Hasta ahora, todos nuestros componentes viven en la misma "página". ¿Pero qué pasa si queremos tener una sección de "Tareas" y una sección de "Directorio" separadas, con sus propias URLs (como /tareas o /directorio)?

Esto se logra con un **Router (Enrutador)**. En React, la librería estándar de oro para esto es **React Router**.

En este tutorial, convertiremos nuestra aplicación de una sola vista en una **Aplicación de Página Única (SPA)** con múltiples "páginas", un menú de navegación y un "Layout" compartido.

Tutorial: Creando una Aplicación Multi-página con React Router (react-router-dom)

Objetivo del Proyecto:

Usaremos los componentes que ya hemos construido (TodoList y UserDirectory) y los colocaremos en sus propias páginas. Crearemos una barra de navegación en nuestro Header para movernos entre ellas sin que la página se recargue por completo.

Paso 1: Instalación

Abre tu terminal en la raíz de tu proyecto (mi-app-modular) y ejecuta:

Bash

npm install react-router-dom

Paso 2: Configurar el BrowserRouter

El BrowserRouter es un componente que "envuelve" toda tu aplicación y le da la capacidad de manejar rutas. El mejor lugar para ponerlo es en src/index.js.

1. Abre src/index.js.
2. Importa BrowserRouter y envuelve tu componente <App /> (y el ThemeProvider):

JavaScript

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import { ThemeProvider } from './context/ThemeContext';
import { BrowserRouter } from 'react-router-dom'; // <-- Importar
```

```
const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(
  <React.StrictMode>
    <BrowserRouter> {/* <-- Envolver la App */}
      <ThemeProvider>
        <App />
      </ThemeProvider>
    </BrowserRouter>
  </React.StrictMode>
);
```

Paso 3: Crear las "Páginas"

Nuestros componentes TodoList y UserDirectory ya son, en esencia, "páginas". Pero necesitamos una página de "Inicio".

1. Crea una nueva carpeta src/components/Home.
2. Dentro, crea Home.js:

JavaScript

```
import React from 'react';


const Home = () => {
  return (
    <div>
      <h2>Bienvenido a la Aplicación de Demostración</h2>
      <p>
        Usa la navegación de arriba para visitar el
        Directorio de Usuarios o la Lista de Tareas.
      </p>
    </div>
```

```
  );
};

export default Home;
```

Paso 4: Crear un "Layout" Compartido con <Outlet />

Este es un concepto **clave**. Queremos que nuestro Header (con el menú y el toggle de tema) aparezca en *todas* las páginas.

Para hacer esto, creamos un componente "Layout" que renderiza el Header y un marcador de posición especial llamado <Outlet />. El Outlet es donde React Router renderizará dinámicamente el componente de la página actual (ya sea Home, TodoList, etc.).

1. Crea una nueva carpeta src/components/Layout.
2. Dentro, crea Layout.js:

JavaScript

```
import React from 'react';
import { Outlet } from 'react-router-dom'; // <-- Importar Outlet
import Header from '../Header/Header';

const Layout = () => {
  return (
    <>
      <Header />
      <main>
        {/* El Outlet renderizará el componente de la ruta hija */}
        <Outlet />
      </main>
    </>
  );
};
```

export default Layout;

<> ... </> es un "Fragmento" de React, nos permite agrupar elementos sin añadir un div extra al DOM.

Paso 5: Definir las Rutas en App.js

El trabajo de App.js va a cambiar drásticamente. Ya no mostrará los componentes directamente. Ahora, su *única* responsabilidad será definir la "tabla de rutas" de nuestra aplicación.

1. Abre src/App.js y reemplaza su contenido con esto:

JavaScript

```
import React, { useContext } from 'react';
import { Routes, Route } from 'react-router-dom';
import './App.css';
import ThemeContext from './context/ThemeContext';

// Importar el Layout y las Páginas
import Layout from './components/Layout/Layout';
import Home from './components/Home/Home';
import TodoList from './components/TodoList/TodoList';
import UserDirectory from './components/UserDirectory/UserDirectory';

function App() {
  const { theme } = useContext(ThemeContext);

  return (
    <div className={`App ${theme}`}>
      {/* El componente <Routes> envuelve todas las rutas */}
      <Routes>
        {/* Esta es una "Ruta de Layout".
          Todas las rutas anidadas dentro se renderizarán DENTRO del <Outlet /> de Layout.
        */}
```

