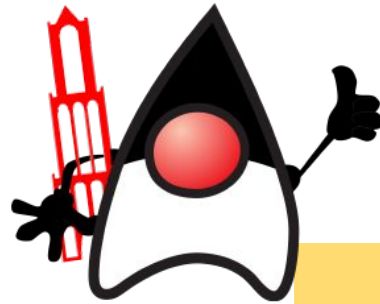




# Curso FullStack Python

Codo a Codo 4.0



# CSS

## *Parte 6*

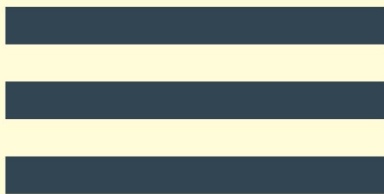
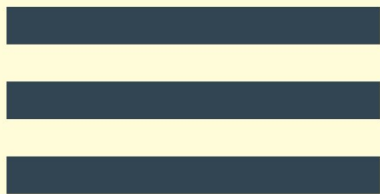
**CSS**



**CSS**



**GRID**

**DISEÑO WEB**



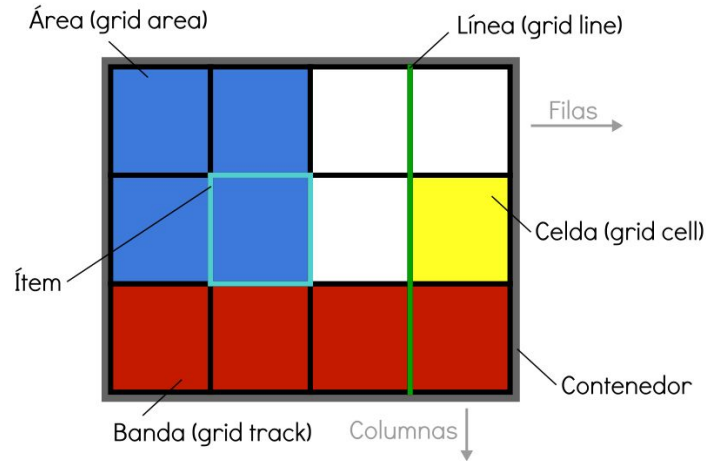


# CSS Grid | Introducción

- CSS Grid es una muy buena opción para **estructurar, organizar** y **diseñar** los elementos de nuestro sitio web, ofreciéndonos un sistema de disposición apropiado de forma nativa en el navegador y a nuestra total disposición.
  - En CSS solemos utilizar múltiples propiedades para estructurar nuestros sitio web, utilizando inline-block, floats, relative o hasta el mismo flexbox que solo utiliza una dimensión.
  - El módulo CSS Grid entra en acción ofreciéndonos un sistema de filas y columnas en **dos dimensiones**, y total libertad de los ítems para organizar nuestros contenido a gusto.
  - ¿Y donde queda el **Responsive Design**? Tampoco es un problema con CSS Grid ya que podemos definir filas y columnas como también redefinirlas a nuestro gusto dentro de un media query.
- 
- 

# CSS Grid

El módulo de diseño de CSS Grid ofrece un sistema de diseño basado en cuadrículas, con filas y columnas, lo que facilita el diseño de páginas web **sin tener que usar flotadores y posicionamiento**.





**Para ampliar:**

[https://www.w3schools.com/css/css\\_grid.asp](https://www.w3schools.com/css/css_grid.asp)



# CSS Grid | Conceptos básicos

- **Grid Container:** es nuestro elemento “Padre”, donde se asigna un `{display:grid;}` y nos permitirá colocar otras propiedades para manipular nuestro diseño.
  - **Grid Item:** son los hijos directos de nuestro container. Estos los manejaremos a nuestra voluntad, nuestras filas y columnas que moveremos a nuestro gusto.
  - **Grid Line:** son las líneas divisorias horizontales y verticales.
  - **Grid Track:** es el espacio entre dos líneas adyacentes. Filas y columnas.
  - **Grid Cell:** nuestras celdas serán el espacio entre dos filas adyacentes y 2 columnas adyacentes.
  - **Grid Area:** espacio rodeado por 4 grid lines.
- 
- 

# CSS Grid | Contenedor



```
<div>
  <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 class="grid-item">7</div>
  <div class="grid-item">8</div>
  <div class="grid-item">9</div>
</div>
```

Elemento padre que contiene uno o más elementos.

