



# Navegación en React

Centro de Servicios y Gestión Empresarial  
SENA Regional Antioquia



@SENAComunica

[www.sena.edu.co](http://www.sena.edu.co)

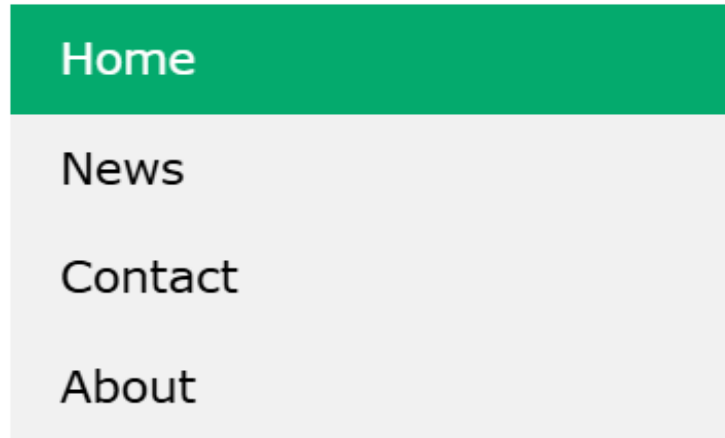
# Concepto – Navegación

- Como React es una **Single Page Application** (SPA), no recarga el navegador cuando cambias de "página".
- Todo pasa en el **cliente**, React simula la navegación entre páginas cambiando la URL del navegador y mostrando distintos componentes según la ruta.
- Para lograr esto, se usa un **router** que controla qué componente renderizar en base a la URL.

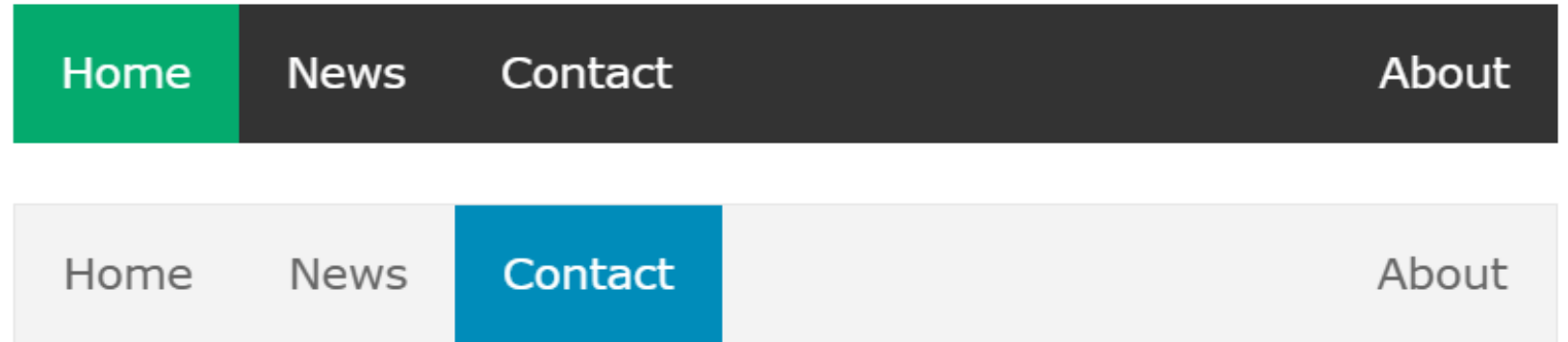
# Concepto – Navegación

La navegación en React hace referencia al proceso de cambiar entre distintas vistas o páginas dentro de una aplicación web. Como React es una biblioteca de JavaScript enfocada en construir interfaces de usuario, no incluye un sistema de enrutamiento integrado.

Vertical



Horizontal





# Concepto – React router

**React Router** es una biblioteca que permite manejar la navegación en aplicaciones React de manera declarativa. Esto facilita crear aplicaciones de una sola página (SPA).

## 1. Instalación de React Router

Para usar React Router, primero hay que instalarlo en el proyecto. En la mayoría de los casos, se instala la última versión de la biblioteca.

