1-¿Qué es Vanilla Js?

2-Buscar 3 FrameWorks de Javascript que se usen en la actualidad

**¿Qué es Vanilla JS?**

Vanilla JS es una broma usada por los programadores y creada por Eric Wastl. Si descargas la librería vanilla.js de la web oficial y la abres, verás que es un fichero vacío.

Por lo tanto, todo lo que “está programado” con vanillaJS, realmente ha sido programado con JavaScript puro y si se elimina el fichero vanilla.js, funcionará igual.

¿De dónde viene Vanilla JS?

El mundo va por tendencias y la programación no iba a ser diferente. Por lo tanto, cuando se pone de moda un framework JavaScript, muchos programadores van corriendo a aprenderlo y usarlo para todo.

El problema es que hay muchos programadores (yo incluído) que consideran que se nos ha ido de las manos, y que se están cogiendo “malas costumbres”. Sobre todo, los nuevos programadores.

Es decir, ha llegado un momento en el que los programadores solamente usan frameworks javascript, sin llegar realmente a saber cómo funciona javascript, ni cómo se programa javascript puro (sin frameworks).

Cada vez los frameworks son más sofisticados, más grandes y cada vez se parecen menos a JavaScript. De hecho, algunos directamente tienen un pseudolenguaje que tienes que compilar, para lo que tienes que tener instalado nodeJS, npm, webpack, etc.

El resultado de ello es que para hacer web con un “hola mundo” tenemos un proyecto que antes de compilar ocupa 230Mb, cuanto con JavaScript puro serían 3kb. Es decir, 76.000 veces más.

Es por ello que, Eric Wastl creó un framework llamado Vanilla JS y la documentación del framework no era más que la oficial de JavaScript.

Por lo que muchos programadores (juniors), empezaron a descargarlo y hacer proyectos en Vanilla JS. Únicamente porque pensaban que era un nuevo framework, sin saber realmente que estaban programando JavaScript puro.

VanillaJS VS frameworks

Entonces, ¿es mala idea usar un framework?

No, los frameworks son geniales para proyectos con un tamaño medio/medio-grande, te permiten ordenar el código y trabajar de una forma ordenada, el problema que intenta hacer relucir VanillaJS son dos:

Primero, no deberías programar bajo un framework javascript si no sabes primero javascript puro.

Y segundo, no siempre se debe de utilizar un framework. Si el proyecto es enano, podemos no usar ningún framework o apoyarnos en librerías como jQuery y/o VueJS.

Por último, es importante recordar que, una web que usa un framework siempre será mucho más lenta (aunque el humano no lo note), que una web que no lo usa.

Por ejemplo, jQuery que es una de las librerías (no llega ni a framework) menos pesada, tarda 400 veces más ejecutar una consulta para recuperar todas las etiquetas html con un class concreto, que si lo hiciéramos con javascript puro, siendo ambos una sola línea de código.

**Frameworks de JS**

Angular

Gestado por Google, [Angular](https://angular.io/) es uno de los frameworks de JS más potentes y la mejor opción cuando se pretende originar **sitios web de una sola página con elementos interactivos.** Entre sus funcionalidades destaca la posibilidad de actualización en tiempo real desde diferentes dispositivos. Es decir, cualquier diseño se cambia de manera simultánea en la web y en la aplicación.

Angular aporta todo lo necesario para adquirir un buen conocimiento de arquitectura por componentes y es una herramienta muy importante a nivel laboral. Una de sus principales ventajas es que **una vez dominado, es más sencillo aprender otros**frameworks, algo que no sucede a la inversa. Por eso, es una herramienta básica para los desarrolladores.

React

No es propiamente un framework, sino una biblioteca que, en este caso, ha sido implementada por Facebook. [React](https://es.reactjs.org/) permite generar **aplicaciones web muy intuitivas**, ofreciendo un marco de trabajo versátil, robusto y muy fiable.

Además de Facebook, también las aplicaciones de **Instagram o Airbnb** están realizadas con esta tecnología.

Vue.js

Se caracteriza por apoyarse en una interfaz de programación de aplicaciones muy simple que, sin embargo, permite desarrollar **elementos dinámicos de interconexiones realmente sofisticadas.** La principal característica de [Vue.js](https://vuejs.org/) es la posibilidad de seleccionar los módulos que al programador le interesan y descartar los demás.

Es una alternativa muy válida a React o Angular y se espera que en algún momento supere a sus competidores en el mercado. La empresa japonesa **Nintendo**emplea Vue en varios de sus sitios web oficiales.