1	2	3
4	5	6
7	8	9

# CSS Grid | Display

Un elemento HTML se transforma en un contenedor de grilla cuando tiene su propiedad **display** seteada en **grid** o **inline-grid**

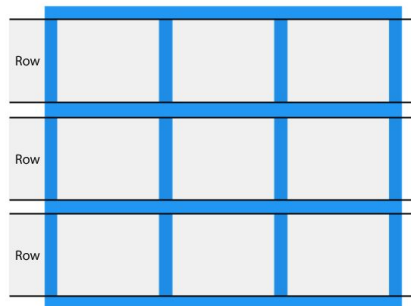
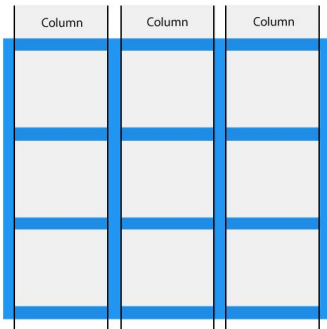
```
.grid-container{  
  display: grid;  
}
```

```
.grid-container{  
  display: inline-grid;  
}
```

**Ejemplos:**  
display: grid  
display: inline-grid

# CSS Grid | Grid Items

Se pueden referenciar por fila o por columna, aunque no es la única forma.

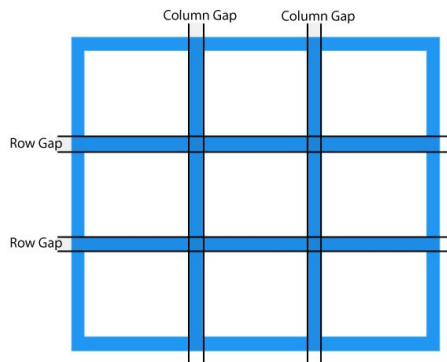




# CSS Grid | Grid Gaps

Es el espacio entre los ítems.

Se pueden ajustar los tamaños de gap con las siguientes propiedades: `grid-column-gap`; `grid-row-gap`; `grid-gap`



**Ejemplos:**

`grid-column-gap`

`grid-row-gap`

`grid-gap`

# CSS Grid | Grid Lines

Hay que referirse a los números de línea para colocar un grid-ítem en un contenedor:

Poner un grid item en la línea de columna 1 y dejarlo finalizar en la línea de columna 3:

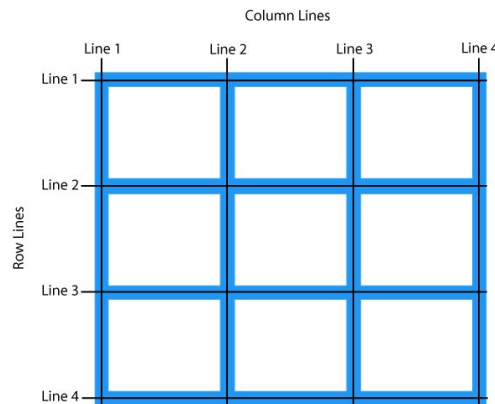
```
.item{  
  grid-column-start: 1;  
  grid-column-end: 3;  
}
```

Ejemplo

Poner un grid item en la línea de fila 1 y dejarlo finalizar en la línea de fila 3:

```
.item{  
  grid-row-start: 1;  
  grid-row-end: 3;  
}
```

Ejemplo



# CSS Grid | Propiedades

Propiedad	Descripción
<u>column-gap</u>	Especifica el espacio entre las columnas.
<u>gap</u>	Una propiedad abreviada para las propiedades de espacio entre filas y espacio entre columnas.
<u>grid</u>	Una propiedad abreviada para las filas de plantilla de cuadrícula, columnas de plantilla de cuadrícula, áreas de plantilla de cuadrícula, filas automáticas de cuadrícula, columnas automáticas de cuadrícula y las propiedades de flujo automático de cuadrícula.
<u>grid-area</u>	Especifica un nombre para el elemento de la cuadrícula o esta propiedad es una propiedad abreviada para las propiedades grid-row-start, grid-column-start, grid-row-end y grid-column-end.
<u>grid-auto-columns</u>	Especifica un tamaño de columna predeterminado.
<u>grid-auto-flow</u>	Especifica cómo se insertan los elementos colocados automáticamente en la cuadrícula.
<u>grid-auto-rows</u>	Especifica un tamaño de fila predeterminado.

