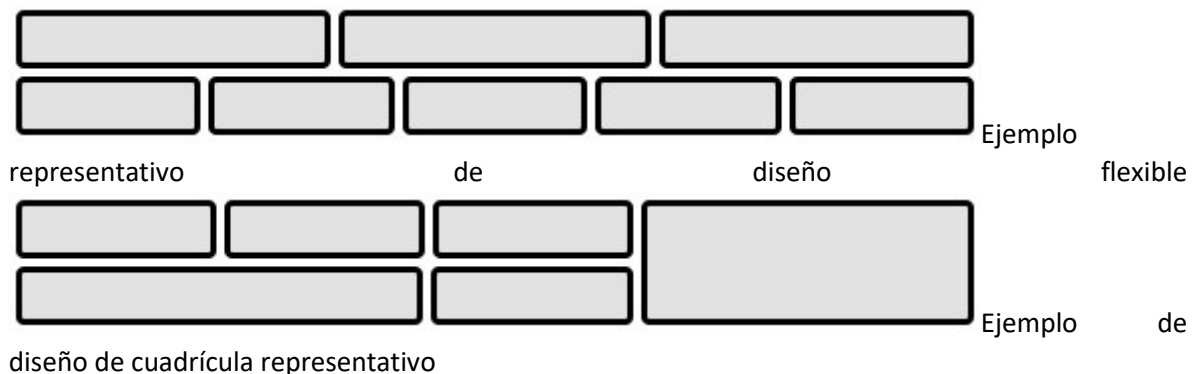


# CSS Grid

**CSS Grid layout** contiene funciones de diseño dirigidas a los desarrolladores de aplicaciones web. El CSS grid se puede utilizar para lograr muchos diseños diferentes. También se destaca por permitir dividir una página en áreas o regiones principales, por definir la relación en términos de tamaño, posición y capas entre partes de un control construido a partir de primitivas HTML.

Al igual que las tablas, el grid layout permite a un autor alinear elementos en columnas y filas. Sin embargo, con CSS grid son posibles muchos más diseños y de forma más sencilla que con las tablas. Por ejemplo, los elementos secundarios de un contenedor de cuadrícula podrían posicionarse para que se solapen y se superpongan, de forma similar a los elementos posicionados en CSS.

Grid Layout es un modelo de diseño para CSS que tiene poderosas capacidades para controlar el tamaño y la posición de las cajas y su contenido. A diferencia del [diseño de caja flexible](#), que está orientado a un solo eje, el diseño de cuadrícula está optimizado para diseños bidimensionales: aquellos en los que se desea alinear el contenido en ambas dimensiones.



Además, debido a su capacidad para colocar elementos explícitamente en la cuadrícula, Grid Layout permite transformaciones dramáticas en la estructura de diseño visual sin requerir los cambios de marcado correspondientes. Al combinar [las consultas de medios](#) con las propiedades CSS que controlan el diseño del contenedor de cuadrícula y sus elementos secundarios, los autores pueden adaptar su diseño a los cambios en los factores de forma del dispositivo, la orientación y el espacio disponible, al tiempo que conservan una estructura semántica más ideal de su contenido en todas las presentaciones.

Aunque muchos diseños se pueden expresar con Grid o Flexbox, cada uno tiene sus especialidades. Grid refuerza la alineación bidimensional, utiliza

un enfoque de arriba hacia abajo para el diseño, permite la superposición explícita de elementos y tiene capacidades de expansión más poderosas. Flexbox se enfoca en la distribución del espacio dentro de un eje, usa un enfoque ascendente más simple para el diseño, puede usar un sistema de ajuste de línea basado en el tamaño del contenido para controlar su eje secundario y se basa en la jerarquía de marcado subyacente para crear diseños más complejos. Se espera que ambas sean herramientas valiosas y complementarias para los autores de CSS.

El nivel de cuadrícula 2 agrega la característica de [subcuadrícula](#) : un eje subcuadrícula es uno que hace coincidir sus líneas de cuadrícula con las líneas en la cuadrícula principal del elemento, y que deriva los tamaños de sus pistas a través de esta integración con la cuadrícula principal.

