



CONCEPTOS REACT

Keren Caldera
Nicol Madera

Contenido

01

Que es react

02

Que es un
componente

03

Que es JSX y
para que sirve

04

Cual es la
funcion de los
estados en react

05

Que son las
props

06

Que son las SPA
y sus ventajas

07

Diferencia entre
vanilla
javascript y
react

QUE ES REACT

React es una biblioteca de JavaScript (no un framework completo) que se utiliza para construir interfaces de usuario (UI), especialmente para aplicaciones web de una sola página (Single Page Applications o SPAs).

Fue desarrollada por Facebook (ahora Meta) y lanzada en 2013. Su principal objetivo es facilitar la creación de interfaces interactivas y eficientes, trabajando con una estructura basada en componentes reutilizables.



¿Por qué usar React?


Es rápido y eficiente gracias al Virtual DOM.

Favorece el desarrollo modular y reutilizable.

Tiene una gran comunidad y ecosistema.

Se usa en muchas empresas grandes (Meta, Netflix, Airbnb, etc.)

jsx

 Copiar código

```
function Saludo() {  
  return <h1>Hola, mundo</h1>;  
}
```

QUE ES UN COMPONENTE

Un componente de React es una pieza reutilizable y autónoma de la interfaz de usuario que React utiliza para construir aplicaciones web. Básicamente, un componente define cómo se debe ver una parte de la UI y cómo debe comportarse.

Características principales:

- Reutilizable: Puedes usar un mismo componente varias veces en diferentes partes de la aplicación.
- Autónomo: Cada componente maneja su propia lógica y estructura.
- Declarativo: Describe qué debe mostrarse en la pantalla basado en los datos (props y estado).

jsx

```
function Saludo() {  
  return <h1>Hola, mundo!</h1>;  
}
```

Componentes funcionales: Son funciones de JavaScript que retornan JSX (el código similar a HTML que React usa).

jsx

```
class Saludo extends React.Component {  
  render() {  
    return <h1>Hola, mundo!</h1>;  
  }  
}
```

Componentes de clase: Son clases de JavaScript que extienden React.Component y tienen un método render().

¿QUÉ ES JSX ?

¿Qué es JSX?

JSX (JavaScript XML) es una extensión de sintaxis para JavaScript que permite escribir código que se parece mucho a HTML dentro de los archivos JavaScript. Es una forma más intuitiva y declarativa de definir la estructura de la interfaz en React.

En lugar de usar funciones o métodos complicados para crear elementos HTML, con JSX puedes escribir algo así:

jsx

```
const elemento = <h1>Hola, mundo!</h1>;
```

- ¿Para qué sirve JSX?
- Facilita la creación de interfaces: Permite definir la estructura visual de los componentes de React de forma clara y legible, muy parecido a escribir HTML.
- Mejora la productividad: Es más rápido y sencillo escribir y entender el código.
- Integra lógica y presentación: Puedes mezclar expresiones de JavaScript dentro de JSX usando llaves {}, lo que permite mostrar datos dinámicos fácilmente.

jsx

```
const nombre = "Ana";  
const saludo = <h1>Hola, {nombre}!</h1>;
```

CUÁL ES LA FUNCIÓN DE LOS ESTADOS EN REACT

¿Cuál es la función del estado en React?

- Guardar información dinámica: El estado permite que un componente "recuerde" información que puede cambiar, como el valor de un input, si un botón está activado, datos recibidos de una API, etc.

Controlar el comportamiento y la apariencia:

- Cuando el estado cambia, React automáticamente vuelve a renderizar el componente para reflejar esos cambios en la interfaz.

Gestionar la interactividad: Por ejemplo, mostrar/ocultar elementos, contar clics, actualizar listas, etc.

```
jsx

import React, { useState } from 'react';

function Contador() {
  // Declara un estado llamado "contador" con valor inicial 0
  const [contador, setContador] = useState(0);

  return (
    <div>
      <p>Has hecho clic {contador} veces</p>
      <button onClick={() => setContador(contador + 1)}>Haz clic</button>
    </div>
  );
}
```

Resumen rápido:

El estado permite que los componentes sean dinámicos y reactivos a la interacción del usuario o a eventos externos.

¿Qué son las props?

¿Qué son las props?

Props (abreviatura de "properties" o propiedades) son un mecanismo que React usa para pasar datos de un componente padre a un componente hijo. Son como los "parámetros" o "atributos" que le das a un componente para que se configure o muestre información específica.