# CSS Grid | Propiedades (continuación)

Propiedad	Descripción
<u>grid-column</u>	Una propiedad abreviada para las propiedades grid-column-start y grid-column-end.
<u>grid-column-end</u>	Especifica dónde terminar el elemento de la cuadrícula.
<u>grid-column-gap</u>	Especifica el tamaño del espacio entre columnas.
<u>grid-column-start</u>	Especifica dónde comenzar el elemento de la cuadrícula.
<u>grid-gap</u>	Una propiedad abreviada para las propiedades grid-row-gap y grid-column-gap.
<u>grid-row</u>	Una propiedad abreviada para las propiedades grid-row-start y grid-row-end.
<u>grid-row-end</u>	Especifica dónde terminar el elemento de la cuadrícula.
<u>grid-row-gap</u>	Especifica el tamaño del espacio entre filas.

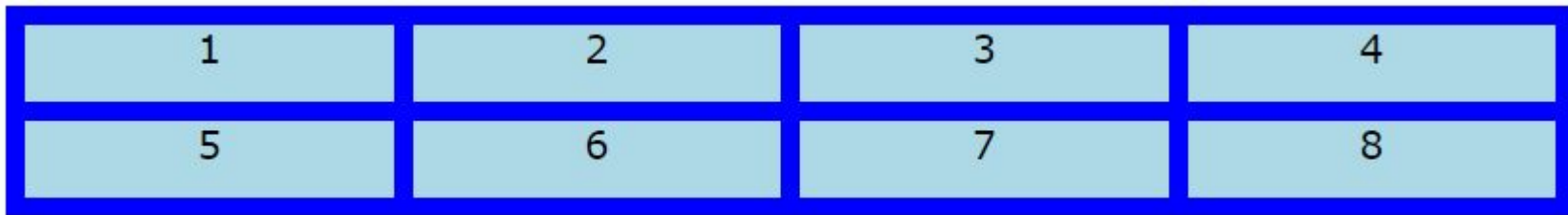
# CSS Grid | Propiedades (continuación)

Propiedad	Descripción
<u>grid-row-start</u>	Especifica dónde comenzar el elemento de la cuadrícula.
<u>grid-template</u>	Una propiedad abreviada para las propiedades de las filas de plantilla de cuadrícula, columnas de plantilla de cuadrícula y áreas de cuadrícula.
<u>grid-template-areas</u>	Especifica cómo mostrar columnas y filas, utilizando elementos de cuadrícula con nombre.
<u>grid-template-columns</u>	Especifica el tamaño de las columnas y cuántas columnas en un diseño de cuadrícula.
<u>grid-template-rows</u>	Especifica el tamaño de las filas en un diseño de cuadrícula.
<u>row-gap</u>	Especifica el espacio entre las filas de la cuadrícula.

**Fuente:** [https://www.w3schools.com/css/css\\_grid.asp](https://www.w3schools.com/css/css_grid.asp)

# CSS Grid | Grid Container

Para que un elemento HTML se comporte como un contenedor de cuadrícula, debemos establecer la propiedad **display** en **grid** (cuadrícula) o **inline-grid** (cuadrícula en línea). Los contenedores de cuadrícula consisten en elementos de cuadrícula, colocados dentro de columnas y filas.



```
<div class="grid-container">
  <div>1</div>
  <div>2</div>
  <div>3</div>
  <div>4</div>
  <div>5</div>
  <div>6</div>
  <div>7</div>
  <div>8</div>
</div>
```

```
.grid-container {
  width: 800px;
  display: grid;
  grid-template-columns: repeat(4, auto);
  grid-gap: 10px;
  background-color: blue;
  padding: 10px;
}
```

```
.grid-container > div {
  background-color: lightblue;
  text-align: center;
  font-size: 20px;
  height: 40px;
}
```

# CSS Grid | grid-template-columns

La propiedad **grid-template-columns** define el número de columnas en su diseño de cuadrícula y puede definir el ancho de cada columna.

El valor es una lista separada por espacios, donde cada valor define el ancho de la columna respectiva.