```
      <Route path="/" element={<Layout />}>

        {/* Rutas Hijas */}
        <Route index element={<Home />} />
        <Route path="tareas" element={<TodoList />} />
        <Route path="directorio" element={<UserDirectory />} />

        {/* Ruta "Catch-all" para 404 (No encontrado) */}
        <Route path="*" element={<h2>Página no encontrada</h2>} />
      </Route>
    </Routes>
  </div>
 );
}

export default App;
```

Desglose de las Rutas:

- **<Routes>**: El contenedor de todas las definiciones de ruta.
- **<Route path="/" element={<Layout />}> ... </Route>**: Esta es la ruta padre. Le decimos: "Para cualquier ruta que comience con /, renderiza el componente Layout".
- **<Route index ... />**: La palabra clave index significa "este es el componente por defecto para la ruta padre (/)".
- **<Route path="tareas" ... />**: Esto se combina con el padre para crear la URL /tareas.
- **<Route path="*" ... />**: El asterisco es un comodín. Si el usuario va a /cualquier-cosa, se mostrará este elemento.

Paso 6: Añadir Navegación con <Link />

Finalmente, necesitamos enlaces en nuestro Header para navegar. **¡Importante!** Nunca uses una etiqueta <a> normal. Una etiqueta <a> recarga toda la página, lo que rompe la experiencia de una SPA.

Usaremos el componente <Link> de React Router.

1. Abre src/components/Header/Header.js.
2. Importa Link y añade la navegación:

JavaScript

```
import React from 'react';
import { Link } from 'react-router-dom'; // <-- Importar Link
import './Header.css';
import ThemeSwitcher from '../ThemeSwitcher/ThemeSwitcher';


const Header = () => {
  return (
    <header className="app-header">
      <div className="logo-nav">
        <h1 className="logo">Mi App</h1>
        <nav>
          {/* Usamos <Link> en lugar de <a href=""> */}
          <Link to="/">Inicio</Link>
          <Link to="/tareas">Tareas</Link>
          <Link to="/directorio">Directorio</Link>
        </nav>
      </div>
      <ThemeSwitcher />
    </header>
  );
};


export default Header;
```

3. (Opcional) Añade un poco de estilo a src/components/Header/Header.css para que se vea bien:

CSS

```
/* ... (estilos existentes) ... */

.app-header {
  /* ... (estilos existentes) ... */
  display: flex;
  justify-content: space-between;
  align-items: center;
}

.logo-nav {
  display: flex;
  align-items: center;
  gap: 30px;
}

.logo {
  margin: 0;
  font-size: 1.5em;
}

nav {
  display: flex;
  gap: 20px;
}

nav a { /* <Link> se renderiza como una <a>, así que podemos estilizarla */
  color: white;
  text-decoration: none;
  font-weight: bold;
```

```
  padding: 5px 0;
  border-bottom: 2px solid transparent;
  transition: border-bottom-color 0.2s;
}

nav a:hover {
  border-bottom-color: #61dafb; /* Color azul de React */
}
```

¡Prueba Final!

Ejecuta npm start. Tu aplicación ahora cargará en la página de "Inicio". Verás el Header con los nuevos enlaces.

Haz clic en "Tareas" o "Directorio". Observa cómo:

1. La URL en tu navegador cambia (a /tareas o /directorio).
2. El contenido de la página cambia *instantáneamente* (el <Outlet /> hace su trabajo).
3. El Header y el botón de tema permanecen en su lugar, ¡porque son parte del Layout!
4. **No hay recarga de página.**

Resumen de Conceptos Clave

- **react-router-dom**: La librería estándar para el enrutamiento en React.
- **BrowserRouter**: El componente que "activa" el enrutamiento en tu aplicación.
- **Routes y Route**: Componentes declarativos para definir qué componente renderizar para qué URL.
- **Link**: El componente para crear navegación interna *sin* recargar la página (reemplaza a <a>).
- **Rutas de Layout y <Outlet />**: Un patrón poderoso para crear partes de la UI compartidas (como encabezados, barras laterales, pies de página) que "envuelven" el contenido de tu página.