

Taller sobre Componentes Frontend: React JS

Jorge Miller Gutierrez Ospina

SENA

ANALISIS Y DESARROLLO DE SOFTWARE – 2721520

Ivan Leonardo Medina Gomez

Septiembre 2024

Introducción

El desarrollo frontend es un aspecto fundamental en la creación de aplicaciones web modernas. Permite construir interfaces de usuario dinámicas, interactivas y eficientes. En este contexto, React JS se ha convertido en una de las bibliotecas más populares para el desarrollo de interfaces, gracias a su enfoque basado en componentes y su capacidad de crear aplicaciones rápidas y escalables.

Objetivo

El objetivo de este documento es proporcionar una explicación detallada sobre las principales características de React JS, una de las bibliotecas más utilizadas para el desarrollo frontend. Además, se busca clarificar conceptos esenciales que permitan un mejor entendimiento de cómo React gestiona las interfaces de usuario y cómo sus componentes facilitan el desarrollo de aplicaciones web modernas.

Características Principales del Framework

1. Diferencia entre React y J

“React” es una biblioteca de JavaScript diseñada para construir interfaces de usuario. Su característica principal es la creación de componentes reutilizables, lo que facilita el desarrollo de aplicaciones web escalables y dinámicas. React se enfoca en el manejo eficiente del DOM (Document Object Model), permitiendo actualizaciones rápidas y re-renderizaciones solo cuando es necesario.

Por otro lado, “JSX (JavaScript XML)” es una extensión de la sintaxis de JavaScript que permite escribir código similar a HTML directamente dentro de los archivos JavaScript. Aunque React puede funcionar sin JSX, este es un estándar en la comunidad, ya que mejora la legibilidad y la estructura del código.

Diferencias clave:

- React es la biblioteca principal, encargada de la lógica y la gestión de componentes.

- JSX es una sintaxis que permite escribir código HTML-like dentro de archivos JavaScript, lo cual se traduce a llamadas JavaScript nativas.

2. ¿Qué son las clases en React?

En React, los componentes pueden ser definidos como “funciones” o como “clases”. Los componentes de clase eran la única manera de crear componentes con estado hasta la introducción de los Hooks en la versión 16.8 de React.

Un “componente de clase” es una estructura que permite definir el estado interno y los ciclos de vida del componente, como ``componentDidMount``, ``componentDidUpdate`` y ``componentWillUnmount``. Estos métodos permiten manejar tareas específicas en cada fase del ciclo de vida de un componente.

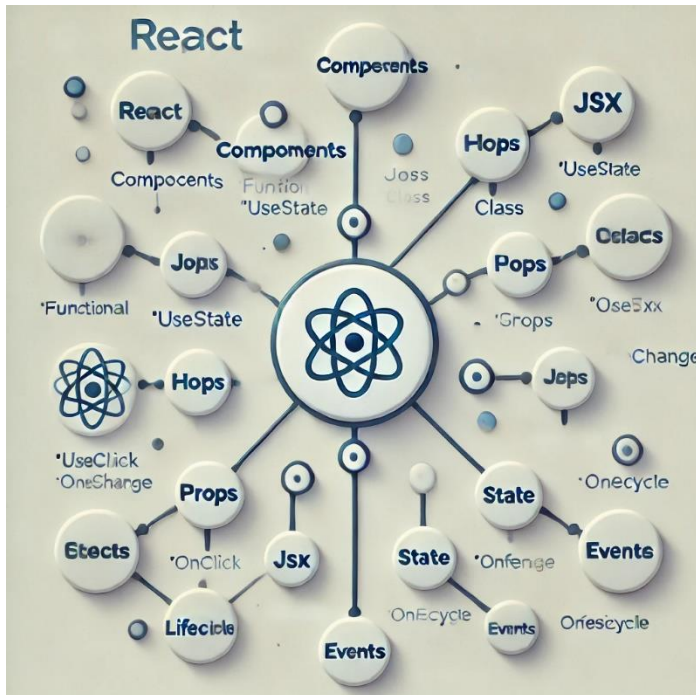
Con el uso de clases, se permite a los componentes almacenar estado interno y manejar eventos del ciclo de vida de manera más estructurada.

3. Principales eventos de React

React ofrece un sistema de manejo de eventos similar al de los navegadores, pero con algunas mejoras que facilitan la interoperabilidad entre navegadores y permiten una mayor eficiencia.

Estos eventos permiten capturar las interacciones del usuario con la interfaz y reaccionar a ellas de manera eficiente.

4. Mapa conceptual de React



Conclusión

React JS ha transformado el desarrollo de frontend al introducir un enfoque basado en componentes que simplifica la construcción y el mantenimiento de interfaces de usuario. La capacidad de React para manejar el DOM de manera eficiente, junto con el uso de JSX, clases, y eventos bien estructurados, permite crear aplicaciones web dinámicas y altamente interactivas. A medida que la tecnología web continúa evolucionando, React sigue siendo una herramienta clave para desarrolladores en todo el mundo.