



Buenos
Aires
Ciudad

Agencia de Habilidades
para el Futuro

<Talento
Tech/>

ReactJS

Clase 04 | React Hooks y Eventos

¡Les damos la bienvenida!



Vamos a comenzar a grabar la clase.

Índice

Layout en React

- Creación de la estructura básica de la aplicación.
- Desarrollo de los primeros componentes reutilizables (Header, Footer, Nav, Main, Gallery).
- Visualización de los componentes en el navegador.
- TalentoLab

React Hooks y Eventos

- **Introducción al manejo del estado local con useState.**
- **Comprendión y manejo de eventos en React (clics, formularios).**
- **Proyecto final Talento Lab**

Uso de useEffect y Peticiones a APIs

- Introducción al manejo de efectos secundarios con useEffect.
- Realización de peticiones a APIs para cargar productos.
- Manejo del estado de carga y errores en aplicaciones React.

Objetivos de la Clase

- 1** Aprender a gestionar el estado local de un componente usando el hook **useState**.
- 2** Manejar eventos comunes en React, como clics y envío de formularios.
- 3** Crear una funcionalidad básica para agregar productos a un carrito de compras.

useState

Introducción al manejo del estado local con useState

¿Qué es useState?

Es un hook de React para manejar el estado local en componentes funcionales.

Fíncionamiento

Declara una variable de estado y una funci&on para actualizarla.

Ejemplo Básico de useState

```
import React, { useState } from 'react';

function Contador() {
  const [contador, setContador] = useState(0);

  return (
    <div>
      <p>Valor del contador: {contador}</p>
      <button onClick={() => setContador(contador +
1)}>Incrementar</button>
    </div>
  );
}

export default Contador;
```



Reglas Importantes de useState



1

No modificar directamente

Usa siempre la función proporcionada por useState para actualizar el estado.

2

No combina estados

Declara múltiples variables de estado por separado si es necesario.

3

Ejecución en cada renderizado

El componente se vuelve a renderizar cuando el estado cambia.

Eventos

Manejo de Eventos en React

Los eventos en React se basan en JavaScript, pero con particularidades para mayor consistencia y seguridad.

Sintaxis camelCase

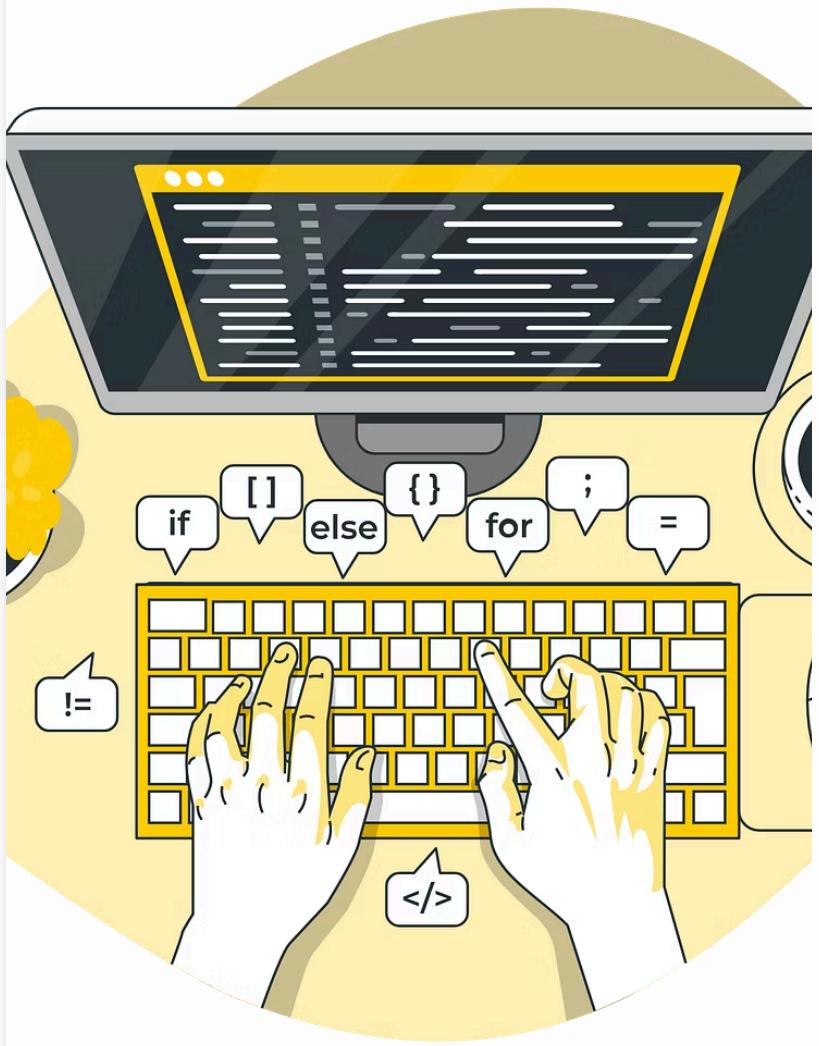
Ejemplo: onClick en lugar de onclick.

Manejadores como funciones

Se definen dentro del componente.

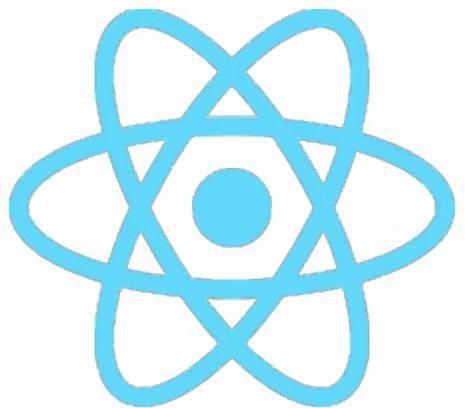
Delegación de eventos

Mejora el rendimiento al no adjuntar listeners directamente al DOM.



