

# React Router: Navegación en SPAs

---

Diego Muñoz

09 de noviembre de 2025

- Entender cómo se maneja la **navegación** en aplicaciones React.
- Aprender los componentes esenciales de **React Router**.
- Configurar rutas básicas y navegación entre páginas.
- Distinguir entre **SPA** y múltiples archivos HTML.

## ¿Por qué usar React Router?

- React solo muestra **un componente raíz** (App).
- Necesitamos un sistema para cambiar “pantallas” sin recargar.
- **React Router** permite rutas declarativas dentro de una SPA.

```
yarn add react-router-dom
```

- Cada “ruta” asocia una **URL** con un **componente**.
- React Router escucha los cambios de URL sin recargar la página.

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

```
<Route path="/about" element={<About />} />
```

## Configuración inicial (Router + Routes)

```
import { BrowserRouter, Routes, Route } from "react-router-dom";
import Home from "../pages/Home";
import About from "../pages/About";

export default function App() {
  return (
    <BrowserRouter>
      <Routes>
        <Route path="/" element={<Home />} />
        <Route path="/about" element={<About />} />
      </Routes>
    </BrowserRouter>
  );
}
```

```
// pages/Home.jsx
export default function Home() {
  return <h2>Inicio</h2>;
}
```

```
// pages/About.jsx
export default function About() {
  return <h2>Acerca de</h2>;
}
```

*Cada archivo representa una “vista” o “pantalla” distinta.*

## Navegación entre páginas

- Link reemplaza los `<a>` normales.
- Cambia la URL sin recargar la página.

```
import { Link } from "react-router-dom";
```

```
export default function NavBar() {  
  return (  
    <nav>  
      <Link to="/">Inicio</Link> |{" "  
      <Link to="/about">Acerca de</Link>  
    </nav>  
  );  
}
```

## Integrar la barra de navegación

```
import { BrowserRouter, Routes, Route } from "react-router-dom";
import NavBar from "../components/NavBar";
import Home from "../pages/Home";
import About from "../pages/About";
export default function App() {
  return (
    <BrowserRouter>
      <NavBar />
      <Routes>
        <Route path="/" element={<Home />} />
        <Route path="/about" element={<About />} />
      </Routes>
    </BrowserRouter>
  );
}
```



- Se definen con `:nombre`.
- Se leen con `useParams()`.

// En App.jsx

```
<Route path="/user/:id" element={<User />} />
```

```
// pages/User.jsx
import { useParams } from "react-router-dom";

export default function User() {
  const { id } = useParams();
  return <h3>Perfil de usuario: {id}</h3>;
}
```

*Ejemplo: /user/42 muestra "Perfil de usuario: 42".*

- `useNavigate()` permite redirigir desde código.

```
import { useNavigate } from "react-router-dom";

export default function Login() {
  const navigate = useNavigate();
  function handleLogin() {
    // validaciones...
    navigate("/dashboard");
  }
  return <button onClick={handleLogin}>Entrar</button>;
}
```

```
<Route path="*" element={<h3>Página no encontrada</h3>} />
```

*Debe ir al final del bloque <Routes>.*

## Estructura recomendada

```
src/  
├─ pages/  
│   ├─ Home.jsx  
│   ├─ About.jsx  
│   └─ User.jsx  
├─ components/  
│   └─ NavBar.jsx  
└─ App.jsx
```

Referir a ejemplo en [GitHub](#).

- **BrowserRouter**: contexto principal de navegación.
- **Routes / Route**: definen qué mostrar por URL.
- **Link**: navegación declarativa.
- **useParams / useNavigate**: lectura y cambio de ruta.
- **SPA real**: una sola página HTML, múltiples vistas.