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

# Concepto – React router

Elemento	¿Qué hace?
<code>&lt;BrowserRouter&gt;</code>	Envoltorio principal que habilita el enrutamiento (usa el History API del navegador)
<code>&lt;Routes&gt;</code>	Contenedor de todas las rutas
<code>&lt;Route&gt;</code>	Define una ruta con un path y el componente a renderizar
<code>&lt;Link&gt;</code> o <code>&lt;NavLink&gt;</code>	Links declarativos (no usan a href, evitan recarga de página)
<code>useNavigate()</code>	Hook para redireccionar programáticamente
<code>useParams()</code>	Hook para leer parámetros dinámicos de la URL

A white icon of an atom, featuring a central circle and three intersecting elliptical orbits, is positioned to the left of the main title.

# Actualización de paquetes

# Actualización de paquetes



```
PS C:\Users\USUARIO\Documents\React\ejemplosNavegacion> npm install
npm warn deprecated inflight@1.0.6: This module is not supported, and leaks memory. Do not use it. Check out lru-cache if you want a good and tested way to coalesce async requests by a key value, which is much more comprehensive and powerful.
npm warn deprecated rimraf@3.0.2: Rimraf versions prior to v4 are no longer supported
npm warn deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm warn deprecated @humanwhocodes/object-schema@2.0.3: Use @eslint/object-schema instead
npm warn deprecated @humanwhocodes/config-array@0.13.0: Use @eslint/config-array instead
npm warn deprecated eslint@8.57.1: This version is no longer supported. Please see https://eslint.org/version-support for other options.
```

```
PS C:\Users\USUARIO\Documents\React\ejemplosNavegacion> npm outdated
```

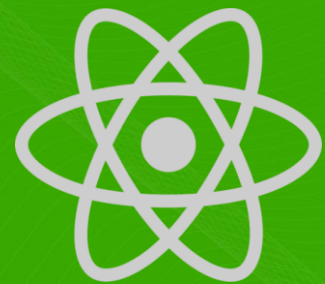
<u>Package</u>	<u>Current</u>	<u>Wanted</u>	<u>Latest</u>	<u>Location</u>	<u>Depended by</u>
eslint	8.57.1	8.57.1	9.15.0	node_modules/eslint	ejemplosNavegacion
eslint-plugin-react-hooks	4.6.2	4.6.2	5.0.0	node_modules/eslint-plugin-react-hooks	ejemplosNavegacion

```
PS C:\Users\USUARIO\Documents\React\ejemplosNavegacion> npm dedupe
```

```
up to date, audited 269 packages in 1s
```

```
103 packages are looking for funding
  run `npm fund` for details
```





# Navegación Externa



# Concepto – Ejemplo

```
import React from 'react';
```

01

```
function HomePage() {  
  return (  
    <div>  
      <h1>Inicio</h1>  
      <p>Bienvenido a la página de inicio</p>  
    </div>  
  );  
}
```

02

```
function AboutPage() {  
  return (  
    <div>  
      <h1>Acerca de</h1>  
      <p>Nosotros somos una empresa dedicada a...</p>  
    </div>  
  );  
}
```

03

# Concepto – Ejemplo

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

04

```
export function App() {  
  return (  
    <Router>  
      <Routes>  
        <Route path="/" element={<HomePage />} />  
        <Route path="/about" element={<AboutPage />} />  
      </Routes>  
    </Router>  
  );  
}
```

05

# Concepto – Elementos

## 2. Conceptos Clave de React Router

A light blue rounded rectangular box, likely a placeholder for an image or diagram.

### Link

- Es un componente que reemplaza los enlaces HTML `<a>`. Se usa para navegar entre rutas sin recargar la página.

