

## 2.2.3. Grid layout y Media Queries



**LENGUAJES DE PROGRAMACIÓN NIVEL 5. UNIDAD 2**

DESARROLLO FRONTEND AVANZADO

## Contenido

Introducción a Grid Layout y Media Queries.....	3
Importancia del Grid Layout en el Diseño Web.....	3
Media Queries y su Función en el Diseño Responsivo .....	4
Conceptos Básicos de Grid Layout .....	5
CSS y Grid Layout.....	6
¿Cómo se trabaja con Grid Layout? .....	7
Uso de Media Queries con Grid Layout .....	11
Integración de Media Queries con Grid- Layout.....	11
Mejores Prácticas y Consideraciones .....	13
Ejemplos y Casos de Uso .....	15
Ejemplos .....	15
Caso de Uso: Diseño de una Página de Inicio .....	16
Conclusiones.....	17
Despedida y Recursos Adicionales .....	18

## Introducción a Grid Layout y Media Queries



El diseño web ha evolucionado enormemente en los últimos años, y dos tecnologías fundamentales que han contribuido significativamente a esta evolución son Grid Layout y Media Queries.

### Importancia del Grid Layout en el diseño web

El Grid Layout, o diseño en cuadrícula, es una herramienta poderosa que permite a los desarrolladores y diseñadores crear diseños complejos y estructurados de manera más eficiente y flexible que nunca. Con Grid Layout, podemos dividir el diseño de una página en filas y columnas, estableciendo así una estructura visual clara y ordenada que facilita la disposición de los elementos de la página de manera precisa y consistente. Esto no solo simplifica el proceso de diseño, sino que también mejora la legibilidad y la experiencia del usuario al proporcionar una organización visual coherente.

## Media Queries y su función en el diseño responsivo

Las Media Queries son una característica fundamental de CSS que nos permiten aplicar estilos específicos según las características del dispositivo que está visualizando la página, como el tamaño de la pantalla, la orientación, la resolución y otros aspectos. En otras palabras, las Media Queries nos permiten crear diseños responsivos que se adaptan automáticamente a diferentes dispositivos y tamaños de pantalla, lo que garantiza una experiencia de usuario óptima en una amplia variedad de dispositivos, desde ordenadores de escritorio hasta teléfonos móviles y tabletas. Mediante el uso inteligente de Media Queries, podemos ajustar el diseño, el tamaño de los elementos, la disposición y otros aspectos visuales para garantizar que nuestro sitio web se vea y funcione de manera excelente en cualquier dispositivo, independientemente de su tamaño o resolución de pantalla.

## Conceptos básicos de Grid Layout



El Grid Layout es una técnica de diseño que nos permite crear diseños de páginas web complejos y flexibles mediante la definición de una estructura de cuadrícula compuesta por filas y columnas. Aquí hay una descripción de los conceptos fundamentales de Grid Layout:

- **Filas y columnas:**

En Grid Layout, la página se organiza en una estructura de filas y columnas. Las filas se definen mediante la propiedad `grid-template-rows`, mientras que las columnas se definen con la propiedad `grid-template-columns`. Podemos establecer el tamaño de las filas y columnas utilizando diferentes unidades, como píxeles, porcentajes o fracciones.

- **Celdas:**

Las celdas son los espacios individuales dentro de la cuadrícula donde colocamos los elementos de nuestra página web. Cada celda ocupa una posición específica en la cuadrícula, definida por su fila y columna. Podemos colocar elementos en una celda específica utilizando la propiedad `grid-row` para especificar la fila y `grid-column` para especificar la columna.

- **Contenedores y elementos:**

En Grid Layout, el contenedor principal que define la cuadrícula se denomina "grid container". Todos los elementos hijos directos de este contenedor se convierten en "grid items" y se colocan dentro de la

cuadrícula según las reglas establecidas. Podemos controlar la ubicación y el tamaño de los elementos utilizando propiedades como `grid-row-start`, `grid-row-end`, `grid-column-start` y `grid-column-end`.

## CSS y Grid Layout

Grid Layout es una capacidad avanzada de CSS que permite crear diseños de páginas web complejos y flexibles mediante la definición de una cuadrícula bidimensional. Con Grid Layout, los desarrolladores web tienen un control preciso sobre la disposición y el posicionamiento de los elementos en una página, lo que facilita la creación de diseños sofisticados y adaptables a diferentes tamaños de pantalla y dispositivos.

