

Diferencias entre las clases "container y container-fluid"

`<div class="container">` Aquí nuestro contenidos acotados en anchura `</div>`

`<div class=container-fluid>` aquí nuestros contenidos al 100% de la página `</div>`

- Las filas siempre se definen dentro de un contenedor de tipo `.container` (anchura fija) o de tipo `.container-fluid` (anchura variable). De esta forma las filas se alinean bien y muestran el padding correcto.
- Las filas se utilizan para agrupar horizontalmente a varias columnas.
- El contenido siempre se coloca dentro de las columnas, ya que las filas sólo deberían contener como *hijos* elementos de tipo columna.
- Bootstrap define muchas clases CSS (como por ejemplo `.row` y `.col-xs-4`) para crear rejillas rápidamente. También existen *mixins* de Less para crear diseños más semánticos.
- La separación entre columnas se realiza aplicando padding. Para contrarrestar sus efectos en la primera y última columnas, las filas (elementos `.row`) aplican márgenes negativos.
- Las columnas de la rejilla definen su anchura especificando cuántas de las 12 columnas de la fila ocupan. Si por ejemplo quieres dividir una fila en tres columnas iguales, utilizarías la clase `.col-xs-4` (el 4 indica que cada columna ocupa 4 de las 12 columnas en las que se divide cada fila).

Materialize CC representa una arquitectura de software, siendo un *framework* dispuesto a facilitar el proceso del diseño web para un escritorio o distintos dispositivos móviles de cualquier tamaño. Se basa en el Material Design y aumenta el ritmo de su desarrollo.

Este programa posee componentes personalizados que sirven como herramientas para la creación de animaciones y transiciones, que hacen más fácil la realización de algún diseño web. Materialize cuenta con archivos JavaScript y CSS, en una versión estándar que necesita poca o hasta quizás ninguna configuración.

Además...

También contiene archivos de documentación para que sus nuevos usuarios tengan a la mano las normas y ejemplos de **cómo usar este framework**, así como también cuenta con una especial interacción con sus desarrolladores, puesto que tiene una ventana donde se pueden observar las preguntas y comentarios de sus usuarios.

A continuación te mostramos cuáles son las ventajas que esta herramienta proporciona a sus usuarios y por qué se recomienda más que otras:



Para desarrollarlo tiene la gran capacidad de ahorro por mucho más tiempo, puesto que utiliza menos **códigos Javascript y CSS**, en comparación con la escritura que se tendría que usar desde el inicio.

1.- En comparación con otras herramientas, esta *framework* para su funcionamiento no utiliza jQuery, pues usa recomendaciones CSS. También posee **guías de diseño que son generalmente usadas por los desarrolladores** como lo son las del *Material Design* de Google. Que hacen más sencillas y fáciles sus aplicaciones, ya que suelen ser más familiares visualmente para sus usuarios, como los de Android quienes la utilizan mayormente para sus aplicaciones modernas.

2.- Sin tener demasiado conocimiento en cuanto al diseño Materialize proporciona una estética muy atractiva así como un aspecto homogéneo para el producto final, aplicando estilos definidos y personalizados.

3.- Como primer paso para la utilización de Materialize CSS, debes incorporarlo a un proyecto colocando los enlaces junto al JavaScript y a la hoja de estilos, todo esto mediante el CDN al que solo debes incluir dos líneas de código en la página.

4.- Recuerda que para descargar el archivo es más recomendable colocarlo en tu propio servidor. Sin embargo, los diseñadores más avanzados con esta herramienta, podrían incorporarlo al sistema de creación bundles que estén utilizando en su flujo de trabajo, así como también instalarlo desde npm.

5.- Al estar totalmente instalado, debes saber las clases de cada elemento o componente para utilizar el diseño Material Design. Si no lo sabes puedes consultarlo en la documentación que proporciona la herramienta.

Comenzamos por explicar cuáles son las ventajas de esta herramienta y por qué usarla en vez de otras similares y quizás más conocidas como Bootstrap.

Primero aclarar de modo general que las ventajas e inconvenientes de Materialize CSS son más o menos las mismas que las de cualquier otro framework CSS, pues todos ofrecen más o menos lo mismo.

En la parte positiva, de los frameworks CSS en general, podemos destacar:

- Capacidad de ahorrar mucho tiempo en la etapa de desarrollo, ya que no escribiremos tanto código CSS y JavaScript, en comparación con lo que tendríamos que escribir si partimos de cero.
- Aplicando un framework CSS conseguimos una estética atractiva para nuestros desarrollos, y lo podemos conseguir aunque no tengamos especialmente mucha mano con el diseño.
- Nuestros diseños serán más consistentes, ya que iremos aplicando una serie de estilos definidos de antemano, lo que ayudará a que el resultado final tenga un aspecto homogéneo.

Con respecto a las desventajas de los frameworks CSS, que también las hay:

- Muchas veces incorporas al proyecto una cantidad de código que realmente llegas a usar sólo en un porcentaje bajo. Esto quiere decir que, al aplicar todos los estilos CSS y el código JavaScript, estarás incluyendo a tu desarrollo más peso del que realmente irás a usar, porque es difícil que llegues a necesitar de todas las interfaces de usuario que el framework te ofrece. Bien es cierto que Materialize CSS se entrega si se desea con su código en Sass, por lo que, si tenemos un conocimiento razonable de este preprocesador, podríamos quitar de manera relativamente fácil las partes de código que no interesan.
- A veces sentirás que desarrollas para el framework, creando código que tiene muchas clases aplicadas a los elementos, que resultarán un verdadero lío si tienes que cambiar de framework o simplemente eliminarlo del proyecto.
- Tu diseño se puede parecer en su base similar al de otros proyectos que usen el mismo framework.

Eres tú como desarrollador el que decide si las ventajas superan a las desventajas, pero lo cierto es que en líneas generales sí resulta interesante, ya que todos apreciamos en el fondo obtener mejores resultados con menos trabajo.

Materialize vs Bootstrap vs Foundation

Materialize CSS es un framework más nuevo que otras alternativas muy conocidas como Bootstrap o Foundation. Esto es interesante porque ha sido capaz de mejorar algunas cosas de herramientas más antiguas. Gracias a ello, Materialize gana la partida por varias razones, pues entre las que podemos destacar.

- Materialize CSS no requiere jQuery para funcionar, aunque si lo tienes lo puedes usar perfectamente en el proyecto y beneficiarte también de la librería Javascript para inicializar más cómodamente los componentes que requieran JS.
- Usa desde el principio recomendaciones CSS que otros frameworks aún no implementan, o han tardado bastante tiempo en incorporar.
- Aplica unas guías de diseño depuradas, Material Design de Google, a las cuales los usuarios están acostumbrados. Esto quiere decir que tus aplicaciones pueden ser más sencillas de usar y resultar más familiares a simple vista. Sobre todo los usuarios de Android estarán muy familiarizados con las guías de diseño "Material", ya que son las usadas de las aplicaciones modernas de Android.

En relación a su peso en Bytes son todos más o menos similares. Materialize ocupa unos 140 KB con su CSS más 180 KB con el JS. Es prácticamente el mismo peso de Bootstrap.