Características de las props:

- Son inmutables dentro del componente que las recibe, es decir, no se pueden cambiar directamente.
- Permiten que un componente sea más reutilizable y configurable.
- Se pasan como un objeto al componente y pueden contener cualquier tipo de dato: números, textos, funciones, objetos, etc

jsx

```
function Saludo(props) {  
  return <h1>Hola, {props.nombre}!</h1>;  
}
```

```
// Usando el componente con una prop llamada "nombre"  
<Saludo nombre="Carlos" />
```

¿Para qué sirven?

Para personalizar componentes.
Para pasar datos y funciones entre componentes.
Para hacer que la UI sea dinámica y flexible.

¿Qué es una SPA?

¿Qué es una SPA?

SPA significa Single Page Application (Aplicación de Página Única). Es un tipo de aplicación web que carga una única página HTML y luego actualiza el contenido dinámicamente sin recargar toda la página cada vez que el usuario navega o interactúa.

En lugar de pedir una página nueva al servidor para cada acción (como en las aplicaciones tradicionales), las SPA usan JavaScript para modificar el contenido y cambiar la “vista” de forma rápida y fluida.

Ejemplos comunes de SPA:

- Gmail
- Facebook
- Twitter

Ventajas de las SPA

1.Velocidad y fluidez:

Como solo se carga una vez la página, las interacciones y cambios son instantáneos y no hay esperas por recargar todo el sitio.

2.Mejor experiencia de usuario:

La navegación se siente más natural y parecida a una aplicación de escritorio o móvil.

3.Menor carga en el servidor:

Solo se cargan datos necesarios mediante APIs, no toda la página, lo que reduce el tráfico y mejora la eficiencia.

Diferencia entre vanilla javascript y react

VANILLA javascript

1. ¿Qué es Vanilla JavaScript?

- Es JavaScript puro, sin librerías ni frameworks adicionales.
- Es el lenguaje base que se ejecuta directamente en el navegador.
- Para manipular el DOM (la estructura HTML de la página), debes usar métodos como `document.getElementById()`, `document.createElement()`, etc.
- Tú eres quien controla toda la lógica para actualizar la interfaz y gestionar eventos.

React

2. ¿Qué es React?

- Es una librería de JavaScript creada por Facebook para construir interfaces de usuario.
- Usa un enfoque declarativo y basado en componentes para construir la UI.
- Permite crear componentes reutilizables que manejan su propio estado y lógica.
- React actualiza automáticamente el DOM usando un sistema llamado Virtual DOM, lo que mejora el rendimiento.

Diferencias clave

Aspecto	Vanilla JavaScript	React
Manipulación DOM	Manual con métodos del DOM	React usa Virtual DOM para optimizar y abstraer cambios
Enfoque	Imperativo (decir cómo hacer todo)	Declarativo (decir qué UI debe mostrarse)
Componentes	No hay una estructura nativa	Basado en componentes reutilizables
Actualización UI	Debes actualizar el DOM tú mismo	React actualiza la UI automáticamente al cambiar el estado o props
Manejo de estado	Gestionado manualmente (variables, eventos)	React tiene hooks y estado integrado para manejar datos dinámicos
Curva de aprendizaje	Más simple para cosas básicas, pero complejo para apps grandes	Puede ser más complejo al principio, pero facilita apps grandes y mantenibles

¿Entonces cuándo usar cada uno?

Vanilla JS: Ideal para proyectos pequeños, scripts simples o cuando no quieres depender de librerías externas.

React: Mejor para aplicaciones grandes, interfaces dinámicas, y cuando quieres trabajar con componentes y aprovechar su ecosistema.



Conclusión

Resumen sencillo sobre React y conceptos clave

React es una herramienta para crear páginas web que funcionan rápido y se ven bien. Usa componentes, que son como bloques pequeños que puedes usar muchas veces para construir la página.

Con JSX puedes escribir código que parece HTML dentro de JavaScript, lo que facilita crear la interfaz. Los estados guardan información que puede cambiar, como cuando haces clic en un botón, y las props son datos que se pasan de un componente a otro para personalizarlos.

React es perfecto para hacer SPA (Aplicaciones de Página Única), que no recargan la página completa cada vez, por lo que la experiencia es más rápida y fluida.

A diferencia de usar solo JavaScript puro (Vanilla JS), React te ayuda a organizar mejor el código y actualizar la pantalla de manera automática y eficiente.



REFERENCIAS
CHATGPT
CLAUDE.IA





Gracias