La especificación de Grid Layout introduce una nueva forma de organizar y estructurar el diseño de una página web, complementando las capacidades existentes de CSS como Flexbox y las técnicas de posicionamiento basadas en flotadores. Mientras que Flexbox está diseñado principalmente para organizar elementos en una sola dimensión (ya sea en una fila o en una columna), Grid Layout permite crear una cuadrícula bidimensional de filas y columnas, lo que proporciona una mayor flexibilidad y control sobre la disposición de los elementos.

Con Grid Layout, los desarrolladores pueden definir fácilmente la estructura de la cuadrícula, especificar el tamaño y la ubicación de las filas y columnas, y colocar los elementos en celdas específicas de la cuadrícula. Esto permite crear diseños complejos con múltiples secciones y áreas, distribuir el espacio de manera uniforme o personalizada, y alinear los elementos de forma precisa.

Una de las principales ventajas de Grid Layout es su capacidad para crear diseños responsivos de manera más eficiente. Al definir la estructura de la cuadrícula y establecer reglas de posicionamiento relativas, los desarrolladores pueden crear diseños que se adapten automáticamente a diferentes tamaños de pantalla y dispositivos, sin necesidad de recurrir a soluciones complicadas de CSS o JavaScript.

## ¿Cómo se trabaja con Grid Layout?

Trabajar con Grid Layout implica seguir una serie de pasos para definir la estructura de la cuadrícula y colocar los elementos en las posiciones deseadas. Aquí hay una descripción de los pasos básicos para trabajar con Grid Layout:

### 1. Crear un contenedor de cuadrícula:

Utiliza CSS para seleccionar el elemento HTML que actuará como contenedor de la cuadrícula y establece su propiedad `display` en `grid` o `inline-grid`. Esto convierte al elemento en un contenedor de cuadrícula, donde puedes organizar los elementos hijos en una cuadrícula.

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

1

### 2. Definir la estructura de la cuadrícula:

Utiliza las propiedades `grid-template-rows` y `grid-template-columns` para especificar el número de filas y columnas de la cuadrícula, así como sus tamaños y distribuciones.

```
.container {
  display: grid;
  grid-template-rows: 100px 200px; /* Dos filas de 100px y 200px respectivamente */
  grid-template-columns: 1fr 2fr; /* Dos columnas con un ancho flexible (fracción) */
}
```

2

---

<sup>1</sup> Accesibilidad

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

<sup>2</sup> Accesibilidad

```
.container {
  display: grid;
  grid-template-rows: 100px 200px; /* Dos filas de 100px y 200px respectivamente */
  grid-template-columns: 1fr 2fr; /* Dos columnas con un ancho flexible (fracción) */
}
```

Este código crea un contenedor de cuadrícula con dos filas de alturas específicas (100px y 200px) y dos columnas que se distribuyen el espacio disponible en proporción 1:2.

3. Colocar elementos en la cuadrícula:

Utiliza las propiedades `grid-row` y `grid-column` en los elementos hijos para especificar en qué celda de la cuadrícula deben colocarse. Puedes definir el inicio y el final de la fila y la columna para cada elemento.

```
.item {  
  grid-row: 1 / 2; /* El elemento ocupa desde la fila 1 hasta la fila 2 */  
  grid-column: 1 / 2; /* El elemento ocupa desde la columna 1 hasta la columna 2 */  
}
```

3

Este código CSS está diseñado para ser aplicado a un elemento con la clase `.item` dentro de un contenedor que utiliza un diseño de cuadrícula (`display: grid`). Aquí está lo que hace cada línea:

- `grid-row: 1 / 2`; indica que el elemento debe ubicarse en la cuadrícula de tal manera que comience en la línea de la cuadrícula 1 (el inicio de la primera fila) y se extienda hasta, pero no incluya, la línea de la cuadrícula 2 (el inicio de la segunda fila). En la práctica, esto significa que el elemento ocupará completamente la primera fila.
- `grid-column: 1 / 2`; especifica que el elemento debe comenzar en la línea de la cuadrícula 1 (el inicio de la primera columna) y extenderse hasta, pero no incluir, la línea de la cuadrícula 2 (el inicio de la segunda columna). Esto implica que el elemento llenará completamente la primera columna.

En conjunto, estas propiedades posicionan el elemento `.item` para que ocupe exactamente el primer celdón de la cuadrícula, definido por la primera fila y la primera columna. No se extenderá más allá de este espacio, lo que significa que ocupará la esquina superior izquierda del

---

<sup>3</sup> Accesibilidad

```
.item {  
  grid-row: 1 / 2; /* El elemento ocupa desde la fila 1 hasta la fila 2 */  
  grid-column: 1 / 2; /* El elemento ocupa desde la columna 1 hasta la columna 2 */  
}
```



contenedor de la cuadrícula si no hay otros elementos que afecten su posición.

