React - R0

Tomas.S.Alsina

E.E.S.T.N°5

7°4ta Informatica

Martin Estanga

5/8/24

INDICE

CUESTIONARIO

Instalar Node.js

Instalar VSCode con extención react VSC "React snippets"

Buscar diferencias entre reacthtml, reactcss, etc, y su estructura

Y armar un listado de buenas normas de programación en react

RESPUESTAS

1. ReactHTML (JSX):

- Utiliza JSX para escribir HTML dentro de JavaScript
- JSX se transpila a llamadas a `React.createElement`
- Mejora la legibilidad y organización del código

2. ReactCSS:

- **Estilos en línea:** Aplicar estilos directamente en los componentes mediante objetos de JavaScript
- **CSS Modules:** Estilos encapsulados y reutilizables
- **Styledcomponents:** Escribir estilos utilizando JavaScript con la biblioteca `styledcomponents`, permitiendo el uso de estilos en componentes de manera encapsulada

3. ReactJS (JavaScript):

- Utiliza JavaScript para manejar la lógica y la interacción en los componentes
- Los métodos y funciones de JavaScript se utilizan para manipular el estado y las propiedades de los componentes

4. React State Management:

- **Estado Local:** Manejado dentro de un componente individual

- **Context API:** Manejo del estado global sin pasar props manualmente
- **Bibliotecas de Gestión del Estado:** Uso de bibliotecas cmo Redux y MobX, para estados complejos

5. React Routing:

- Maneja la navegación y rutas en aplicaciones React
- Facilita la creación de aplicaciones de una sola página (SPA)

BUENAS NORMAS DE PROGRAMACIÓN DE REACT

1. Organización del Proyecto:

- Mantener una estructura de carpetas clara y coherente
- Separar los componentes en carpetas específicas

2. Componentes Funcionales y Hooks:

- Preferir componentes funcionales sobre componentes de clase
- Utilizar hooks para manejar el estado y los efectos secundarios

3. División de Componentes:

- Crear componentes pequeños y reutilizables
- Seguir el principio de responsabilidad única

4. Uso de PropTypes:

- Validar las props utilizando `PropTypes` para mejorar la robustez y legibilidad del código

5. Estilos:

- Preferir `CSS Modules` o `styledcomponents` para evitar conflictos de estilos
- Mantener la separación de preocupaciones entre lógica y presentación

6. Manejo del Estado:

- Mantener el estado lo más cerca posible de donde se necesita
- Utilizar `Context` para manejar el estado global cuando sea necesario

7. Estructura del JSX:

- Mantener el JSX limpio y legible
- Evitar la lógica compleja dentro del JSX, moverla a funciones auxiliares

8. Testing:

- Escribir pruebas para los componentes utilizando bibliotecas como `Jest` y `React Testing Library`
- Asegurarse de cubrir tanto pruebas unitarias como pruebas de integración

9. Accesibilidad:

- Asegurarse de que los componentes sean accesibles
- Utilizar etiquetas semánticas de HTML y atributos `aria` cuando sea necesario

10. Documentación:

- Documentar los componentes
- Mantener un README actualizado con instrucciones claras para desarrollar y desplegar la aplicación

BIBLIOGRAFIA

Chat GPT: chatgpt.com