



Buenos
Aires
Ciudad

Agencia de Habilidades
para el Futuro

<Talento
Tech />

React JS

**Clase 06 | Router - Rutas Estáticas y
Dinámicas**

¡Les damos la bienvenida!



Vamos a comenzar a grabar la clase.

Índice

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.

Router - Rutas Estáticas y Dinámicas

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

Rutas Dinámicas y Protegidas

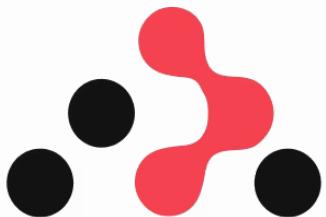
- Creación de rutas dinámicas (detalle de productos).
- Implementación de rutas protegidas (carrito, administración de productos).
- Redirección de usuarios no autenticados.

Objetivos de la Clase

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

React Router

¿Qué es React Router?



React Router

Definición

React Router es una librería estándar para el manejo de rutas en aplicaciones React.

Beneficios

Permite la navegación entre vistas sin recargar la página, mejorando la experiencia del usuario.

Instalación de React Router

Paso 1

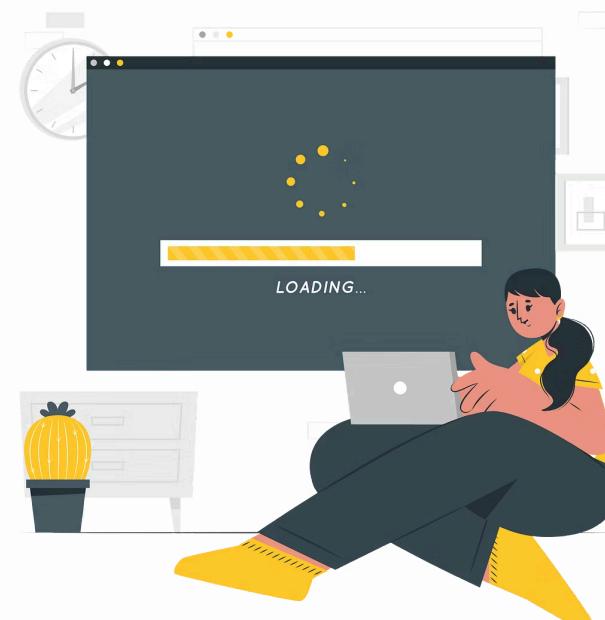
Abre tu terminal en el directorio del proyecto.

Paso 2

Ejecuta el comando: `npm install react-router-dom`

Paso 3

Espera a que se complete la instalación.



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 de la Configuración



BrowserRouter

Habilita el enrutamiento en tu aplicación.