Ejemplo de Manejo de Clics

```
export default function Boton() {
  function manejarClick() {
    alert('Botón clickeado!');
  }
  return <button onClick={manejarClick}>Hacer
clic</button>;
}
```



Eventos Comunes en React

onClick

Se activa cuando un elemento es clickeado.

onChange

Detecta cambios en inputs de formularios.

onSubmit

Se dispara cuando un formulario es enviado.

onMouseOver / onMouseOut

Para manejar eventos del mouse.

Formularios

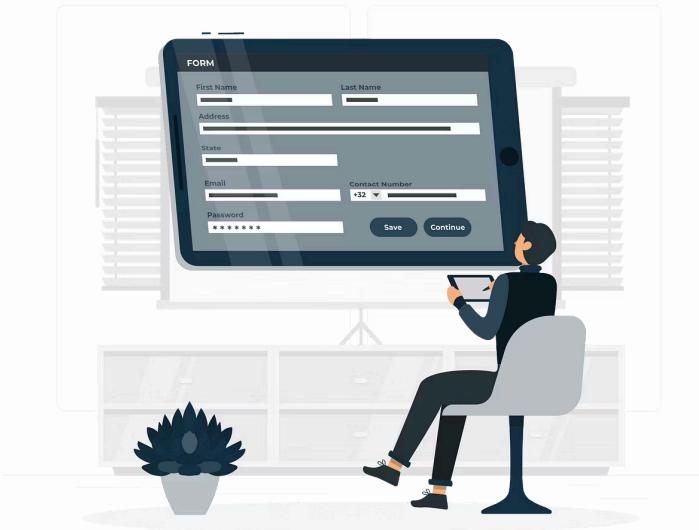
Formularios Controlados en React

¿Qué son?

Formularios donde el estado de los elementos se maneja a través de variables de estado.

Ventajas

- Facilita la validación en tiempo real
- Permite formatear datos antes de enviarlos
- Centraliza la lógica en el estado del componente



Ejemplo de Formulario Controlado

```
import React, { useState } from 'react';
export default function Formulario() {
  const [nombre, setNombre] = useState('');
  function manejarEnvio(evento) {
    evento.preventDefault();
    alert(`Formulario enviado por: ${nombre}`);
  }
  return (
    <form onSubmit={manejarEnvio}>
      <input
        type="text"
        value={nombre}
        onChange={(e) => setNombre(e.target.value)}
        placeholder="Ingresa tu nombre"
      />
      <button type="submit">Enviar</button>
    </form>
  );
}
```


Reflexión final

Aprendimos dos conceptos clave en React: la gestión del estado local con **useState** y el manejo de eventos, ambos esenciales para crear aplicaciones web dinámicas e interactivas.

La gestión del estado nos permitió almacenar y manipular datos dinámicos, mientras que el manejo de eventos nos ayudó a interactuar con el usuario de manera efectiva. Este enfoque modular y práctico fortalece las bases necesarias para avanzar hacia aplicaciones más sofisticadas.



Preguntas para Reflexionar



Ventajas de React

¿Qué beneficios ofrece React al manejar eventos comparado con JavaScript puro?

Aprendizajes de Formularios

¿Qué aprendiste al trabajar con formularios en React?
¿Qué desafíos enfrentaste?

Extensiones del Proyecto

¿Cómo podrías extender la aplicación para incluir funcionalidades como eliminar productos individuales del carrito?

Próximos Pasos

1

useEffect

Introducción al hook useEffect para manejar efectos secundarios.

2

Peticiones a APIs

Aprender a realizar peticiones para cargar productos desde una API.

3

Manejo de Estados Complejos

Gestionar estados de carga y errores en la aplicación.

Recursos Adicionales

1 [Documentación oficial de React - useState](#)

2 [React Events Handling - Documentación](#)

3 [Guía completa sobre formularios en React](#)

4 [Ejercicios prácticos sobre React Hooks](#)



Ejercicios Prácticos



Tu Primer Trabajo en TalentoLab

¡Felicitaciones! Tras haber realizado los ejercicios prácticos, has sido seleccionado para unirte a TalentoLab.

Serás parte del equipo responsable de crear un eCommerce para un cliente.



Ejercicio Práctico

Obligatorio



Descripción de tu tarea inicial:

Tu tarea inicia crear una aplicación que permita a los usuarios agregar productos a un carrito de compras y visualizar la lista de productos seleccionados.



Ejercicio Práctico

Obligatorio

Requisitos del Proyecto:

1 Listar Productos

Crear un componente para mostrar los productos disponibles.

2 Estado del Carrito

Usar useState para manejar el estado del carrito de compras.

3 Agregar Productos

Implementar un evento de clic para añadir productos al carrito.

4 Mostrar Carrito

Crear un componente para visualizar los productos seleccionados.

5 Crear un Layout del eCommerce

(Header, Footer, Main, Nav)



Ejercicio Práctico

Obligatorio

Pautas Generales

1

Información de Productos

La lista debe incluir al menos nombre y precio de cada producto.



2

Actualización Dinámica

El estado del carrito debe actualizarse al agregar cada producto.

3

Carrito Vacío

Mostrar un mensaje cuando no haya productos seleccionados.



Ejercicio Práctico

Obligatorio

Desafío Adicional

Botón "Vaciar Carrito"

Implementar una función para eliminar todos los productos seleccionados del carrito.



¡Nuevo cuestionario en Campus!

- (i) No olvides que los cuestionarios son de carácter obligatorio para poder avanzar con la cursada.