

¿Es mejor utilizar JavaScript y CSS sin librerías/frameworks, o estas herramientas ofrecen una mejor experiencia para el programador y el usuario final?

Tras analizar diversas fuentes y reflexionar sobre la experiencia práctica en el desarrollo web, estoy completamente a favor del uso de librerías y frameworks en el frontend. Considero que estas herramientas no solo optimizan el proceso de programación, sino que también mejoran significativamente la experiencia del usuario final.

¿Por qué son importantes estas herramientas?

En la actualidad, los sitios web requieren ser dinámicos, visualmente atractivos, responsivos y funcionales en múltiples navegadores y dispositivos. Hacer todo esto desde cero con solo JavaScript y CSS puro es posible, pero consume más tiempo, requiere mayor esfuerzo y complica la mantenibilidad del código a largo plazo. Por eso existen librerías y frameworks, cuyo propósito es agilizar el desarrollo sin sacrificar la calidad.

Librerías y frameworks investigados

jQuery

Fue una de las primeras librerías populares de JavaScript. Simplifica operaciones como la manipulación del DOM, eventos, animaciones y AJAX. Aunque su uso ha disminuido con la llegada de JavaScript moderno, sigue siendo útil en proyectos heredados o cuando se necesita compatibilidad con navegadores antiguos.

Bootstrap

Es un framework CSS que incluye estilos y componentes prediseñados como botones, formularios y menús de navegación. Facilita el diseño de interfaces atractivas y adaptables, ahorrando tiempo de maquetación y asegurando consistencia visual. Su sistema de cuadrícula (grid) es especialmente útil para lograr diseños responsivos rápidamente.

React

Esta librería, desarrollada por Meta (antes Facebook), permite crear interfaces dinámicas mediante el uso de componentes reutilizables. Utiliza un modelo de DOM virtual que mejora el rendimiento y tiene una comunidad muy activa. Es ampliamente utilizada en aplicaciones web modernas, desde redes sociales hasta plataformas de comercio electrónico.

Mi postura: a favor del uso de librerías y frameworks

Desde mi punto de vista, trabajar sin estas herramientas sería como construir un edificio sin maquinaria: se puede, pero no es lo más eficiente ni recomendable en proyectos reales. Las librerías y frameworks:

- Promueven buenas prácticas de desarrollo.

- Reducen la cantidad de código repetitivo.
- Mejoran la legibilidad y organización del proyecto.
- Aumentan la velocidad de desarrollo.
- Mejoran la escalabilidad y mantenibilidad.

Además, la mayoría de las empresas y proyectos actuales requieren experiencia con estas tecnologías, por lo que usarlas también incrementa nuestras oportunidades en el ámbito profesional.

Reflexión final

Las herramientas como jQuery, Bootstrap, React o Vue.js no sustituyen el conocimiento de HTML, CSS y JavaScript, sino que lo complementan y lo potencian. Usarlas no es hacer “trampa” ni tomar el camino fácil, sino aprovechar los avances tecnológicos para entregar mejores productos y soluciones.

Referencias (formato APA):

- Bootstrap. (s.f.). *Introduction*. Bootstrap. <https://getbootstrap.com/docs/5.3/getting-started/introduction/>
- jQuery Foundation. (s.f.). *jQuery*. <https://jquery.com/>
- Meta. (2024). *React – A JavaScript library for building user interfaces*. <https://react.dev/>
- Mozilla Developer Network. (2024). *JavaScript frameworks overview*. [https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Client-side JavaScript frameworks](https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks)
- Vue.js. (s.f.). *The Progressive JavaScript Framework*. <https://vuejs.org/>

PROPUESTA 2:

Título sugerido: *¿Vale la pena usar frameworks y librerías en el desarrollo web?*

Hola a todos.

Quiero compartir mi punto de vista sobre esta pregunta que me parece muy interesante: ¿Es mejor usar solo JavaScript y CSS puro, o conviene apoyarnos en librerías y frameworks como React, Bootstrap o jQuery?

En mi opinión, sí vale mucho la pena usar estas herramientas, ya que en proyectos reales hacen que el desarrollo sea más rápido, más organizado y que el resultado final sea más funcional y atractivo para los usuarios. Claro que es importante saber bien JavaScript y CSS desde cero, pero usar frameworks no significa que uno no sepa programar, sino que aprovecha lo que ya está hecho para enfocarse en resolver problemas más grandes.

Durante esta actividad investigué un poco más a fondo algunas herramientas muy conocidas:

jQuery:

Aunque ya no se usa tanto como antes, jQuery fue muy útil porque resolvía problemas de compatibilidad entre navegadores y hacía que cosas como manejar eventos o modificar el DOM fueran mucho más sencillas. Todavía se encuentra en muchos proyectos antiguos.

Bootstrap:

Es muy útil para el diseño visual. Con su sistema de grid y sus componentes ya hechos, puedes armar interfaces limpias y responsivas sin tener que escribir todo desde cero. A mí me gusta porque ayuda a mantener coherencia en el diseño y ahorra tiempo.

React:

Esta es una de las herramientas más usadas hoy en día. Está hecha por Meta (antes Facebook) y sirve para crear interfaces dinámicas con componentes reutilizables. Es más compleja al principio, pero te permite hacer aplicaciones muy completas y rápidas.

Después de revisar estas herramientas, me quedó claro que **usar frameworks y librerías no solo beneficia al programador, sino también al usuario final**, porque ayudan a crear productos más modernos, accesibles y eficientes. Además, el mercado laboral también lo exige; hoy en día muchas empresas piden experiencia en estas tecnologías.

Para mí, lo ideal es tener una base sólida en HTML, CSS y JavaScript, pero también **saber cuándo usar una herramienta que te va a ahorrar tiempo y esfuerzo sin perder calidad**. Así como en la vida usamos calculadoras, editores de texto o apps para organizarnos, en el desarrollo web tiene sentido usar herramientas que nos faciliten las cosas.

Fuentes consultadas:

- Bootstrap. (s.f.). *Introduction*. <https://getbootstrap.com/docs/5.3/getting-started/introduction/>
- jQuery Foundation. (s.f.). *jQuery*. <https://jquery.com/>

- React. (2024). *A JavaScript library for building user interfaces*. <https://react.dev/>
- Vue.js. (s.f.). *The Progressive JavaScript Framework*. <https://vuejs.org/>
- Mozilla Developer Network. (2024). *Client-side JavaScript frameworks*. https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks

Quedo atento a leer sus opiniones, comparar ideas y seguir aprendiendo entre todos.
¡Saludos!