

TALLER SOBRE COMPONENTES FRONT-END
GA7-220501096-AA4-EV01

EDISSON ESTEVE RODRIGUEZ ORDOÑEZ
ANA MARÍA RUIZ SALCEDO
MOISES BUENAÑO TUBERQUIA

TECNÓLOGO EN ANÁLISIS Y DESARROLLO DE SOFTWARE

FICHA: 2547406

INSTRUCTOR
FEDERICO HIGUITA GARNICA

SENA

BOGOTA D.C.



ASPECTO 1: DIFERENCIA ENTRE REACT Y JSX

REACT:

React es una biblioteca de JavaScript desarrollada por Facebook para construir interfaces de usuario interactivas y reutilizables.

Proporciona una arquitectura basada en componentes que facilita la creación y el mantenimiento de aplicaciones web complejas.

React permite manipular la interfaz del usuario mediante el concepto de "reactividad", es decir, cuando los datos cambian, solo las partes afectadas de la interfaz se actualizan automáticamente.

Es independiente del navegador, lo que significa que se puede utilizar tanto en el lado del cliente como en el servidor.

JSX:

JSX es una extensión de sintaxis utilizada con React que permite escribir código JavaScript con sintaxis similar a HTML/XML.

JSX facilita la creación de componentes de React y mejora la legibilidad del código al combinar el código JavaScript con la estructura del árbol de componentes.

JSX debe ser transformado en JavaScript válido antes de que los navegadores puedan interpretarlo, y esto generalmente se realiza utilizando herramientas como Babel.

En resumen, React es la biblioteca principal utilizada para crear componentes y manejar la lógica de la interfaz de usuario, mientras que JSX es una extensión de sintaxis que ayuda a definir cómo se deben renderizar estos componentes en la interfaz de usuario.

ASPECTO 2: ¿QUÉ SON CLASES EN REACT?

En versiones anteriores de React, las clases eran la forma principal de definir componentes. Una clase de React es una forma de crear un componente utilizando la sintaxis de clases de JavaScript. Para definir un componente mediante una clase, debemos extender la clase `React.Component` y definir un método llamado `render()`, que devuelve el código JSX que queremos que se represente en la interfaz de usuario.

Ejemplo de un componente de React utilizando clases:

```
import React from 'react';

class MiComponente extends React.Component {
  render() {
    return <div>Hola, soy un componente de React!</div>;
  }
}

export default MiComponente;
```

En las versiones más recientes de React, se ha introducido una nueva forma de definir componentes utilizando funciones, conocida como "componentes funcionales". Los componentes funcionales se han vuelto más populares debido a su simplicidad y legibilidad, y es común usarlos en lugar de las clases para definir componentes en React. A partir de React 16.8, los hooks permiten que los componentes funcionales tengan estado y utilicen características previamente reservadas para las clases, lo que ha llevado a una mayor adopción de los componentes funcionales.

ASPECTO 3: PRINCIPALES EVENTOS DE REACT

En React, los eventos son manejadores de funciones que se utilizan para capturar interacciones del usuario con los elementos de la interfaz de usuario. Algunos de los principales eventos de React son:

onClick: Se activa cuando un elemento se hace clic.

onChange: Se activa cuando un valor de entrada cambia, como en un campo de texto.

onKeyPress: Se activa cuando se presiona una tecla mientras un elemento tiene el foco.

onSubmit: Se activa cuando se envía un formulario.

onMouseEnter: Se activa cuando el cursor del mouse entra en un elemento.

onMouseLeave: Se activa cuando el cursor del mouse sale de un elemento.

onFocus: Se activa cuando un elemento recibe el foco.

onBlur: Se activa cuando un elemento pierde el foco.

Hay muchos otros eventos disponibles en React para cubrir una amplia gama de interacciones. Los eventos se utilizan comúnmente para actualizar el estado de los componentes o invocar acciones adicionales en función de las acciones del usuario.

ASPECTO 4: MAPA CONCEPTUAL DE REACT