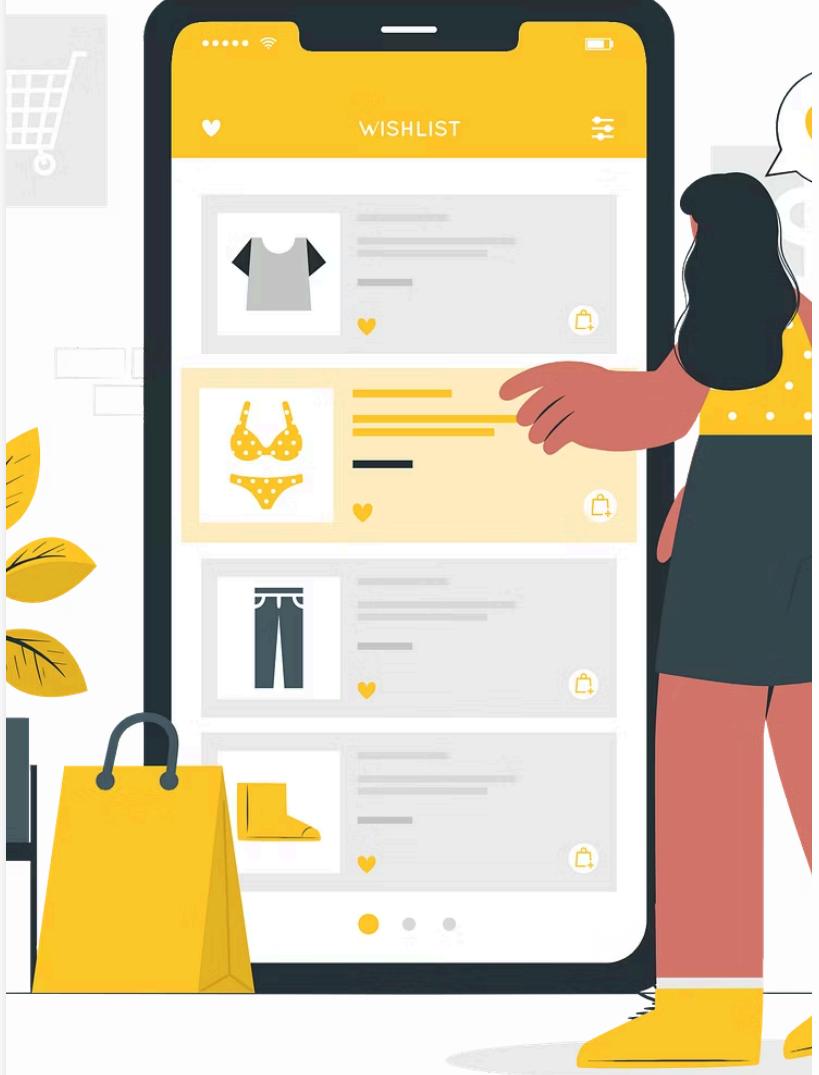
Envolviendo App

Permite usar rutas dentro de la aplicación.



Componente Router

Proporciona contexto de enrutamiento a la aplicación.



Rutas Estáticas

Rutas que no dependen de datos dinámicos y se configuran de manera fija.

Ejemplos

Página de inicio, lista de productos, página de contacto.

Configuración de Rutas Estáticas

```
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>
  );
}
export default App;
```

Explicación de Rutas Estáticas

Routes

Agrupar las rutas y asegurarse de que solo una coincide y se renderice

path

define la URL asociada

element

Aquí debes pasar el componente como un JSX, envuelto en {}

Route

Asocia una ruta específica con un componente.

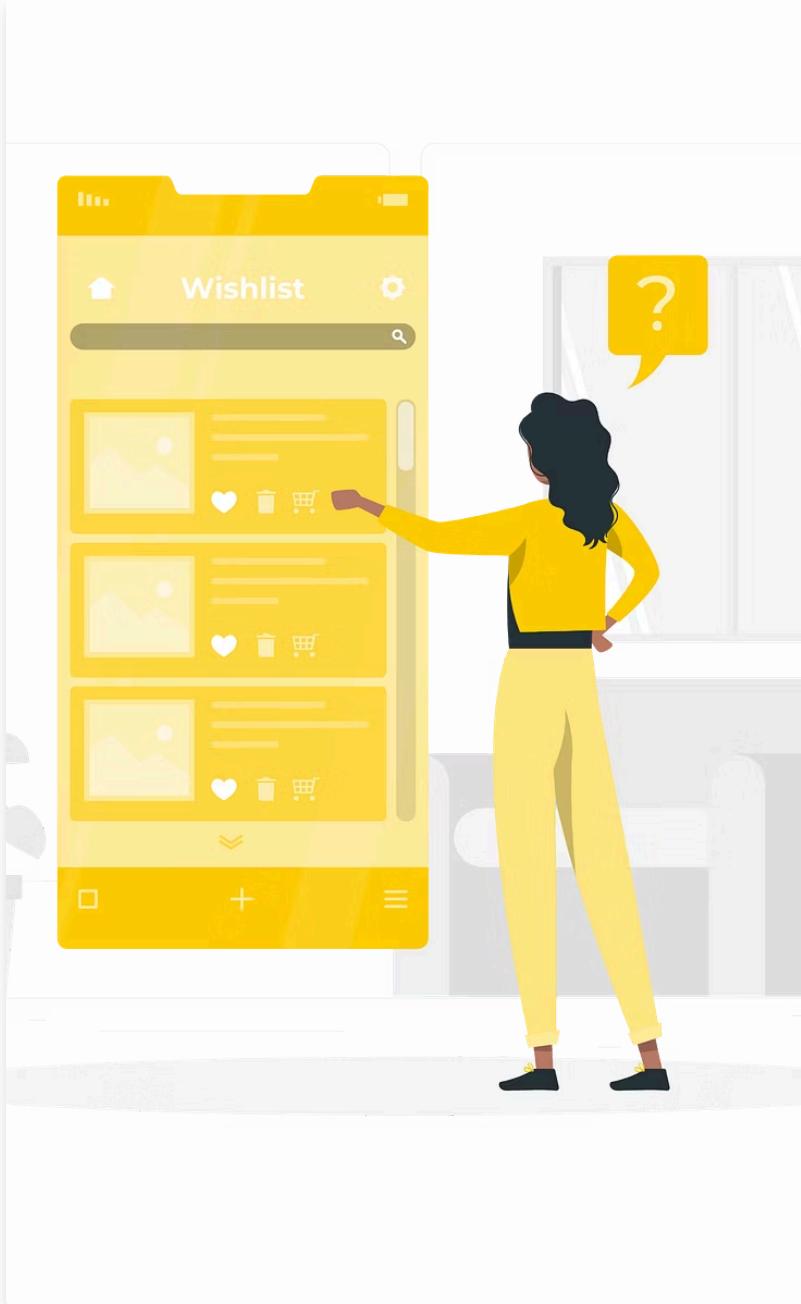
Componente Inicio

```
import React from 'react';

function Inicio() {
  return <h1>Bienvenido a la Página de Inicio</h1>;
}

export default Inicio;
```



