

Introducción al Desarrollo de Sistemas

[Área personal](#) / [Mis cursos](#) / [IDS](#) / [Tema 4 - Reglas de Estilo CSS \(Cascade Style Sheets \)](#)
/ [Práctica: Introducción al manejo de Frameworks CSS](#)

Práctica: Introducción al manejo de Frameworks CSS

Un framework CSS, es uno ó más archivos .CSS que definen un conjunto de clases con un lineamiento de estilo determinado, que sigue un conjunto de políticas de diseño estético en común y que proveen un esquema de reúso para la construcción de interfaces gráficas complejas.

¿Porqué existen?

Porque si bien en nuestro ejemplo de la calculadora, el estilo visual se va conformando por definir la clase correspondiente para cada sector, imagínate el trabajo que puede resultar para una aplicación grande. ¿Sería considerable? ¿No es así?. Los frameworks están pensados para **agilizar el desarrollo de interfaces gráficas**, reducir el tiempo, la probabilidad de error del programador/diseñador y manteniendo coherencia estética.

Sin embargo, no están exentos de **ventajas y desventajas**. Recuerden siempre que no existen herramientas mágicas, **todas introducen un costo y un beneficio**, lo importante siempre para desarrollar software es **entrenar nuestro criterio** para determinar dichos costos, y saber balancear qué adoptar en cada caso, asumiendo la responsabilidad de la decisión.

Balance

Ventajas

- Proporcionar una forma fácil y rápida de implementar interfaces gráficas.
- Aseguran alto grado de compatibilidad visual entre diferentes navegadores Web. (Portabilidad)
- Posee fiabilidad, se asume que el framework de calidad, está bien probado y nos asegura que no posee errores.
- Coherencia estética

Desventajas

- Existen prácticamente tantos frameworks de CSS como programadores de interfaces gráficas.
- Algunos son intrusivos, y la adopción de los mismos obliga al programador a seguir convenciones y documentación rigurosa.
- Muchos de ellos reinventan funcionalidades que ya existen, dado que algunos fueron desarrollados cuando la tecnología Web no estaba en un alto nivel de madurez.
- Algunos tienen adaptación a diferentes tamaños de pantalla (Diseños Responsivos/Responsive).
- Algunos incluyen ecosistemas de código y dependencias muy costosas en términos de performance.
- Las migraciones entre versiones de frameworks, son costosísimas.
- Las migraciones entre diferentes frameworks es inviable.
- Si un framework deja de dar soporte, tu interfaz gráfica muere con el framework.
- No es posible modificar un framework, se lo usa tal como viene de fábrica adhiriendo reglas.
- Muchas veces están sobredimensionados en funcionalidades que no necesitamos y que no se pueden remover.

Hasta aquí, podríamos decir que parecen más las desventajas que las ventajas. En realidad, sí, la mayoría aportan más problemas que soluciones a la larga.

Generalmente es la parte más volátil del software, la que más rápido se descarta y se hace obsolescente. Sin embargo, la ventaja de la coherencia visual y la agilización al no crear toda la infraestructura base, que es muy costosa, balancea el promedio.

Tipos de frameworks CSS

En reglas generales, lo que hoy existe en el mercado son en principio, frameworks CSS de **pago y gratuitos**.

No necesariamente los de pago son mejores que los gratuitos, a veces los frameworks pagos se paga el soporte, en otros casos, sí, proveen características que no se encuentran en los frameworks gratuitos, ó tienen un diseño estético mucho más desarrollado con mucha más carga de trabajo detrás.

Separando esta primera línea de frameworks, existen en líneas generales las siguientes ramas:

Frameworks CSS puros

Estos son los frameworks CSS menos intrusivos, y los más "nobles". Dado que el programador sólo debe utilizar los nombres de clases adecuados en los elementos gráficos que necesita, y no mucho más. La ventaja es que suelen ser muy livianos, algunos incluso son responsivos, son eficientes. La desventaja es que tienen limitaciones cuando uno necesita construir elementos gráficos personalizados, no suelen poseer animaciones, no son las mejores en términos de diseño estético.

Generalmente se pueden utilizar cuando los sistemas no tienen requerimientos fuertes de diseño gráfico, sistemas más del tipo funcional.