# Concepto – Ejemplo

```
import React from 'react';
import { BrowserRouter as Router, Route, Routes, Link } from 'react-router-dom';

export function App() {
  return (
    <Router>
      <nav>
        <Link to="/">Inicio</Link>
        <Link to="/about">Acerca de</Link>
      </nav>
      <Routes>
        <Route path="/" element={<HomePage />} />
        <Route path="/about" element={<AboutPage />} />
      </Routes>
    </Router>
  );
}
```

01

02

# Concepto – Elementos

## 3. Parámetros de Ruta y Navegación Dinámica

React Router permite definir rutas dinámicas mediante parámetros, útiles cuando una página depende de un valor variable, como el ID de un producto o usuario.

Para acceder al parámetro, se usa el hook **useParams**; es una herramienta que permite acceder a los parámetros de la URL en una aplicación de React. Este hook es particularmente útil cuando una ruta contiene valores dinámicos, y quieres extraer esos valores para usarlos dentro de un componente.

# Concepto – Elementos

## 3. Parámetros de Ruta y Navegación Dinámica

### Enviar el parámetro - Sintaxis:

```
<Link to={` /product/${productId}`} >Productos</Link>
```

### Definir la ruta para capturar el parámetro - Sintaxis:

```
<Route path="/product/:productId" element={<ProductPage />} />
```

### Recibir el parámetro en el componente - Sintaxis:

```
import { useParams } from 'react-router-dom';  
const { productId } = useParams();
```

# Concepto – Ejemplo

```
import React from 'react';
```

01

```
function ProductPage() {  
  const { productId } = useParams();  
  return (  
    <div>  
      <h1>Producto {productId}</h1>  
      <p>Información del producto {productId}</p>  
    </div>  
  );  
}
```

02

```
export function App() { . . . }
```

02



# Concepto — Ejemplo

```
import React from 'react';
import { BrowserRouter as Router, Route, Routes, Link, useParams } from 'react-router-dom';
```

01

```
function ProductPage() { . . . }
```

02

```
export function App() {
  const productId = 123;
  return (
    <Router>
      <nav>
        <Link to={` /product/${productId}`} >Productos</Link>
      </nav>
      <Routes>
        <Route path="/product/:productId" element={ <ProductPage /> } />
      </Routes>
    </Router>
  );
}
```

03

# Concepto – Elementos

## 4. Navegación Programática

A veces necesitamos redirigir al usuario después de una acción (como el envío de un formulario). Para esto, usamos el hook **useNavigate** para navegar programáticamente.

### Sintaxis:

```
import { useNavigate } from 'react-router-dom';  
const navigate = useNavigate();  
navigate('/dashboard');
```

# Concepto – Ejemplo

```
import React from 'react';
import { BrowserRouter as Router, Route, Routes, Link, useParams, useNavigate }
from 'react-router-dom';
```

01

```
function HomePage() {
  const navigate = useNavigate();

  const handleLogin = () => {
    navigate('/dashboard');
  };
  return (
    <div>
      <h1>Inicio</h1>
      <p>Bienvenido a la página de inicio</p>
      <br />
      <button onClick={handleLogin}>Iniciar Sesión</button>
    </div>
  );
}
```

02

# Concepto — Ejemplo

```
import React from 'react';  
import { BrowserRouter as Router, Route, Routes, Link, useParams, useNavigate }  
from 'react-router-dom';
```

01

```
function HomePage() { . . . }
```

02

```
function DashboardPage() {  
  return (  
    <p>Panel de control</p>  
  );  
}
```

03

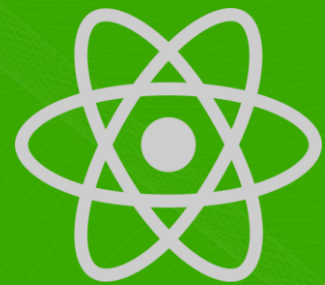
```
export function App() {  
  const productId = 123;  
  return (  
    <Router>  
      <Routes>  
        <Route path="/dashboard" element={<DashboardPage />} />  
      </Routes>  
    </Router>  
  );  
}
```

04

# Ventajas – React router

## Ventajas de React Router

- **SPA sin recarga de página:** Permite crear una experiencia de navegación sin refrescar la página, mejorando la UX.
- **Rutas dinámicas:** Facilita el manejo de contenido dinámico y rutas anidadas.
- **Control completo sobre la navegación:** A través de hooks y componentes, React Router ofrece flexibilidad para la gestión de rutas y redirecciones.