## 1.1. Antecedentes y motivación

First name:	<input type="text"/>	Department: <div>Finance Human Resources Marketing Payroll Shipping</div>
Last name:	<input type="text"/>	
Address:	<input type="text"/>	
<div>Back Cancel Next</div>		

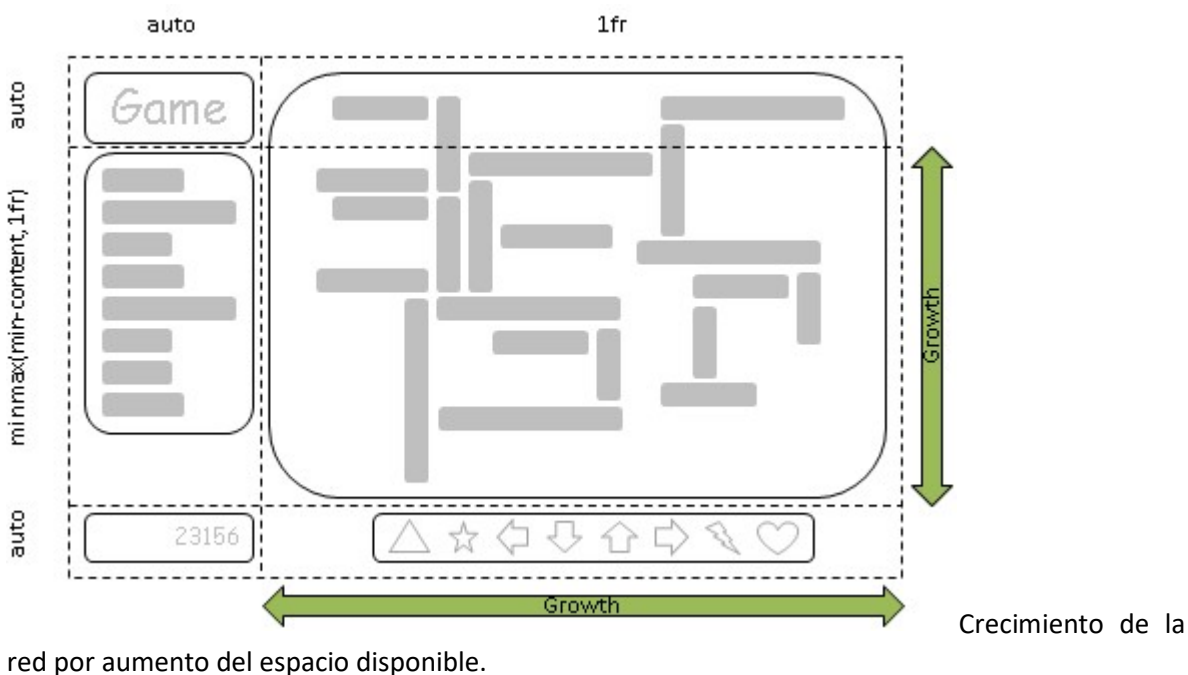
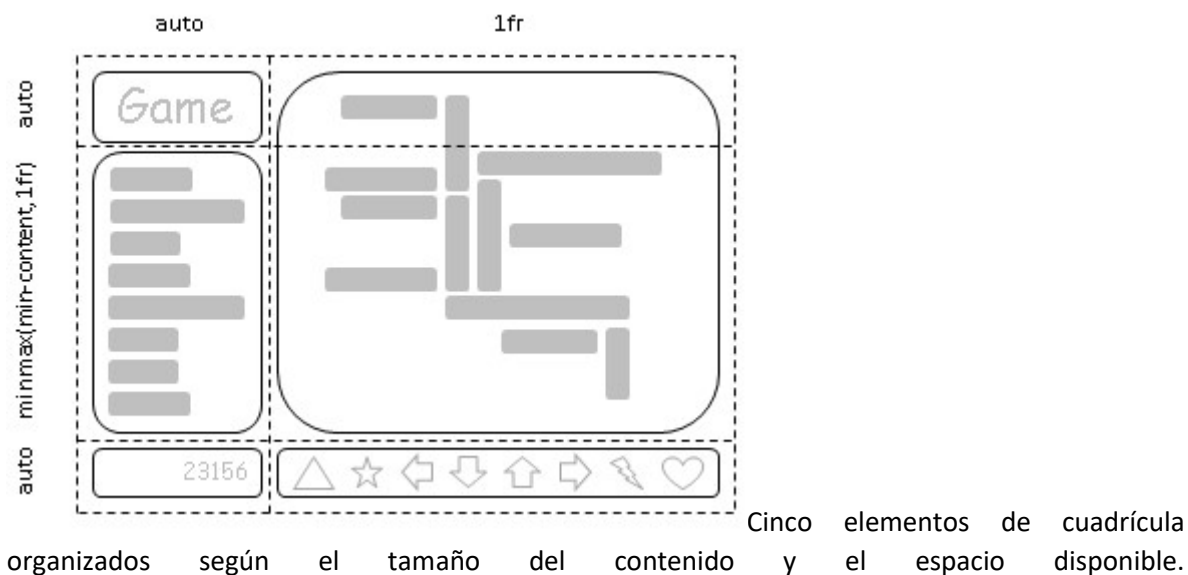
Ejemplo de diseño de

aplicación que requiere alineación horizontal y vertical.

