

«Talento Tech»

React JS

Clase 06



Clase N° 6 | Router - Rutas Estáticas y Dinámicas

Índice:

- Instalación y configuración de React Router.
 - Creación de rutas estáticas (inicio, lista de productos).
 - Uso de Link para navegar entre componentes.
-

Objetivos de la Clase:

- Instalar y configurar React Router en una aplicación React.
- Crear rutas estáticas para navegar entre diferentes vistas como inicio y lista de productos.
- Utilizar el componente Link para permitir la navegación entre componentes sin recargar la página.

Instalación y Configuración de React Router



React Router

¿Qué es React Router?

React Router es una librería estándar para el manejo de rutas en aplicaciones React. Permite la navegación entre diferentes vistas de una aplicación sin necesidad de recargar la página, mejorando la experiencia del usuario.

Pasos para instalar React Router:

Ejecuta el siguiente comando para instalar React Router en tu proyecto:

```
npm install react-router-dom
```

Configura React Router en tu archivo `index.js` o `App.js`:

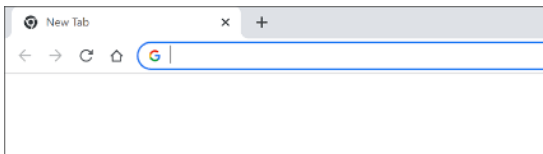
```
import React from 'react';
import ReactDOM from 'react-dom/client';
import App from './App';
import { BrowserRouter as Router } from 'react-router-dom';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <Router>
      <App />
    </Router>
  </React.StrictMode>
);
```

Explicación:

- **BrowserRouter** es el componente que habilita el enrutamiento en tu aplicación.
- Todo tu componente **App** estará envuelto por **Router**, lo que te permitirá usar rutas dentro de la aplicación.

Creación de Rutas Estáticas (Inicio, Lista de Productos)



¿Qué son las rutas estáticas?

Las rutas estáticas son aquellas que no dependen de datos dinámicos y se configuran de manera fija en la aplicación. Por ejemplo,

una página de inicio o una lista de productos.

Configurar rutas estáticas:

1. Definir las rutas en **App.js**

Usa el componente **Route** para definir qué componentes se deben renderizar para cada ruta específica.

```
import React from 'react';
import { BrowserRouter as Router, Routes, Route } from
'react-router-dom';
import Inicio from './Inicio';
import Productos from './Productos';

function App() {
  return (
    <Router>
      <div>
        <Routes>
          <Route path="/" element={<Inicio />} />
          <Route path="/productos" element={<Productos />} />
        </Routes>
      </div>
    </Router>
  );
}
```

```

        </div>
      </Router>
    );
  }

export default App;

```

Explicación:

1. **Routes:** Reemplaza el componente **Switch** de versiones anteriores de React Router. Su propósito es agrupar las rutas y asegurarse de que solo una coincida y se renderice.
2. **Route:** Se usa para definir una ruta específica y asociarla con un componente.
 - La propiedad **path** define la URL asociada.
 - La propiedad **element** reemplaza a **component** en React Router v6. Aquí debes pasar el componente como un JSX, envuelto en **{}**.
3. **Exactitud de rutas:** En React Router v6, las rutas son exactas por defecto, por lo que no necesitas usar **exact**.

Crear los componentes de las rutas:

- Componente **Inicio.js**:

```

import React from 'react';

function Inicio() {

  return <h1>Bienvenido a la Página de Inicio</h1>;

}

export default Inicio;

```

Componente **Productos.js**:

```
import React from 'react';

function Productos() {
  return <h1>Lista de Productos</h1>;
}

export default Productos;
```

Uso de Link para Navegar entre Componentes

¿Qué es el componente Link?

El componente **Link** es utilizado para crear enlaces de navegación dentro de una aplicación React. A diferencia de los enlaces tradicionales (`<a>`), **Link** permite la navegación sin recargar la página.

