grid-template-columns, grid-template-rows, gap, entre otras.

Ejemplo básico:

```
.container {  
    display: grid;  
    grid-template-columns: repeat(3, 1fr);  
    /* Tres columnas de igual tamaño */  
    gap: 10px; /* Espacio entre elementos */  
}
```

HTML:

```
html  
  
<div class="container">  
    <div>Item 1</div>  
    <div>Item 2</div>  
    <div>Item 3</div>  
</div>
```

1.2 Flexbox

Concepto: Flexbox es un sistema de diseño unidimensional, ideal para alinear y distribuir elementos en una dirección (horizontal o vertical). Es útil para elementos que necesitan ajustarse dinámicamente al espacio disponible, como menús o secciones en una página.

Características:

- **Unidimensional:** Opera en una sola dirección (filas o columnas).
- **Flexible:** Los elementos pueden crecer o reducirse para llenar el espacio disponible.

- **Propiedades clave:** display: flex, justify-content, align-items, flex-direction, entre otras.

Ejemplo básico:

```
.container {  
    display: flex;  
    justify-content: space-between;  
    /* Espacio entre los elementos */  
}
```

HTML:

```
html  
  
<div class="container">  
    <div>Elemento 1</div>  
    <div>Elemento 2</div>  
    <div>Elemento 3</div>  
</div>
```

2. Uso Práctico de Flexbox para Alinear y Distribuir Elementos

Flexbox permite organizar los elementos en filas o columnas y ajustar su posición y tamaño dentro del contenedor. A continuación se explican las propiedades clave para alinear y distribuir elementos.

Propiedades Clave de Flexbox para Alineación y Distribución

- **justify-content:** Define cómo se distribuyen los elementos a lo largo del eje principal.
 - Valores comunes: flex-start, center, space-between, space-around.

Ejemplo:

```
.container {  
    display: flex;  
    justify-content: space-around;  
    /* Espacio alrededor de los elementos */  
}
```

- **align-items:** Controla la alineación de los elementos a lo largo del eje perpendicular (vertical si la dirección es row).
- Valores comunes: flex-start, center, flex-end, stretch.

Ejemplo:

```
.container {  
    display: flex;  
    align-items: center;  
    /* Centra los elementos verticalmente */  
}
```

- **flex-direction:** Define la dirección de los elementos dentro del contenedor.
- Valores comunes: row, column, row-reverse, column-reverse.

Ejemplo:

```
.container {  
    display: flex;  
    flex-direction: column;  
    /* Organiza los elementos en una columna */  
}
```

3. Ejercicios Prácticos

Ejercicio 1: Crear un Menú de Navegación con Flexbox

1. Crea un contenedor con varios elementos <div> que simulen los enlaces de un menú de navegación.
2. Usa display: flex y justify-content: space-around para distribuir el espacio uniformemente entre los enlaces.

Ejercicio 2: Crear una Galería de Imágenes con Grid

1. Diseña una galería con múltiples imágenes utilizando CSS Grid.
2. Define tres columnas en pantallas grandes y reorganízalo a una columna en pantallas pequeñas.

