

## Crear aplicación React (CRA)

CRA es una herramienta de desarrollo introducida por el equipo de React para ayudar a los desarrolladores a crear aplicaciones de React.

Esta herramienta configura un entorno de desarrollo con todas las configuraciones y dependencias necesarias preconfiguradas, lo que permite a los desarrolladores centrarse en la creación de sus componentes y aplicaciones de React.

CRA proporciona una cadena de herramientas de compilación estándar basada en Webpack, que incluye características como la recarga de módulos, la compatibilidad con la sintaxis de ES6+ y las compilaciones listas para producción.

Es una solución fiable y robusta para iniciar proyectos de React, especialmente para los principiantes que quieren desarrollar aplicaciones pequeñas y medianas.

Para crear un proyecto de React con la herramienta CRA, debe dar el siguiente comando:

**`npx create-react-app your-app-name`**

## Vite

Evan You, el creador de Vue.js, creó la herramienta Vite. Está diseñado para ser una herramienta de compilación de desarrollo rápido para proyectos de JavaScript, incluidos varios marcos de JavaScript.

Vite aprovecha las importaciones modernas de módulos ES y el soporte nativo del navegador para módulos ES para proporcionar un reemplazo extremadamente rápido de módulos en caliente y tiempos de inicio del servidor durante el desarrollo.

A diferencia de CRA, que usa Webpack, Vite usa Rollup para agrupar, lo que resulta en tiempos de compilación más rápidos y un mejor rendimiento, especialmente para proyectos más grandes.

Vite también es compatible con varias funciones modernas de JavaScript listas para usar, como los preprocesadores TypeScript, JSX y CSS, sin configuración adicional.

La velocidad y la simplicidad de Vite lo convierten en una opción atractiva para los desarrolladores que buscan una experiencia de desarrollo más eficiente, especialmente a medida que los proyectos crecen en tamaño y complejidad.

Para crear un proyecto de React con Vite, debe usar el siguiente comando:

**`npm create vite@latest`**

El comando anterior le presentará diferentes marcos y bibliotecas. Debe seleccionar React para crear un proyecto de React.

## Kit de herramientas Redux

El kit de herramientas Redux se puede instalar usando npm

**`npm install @reduxjs/toolkit.`**

## INTRODUCCION A REACT Y COMPONENTES DE CLASE

| Term                 | Definition   |
|----------------------|--|
| App.jsx              | El componente raíz de la aplicación React.   |
| Babel                | Una herramienta utilizada para compilar código jsx en el navegador.  |
| Class                | Una plantilla o plano para crear Objetos.  |
| const                | Permite declarar constantes cuya Los valores no se pueden cambiar.   |
| eslintrc.js          | Un archivo de configuración utilizado por ESLint, que es una herramienta de linting popular para JavaScript y la extensión .cjs, indica que es un módulo de CommonJS.  |
| ECMAScript 6 (ES6)   | Una organización de normalización que crea una amplia gama de normas mundiales de tecnología de la información y las comunicaciones. JavaScript se adhiere a la especificación ES6 de Ecma, que salió en 2015. |
| Framework            | Una plataforma integral de desarrollo de software que proporciona una base y estructura para crear aplicaciones completas.   |
| Front-end framework  | Un marco que se centra en la creación del lado orientado al usuario de las aplicaciones web.   |
| .gitignore           | Un archivo que especifica qué archivos y directorios deben ser omitidos por el control de versiones de Git.  |
| Hook                 | Una característica que permite a los desarrolladores manejar los estados y otras características de React sin problemas, eliminando la necesidad de codificación basada en clases.                             |
| Index.html           | Una parte fundamental de los proyectos de desarrollo web, sirviendo como punto de entrada para las aplicaciones web.   |
| JavaScript XML (JSX) | Una extensión de sintaxis de JavaScript que permite código similar a HTML dentro de JavaScript.  |
| let                  | Permite restringir el alcance de las variables dentro de la manzana donde se declaran.   |
| main.jsx             | El punto de entrada para la aplicación React.  |
| Node modules         | Un directorio que contiene todas las dependencias instaladas a través de npm o yarn.   |
| package.json         | Un archivo que contiene metadatos sobre el proyecto y las dependencias.  |
| Public directory     | Un directorio que contiene activos estáticos como archivos HTML, imágenes y fuentes.   |

|                |  |
|----------------|--|
| React          | Una biblioteca JavaScript de código abierto que facilita la creación de interfaces de usuario dinámicas e interactivas para aplicaciones web.                                      |
| README.md      | Un archivo que proporciona información sobre el proyecto, incluidas las instrucciones de configuración, las directrices de uso y la documentación del proyecto.                    |
| src directory  | Un directorio que contiene el código fuente de la aplicación React.  |
| Subclass       | La clase que hereda otra clase.  |
| Superclass     | La clase que hereda la subclase.   |
| this           | Una referencia al objeto actual.   |
| var            | Una variable que se puede usar o hacer referencia desde cualquier parte del código.  |
| Vite           | Una herramienta de compilación que acelera la creación y el desarrollo de proyectos web al hacer un uso óptimo de cómo los navegadores actuales manejan los módulos de JavaScript. |
| vite.config.js | Un archivo que contiene los ajustes de configuración de la herramienta de compilación de Vite.   |

