

# CSS - Grid

**Carrera  
Programador  
full-stack**

# ¿Qué es Grid?

**CSS Grid es un sistema de maquetación web que divide la página en una cuadrícula o rejilla (*grid*) a partir de la cual se pueden posicionar los diferentes elementos de manera más sencilla, versátil y coherente.**

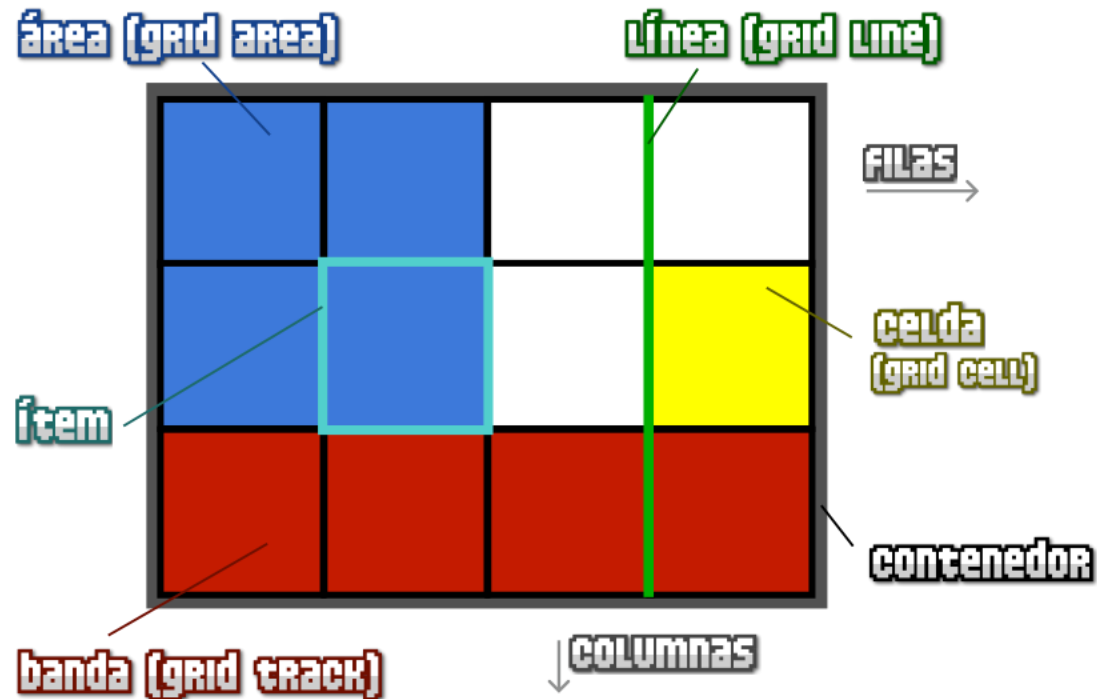
**Su practicidad y sus múltiples ventajas lo han convertido en un estándar.** Es decir, casi cualquier navegador soporta e interpreta este tipo de código.

# Ventajas

- **Mucha más flexibilidad:** te permite controlar los elementos en las 2 dimensiones y con total libertad.
- **Menos código y menos bugs:** se reduce considerablemente el código empleado, lo que conlleva revisiones más sencillas y menor probabilidad de bugs.
- **Optimización de recursos:** al ser código más simple y consistente se economiza tanto en tiempo como en recursos necesarios para mostrar la página.
- **Responsive más sencillo:** permite crear elementos dinámicos que se adaptan a diferentes tamaños o resoluciones sin complicaciones.

# Conceptos básicos

Para crear diseños basados en **Grid CSS** necesitaremos tener en cuenta una serie de conceptos que utilizaremos a partir de ahora y que definiremos a continuación:



# Conceptos básicos

- **Contenedor:** El elemento padre contenedor que definirá la cuadrícula o rejilla.
- **Ítem:** Cada uno de los hijos que contiene la cuadrícula (elemento contenedor).
- **Celda (grid cell):** Cada uno de los cuadritos (unidad mínima) de la cuadrícula.
- **Area (grid area):** Región o conjunto de celdas de la cuadrícula.
- **Banda (grid track):** Banda horizontal o vertical de celdas de la cuadrícula.
- **Línea (grid line):** Separador horizontal o vertical de las celdas de la cuadrícula.

Para activar la cuadrícula **grid** hay que utilizar sobre el elemento contenedor la propiedad `display` y especificar uno de los dos valores que queramos utilizar: **grid**

# Definir filas y columnas

En **Grid CSS**, la forma principal de definir una cuadrícula es indicar el tamaño de sus filas y sus columnas de forma explícita. Para ello, sólo tenemos que usar las propiedades CSS “grid-template-column” y “grid-template-row”

Propiedad	Valor	Descripción
grid-template-columns	[col1] [col2] ...	Establece el <b>SIZE</b> de cada columna (col 1, col 2...).
grid-template-rows	[fila1] [fila2] ...	Establece el <b>SIZE</b> de cada fila (fila 1, fila 2...).

```
.grid {  
  display: grid;  
  grid-template-columns: 50px 300px;  
  grid-template-rows: 200px 75px;  
}
```

# Unidad fracción restante (fr)

En el ejemplo anterior he utilizado **píxels** como unidades de las celdas de la cuadrícula, sin embargo, también podemos utilizar otras unidades (o incluso combinarlas): porcentajes, la palabra clave **auto** (que obtiene el tamaño restante) o la unidad especial de grid **fr** (fracción restante), que explicaremos a continuación.

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

# Grid Areas

Hasta ahora hemos visto como definir cuadrículas **Grid CSS** donde definimos sus filas y sus columnas.

Sin embargo, no es la única forma de definir cuadrículas en CSS. Si necesitamos un poco más de flexibilidad a la hora de definir un grid, podemos utilizar una funcionalidad denominada **Grid por áreas**, que permite de forma casi gráfica y visual, definir mediante texto la ubicación y forma que van a tener las celdas de nuestra cuadrícula.

Propiedad	Descripción
<code>grid-template-areas</code>	Indica la disposición de las áreas en el grid. Cada texto entre comillas simboliza una fila.
<code>grid-area</code>	Indica el nombre del área. Se usa sobre ítems hijos del grid.



# Ejemplo

## CSS

```
.container {  
  display: grid;  
  grid-template-areas: "head head"  
                       "menu main"  
                       "foot foot";  
  
  grid-template-columns: 1fr 1fr;  
  grid-template-rows: 100px 200px 100px;  
}
```

```
.item-1 { grid-area: head; background: blue; }  
.item-2 { grid-area: menu; background: red; }  
.item-3 { grid-area: main; background: green; }  
.item-4 { grid-area: foot; background: orange; }
```

## HTML

```
<div class="container">  
  <div class="item item-1"></div>  
  <div class="item item-2"></div>  
  <div class="item item-3"></div>  
  <div class="item item-4"></div>  
</div>
```