Si desea que su diseño de cuadrícula contenga 4 columnas, especifique el ancho de las 4 columnas, o "auto" si todas las columnas deben tener el mismo ancho.

Se pueden establecer anchos en px, unidades relativas o %, aunque es recomendable utilizar la medida **fr**.

## Medida en px:

1	2	3	
4	5	6	

```
grid-template-columns: 300px 200px 400px;
```

# CSS Grid | grid-template-columns

Medida en em: `grid-template-columns: 40em 20em 30em; /*1em=16px*/`

1	2	3
4	5	6

Medida en %: `grid-template-columns: 50% 35% 15%;`

1	2	3
4	5	6

Medida automática: `grid-template-columns: auto auto auto;`

1	2	3
4	5	6

Si tiene más de 4 elementos en una cuadrícula de 4 columnas, la cuadrícula agregará automáticamente una nueva fila para colocar los elementos.

# CSS Grid | grid-template-columns

## Unidad fracción restante (fr)

La unidad especial de Grid CSS **fr** (*fraction*) simboliza una fracción de espacio restante en el grid.

1	2	3
4	5	6

```
grid-template-columns: 0.5fr 2fr 1fr;
```

## La expresión repeat():

Se puede utilizar la expresión **repeat()** para indicar repetición de valores, indicando el número de veces que se repiten y el tamaño en cuestión.

La expresión a utilizar sería la siguiente: **repeat([número de veces], [valor o valores]):**

1	2	3
4	5	6

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



# CSS Grid | grid-template-rows

La propiedad **grid-template-rows** define la altura de cada fila.

El valor es una lista separada por espacios, donde cada valor define el alto de la fila respectiva.

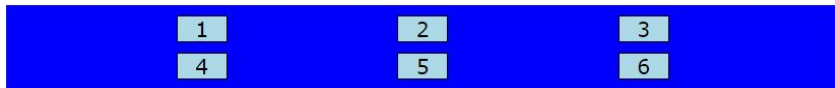
1	2	3
4	5	6

```
grid-template-rows: 80px 200px;
```

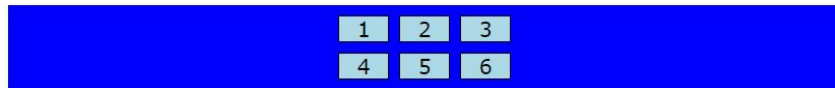
# CSS Grid | Propiedad justify-content

La propiedad **justify-content** se utiliza para alinear toda la cuadrícula dentro del contenedor.

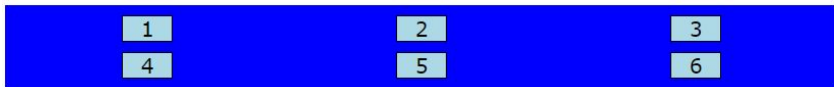
Nota: El ancho total de la cuadrícula debe ser menor que el ancho del contenedor para que la propiedad justify-content tenga algún efecto.



