* React es una biblioteca de JavaScript para construir interfaces de usuario (UI) interactivas.
* Es mantenido por Facebook y una comunidad de desarrolladores.
* Se utiliza para crear aplicaciones web de una sola página (SPA) y aplicaciones móviles.

**Componentes en React**

* React se basa en el concepto de componentes, que son bloques de construcción reutilizables para interfaces de usuario.
* Los componentes pueden ser funcionales o de clase.
* Los componentes funcionales son funciones de JavaScript que devuelven JSX (JavaScript XML), mientras que los componentes de clase son clases de JavaScript que extienden **React.Component**.

**JSX (JavaScript XML)**

* JSX es una sintaxis que permite escribir HTML dentro de JavaScript.
* Facilita la creación de componentes de React al proporcionar una sintaxis familiar similar a HTML.
* JSX se compila en llamadas a funciones JavaScript regulares durante el tiempo de compilación.

**Estado y Propiedades (Props)**

* El estado (**state**) es un objeto que contiene datos relevantes para un componente.
* Las propiedades (**props**) son entradas que un componente recibe de su componente padre.
* El estado y las props pueden ser utilizados para pasar datos entre componentes y para manejar la interacción y la actualización de la interfaz de usuario.

**Ciclo de Vida del Componente**

* Los componentes de clase de React tienen un ciclo de vida que consta de diferentes etapas, como montaje, actualización y desmontaje.
* Algunos métodos del ciclo de vida importantes son **componentDidMount()**, **componentDidUpdate()** y **componentWillUnmount()**.
* Estos métodos nos permiten ejecutar código en momentos específicos del ciclo de vida del componente, como después de que el componente se haya montado en el DOM o justo antes de que se desmonte.

**REACT Y APIS**

* React es una biblioteca de JavaScript utilizada para construir interfaces de usuario (UI) interactivas.
* Las APIs (Interfaces de Programación de Aplicaciones) son conjuntos de reglas y protocolos que permiten que diferentes aplicaciones se comuniquen entre sí.
* Usar React con APIs nos permite integrar datos dinámicos en nuestras aplicaciones web.

**Consumir una API en React**

* Para consumir una API en React, generalmente utilizamos el método **fetch()** o bibliotecas como Axios.
* **fetch()** es una función incorporada en los navegadores modernos que nos permite hacer solicitudes HTTP.
* Axios es una biblioteca de JavaScript que nos permite realizar solicitudes HTTP desde el navegador o desde Node.js.

**Ejemplo de Consumo de API en React**

* Vamos a crear una aplicación React que consuma una API de muestra y muestre los datos en nuestra interfaz de usuario.
* Usaremos la API JSONPlaceholder, que proporciona datos de prueba simulados para fines de desarrollo.
* Por ejemplo, podemos obtener una lista de usuarios de la API y mostrarlos en nuestra aplicación React.

**Pasos para Consumir una API en React**

1. Instalar Axios (si lo usamos): **npm install axios**.
2. Importar Axios en nuestro componente React.