

# ReactJS

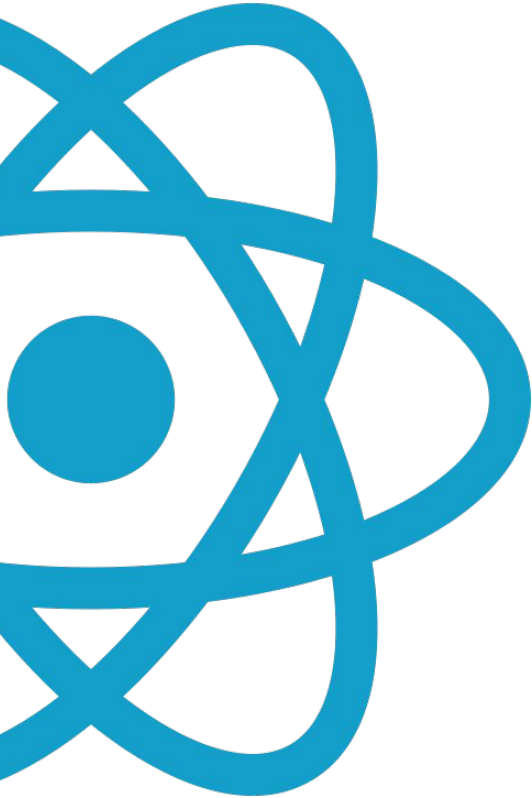
SPA con React Router DOM





# ¿Qué es una Single Page Applications(SPA)?

- Son aplicaciones web que despliegan todo su contenido en una sólo página web de manera dinámica, sin recarga de la misma o una petición de una página entera al servidor.
- En un proyecto de React, la librería más usada para crear Single Page Applications es **React Router**



# React Router DOM

Es la librería más popular para hacer enrutamiento en aplicaciones de React, implementando funcionalidades de navegación en una misma página.

Intercepta las peticiones que se hacen al servidor e inyecta el contenido dinámicamente (sin actualizar la página) renderizando los componentes según cómo hayamos configurado el enrutamiento de nuestro proyecto con la librería.

## *Instalación por cli:*

```
npm install react-router-dom@6
```

<https://www.freecodecamp.org/espanol/news/tutorial-de-react-router-version-6-como-navegar-a-otros-componentes-y-configurar-un-enrutador/>

## Configurando Router

Importamos el Componente **<BrowserRouter>** y envolvemos el componente de la aplicación en él (*Higher Order Component*)

```
import React from 'react'
import ReactDOM from 'react-dom/client'
import App from './App.tsx'
import './index.css'
import { BrowserRouter } from 'react-router-dom'

const rootElement = document.getElementById('root') as HTMLElement

ReactDOM.createRoot(rootElement).render(
  <React.StrictMode>
    <BrowserRouter>
      <App />
    </BrowserRouter>
  </React.StrictMode>
)
```

# Configurando Router

En el componente de la Aplicación Envuelta por el **BrowserRouter**, declaramos las rutas usando los componentes **<Routes>** y **<Route>** como se muestra abajo

```
export const App = (): JSX.Element => {  
  return (  
    <>  
      <AuthProvider>  
        <header>  
          <h1>React Router</h1>  
          <Nav />  
        </header>  
        <main>  
          <Routes>  
            <Route path="/" element={<Home />} />  
            <Route path="/login" element={<Login />} />  
            <Route path="/facts" element={<PrivatedRoute><FactsPage /></PrivatedRoute>} />  
            <Route path="/search-page" element={<PrivatedRoute><SearchPage /></PrivatedRoute>} />  
            <Route path="/tacos/:name" element={<PrivatedRoute><Taco /></PrivatedRoute>} >  
              <Route path="details" element={<TacoDetails />} /> { /* This is the Outlet */}  
            </Route>  
            <Route path="*" element={<NotFound />} />  
          </Routes>  
        </main>  
      </AuthProvider>  
    </>  
  )  
}
```

El componente **<Route>** recibe la propiedad **path** con el nombre de la ruta en la cual el componente que recibe en la propiedad **element** se mostrará.

## Navegando a las rutas

Los componente **<NavLink>** y **<Link>** permiten navegar a las rutas. Reciben la propiedad **to** para indicar a qué ruta enviará determinado link.

```
const Nav: React.FC = () => {  
  return (  
    <nav>  
      <ul>  
        <li>  
          <NavLink to="/" >  
            Home  
          </NavLink>  
        </li>  
        <li>  
          <NavLink to="/search-page">  
            Search Page  
          </NavLink>  
        </li>  
        <li>  
          <NavLink to="/facts" >  
            Facts  
          </NavLink>  
        </li>  
      </ul>  
    </nav>  
  )  
}
```

```
const Taco: React.FC = () => {  
  // todo: useParams() helps to get dynamic url segments keys /:segment  
  const { name } = useParams()  
  console.log(name)  
  return (  
    <div>  
      <h2>Taco</h2>  
      <h3>{name}</h3>  
      <Link to='details'>Ir a detalles</Link>  
      <Outlet /> { /* This is TacoDetails */}  
    </div>  
  )  
}
```

## Hay más!

Otras funcionalidades:

**useNavigate()** que permite navegar a una ruta de manera programaticamente.

**useParams()** que nos permite obtener el segmento dinámico que enviamos a una ruta en específico.

**useLocation()** que nos permite obtener la URL actual y sus parametros.

## Ver Documentación de React Router v6

<https://reactrouter.com/en/main>