1. **Usar Link en el componente de navegación:** Crea un componente de navegación que incluya los enlaces para navegar entre las rutas estáticas.

```
import React from 'react';

import { Link } from 'react-router-dom';

function Navbar() {

  return (

    <nav>

      <ul>

        <li><Link to="/">Inicio</Link></li>

        <li><Link to="/productos">Productos</Link></li>

      </ul>

    </nav>

  );
}
```

```

    </nav>

  );
}

export default Navbar;

```

Explicación:

- El componente **Link** tiene la propiedad **to** que define la ruta a la que se debe dirigir.
- Al hacer clic en un **Link**, React Router maneja la navegación sin recargar la página.

Incluir **Navbar en **App.js**:** En tu archivo **App.js**, importa y usa el componente **Navbar** para que los enlaces estén disponibles en toda la aplicación.

```

import React from 'react';
import { BrowserRouter as Router, Routes, Route } from
'react-router-dom';
import Navbar from './Navbar';
import Inicio from './Inicio';
import Productos from './Productos';

function App() {
  return (
    <Router>
      <div>
        <Navbar />
        <Routes>
          <Route path="/" element={<Inicio />} />
          <Route path="/productos" element={<Productos />} />
        </Routes>
      </div>
    </Router>
  );
}

export default App;

```


TalentoLab - Proyecto final



Descripción de tu tarea:



En este ejercicio, vas a mejorar la navegación de tu aplicación eCommerce integrando rutas para ver algún tipo de navegación básica en el sitio web. Para ello, se usará React Router y se agregarán nuevas rutas para distintas secciones de la página. Además, optimizarás la navegación para que el usuario pueda ir de una lista de productos a otras secciones y viceversa.

Requisitos:

1. Implementación de rutas :

- Usa React Router para crear rutas, donde cada sección (About, Contact, etc) tendrá su propia página.

2. Crear componente para cada sección:

- Crea un componente **About** que tenga una descripción de la marca del eCommerce. Otro componente **Contact** que tendrá un form que armaremos más adelante, donde el usuario podrá ponerse en contacto con la empresa de nuestro cliente.

3. Navegar entre productos:

- Usa el componente **Link** para permitir la navegación entre las secciones del sitio web.

4. Estado de carga y manejo de errores:

- Mantén el estado de carga y de errores en la página donde se vean los productos.
- Muestra un mensaje de "Cargando productos..." mientras los datos están siendo obtenidos y maneja posibles errores con un mensaje adecuado.

Pautas Generales:

- **Estructura del proyecto:** Continúa con el proyecto anterior. Agrega el componente de cada sección nueva.
- **Componentes:** Asegúrate de que los componentes sean reutilizables y están bien organizados.
- **Diseño:** Mantén el diseño profesional utilizando Bootswatch o cualquier otra alternativa de diseño responsivo.

Reflexión final

Aprendimos a instalar y configurar React Router para manejar la navegación entre diferentes componentes en nuestra aplicación React. Vimos cómo crear rutas estáticas para el inicio y la lista de productos, y cómo usar el componente `Link` para permitir una navegación fluida entre ellas sin recargar la página.

Al dominar React Router, podemos construir aplicaciones más dinámicas y completas, donde el usuario puede navegar entre diferentes vistas sin perder el estado de la aplicación. Esta es una habilidad esencial para desarrollar aplicaciones React de una sola página (SPA).

Materiales y Recursos Adicionales:

- ★ [Documentación oficial de React Router](#)
- ★ [React Router - Guía para Principiantes](#)

Preguntas para Reflexionar:

- ¿Qué sucedería si no utilizamos `exact` en la ruta de inicio?
- ¿Por qué es importante usar `Link` en lugar de `<a>` para la navegación?
- ¿Cómo podrías organizar una aplicación con muchas rutas? ¿Qué patrón de diseño seguirías?

Próximos Pasos:

- Creación de rutas dinámicas (detalle de productos).
- Implementación de rutas protegidas (carrito, administración de productos).
- Redirección de usuarios no autenticados.



Buenos Aires
aprende
Agencia de Políticas para el Futuro

BA Buenos
Aires
Ciudad