

«Talento Tech»

React JS

Clase 02



Clase N° 2 | JSX y Componentes

Índice:

- Repaso de JSX y componentes funcionales.
 - Creación de componentes más complejos.
 - Bienvenida a TalentoLab
 - Situación Inicial en TalentoLab
-

Objetivos de la Clase:

- Comprender la importancia y las ventajas del uso de JSX.
- Construir componentes más complejos que utilicen estructuras dinámicas.
- Aprender a utilizar **props** para pasar datos entre componentes.
- Practicar la reutilización y personalización de componentes.

Repaso de JSX y componentes funcionales.



En la clase anterior vimos cómo JSX nos permite escribir código que combina HTML y JavaScript para describir la interfaz de usuario. JSX es un atajo elegante y más intuitivo que el uso directo de `React.createElement`.

Por ejemplo:

```
const Bienvenida = () => <h1>¡Hola, Mundo!</h1>;
```

Además, vimos que un **componente funcional** es una función de JavaScript que devuelve un bloque de JSX. Los componentes son las piezas fundamentales de cualquier aplicación React. Hoy daremos un paso más allá y exploraremos cómo construir componentes más complejos.

Creación de componentes más complejos

Un componente puede incluir lógica y múltiples elementos para formar interfaces más sofisticadas. Por ejemplo, un componente que lista usuarios:

```
function ListaUsuarios() {  
  const usuarios = ['Ana', 'Luis', 'Pedro', 'María'];  
  return (  
    <ul>  
      {usuarios.map(usuario => (  
        <li key={usuario}>{usuario}</li>  
      ))}  
    </ul>  
  );  
}
```

Aquí usamos la función `map` para recorrer un array y generar un elemento `` por cada usuario. Este ejemplo demuestra cómo combinar JSX con lógica de JavaScript para producir contenido dinámico.

Creación de componentes reutilizables con props

Las props permiten construir componentes genéricos que funcionan en diferentes contextos. Por ejemplo, un botón personalizado:

```
function Boton({ texto, color }) {  
  const estilo = { backgroundColor: color, color: 'white', padding:  
'10px', border: 'none' };  
  return <button style={estilo}>{texto}</button>;  
}
```

Al usar este componente:

```
function App() {  
  return (  
    <div>  
      <Boton texto="Aceptar" color="green" />  
      <Boton texto="Cancelar" color="red" />  
    </div>  
  );  
}
```

Con un solo componente, podemos generar botones con estilos y textos distintos.

Reflexión Final

En esta clase, exploramos el potencial de JSX y los componentes funcionales, comprendiendo cómo React combina lo mejor de HTML y JavaScript para construir interfaces dinámicas y reutilizables. Aprendimos que JSX simplifica el desarrollo al permitir una integración fluida entre estructura y lógica, haciendo que el código sea más claro y mantenible.

También introdujimos las **props**, una herramienta fundamental para personalizar y reutilizar componentes en diferentes contextos. Esto nos permitió crear interfaces más flexibles y adaptables, sentando las bases para el desarrollo de aplicaciones escalables y eficientes.

¡Les damos la bienvenida a TalentoLab!



En **TalentoLab**, convertimos ideas en herramientas digitales innovadoras y confiables, ofreciendo servicios de desarrollo con un enfoque en la calidad y eficiencia. Nuestro compromiso es desarrollar soluciones que optimicen procesos y potencien negocios, combinando creatividad, tecnología de punta y un compromiso absoluto con la satisfacción de nuestros clientes. Tu éxito es nuestra prioridad, y nuestras soluciones están diseñadas para marcar la diferencia.

Equipo TalentoLab:



Silvia
Product Owner



Luis
Diseñador UX UI



Matias
Desarrollador



Sabrina
Desarrolladora

Situación Inicial en TalentoLab.

¡Únete al equipo TalentoLab! 🚀

La familia de **TalentoLab** está buscando un nuevo integrante para colaborar en el desarrollo de un emocionante producto innovador. Buscamos personas con un perfil **proactivo**, **autodidacta e innovador**, apasionadas por el aprendizaje continuo y con ganas de afrontar grandes desafíos.



Hemos preparado una serie de **ejercicios prácticos** diseñados para evaluar tus habilidades técnicas. Estos retos nos permitirán conocerte mejor y comprobar cómo aplicas tus conocimientos en situaciones reales.

Ejercicio Práctico:

1. Crea un componente que reciba un array de productos como prop y los muestre en una lista ordenada.

Ejemplo de array:

```
const productos = ['Manzanas', 'Peras', 'Naranjas'];
```

2. Crea un componente **Tarjeta** que reciba props para mostrar un título, una descripción y un botón personalizado.

Ejemplo de uso:

```
<Tarjeta  
  
  titulo="Oferta especial"  
  
  descripcion="20% de descuento en todos los productos"  
  
  botonTexto="Ver más"  
  
/>
```


3. Usa el componente **Boton** que creamos para estilizar diferentes acciones en una página.
-

Material y Recursos Adicionales:

- ★ [Documentación oficial de React](#)
 - ★ [Guía sobre props en React](#)
-

Preguntas para Reflexionar:

- ¿Qué ventajas observas al usar props en componentes React?
 - ¿Cómo podrías reutilizar un componente en diferentes partes de una aplicación?
 - ¿Cuál es la diferencia entre un componente estático y uno dinámico que utiliza props?
-

Próximos Pasos:

- Creación de la estructura básica de la aplicación.
- Primeros componentes reutilizables (Header, Footer).
- Visualización de los componentes en el navegador.



Buenos Aires
aprende
Agencia de Políticas para el Futuro

BA Buenos
Aires
Ciudad