REACT JS DESDE CERO

SOMOS Y FORMAMOS EXPERTOS EN T.I

Metodología

100% PRACTICO

Duración

30 HRS.

ACERCA DEL CURSO

OBJETIVOS PRINCIPALES

- **Comprender los fundamentos de React.js, incluyendo componentes, JSX, estado y props, para construir interfaces de usuario interactivas y dinámicas.**
- **Aprender a utilizar los Hooks de React para gestionar el estado, efectos y otros aspectos en componentes funcionales.**
- **Aprender a realizar peticiones a APIs REST, manejar respuestas y mostrar datos en la interfaz de usuario.**
- **Dominar el enrutamiento en una aplicación React utilizando React Router para crear experiencias de navegación fluidas.**
- **Implementar autenticación de usuario utilizando OAuth, asegurando rutas protegidas y controlando el acceso a recursos.**
- **Crear formularios dinámicos y controlados, validar datos y enviarlos a APIs, manejando respuestas y errores.**
- **Aprender a manejar errores en peticiones a APIs y cómo brindar retroalimentación útil a los usuarios.**

OBJETIVOS SECUNDARIOS

- **Explorar patrones de diseño y buenas prácticas en el desarrollo de aplicaciones con React.**
- **Obtener una base sólida para explorar conceptos más avanzados de React en el futuro.**
- **Conocer la arquitectura de una aplicación web completa.**

PREREQUISITOS

- **Conocimientos básicos de HTML, CSS y JavaScript.**
- **Familiaridad con conceptos de programación orientada a objetos.**
- **Deseable pero no obligatorio: experiencia previa con algún framework o biblioteca de JavaScript.**
- **Se recomienda el curso Javascript Desde Cero**
  ¿Cuál es mi nivel en programación? clic aquí

¡NUNCA DEJES DE APRENDER!

1.- Introducción a React.js

1.1 ¿Qué es React.js? 1.2 Ventajas de usar React.js en el desarrollo web. 1.3 Componentes y su reutilización. 1.4 Primeros pasos con create-react-app.

2.- Fundamentos de JSX y Componentes

2.1 Introducción a JSX. 2.2 Creación y renderizado de componentes. 2.3 Props y su uso para pasar datos.

3.- Manejo de Estado y Ciclo de Vida

3.1 Uso del estado en componentes. 3.2 Ciclo de vida de los componentes de clase. 3.3 Actualización y montaje de componentes.

4.- Eventos y Manipulación de Datos

4.1 Manejo de eventos en React. 4.2 Actualización del estado y renderizado condicional.

5.- Listas y Keys

5.1 Renderizado de listas dinámicas. 5.2 Importancia de las keys. 5.3 Iteración y renderizado de componentes en listas.

6.- Formularios y Componentes Controlados

6.1 Creación de formularios en React. 6.2 Componentes controlados y no controlados. 6.3 Validación básica de formularios.

7.- Comunicación entre Componentes

7.1 Patrones de comunicación entre componentes. 7.2 Elevación de estado (lifting state up). 7.3 Context API para el manejo global del estado.

8.- Enrutamiento con React Router

8.1 Configuración de rutas con React Router. 8.2 Navegación entre componentes. 8.3 Paso de parámetros y rutas anidadas.

9.- Introducción a Hooks

9.1 ¿Qué son los Hooks? 9.2 Uso de useState y useEffect. 9.3 Conversión de componentes de clase a componentes funcionales.

10.- Estilización de Componentes

10.1 Uso de estilos en línea y clases de CSS. 10.2 Styled-components y su enfoque en los componentes. 10.3 Creación de componentes estilizados.

11.- Consumo de API REST

11.1 Introducción a las peticiones HTTP. 11.2 Uso de la función fetch para obtener datos. 11.3 Manejo de respuestas JSON. 11.4 Renderizado de datos de una API externa.

12.- Autenticación con OAuth

12.1 Conceptos básicos de OAuth. 12.2 Integración de OAuth en aplicaciones React. 12.3 Flujo de autenticación con OAuth 2.0. 12.4 Uso de bibliotecas como react-oauth para la autenticación.

13.- Protección de Rutas y Autorización

13.1 Protección de rutas para usuarios autenticados. 13.2 Control de acceso y autorización en la aplicación. 13.3 Redireccionamiento y rutas privadas.

14.- Trabajo con Formularios Dinámicos y Envío de Datos

14.1 Creación de formularios dinámicos. 14.2 Generación de campos de formulario a partir de datos. 14.3 Validación y manejo de envíos a la API. 14.4 Manipulación de respuestas del servidor.

15.- Manejo de Errores y Carga Condicional

15.1 Manejo de errores en peticiones a la API. 15.2 Indicadores de carga y renderizado condicional. 15.3 Uso de componentes de carga y errores.

16.- Proyecto Final

10.1 Construcción de Servicios para consumo de API REST
10.2 Optimización de rendimiento y carga

TecGurus
Somos y formamos expertos en T.I.

CERTIFICADO DIGITAL

Obtén una constancia que avala tu preparación, si cumples con la asistencia a tu capacitación y elaboras el proyecto final de cada curso, bootcamp o diplomado.

Registrado por la Secretaria del Trabajo y Previsión Social (México).

¡Te esperamos!

55 5211 6931

+52 55 6186 8835

TecGurusNet