**space-evenly**



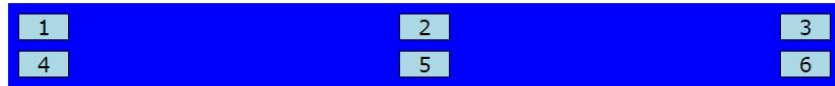
**center**



**space-around**



**start**



**space-between**

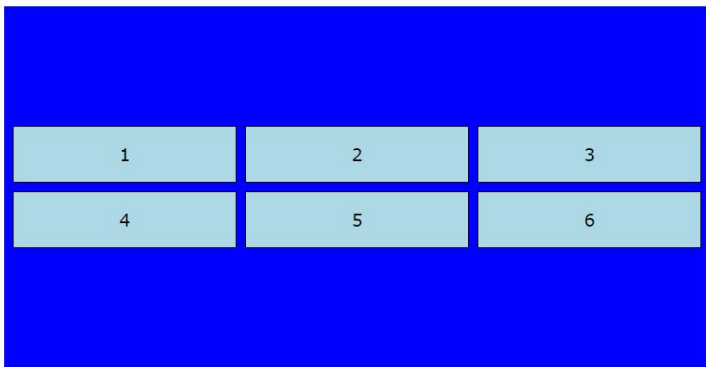


**end**

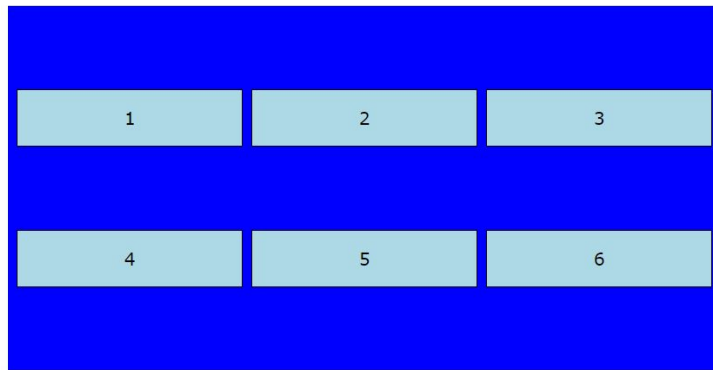
# CSS Grid | Propiedad align-content

La propiedad **align-content** se usa para alinear verticalmente toda la cuadrícula dentro del contenedor.

Nota: La altura total de la cuadrícula debe ser menor que la altura del contenedor para que la propiedad align-content tenga algún efecto.

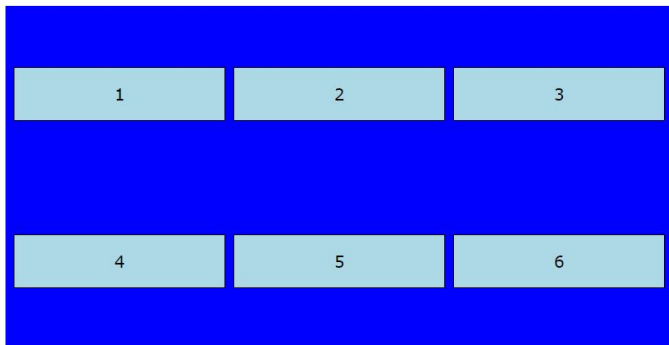


**center**

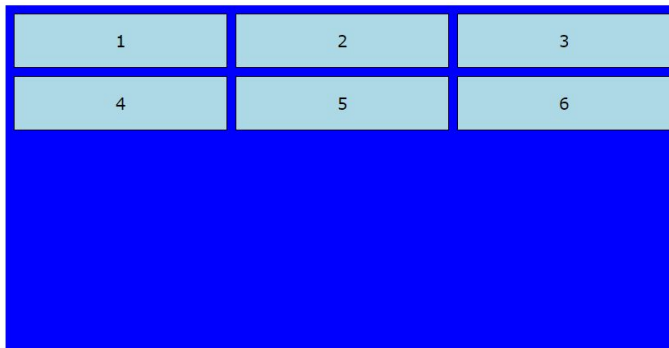


**space-evenly**

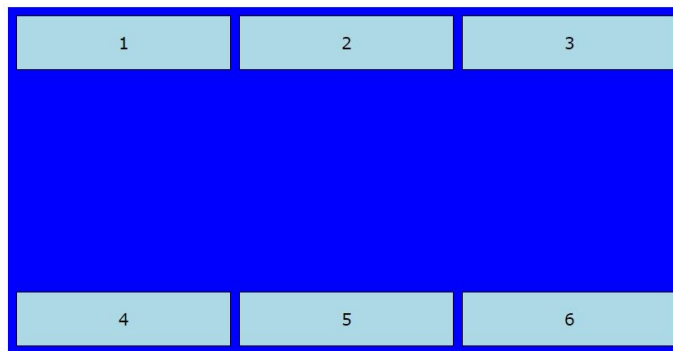
# CSS Grid | Propiedad align-content



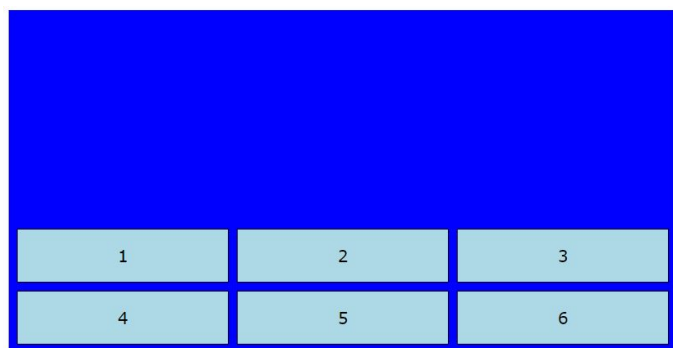
**space-around**



**start**



**space-between**



**end**

# CSS Grid | Propiedad grid-area

La propiedad **grid-area** especifica el tamaño y la ubicación de un elemento de cuadrícula en un diseño de cuadrícula, y es una propiedad abreviada para las siguientes propiedades: *grid-row-start*, *grid-column-start*, *grid-row-end*, *grid-column-end*.