4. Controlar la alineación y el espaciado:

Utiliza propiedades como `justify-items`, `align-items`, `justify-content` y `align-content` para alinear y distribuir los elementos dentro de la cuadrícula según sea necesario.

```
.container {
  display: grid;
  justify-items: center; /* Centra los elementos horizontalmente */
  align-items: center; /* Centra los elementos verticalmente */
}
```

4

Este código establece y garantiza que todos los elementos dentro del contenedor de cuadrícula `.container` se centren completamente dentro de sus celdas asignadas, tanto horizontal como verticalmente. Este es un patrón comúnmente utilizado para alinear y distribuir elementos de manera uniforme dentro de un diseño de cuadrícula, proporcionando una apariencia ordenada y estéticamente agradable.

5. Agregar elementos adicionales y ajustar la cuadrícula según sea necesario:

Puedes agregar más elementos a la cuadrícula y ajustar su tamaño y posición según lo requiera el diseño de tu página web. Además, puedes usar propiedades como `grid-template-areas` para definir áreas dentro de la cuadrícula y asignar elementos a esas áreas de manera semántica.

---

<sup>4</sup> Accesibilidad

```
.container {
  display: grid;
  justify-items: center; /* Centra los elementos horizontalmente */
  align-items: center; /* Centra los elementos verticalmente */
}
```

```
.container {
  display: grid;
  grid-template-areas:
    "header header"
    "sidebar main"
    "footer footer";
}
```

5

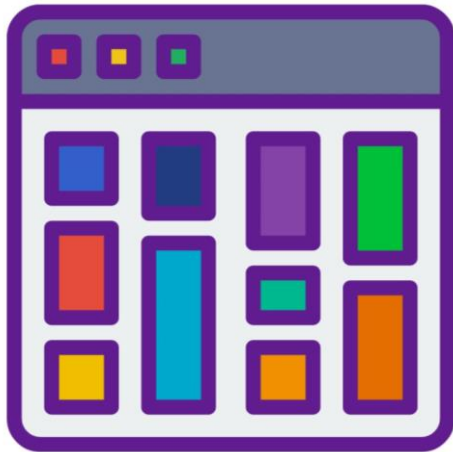
Este diseño crea un contenedor de cuadrícula con tres filas. La primera y la tercera fila tienen una única área que se extiende a lo largo de dos columnas, destinada para el header y el footer, respectivamente. La segunda fila divide el espacio en dos áreas: sidebar y main, destinadas para una barra lateral y un contenido principal, respectivamente.

---

<sup>5</sup> Accesibilidad

```
.container{
  display: grid;
  grid-template-areas:
    "header header"
    "sidebar main"
    "footer footer";
}
```

## Uso de Media Queries con Grid Layout



En la actualidad, la diversidad de dispositivos y tamaños de pantalla utilizados para acceder a sitios web ha aumentado significativamente. Esto plantea un desafío para los desarrolladores web: ¿cómo garantizar una experiencia consistente y agradable para los usuarios independientemente del dispositivo que utilicen? Aquí es donde entran en juego tecnologías como Grid Layout y Media Queries.

El Grid Layout ofrece una manera poderosa y flexible de diseñar interfaces web, permitiendo organizar el contenido en filas y columnas de una manera más precisa y controlada. Por otro lado, las Media Queries permiten aplicar estilos específicos basados en las características del dispositivo, como su tamaño de pantalla.

La combinación perfecta.

### Integración de Media Queries con Grid Layout

Integrar Media Queries con Grid Layout es una práctica fundamental para crear diseños web responsivos y adaptables a diferentes dispositivos y tamaños de pantalla. Las Media Queries permiten aplicar reglas de estilo específicas según las características del dispositivo, como el ancho de la pantalla, la orientación o la resolución, lo que permite ajustar el diseño de la página según las necesidades del usuario.

Para utilizar Media Queries con Grid Layout, primero se define la estructura de la cuadrícula utilizando las propiedades de Grid, como `grid-template-`

columns y grid-template-rows, para establecer el número y el tamaño de las filas y columnas. Una vez definida la cuadrícula base, se pueden utilizar Media Queries para decidir qué grid debe utilizarse según el dispositivo o tamaño de la pantalla.

Por ejemplo, supongamos que queremos que nuestra cuadrícula tenga un diseño de dos columnas en pantallas grandes y un diseño de una sola columna en pantallas más pequeñas. Podríamos definir la cuadrícula de la siguiente manera:

```
.container {
  display: grid;
  grid-template-columns: 1fr 1fr; /* Diseño de dos columnas */
}

@media screen and (max-width: 768px) {
  .container {
    grid-template-columns: 1fr; /* Diseño de una sola columna en pantallas más pequeñas */
  }
}
```

6

En este ejemplo, y en la parte de abajo, tras @media, utilizamos una Media Query con una condición de max-width: 768px para aplicar un diseño de una sola columna cuando el ancho de la pantalla es igual o menor a 768 píxeles. En el caso de que el ancho de pantalla sea mayor se utiliza el diseño a dos columnas definido arriba.

