

Frontend: React y Tailwind

En el desarrollo del frontend tuve que revisar distintas opciones de frameworks y formas de trabajar con estilos. Después de comparar, llegué a la conclusión de que la combinación de React con Tailwind CSS es la que mejor se adapta a lo que necesitamos en el proyecto.

React frente a otras alternativas

Lo primero que me llamó la atención de React es la manera en que permite organizar la interfaz en componentes. Esto hace que el código no se vuelva tan enredado y que las partes de la aplicación se puedan reutilizar en distintos lugares. A diferencia de otros frameworks más cerrados, React tiene un ecosistema enorme y un montón de librerías que facilitan cosas como la navegación o el manejo de datos desde APIs.

Otra cosa importante es el **manejo de estados**. Con los hooks, por ejemplo **useState** o **useEffect**, se puede actualizar la información en pantalla de forma rápida y sin necesidad de recargar todo. Además, el uso del Virtual DOM hace que la aplicación sea más eficiente porque solo actualiza lo que realmente cambia.

Tailwind frente a otros sistemas de estilos

En cuanto al diseño, probé distintos enfoques, pero **Tailwind** fue el que más me convenció. La forma en que trabaja con **clases utilitarias** es muy práctica: en lugar de tener que escribir hojas de estilo largas, uno va aplicando los estilos directamente sobre el HTML o JSX. Esto al principio parece raro, pero ahorra mucho tiempo.

Otra ventaja es que el diseño responsivo se vuelve más fácil. Con simples prefijos como **sm:** o **lg:** se pueden adaptar los componentes a distintas pantallas sin complicarse con media queries tradicionales. También noté que ayuda a mantener un estilo uniforme, porque todo parte de una misma configuración de colores, tipografías y tamaños.

Conclusión

Lo que aprendí con este trabajo es que el frontend no es solo “hacer que se vea bonito”, sino lograr que la aplicación sea interactiva, clara y fácil de mantener. React aporta la lógica y la organización, mientras que Tailwind ayuda a construir un diseño limpio y adaptable sin perder tiempo en detalles repetitivos. Al final, esta combinación me parece la más completa para proyectos que necesitan crecer y mantenerse en el tiempo.

