Routeo en React

De la misma forma que manejamos el módulo de Route en Express, React habilita su módulo para poder a partir de una url y parámetros gestionar la visualización de módulos de nuestra App.

Paso 1: Instalar el módulo

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

Se instala de manera local en nuestro proyecto de React.

Paso 2: Importar el módulo en nuestra App

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

Paso 3: Analizando el módulo, vemos para que sirve cada uno de los componentes en nuestro proyecto.

Hay tres tipos de componentes en React Router: componentes del router, componentes de coincidencia de rutas y componentes de navegación.

Router

En el núcleo de cada aplicación React Router debería haber un componente *router*. Para proyectos web, react-router-dom proporciona routers <BrowserRouter> y <HashRouter>. Ambos crearán un objeto de historia especializado para usted. En general, debe usar un <BrowserRouter> si tiene un servidor que responde a las solicitudes y un <HashRouter> si está usando un servidor de archivos estático.

Route matching

Hay dos componentes de coincidencia de ruta: *<Route>* y *<Switch>*.

El *Route matching* se realiza comparando la prop de ruta de un <Route> con el nombre de ruta de la ubicación actual. Cuando una <Route> coincide, mostrará su contenido y cuando no coincida, dará nulo.

Un <Route> sin path siempre coincidirá.

```
<Route path='/about' component={About}/> // renders <About/>
<Route path='/contact' component={Contact}/> // renders null
<Route component={Always}/> // renders <Always/>
```

De manera recíproca a lo mencionado antes:

```
<Switch>
  <Route exact path='/' component={Home}/>
  <Route path='/about' component={About}/>
  <Route path='/contact' component={Contact}/>
  {/* when none of the above match, <NoMatch> will be rendered */}
  <Route component={NoMatch}/>
</Switch>
```

Se puede incluir un <Route> en cualquier lugar donde se desee presentar contenido según la ubicación. A menudo tendrá sentido enumerar una cantidad de <Route> posibles una al lado de la otra. El componente <Switch> se usa para agrupar <Route> s juntos.

```
<Switch>
  <Route exact path='/' component={Home}/>
  <Route path='/about' component={About}/>
  <Route path='/contact' component={Contact}/>
</Switch>
```

El <Switch> no es necesario para agrupar <Route> s, pero puede ser bastante útil. Un <Switch> iterará sobre todos sus elementos secundarios <Route> y solo representará el primero que coincida con la ubicación actual. Esto ayuda cuando los paths de varias rutas coinciden con el mismo pathname, al animar transiciones entre rutas, y al identificar cuando ninguna ruta coincide con la ubicación actual (para que pueda representar un componente "404").

Navegación

React Router proporciona un componente <Link> para crear enlaces en la app. Donde sea que represente un <Link>, se generará un anchor (<a>) en el código HTML.

El <NavLink> es un tipo especial de <Link> que puede calificarse a sí mismo como "activo" cuando su prop para coincide con la ubicación actual.