

PROGRAMA DE ESTUDIOS

DESARROLLO DE SISTEMAS DE INFORMACIÓN FRONTEND ORIENTADO A SERVICIOS

Tema

**SISTEMA DE CUADRICULA CON CSS
GRID PARTE II**

SISTEMA DE CUADRICULA CON CSS GRID PARTE II

CSS Grid es una herramienta poderosa para crear diseños de cuadrícula avanzados, y en esta segunda parte vamos a profundizar en algunas propiedades que permiten un control más detallado sobre el diseño de páginas web.

1. Uso Avanzado de CSS Grid

a. Propiedad grid-template

- **Concepto:** La propiedad grid-template es una abreviatura que permite definir tanto grid-template-rows como grid-template-columns en una sola línea, lo que permite estructurar rápidamente la cuadrícula.
- **Características:**
 - Define el número de filas y columnas y sus tamaños en la cuadrícula.
 - Facilita la creación de plantillas de diseño complejas especificando patrones de tamaño.

Ejemplo:

CSS

```
.contenedor {  
    display: grid;  
    grid-template: 100px 200px / 1fr 2fr 1fr;  
}
```

Este código crea una cuadrícula con dos filas (100px y 200px) y tres columnas (1fr, 2fr y 1fr).

b. Propiedad grid-gap (o gap)

- **Concepto:** Define el espacio entre filas y columnas en la cuadrícula.
- **Características:**
 - Se puede usar como gap, row-gap (espacio entre filas) o column-gap (espacio entre columnas).
 - Facilita el control del espaciado entre los elementos de la cuadrícula.

Ejemplo:

CSS

```
.contenedor {  
    display: grid;  
    grid-template-columns: repeat(3, 1fr);  
    gap: 20px;  
}
```

En este ejemplo, el espacio entre cada elemento en la cuadrícula será de 20px.

c. Unidad fr

- **Concepto:** fr (fracción) es una unidad de medida específica de CSS Grid que asigna una proporción del espacio disponible a una columna o fila.
- **Características:**
 - Es dinámica y ajusta el tamaño de los elementos en función del espacio disponible.

- Ideal para crear diseños responsivos.

Ejemplo:

CSS

```
.contenedor {  
    display: grid;  
    grid-template-columns: 1fr 2fr 1fr;  
}
```

En este ejemplo, la segunda columna será el doble del tamaño de la primera y la tercera.

2. Plantillas de Diseño Complejas

CSS Grid permite crear diseños avanzados al combinar propiedades como grid-template-areas para definir áreas específicas que facilitan la disposición de elementos en patrones complejos.

Propiedad grid-template-areas

- **Concepto:** Define áreas dentro de la cuadrícula mediante nombres que representan diferentes secciones del diseño.
- **Características:**
 - Permite asignar nombres a partes de la cuadrícula, creando un diseño más intuitivo.
 - Cada área puede abarcar varias celdas en filas y columnas.

Ejemplo:

CSS

```
.contenedor {  
    display: grid;  
    grid-template-areas:  
        "header header header"  
        "sidebar content content"  
        "footer footer footer";  
    grid-template-rows: 100px 1fr 50px;  
    grid-template-columns: 200px 1fr 1fr;  
}  
  
.header { grid-area: header; }  
.sidebar { grid-area: sidebar; }  
.content { grid-area: content; }  
.footer { grid-area: footer; }
```

En este diseño, el área de header ocupa toda la primera fila, sidebar ocupa la primera columna, content ocupa el resto de las columnas en la segunda fila, y footer ocupa toda la tercera fila.

Propiedad grid-auto-flow

- **Concepto:** Controla cómo los elementos se colocan automáticamente en la cuadrícula.

• Características:

- Determina si los elementos se colocan en filas o columnas automáticas.
- Puede tomar los valores row, column, dense, o combinaciones de ellos.

Ejemplo:

CSS

```
.contenedor {  
    display: grid;  
    grid-template-columns: repeat(3, 1fr);  
    grid-auto-flow: column;  
}
```

Aquí, los elementos se colocarán automáticamente en columnas, llenando cada una antes de pasar a la siguiente.

Ejercicios Prácticos**Ejercicio 1: Crear un Layout de Blog con CSS Grid**

- **Objetivo:** Crear una cuadrícula con encabezado, barra lateral, contenido principal y pie de página.
- **Instrucciones:**
 - Define un contenedor grid-container y asigna áreas para el encabezado, barra lateral, contenido y pie de página.
 - Utiliza grid-template-areas para estructurar el diseño.
 - Asegúrate de usar gap para espaciar los elementos.

Ejercicio 2: Galería de Imágenes Responsiva con fr y gap

- **Objetivo:** Crear una galería de imágenes que se ajuste automáticamente al tamaño del contenedor.
- **Instrucciones:**

1. Crea un contenedor gallery y usa grid-template-columns con unidades fr para que se ajuste automáticamente.
2. Usa gap para añadir espacio entre las imágenes.
3. Configura el diseño para que tenga 3 columnas en pantallas grandes y se reduzca a 1 columna en pantallas pequeñas.