A woman with dark hair tied back, wearing a yellow long-sleeved top and yellow pants, stands facing a large smartphone screen. The phone displays a 'Wishlist' app interface with three items in a grid. Each item has a heart icon, a trash bin icon, and a shopping cart icon. The woman is pointing her right index finger at the shopping cart icon of the top item. A yellow speech bubble containing a question mark is positioned above her head. The background is a light grey.

Componente Productos

```
import React from 'react';

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

export default Productos;
```

Componente Link



Componente Link

El componente Link es utilizado para crear enlaces de navegación dentro de una aplicación **React**.

Ventaja

Permite la navegación sin recargar la página, mejorando la experiencia del usuario.

Uso de Link en Navegación

```
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>
  );
}
export default Navbar;
```



Explicación del Componente Link

Propiedad to

Define la ruta a la que se debe dirigir.

Manejo de Navegación

React Router maneja la navegación sin recargar la página al hacer clic.

Inclusión de Navbar en App.js

```
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;
```



⚠ Importante: El Router debe montarse en `main.jsx` o en `App.jsx`, pero no en ambos a la vez.

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).



Preguntas para Reflexionar



Patrón de diseño

¿Cómo podrías organizar una aplicación con muchas rutas?
¿Qué patrón de diseño seguirías?

Uso de exact

¿Qué sucedería si no lo utilizamos en la ruta de inicio?

Link vs

¿Por qué es importante usar Link para la navegación?

Próximos Pasos

1

Rutas Dinámicas

Creación de rutas para detalle de productos.

2

Rutas Protegidas

Implementación para carrito y administración de productos.

3

Redirección

Manejo de usuarios no autenticados.

Recursos Adicionales



[Documentación oficial de React Router](#)



[React Router – Guía para Principiantes](#)



Ejercicios Prácticos



TalentoLab – Proyecto Final

Ahora que ya has integrado una API y optimizado el manejo de estados en tu eCommerce, el equipo de TalentoLab ha decidido asignarte una nueva responsabilidad.



Ejercicio Práctico

Obligatorio



Silvia

Product Owner

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.



Ejercicio Práctico

Obligatorio

Requisitos:

 **Implementación de rutas:**

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

 **Crear componente de detalle de producto:**

- 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.

 **Navegar entre productos:**

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



Ejercicio Práctico

Obligatorio

Requisitos:



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.

Ejercicio Práctico

Obligatorio

Pautas Generales:



- **Estructura del proyecto:** Continúa con el proyecto anterior. Agrega el componente de detalles y la funcionalidad de rutas dinámicas.
- **Componentes:** Asegúrate de que los componentes Productos y ProductoDetalle sean reutilizables y estén bien organizados.



¡Nuevo cuestionario en Campus!

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