---

<sup>6</sup> Accesibilidad

```
.container {
  display: grid;
  grid-template-columns: 1fr 1fr; /* Diseño de dos columnas */
}

@media screen and (max-width: 768px) {
  .container {
    grid-template-columns: 1fr; /*Diseño de una sola columna en pantallas mas pequeñas */
  }
}
```

## Mejores prácticas y consideraciones



Al diseñar layouts con Grid Layout y Media Queries, es importante tener en cuenta algunas mejores prácticas y consideraciones para garantizar un resultado óptimo en términos de rendimiento, accesibilidad y usabilidad:

- **Planificación y diseño responsivo:** Antes de comenzar a implementar un layout con Grid Layout, es crucial realizar una planificación adecuada del diseño teniendo en cuenta los diferentes dispositivos y tamaños de pantalla. Esto implica pensar en cómo se verá el contenido en dispositivos de diferentes anchos y alturas y cómo se adaptará el diseño para proporcionar una experiencia de usuario coherente en todos ellos.
- **Uso semántico de Grid:** Aprovecha al máximo la semántica de HTML al utilizar Grid Layout. Utiliza elementos HTML apropiados como contenedores para filas y columnas en lugar de divs genéricos para mejorar la accesibilidad y la comprensión del código.
- **Mantenimiento del código:** Mantén el código limpio y organizado al utilizar Grid Layout y Media Queries. Usa comentarios descriptivos para documentar el propósito de diferentes secciones del código y agrupa reglas relacionadas para facilitar su mantenimiento y comprensión.
- **Pruebas en dispositivos reales:** Realiza pruebas exhaustivas en una variedad de dispositivos reales para asegurarte de que el diseño responde de manera adecuada en diferentes tamaños de pantalla y

dispositivos. Esto te permitirá identificar y solucionar cualquier problema de diseño o rendimiento antes de lanzar el sitio web.

- **Accesibilidad:** Asegúrate de que tu diseño sea accesible para todos los usuarios, incluidos aquellos con discapacidades visuales o de movilidad. Utiliza etiquetas adecuadas, atributos aria y técnicas de diseño responsivo accesible para garantizar que el contenido sea fácilmente accesible y utilizable para todos.
- **Rendimiento:** Optimiza el rendimiento del sitio web al minimizar el uso de recursos innecesarios y optimizar la carga de la página. Esto incluye la minimización de solicitudes HTTP, la optimización de imágenes y el uso eficiente de CSS y JavaScript para reducir el tiempo de carga y mejorar la experiencia del usuario.

Al seguir estas mejores prácticas y consideraciones, podrás aprovechar al máximo las capacidades de Grid Layout y Media Queries para crear diseños web atractivos, responsivos y accesibles que ofrezcan una experiencia de usuario excepcional en una variedad de dispositivos y tamaños de pantalla.

## Ejemplos y casos de uso

En el diseño web moderno, la capacidad de crear diseños responsivos y adaptables es esencial para garantizar una experiencia de usuario óptima en una amplia variedad de dispositivos y tamaños de pantalla. En este epígrafe, exploraremos cómo utilizar Grid Layout y Media Queries, dos potentes herramientas de CSS, para lograr este objetivo. Grid Layout nos permite crear diseños de manera más eficiente y flexible, mientras que Media Queries nos permite aplicar estilos específicos según el tamaño y las características del dispositivo.

A lo largo de este contenido, examinaremos ejemplos concretos y casos de uso prácticos que ilustran cómo estas técnicas pueden ser implementadas en proyectos reales para crear diseños web atractivos y funcionales en cualquier entorno.

### Ejemplos

- **Diseño de un Blog Responsivo**

En un proyecto de blog, se puede utilizar Grid Layout para organizar el contenido de manera eficiente. Por ejemplo, se pueden crear áreas específicas para la publicación, la barra lateral y los elementos de navegación.

Con Media Queries, se puede ajustar la disposición de estas áreas en función del tamaño de la pantalla. Por ejemplo, en pantallas más pequeñas, se puede cambiar la disposición para que la barra lateral se coloque debajo del contenido principal para mejorar la legibilidad en dispositivos móviles.

