

UNIVERSIDAD MARIANO GÁLVEZ DE GUATEMALA
FACULTAD DE INGENIERÍA EN SISTEMAS DE
INFORMACIÓN Y CIENCIAS DE LA
COMPUTACIÓN
CUILAPA, SANTA ROSA

Curso: Desarrollo Web

Ing. José Luis Xiloj



ACTIVIDAD DEL TEMA - WEB FORMS

Leilany Yanira Mejía Carrillo

1590-16-3786

Sección "A"



INTRODUCCIÓN

Flexbox y Grid son dos módulos de diseño CSS que se han convertido en la tendencia principal durante los últimos años. Ambos nos permiten crear diseños complejos que antes sólo eran posibles mediante la aplicación de trucos con CSS y/o JavaScript. Flexbox y CSS Grid comparten múltiples similitudes y pueden resolver muchos diseños con ambos. Sin embargo, cuándo usar, es otra pregunta.

FLEXBOX



Conceptos básicos y terminología

Dado que flexbox es un módulo completo y no una sola propiedad, implica muchas cosas, incluido su conjunto completo de propiedades. Algunos de ellos están destinados a establecerse en el contenedor (elemento principal, conocido como "contenedor flexible"), mientras que otros están destinados a establecerse en los elementos secundarios (dichos "elementos flexibles").

Si el diseño "regular" se basa en direcciones de flujo en bloque y en línea, el diseño flexible se basa en "direcciones de flujo flexible". Eche un vistazo a esta figura de la especificación, que explica la idea principal detrás del diseño flexible.

Estructura

La estructura de Flexbox se compone de contenedores padre e hijos (Contenedor-Flex y Elementos-Flex respectivamente)



- El Contenedor-Flex es nuestro contenedor padre, y es él quién va a contener a todos los elementos hijos a los cuáles queremos posicionar.
- El Elemento-Flex es nuestro elemento hijo, el cual utilizará todo el espacio disponible para ubicarse de acuerdo con las propiedades a las cuales este sometido junto a los demás Elementos-Flex contenidos.

Cada diseño flex esta compuesto por dos ejes:

- Un *eje principal* que define el posicionamiento horizontal de los Elementos-Flex.

- Un *eje secundario* que define el posicionamiento vertical de los Elementos-Flex.

El Contenedor-Flex contiene direcciones que definen el origen y el fin del flujo de Elementos-Flex:

- Inicio principal
- Fin principal
- Inicio secundario
- Fin secundario

Cada Elemento-Flex está definido por sus dimensiones, anchura (tamaño principal) y altura (tamaño secundario) que dependen del eje principal y el eje secundario.

Propiedades

Si queremos utilizar las propiedades de Flexbox tendremos que definirlo mediante la propiedad *display* y su valor *flex* o *inline-flex* dentro del selector que nosotros definamos que es nuestro elemento padre o Contenedor-Flex.

```
.contenedor-padre {  
    display: flex;  
}
```

En este artículo solo daremos un pantallazo de las principales propiedades del elemento padre con una breve descripción en cada una.

¿Para qué sirve CSS Flexbox?

Una de las principales funciones de la hoja de estilo en cascada o CSS es maquetar todos los elementos de la página web: el texto, las imágenes y los botones se pueden organizar en detalle, determinando dónde debe aparecer cada elemento en la pantalla hasta el más mínimo píxel. No obstante, esto solo es posible si se conoce el tamaño de la pantalla y su relación de aspecto. En la era del **Internet móvil**, con su gran variedad de opciones en ese sentido (los smartphones suelen cambiar de orientación al inclinarlos), no es posible lograr un resultado convincente con las cajas rígidas típicas del CSS.

El flexbox (más concretamente, CSS Flexible Box Layout o diseño de caja flexible CSS) funciona de manera más inteligente y dinámica: el diseño se adapta de forma flexible a la pantalla donde se muestra, siguiendo el concepto del diseño receptivo. El espacio se llena o los elementos se desplazan de modo que todo permanezca visible. Para lograrlo sin desmantelar el diseño por completo, flexbox funciona

con **dos ejes**: el eje principal, que suele ser el horizontal, y el eje transversal o vertical. Mediante estos ejes, los elementos se organizan dentro de la caja y se distribuyen en relación unos con otros. Una vez hecho esto, CSS Flexbox no tiene más que garantizar que el espacio que hay alrededor de estos elementos se llene correctamente.

Beneficios de Flexbox

Flexbox nos permite alinear los contenidos tanto horizontal como verticalmente de forma mucho más sencilla de lo que se venía haciendo hasta ahora, dejando a un lado propiedades como float o position que podían llegar a dar conflictos.

Con Flexbox la alineación es mucho más fácil con propiedades como Justify-content o Align-items, como veremos más adelante.

El alineado vertical siempre ha sido un problema pero con Flexbox esto se resuelve de forma mucho más sencilla e intuitiva.

Esta metodología es también aplicable a la creación y personalización de plantillas de WordPress. Editando tanto los archivos que pintan los contenidos como, por ejemplo el single.php, o los archivos de hojas de estilos como el style.css, podemos dar a nuestra web en WordPress el aspecto que deseemos.