Ejemplos:

W3.CSS : <https://www.w3schools.com/w3css/default.asp>

Bulma Framework: <https://bulma.io/>

PureCSSFramework: <http://www.purecssframework.com/>

Frameworks CSS/JS

Estos frameworks, refuerzan las limitaciones de cuestiones que no pueden resolverse fácilmente con CSS. En este caso JavaScript colabora con CSS para habilitar varias cosas. Generalmente se utilizan cuando los elementos gráficos estándar de HTML y las combinaciones entre ellos, no permiten obtener el resultado deseado. Un ejemplo muy común es un cargador de archivos múltiples. El estándar de HTML solo permite un único archivo. Otro ejemplo, son tablas de datos. No es posible clicar en una columna y que las filas se ordenen de manera descendente/ascendente, etc. Este tipo de cuestiones necesitan colaboración de código JS. Se utilizan estos frameworks cuando el programador no quiere perder tiempo en diseñar estos componentes. La desventaja es que al introducir código de programación, éste empieza a poner restricciones al código propio condicionándolo en el estilo.

Ejemplos:

Bootstrap: <https://getbootstrap.com/>

Foundation: <https://foundation.zurb.com/>

Uikit: <https://getuikit.com/>

Frameworks CSS/JS + Herramientas Externas complementarias

Resuelve limitaciones de los frameworks híbridos. Generalmente una de las falencias es la modificación dinámica del CSS. ¿Alguna vez has visto páginas que permiten visualizarse en modo oscuro, claro, u otra combinación, de manera rápida clickeando en un botón? Esas variaciones de valores de propiedades CSS, se gestionan con dependencias externas que permiten generar dinamismo visual.

Ejemplos:

Vue.js: <https://vuejs.org/>

AlpineJS/Talwind: <https://github.com/alpinejs/alpine>

Frameworks CSS / Frameworks JS + Herramientas Externas Complementarias

A todo esto, siempre quedan requerimientos fuera. Cuando además de todas las necesidades visuales, se requieren comunicaciones

entre elementos gráficos, respuestas visuales controladas por eventos, evitar recargas de código HTML/CSS, y además se persigue una línea de diseño siguiendo patrones de diseño puntuales, se introducen estos ecosistemas de herramientas. La ventaja es que como son muy adoptados, tienen comunidades muy grandes detrás, sin embargo son complejos, se necesita un estudio de documentación enorme para tener cierta productividad. Ya no solo a este nivel, toda la aplicación visual queda aferrada al diseño del framework, sino que hasta pueden obligar a escribir en un pseudo-código de programación nuevo no estandarizado.

Ejemplos:

Angular: <https://angular.io/>

React: <https://es.reactjs.org/>

Actividad

Para nuestra actividad, sólo nos limitaremos a una parte mínima de un framework CSS puro. En nuestro caso vamos a elegir W3.CSS, que tiene la curva de aprendizaje más simple, y que bien utilizado da resultados estéticos muy aceptables.

Recurso: <https://www.w3schools.com/w3css/default.asp>

Analiza el código de ejemplo del formulario construido con W3.CSS:

https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_examples_login

A partir de éste, convierte el último formulario Web (POST) del tema anterior, y adáptalo para que tenga una estética similar a la siguiente:

Create Account

First name *

Last name

Email*

Password *

Confirm Password *

Gender:

☐ Male ☐ Female

Hobbies

☐ Music ☐ Sports ☐ Travel ☐ Movies

Source of Income

Employed

Income

20K

Upload Profile Picture

Choose File No file chosen

Age

18

Bio



En batan.coop educación consideramos primordial la agilidad del proceso comunicativo, cualquier inquietud, duda y aporte que quieras hacernos llegar será motivo de crecimiento para nosotros y mejorará el modo en que podemos ayudarte a través de este servicio.

INFO

[Batan.Coop](#)
[Ecolan Soluciones Tecnológicas](#)
[Servicio Educativo Caraludmé](#)

CONTÁCTANOS

Julián Ríos 4215, Batán, Buenos Aires, Argentina. Julián Ríos 4215. Código Postal: 7601
 ☎ Phone : (0223) 464-3000
 ✉ E-mail : info@cooperativabatan.com.ar

GET SOCIAL



Create

Sumario de calificaciones

Re	No mostrado a los estudiantes	No
De	Participantes	23

VER/CALIFICAR TODAS LAS ENTREGAS

Calificación

◀ Práctica : Propiedades CSS

Ir a...

Resolución parcial de actividad -
 Propiedades CSS (7/8/2020) ▶