A medida que los sitios web evolucionaron de documentos simples a aplicaciones complejas e interactivas, las técnicas para el diseño de documentos, por ejemplo, flotantes, no fueron necesariamente adecuadas para el diseño de aplicaciones. Mediante el uso de una combinación de tablas, JavaScript o mediciones cuidadosas en elementos flotantes, los autores descubrieron soluciones para lograr los diseños deseados. Los diseños que se adaptaban al espacio disponible a menudo eran frágiles y resultaban en un comportamiento contrario a la intuición a medida que el espacio se reducía. Como alternativa, los autores de muchas aplicaciones web optaron por un diseño fijo que no puede aprovechar los cambios en el espacio de representación disponible en una pantalla.

Las capacidades del diseño de cuadrícula abordan estos problemas. Proporciona un mecanismo para que los autores dividan el espacio disponible para el diseño en columnas y filas utilizando un conjunto de comportamientos de tamaño predecibles. Luego, los autores pueden colocar y dimensionar con precisión los elementos del bloque de construcción de su aplicación en las [áreas de la cuadrícula](#) definidas por las intersecciones de estas columnas y filas. Los siguientes ejemplos ilustran las capacidades de adaptación del diseño de cuadrícula y cómo permite una separación más limpia de contenido y estilo.

### 1.1.1. Adaptación de diseños al espacio disponible



El diseño de cuadrícula se puede utilizar para cambiar el tamaño de los elementos de forma inteligente dentro de una página web. Las figuras adyacentes representan un juego con cinco componentes principales en el diseño: el título del juego, el área de estadísticas, el tablero de juego, el área de puntuación y el área de control. La intención del autor es dividir el espacio para el juego de manera que:

- El área de estadísticas siempre aparece inmediatamente debajo del título del juego.
- El tablero de juego aparece a la derecha de las estadísticas y el título.
- La parte superior del título del juego y el tablero de juego siempre deben estar alineados.
- La parte inferior del tablero de juego y la parte inferior del área de estadísticas se alinean cuando el juego alcanza su altura mínima. En todos los demás casos, el tablero de juego se estirará para aprovechar todo el espacio disponible.
- Los controles están centrados debajo del tablero de juego.
- La parte superior del área de puntuación se alinea con la parte superior del área de controles.
- El área de puntuación está debajo del área de estadísticas.
- El área de puntuación está alineada con los controles debajo del área de estadísticas.

El siguiente ejemplo de diseño de cuadrícula muestra cómo un autor puede lograr todas las reglas de tamaño, ubicación y alineación de forma declarativa.

## Ejemplo sencillo

El siguiente ejemplo muestra un grid de tres columnas con filas nuevas creadas con un mínimo de 100 píxeles y un máximo automático. Los elementos se han colocado en el grid utilizando posicionamiento en línea.

HTML

```
<div class="wrapper">  
  
  <div class="one">One</div>
```

```
<div class="two">Two</div>

<div class="three">Three</div>

<div class="four">Four</div>

<div class="five">Five</div>

<div class="six">Six</div>

</div>
```

Copy to Clipboard  
CSS

```
.wrapper {

  display: grid;

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

  grid-gap: 10px;

  grid-auto-rows: minmax(100px, auto);

}

.one {

  grid-column: 1 / 3;

  grid-row: 1;

}

.two {

  grid-column: 2 / 4;

  grid-row: 1 / 3;

}

.three {
```

```
    grid-column: 1;

    grid-row: 2 / 5;
}

.four {

    grid-column: 3;

    grid-row: 3;
}

.five {

    grid-column: 2;

    grid-row: 4;
}

.six {

    grid-column: 3;

    grid-row: 4;
}
```

(Etemad & Tab Atkins Jr, 2021)

## Referencias

Etemad, E., & Tab Atkins Jr, R. A. (2021). Obtenido de <https://drafts.csswg.org/css-grid/>