GRID



Grid es uno de los **nuevos valores de HTML5** que podemos dar a la propiedad display de un elemento, y que nos permite maquetar las páginas web de una manera mucho más fácil que se hacía con la forma tradicional.

Especificaciones

Especificaciones	Estado	Comentario
CSS Grid Layout La definición de 'grid' en esta selección.	Candidato a recomendación	Definición inicial

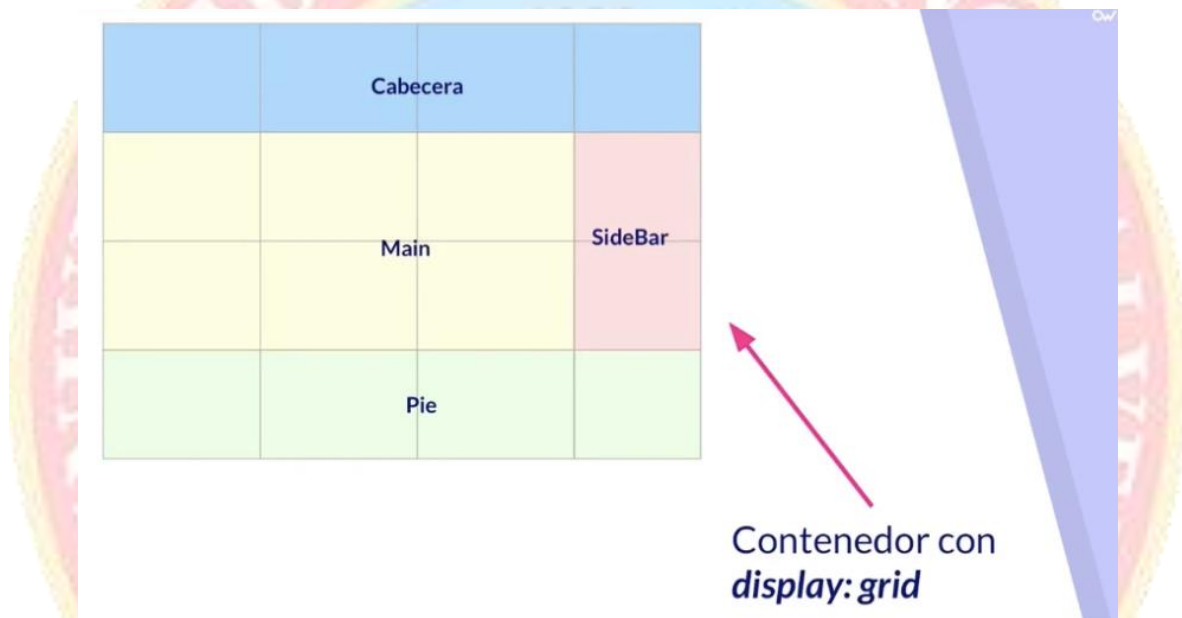
Por qué Grid

Con Grid hacemos que un elemento contenga la propiedad `display:grid`, lo que nos permitirá maquetar de una forma mucho más fácil y potente que con cualquier otro tipo de propiedades.

Esto es así porque, al contrario que en **otras formas de maquetación**, en Grid podemos establecer tanto las filas como las columnas que va a tener el diseño, maquetación o layout, además no sólo en una dimensión, como con Flexbox, sino que vamos a poder especificar la estructura en dos dimensiones.

Ejemplo de uso de Grid

Utilizando **propiedades CSS** vamos a poder especificar estructuras de este tipo:



En esta estructura tenemos cuatro filas y cuatro columnas, una cabecera que ocupa la primera columna entera, una zona main que ocupará seis celdas, el pie que ocupará la última fila y una barra lateral que ocupará dos celdas.

Todo lo anterior lo podemos hacer utilizando únicamente **propiedades CSS**, dando al contenedor que tendrá esa estructura la propiedad `display:grid`.

Aspecto a tener en cuenta

Hay que tener cuidado cuando se utiliza esta propiedad, porque hoy en día no todos los navegadores la soportan. No obstante, hay que decir que la gran mayoría si lo hace, como por ejemplo Chrome, Safari, Firefox, Opera y todos los más usados, aunque pueden existir navegadores de ciertos dispositivos dónde no funcione.

Compatibilidad en navegadores

	PC						Móvil					
	Cromo	Borde	Firefox	explorador de Internet	Ópera	Safari	cromo androide	Firefox para Android	Ópera Android	Safari en iOS	Samsung Internet	WebView Android
grid	✓ 57	✓ dieciséis	✓ 52	✗ No	✓ 44	✓ 10.1	✓ 57	✓ 52	✓ 43	✓ 10.3	✓ 6.0 *	✓ 57

✓ Apoyo total ✗ Sin soporte * Ver notas de implementación.





CONCLUSIONES

- El diseño de cuadrícula es el '**diseño primero**'le ayuda a crear el diseño exterior y, por lo tanto, puede crear un diseño complejo y receptivo.
- Flexbox es de gran ayuda cuando se trata de alinear contenido y mover bloques.
- La cuadrícula CSS funciona tanto para filas como para columnas.
- Flexbox funciona mejor solo con filas o columnas.
- Ahorrará más tiempo y será más útil cuando ambos se utilicen al mismo tiempo.