

BUENAS PRÁCTICAS EN REACT

1. Diseña componentes simples, reusables y de única responsabilidad

ReactJS es muy flexible y los desarrolladores pueden crear los componentes que quieran, pero esa flexibilidad se puede descontrolar, y jugar en contra de los principios de ingeniería aplicados al código.

Antes de tirar código piensa qué necesidades funcionales debes suplir y divide el problema en múltiples responsabilidades. Posiblemente, ese componente será reusable a futuro, entonces desarróllalo entendible, intuitivo y fácil de evolucionar.

2. Identifica similitudes, reescribe código y sé concreto

No repitas código. Cuando el código va creciendo, gracias a nuevas funcionalidades, el programador debe identificar constantemente patrones que le permitan simplificar instrucciones.

3. Estructura ordenadamente tus carpetas para que cualquiera entienda el proyecto

Trata de que todos los archivos relacionados a un componente estén en una misma carpeta, nombrada de manera entendible según su responsabilidad en el código.

Pensá que tu proyecto debe estar listo para que cualquier desarrollador pueda entender la relación de los componentes, viendo la estructura de carpetas y asociandola con la interfaz del proyecto.

4. Usa los React Hooks

Las últimas versiones de React (a partir de la React 16.8 Hooks : 90% código más conciso) incluyeron el concepto de los React Hooks, que facilitan el manejo organizado de estados dentro de los componentes.

5. Ubica los estados, variables y funciones según el contexto

Como desarrollador, constantemente debes pensar en preguntas como: ¿de quién es la responsabilidad funcional? ¿qué componente general ordena los estados principales de la app? ¿mi código es autoexplicativo y claro para saber dónde está cada funcionalidad?

Todas estas preguntas te van a ayudar a responder los problemas de contexto de las variables, estados y métodos.

6. Importa lo necesario y nombra adecuadamente los componentes

La performance y mantenibilidad del código dependen de las decisiones que se van tomando de desarrollo desde que se empieza a diseñar el código. Piensa lo mínimo que necesitas para que el componente desarrolle su función e importa solo lo que uses.

7. Revisa y soluciona constantemente los warnings de la consola

Al correr tu proyecto local React muestra en consola, en tiempo de ejecución, advertencias o warnings mientras desarrollas tu código. Dichas advertencias no impiden que la aplicación siga ejecutándose, porque aún no son errores, simplemente son reflejo de que tu código es propenso a futuros errores. Si se da solución a las advertencias, tendrás un código más limpio, profesional, con mejor performance y más mantenible.

BIBLIOGRAFIA

<https://es.linkedin.com/pulse/7-buenas-pr%C3%A1cticas-para-programar-en-reactjs-maradei-recruiting->