- **Diseño de una Tienda en Línea**

En una tienda en línea, Grid Layout puede ser utilizado para crear una cuadrícula de productos que se ajuste dinámicamente al tamaño de la pantalla.

Con Media Queries, se puede cambiar el número de columnas en la cuadrícula o reorganizar los elementos para que se vean mejor en dispositivos móviles. Por ejemplo, se puede mostrar una sola columna de productos en dispositivos móviles para facilitar la navegación y la compra.

- **Diseño de un Tablero de Administración**

En un tablero de administración, Grid Layout puede ser utilizado para organizar widgets y paneles de información de manera eficiente.

Con Media Queries, se puede ajustar la disposición de los elementos para que se adapten a diferentes tamaños de pantalla y dispositivos.

Por ejemplo, se puede reorganizar los paneles y ocultar elementos menos importantes en pantallas más pequeñas para mejorar la experiencia de usuario en dispositivos móviles.

### Caso de uso: Diseño de una página de inicio

En una página de inicio, Grid Layout y Media Queries pueden utilizarse para crear un diseño visualmente atractivo y funcional. Por ejemplo, se puede utilizar Grid Layout para organizar secciones de contenido como encabezados, imágenes destacadas, testimonios y llamadas a la acción en un diseño de varias columnas. Con Media Queries, se puede ajustar la disposición y el tamaño de estos elementos para que se vean bien en una variedad de dispositivos y tamaños de pantalla, asegurando una experiencia de usuario consistente y atractiva en todas partes.

Estos ejemplos ilustran cómo Grid Layout y Media Queries pueden ser utilizados de manera efectiva para crear diseños web responsivos y adaptables en una variedad de proyectos y casos de uso. Al aprovechar estas tecnologías, los desarrolladores pueden crear experiencias de usuario excepcionales que se adapten a las necesidades y preferencias de los usuarios en diferentes dispositivos y contextos de uso.



## Conclusiones

En este módulo, hemos explorado en profundidad el uso de Grid Layout y Media Queries en el diseño web moderno. Hemos aprendido sobre los conceptos básicos de Grid Layout, cómo trabajar con él para crear estructuras de diseño flexibles y eficientes, así como su integración con Media Queries para lograr diseños responsivos y adaptables. Además, hemos discutido las mejores prácticas, consideraciones importantes y ejemplos de casos de uso para aplicar estas técnicas de manera efectiva en proyectos web reales.

*Espero que este módulo haya sido útil para comprender mejor cómo utilizar Grid Layout y Media Queries en sus proyectos. Animo a todos los estudiantes a practicar y experimentar con estas herramientas, ya que la experiencia práctica es fundamental para dominarlas. **¡No tengan miedo de explorar y probar nuevas ideas en sus propios proyectos web!***

*Recuerden que el diseño web es un campo en constante evolución, por lo que es importante mantenerse actualizado con las últimas tendencias y técnicas. Continúen aprendiendo, practicando y desarrollando sus habilidades para convertirse en diseñadores web más competentes y efectivos. **¡Les deseo mucho éxito en sus futuros proyectos!***

## Despedida y recursos adicionales

**Ha sido un placer recorrer juntos este módulo dedicado al Grid Layout y Media Queries.** Espero que hayas encontrado útiles los conocimientos adquiridos y que estes deseando aplicarlos en tus propios proyectos web.

Recuerda que el diseño web es un campo emocionante y en constante evolución, y dominar técnicas como el Grid Layout y las Media Queries te permitirá crear sitios web modernos, flexibles y adaptativos que se destaquen en el mundo digital.

Como despedida, me gustaría compartir contigo algunos recursos adicionales que pueden ser de gran ayuda para continuar profundizando en este tema:

- Tutoriales en línea: Plataformas como YouTube, Udemy y Coursera ofrecen una amplia variedad de tutoriales gratuitos y de pago sobre Grid Layout y Media Queries. Busquen aquellos que se adapten mejor a su estilo de aprendizaje y nivel de experiencia.
- Documentación oficial: Consulten la documentación oficial de CSS para obtener información detallada sobre Grid Layout y Media Queries. Esta es una excelente fuente de referencia que les ayudará a comprender mejor cómo utilizar estas técnicas en sus proyectos.
- Herramientas para el diseño responsivo: Utilicen herramientas como Browser DevTools, CodePen, o Flexbox Froggy para practicar y experimentar con Grid Layout y Media Queries. Estas herramientas les permitirán visualizar instantáneamente los cambios en el diseño en diferentes dispositivos y tamaños de pantalla.

*Recuerda siempre mantenerte curioso, seguir aprendiendo y practicando. ¡Te deseo mucho éxito en tus futuros proyectos!*

***¡Hasta pronto!***