# Protección de Rutas

# Protección de rutas

```
import { Navigate } from 'react-router-dom';
import { useAuth } from '../shared/contexts/AuthContext';

function ProtectedRoute({ children }) {
  const { user } = useAuth();

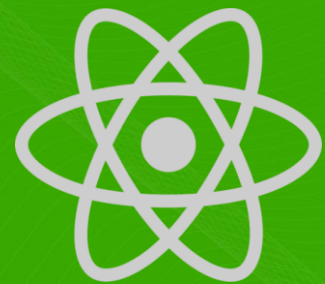
  if (!user) {
    return <Navigate to="/login" replace />;
  }
  return children;
}
```

```
<Route path="/dashboard" element={
  <ProtectedRoute>
    <Dashboard />
  </ProtectedRoute>
} />
```

01

02





# Navegación Interna

# Paquete – react-scroll

## 1. Instalar el paquete

```
npm install react-scroll
```

Una vez instalado **react-scroll**, puedes usar componentes como **Link** y **Element** para implementar desplazamientos animados entre secciones de tu aplicación.

- **<Link>**: Permite enlazar a una sección específica definida por el atributo **name** de un componente **Element**.
- **<Element>**: Define las secciones de destino en la página.

# Paquete – react-scroll

```
/project-app
├── /src
│   ├── /components
│   │   ├── Header.jsx
│   │   ├── Contenido.jsx
│   │   ├── Footer.jsx
│   │   └── Principal.jsx
│   ├── main.jsx
│   └── index.css
└── package.json
```

**Paso 01:** Importar todo lo que exporta el paquete **react-scroll** y se almacena en un objeto llamado **Scroll**.

**Paso 02:** Desestructurar el objeto **Scroll** para obtener específicamente la función **Link** / **Element**.

```
import * as Scroll from 'react-scroll';

const { Link } = Scroll;

const { Element } = Scroll;
```

# Paquete – react-scroll

## Paso 03: Parametrizar la función Link

```
<Link
  to="inicio"
  spy={true}
  smooth={true}
  duration={500}
  offset={-70}
>
  Inicio
</Link>
```

### Resumen:

- **<Link to="inicio" ... >**: Define el destino al que se desplazará la página.
- **spy={true}**: Vigila la posición para resaltar el enlace activo.
- **smooth={true}**: Realiza un desplazamiento suave entre las secciones.
- **duration={500}**: Establece que el desplazamiento dure 500 milisegundos.
- **offset={-70}**: Ajusta la posición de destino para que se detenga 70 píxeles antes de la sección especificada, útil cuando hay un encabezado fijo.

# Package – react-scroll

```
import React from 'react'
import * as Scroll from 'react-scroll';
const { Link } = Scroll;
export function Header() {
  return (
    <header>
      <nav>
        <ul>
          <li>
            <Link to="inicio" spy={true} smooth={true} duration={500}
              offset={-70}>
              Inicio
            </Link>
          </li>
        </ul>
      </nav>
    </header>
  )
}
```

# Paquete – react-scroll

```
import React from 'react'

import * as Scroll from 'react-scroll';
const { Element } = Scroll;

export function Contenido() {
  return (
    <main>
      <Element name="inicio">
        <h2>Inicio</h2>
        <p>Contenido de la Pagina de Inicio</p>
      </Element>
    </main>
  )
}
```





# GRACIAS

Presentó: Alvaro Pérez Niño  
Instructor Técnico

Correo: [aperezn@misena.edu.co](mailto:aperezn@misena.edu.co)

<http://centrodeserviciosygestionempresarial.blogspot.com/>

Línea de atención al ciudadano: 01 8000 910270

Línea de atención al empresario: 01 8000 910682



@SENAComunica

[www.sena.edu.co](http://www.sena.edu.co)