## Gestión de **HOOKS** y formularios

### useState()

`useState()` hook puede administrar estados del componente de la función React donde puede declarar cualquier tipo de datos, por ejemplo, booleano, objeto, matriz, cadena.

|    |   |
|----|---|
| 1  | import React, { useState, useEffect } from 'react'; |
| 2  | function SideEffect() {                             |
| 3  | const [empId, setEmpId] = useState(100);            |
| 4  | return (  |
| 5  | <div>   |
| 6  | <p>{empId}</p>                                      |
| 7  | </div>  |
| 8  | );  |
| 9  | }   |
| 10 | export default SideEffect;                          |
| 11 |   |
| 12 |   |

## useEffect()

`useEffect` es un gancho de React que le permite realizar efectos secundarios en componentes funcionales. Un efecto secundario se refiere a cualquier operación que necesite ejecutar tan pronto como se cargue la página sin llamar a esas operaciones/funcionalidades por separado, como obtener datos de una API.

```
1
2 import React, { useState, useEffect } from 'react';
3 function SideEffect() {
4   const [foods, setFoods] = useState([]);
5   useEffect(() => {
6     fetch('https://api.npoint.io/d542b9ad99f501ab3dbf')
7       .then(response => response.json())
8       .then(data => {
9         console.log(data);
10        setFoods(data);
11      })
12      .catch(error => console.error('Error fetching users:', error));
13  }, []); // Empty dependency array means this effect runs only once when the component mounts
14  return (
15    <div>
16      <h1>Food List</h1>
17      <ul>
18        {foods.map((food)=>{
19          return (<
20            <li><h1>{food.name}</h1></li>
21            <p>food.description</p>
22            <p>food.price</p>
23            <p>food.category</p>
24            <p>food.ingredients</p>
25            <img src={food.image_url} alt="" height='100px' width='100px' />
26          </>
27        )
28        }}}
29      </ul>
30    </div>
31  );
32 }
33 export default SideEffect;
```

## fetch api method

puede obtener datos mediante la API.

```
1 const apiUrl = 'https://jsonplaceholder.typicode.com/posts';
2 fetch(apiUrl)
3   .then(response => response.json())
4   .then(data => {
5     console.log(data);
6   })
7   .catch(error => {
8     console.error('There was a problem with the fetch operation:', error);
9   });
10
```

## Custom hook

Puede utilizar ganchos personalizados en cualquier otro componente. En este fragmento de código, hay un componente de función conocido como `UseToggle`, que sirve como un gancho personalizado y otro componente de la función `ToggleButton`, que usará este gancho personalizado.

```
3 import { useState } from 'react';
4 import UseToggle from './UseToggle';
5 function ToggleButton() {
6   const [isToggled, toggle] = UseToggle(false);
7
8   return (
9     <div>
10       <h1>Toggle Button</h1>
11       <button onClick={toggle}>
12         {isToggled ? 'ON' : 'OFF'}
13       </button>
14     </div>
15   );
16 }
17 export default ToggleButton;
18
19 //UseToggle.jsx
20 import { useState } from "react";
21
22 function UseToggle(initialValue = false) {
23   const [value, setValue] = useState(initialValue);
24
25   const toggle = () => {
26     setValue(!value);
27   };
28
29   return [value, toggle];
30 }
31
32 export default UseToggle
```

## axios api method

El método Axios puede obtener datos mediante la API.

```
1 import axios from 'axios';
2 const apiUrl = 'https://jsonplaceholder.typicode.com/posts';
3 axios.get(apiUrl)
4   .then(response => {
5     console.log(response.data);
6   })
7   .catch(error => {
8     console.error('There was a problem with the fetch operation:', error);
9   });
10
11
```

## onChange

El atributo de evento se usa a menudo en HTML y React para rastrear cuándo cambia el valor de un campo de entrada, como una entrada de texto. El evento onChange se produce cuando un usuario escribe algo en un campo de entrada. Este atributo le permite registrar y controlar los cambios

```
1 import React, { useState } from 'react';
2 function FormData() {
3   const [empName, setEmpName] = useState('');
4   const handleChange = event => {
5     setEmpName(event.target.value);
6   };
7   const handleSubmit = event => {
8     event.preventDefault();
9     console.log('Form submitted:', empName);
10  };
11
12  return (
13    <div>
14      <h2>My Form</h2>
15      <form onSubmit={handleSubmit}>
16        <label>
17          Input:
18          <input type="text" value={empName} onChange={handleChange} />
19        </label>
20        <button type="submit">Submit</button>
21      </form>
22    </div>
23  );
24 }
25
26 export default FormData;
```