Actividad:

Taller sobre componentes frontend

GA7-220501096-AA4-EV01

Aprendiz:

Wilmer Jair Espinosa Silva

CC: 1.095.910.391

Instructor:

ISRAEL ARBONA GUERRERO

Servicio Nacional de aprendizaje-SENA

Curso: TECNOLOGÍA EN ANÁLISIS Y DESARROLLO DE SOFTWARE

Ficha: 2455285

Con base en lo visto en el componente formativo “Desarrollo de frontend con React JS”, realizar un documento que responda los aspectos descritos a continuación:

1. Diferencia entre React y JSX

   RTA:React es una biblioteca JavaScript para construir aplicaciones web. JSX es una sintaxis utilizada en React para describir la estructura y contenido de la interfaz de usuario. Mientras que React es una librería, JSX es una herramienta que se utiliza dentro de React. Por lo tanto, JSX es una parte importante de React pero no es React en sí mismo.

   - React es una biblioteca JavaScript que permite a los desarrolladores crear aplicaciones web interactivas y dinámicas. React se centra en la construcción de componentes de interfaz de usuario, lo que significa que permite a los desarrolladores escribir componentes individuales que representan diferentes partes de la aplicación y luego combinarlos para crear la aplicación completa. React utiliza un enfoque basado en componentes para la construcción de aplicaciones, lo que significa que se divide en pequeñas piezas que se pueden reutilizar y manejar de forma independiente.
   - JSX, por otro lado, es una sintaxis utilizada en React para describir la estructura y contenido de la interfaz de usuario. En lugar de escribir HTML y JavaScript por separado, los desarrolladores pueden utilizar JSX para escribir ambos en un solo archivo. La sintaxis de JSX es similar a HTML y es una forma de describir la estructura y contenido de la interfaz de usuario. La principal ventaja de usar JSX es que permite una escritura más clara y concisa del código, y también permite a los desarrolladores trabajar con componentes de manera más intuitiva.
   - En resumen, React es una biblioteca JavaScript para construir aplicaciones web, mientras que JSX es una sintaxis utilizada dentro de React para describir la estructura y contenido de la interfaz de usuario. Aunque JSX es una parte importante de React, son dos cosas diferentes y React puede utilizarse sin JSX, aunque esto es poco común.

2.¿Qué son clases en React?

RTA:Las clases en React son una forma de definir componentes de interfaz de usuario en React. Un componente de React puede ser una clase que extienda de la clase React.Component. Un componente de clase en React tiene su propio estado y métodos, y se puede renderizar en la pantalla. Al utilizar un componente de clase, los desarrolladores pueden tener un mayor control sobre la lógica y el estado de un componente, lo que permite la creación de aplicaciones más complejas.

Acontinuacion un ejemplo de un componente de clase en React:

```
class ExampleComponent extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      message: 'Hello, world!'
    };
  }
  render() {
    return (
      <div>
        {this.state.message}
      </div>
    );
  }
}
```

En este ejemplo, **ExampleComponent** es un componente de clase que extiende de **React.Component**. El componente tiene un constructor donde se inicializa su estado, y un método **render** que devuelve el HTML que se debe mostrar en la pantalla.

3.Principales eventos de React

RTA: Los eventos en React son acciones que ocurren en la aplicación, como un clic en un botón, una entrada en un formulario, etc. Algunos de los eventos más comunes en React son:

1. onClick: se activa cuando se hace clic en un elemento
2. onChange: se activa cuando el valor de un elemento de formulario cambia
3. onSubmit: se activa cuando se envía un formulario
4. onMouseEnter: se activa cuando el cursor del mouse entra en un elemento
5. onMouseLeave: se activa cuando el cursor del mouse sale de un elemento
6. onKeyDown: se activa cuando se presiona una tecla en un elemento
7. onFocus: se activa cuando un elemento obtiene el foco
8. onBlur: se activa cuando un elemento pierde el foco

Estos son solo algunos ejemplos de los eventos que se pueden utilizar en React. Hay muchos otros eventos disponibles, y los desarrolladores pueden crear sus propios eventos personalizados también. Los eventos se pueden manejar en React utilizando funciones de manejo de eventos, que se ejecutan cuando se activa un evento específico.

4. Mapa conceptual de React .

RTA: Un mapa conceptual de React puede incluir los siguientes elementos:

1. Virtual DOM: una representación en memoria de la estructura del documento que se renderiza en la pantalla. React utiliza el Virtual DOM para optimizar el rendimiento de la aplicación al actualizar solo las partes que necesitan cambiar.
2. Components: los componentes son los bloques de construcción de una aplicación en React. Un componente puede ser una clase o una función, y puede tener su propio estado y métodos.
3. JSX: una extensión de JavaScript que permite escribir HTML directamente en el código JavaScript. JSX se utiliza para escribir la estructura de un componente de React.

4. Props: las propiedades son argumentos que se pasan a un componente. Las propiedades se utilizan para pasar datos a un componente desde su componente padre.
5. State: el estado es un objeto que contiene los datos dinámicos de un componente. El estado se puede actualizar y re-renderizar el componente en consecuencia.
6. Lifecycle methods: los métodos de ciclo de vida son métodos que se ejecutan en diferentes momentos del ciclo de vida de un componente, como cuando se monta o se desmonta un componente.
7. Event handling: la manipulación de eventos es la forma en que React maneja acciones que ocurren en la aplicación, como clics en botones o entradas en formularios.
8. Redux: Redux es una librería de gestión de estado que se utiliza a menudo con React para manejar el estado global de una aplicación.