



CONTEXTUALIZACIÓN REACT

**DAHIANA RODRIGUEZ
NICOL LINARES**

CONTENIDO

- 01 ¿Qué es react?
- 02 ¿Qué es un componente?
- 03 ¿Qué es JSX y para qué sirve?
- 04 ¿Cuál es la función de los estados en React?
- 05 ¿Qué son las props?
- 06 ¿Qué son las SPA y sus ventajas?
- 07 Diferencias entre vanilla, JavaScript y React

¿QUÉ ES REACT?

React es una biblioteca de JavaScript, creada por Facebook, que se utiliza para construir interfaces de usuario (UI) dinámicas y de alto rendimiento. Sirve principalmente para desarrollar la parte visual de las aplicaciones web, lo que permite crear experiencias de usuario fluidas, rápidas e interactivas.

CARACTERÍSTICAS

1. Basado en componentes: Las interfaces de usuario en React se construyen a partir de componentes, que son piezas de código independientes, reutilizables y aisladas.
2. Virtual DOM: Para mejorar el rendimiento, React utiliza una representación virtual del DOM (Document Object Model). En lugar de manipular el DOM real, que puede ser lento, React crea una copia ligera en memoria.
3. JSX: React utiliza una extensión de sintaxis de JavaScript llamada JSX, que permite escribir código similar a HTML directamente dentro de los archivos de JavaScript.
4. Declarativo: A diferencia de la programación imperativa, donde se describe paso a paso cómo cambiar la interfaz, React utiliza un enfoque declarativo.

¿QUÉ ES UN COMPONENTE?

En programación, y específicamente en React, un componente es una pieza de código independiente, reutilizable y aislada que define tanto la lógica como la estructura (la parte de la interfaz de usuario) de una parte específica de una aplicación.

Piensa en una interfaz de usuario como una casa:

- La casa entera sería la aplicación.
- Un componente sería cada uno de los elementos de la casa, como una puerta, una ventana, una silla o una pared.
- Al igual que una ventana puede ser usada en varias habitaciones, un componente puede ser reutilizado en diferentes partes de la aplicación.
- Un componente de puerta tendría su propia lógica (abre y cierra) y su propia estructura (el marco, el picaporte), sin afectar a la lógica o estructura de la ventana.

¿QUÉ ES JSX Y PARA QUÉ SIRVE?

JSX, que significa JavaScript XML, es una extensión de sintaxis para JavaScript. Permite escribir código similar a HTML directamente en un archivo de JavaScript. En el contexto de React, JSX es la forma más común y recomendada de describir la estructura de la interfaz de usuario de un componente.

```
// Sin JSX (método menos común)
const elemento = React.createElement(
  'h1',
  null,
  '¡Hola, mundo!'
);
```

```
// Con JSX (método recomendado y más legible)
const elementoJSX = <h1>¡Hola, mundo!</h1>;
```



¿CUÁL ES LA FUNCIÓN DE LOS ESTADOS EN REACT?

En React, la principal función de los estados es permitir a los componentes gestionar y almacenar datos dinámicos que pueden cambiar con el tiempo. Piénsalo como la "memoria interna" de un componente, la cual determina su comportamiento y la apariencia de la interfaz de usuario en un momento dado.

Cuando el estado de un componente cambia, React detecta automáticamente esta modificación y vuelve a renderizar solo los elementos afectados en el DOM virtual. Esto garantiza que la interfaz de usuario se actualice de forma eficiente y sincronizada con los datos más recientes.

¿QUÉ SON LAS PROPS?

En React, las props (abreviatura de "properties" o propiedades) son un mecanismo para pasar datos de un componente padre a un componente hijo. Sirven como argumentos o parámetros que se le pasan a un componente para que pueda renderizar información dinámica de forma flexible y reutilizable.

```
// La función recibe las props como argumento (un objeto)
function Tarjeta(props) {
  return (
    <div style={{ border: "1px solid #ccc", padding: "16px",
margin: "16px" }}>
      <h3>{props.titulo}</h3>
      <p>{props.descripcion}</p>
    </div>
  );
}
```




¿QUÉ SON LAS SPA Y SUS VENTAJAS?

Es una aplicación web que carga todos sus recursos (HTML, CSS y JavaScript) en una única carga inicial. A medida que el usuario interactúa con ella, el contenido se reescribe dinámicamente, en lugar de recargar la página completa, lo que crea una experiencia fluida, similar a la de una aplicación de escritorio.

DIFERENCIAS ENTRE VANILLA, JAVASCRIPT Y REACT

JavaScript es el lenguaje base.

Vanilla JavaScript es JavaScript puro, sin usar librerías ni frameworks.

React es una librería creada con JavaScript para construir interfaces de usuario de forma más rápida y organizada.

Vanilla JavaScript requiere más código para tareas complejas, mientras que React simplifica esas tareas.

React usa componentes y actualiza la interfaz automáticamente; Vanilla JavaScript no.

**MUCHAS
GRACIAS**