La propiedad del área de la cuadrícula también se puede utilizar para asignar un nombre a un elemento de la cuadrícula. A continuación, se puede hacer referencia a los elementos de cuadrícula con nombre mediante la propiedad *grid-template-areas* del contenedor de cuadrícula.

**Para ampliar:** [https://www.w3schools.com/cssref/pr\\_grid-area.asp](https://www.w3schools.com/cssref/pr_grid-area.asp)

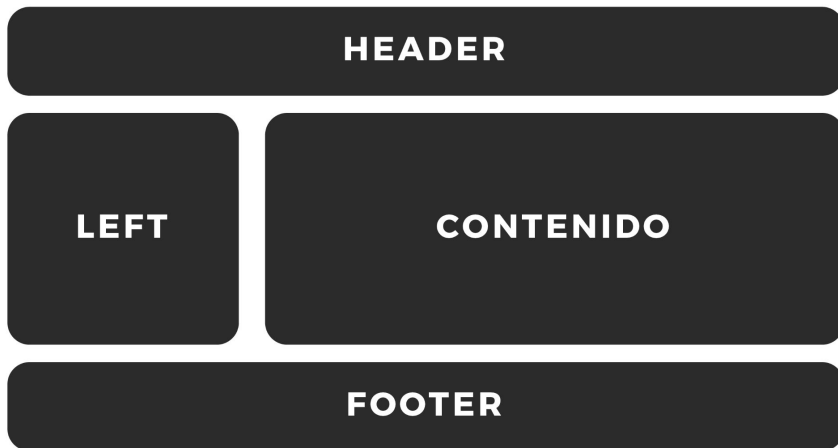


Ver ejemplos:

- [grid-area1.html](#)
- [grid-area2.html](#)
- [grid-area3.html](#)
- [grid-template-area.html](#)

# CSS Grid | Ejemplo

- En el siguiente link podrán ver cómo crear un pequeño proyecto con CSS Grid.
- La idea es que traten de realizar el contenido de la imagen para entender un poco mejor cómo funciona CSS Grid.
  - **Ejemplo:**  
<https://medium.com/sue%C3%B1os-graficos/css-grid-la-mejor-opci%C3%B3n-para-crear-dise%C3%B1os-web-b1b7b8735566>



# CSS Grid | Más información:

## Documentación:

- <https://lenguajecss.com/css/maquetacion-y-colocacion/grid-css/>
- [https://www.w3schools.com/css/css\\_grid.asp](https://www.w3schools.com/css/css_grid.asp)
- [https://www.w3schools.com/css/css\\_grid\\_container.asp](https://www.w3schools.com/css/css_grid_container.asp)
- [https://www.w3schools.com/css/css\\_grid\\_item.asp](https://www.w3schools.com/css/css_grid_item.asp)
- [https://www.w3schools.com/cssref/pr\\_grid-area.asp](https://www.w3schools.com/cssref/pr_grid-area.asp)
- [https://developer.mozilla.org/es/docs/Web/CSS/CSS\\_Grid\\_Layout/Conceptos\\_B%C3%A1sicos\\_del\\_Posicionamiento\\_con\\_Rejillas](https://developer.mozilla.org/es/docs/Web/CSS/CSS_Grid_Layout/Conceptos_B%C3%A1sicos_del_Posicionamiento_con_Rejillas)

## Material multimedia:

- *CSS Grid desde cero:* <https://youtu.be/n1Lnar9mTF8>
- *CSS GRID Página Web Responsive:* [https://youtu.be/9w3gy2dYN\\_E?t=1](https://youtu.be/9w3gy2dYN_E?t=1)

# Herramientas

## Firefox Browser Developer Edition:

<https://www.mozilla.org/es-ES/firefox/developer/>

