# Módulo 4: Introducción a CSS Grid (10:45 – 11:30)

## 🎯 Objetivos del módulo

* Comprender el concepto y la necesidad de CSS Grid.
* Diferenciar entre **layouts unidimensionales** (Flexbox) y **bidimensionales** (Grid).
* Aprender a definir filas, columnas y espacios.
* Aplicar reglas básicas para maquetar contenido visualmente con precisión.

## 🧠 ¿Qué es CSS Grid?

**CSS Grid Layout** es un sistema de maquetado CSS diseñado para **distribuir elementos en filas y columnas**, con total control sobre su ubicación.

Mientras que **Flexbox** trabaja mejor para distribuir ítems **en una única dirección (horizontal o vertical)**, **Grid** funciona en **ambas dimensiones a la vez**.

## 🔳 Conceptos clave

### 📦 Grid Container (contenedor de grid)

Para empezar a usar Grid, hay que declarar en el contenedor:

css

CopiarEditar

display: grid;

A partir de ahí, sus hijos serán tratados como **celdas de una grilla**.

### 📐 Definiendo columnas y filas

css

CopiarEditar

.grid-container {

display: grid;

grid-template-columns: 1fr 1fr 1fr; /\* 3 columnas de igual tamaño \*/

grid-template-rows: auto auto; /\* 2 filas automáticas \*/

gap: 10px; /\* espacio entre columnas y filas \*/

}

* 1fr: una fracción del espacio disponible.
* auto: adapta la fila o columna al contenido.
* gap: reemplaza row-gap y column-gap.

## 🧱 Elementos del Grid

Cada hijo directo del contenedor se convierte en una **celda del grid**.

html

CopiarEditar

<div class="grid-container">

<div class="grid-item">1</div>

<div class="grid-item">2</div>

<div class="grid-item">3</div>

<div class="grid-item">4</div>

<div class="grid-item">5</div>

<div class="grid-item">6</div>

</div>

### Estilo para los ítems

css

CopiarEditar

.grid-item {

background-color: lightblue;

padding: 20px;

text-align: center;

border: 1px solid #333;

}

## 🧲 Posicionamiento de ítems

Los elementos hijos se pueden ubicar de forma precisa usando:

css

CopiarEditar

grid-column: 1 / 3; /\* Desde la columna 1 hasta antes de la 3 \*/

grid-row: 2 / 4; /\* Desde la fila 2 hasta antes de la 4 \*/

También se puede usar span para ocupar varias celdas:

css

CopiarEditar

grid-column: span 2; /\* Ocupa 2 columnas \*/

## 🧭 ¿Qué se puede lograr con Grid?

✔ Galerías de imágenes  
✔ Tablas de precios  
✔ Dashboards (paneles de administración)  
✔ Secciones de contenido responsive

## 💡 ¿Flexbox o Grid?

| Característica | Flexbox | Grid |
| --- | --- | --- |
| Dirección | Unidimensional | Bidimensional |
| Flujo | Ideal para listas o menús | Ideal para cuadrículas |
| Control de espacio | Reparto automático | Ubicación precisa |
| Alineación | Eje principal o transversal | Filas y columnas |

👉 **Usalos juntos**. Por ejemplo: un grid para la estructura general del sitio y Flexbox dentro de cada sección.

## 🧪 Ejemplo práctico

### HTML

html

CopiarEditar

<div class="grid-container">

<div class="grid-item">1</div>

<div class="grid-item">2</div>

<div class="grid-item">3</div>

<div class="grid-item">4</div>

<div class="grid-item">5</div>

<div class="grid-item">6</div>

</div>

### CSS

css

CopiarEditar

.grid-container {

display: grid;

grid-template-columns: repeat(3, 1fr);

gap: 10px;

}

.grid-item {

background-color: salmon;

padding: 20px;

text-align: center;

}

🖼 Resultado: 3 columnas con 2 filas automáticas, 6 cajas organizadas y con espacio entre ellas.