
# Coding Bootcamp

## Sprint 4

# Routes en React

# Routes

Por rutas se hace referencia a las diferentes URLs que puede tener una aplicación.

Ej: para un *e-commerce* podemos tener las siguientes rutas:

- https://onlineshopping.com/ – para acceder a la home.
- https://onlineshopping.com/**shopping-cart** – para acceder al carrito de compras.

React no tiene una manera nativa de manejar rutas. Existen varias librerías (externas) para hacerlo. Algunas de las más conocidas son:

- React Router.
- Redux First Router.
- React Navigation (React Native).
- React Mini Router.
- Etc.

# React Router v5

# React Router v5 (1/10)

Durante este curso se utilizará **React Router v5**, que es la librería de *routing* más utilizada. Además, tiene una muy buena documentación.

En React Router, la mayoría de los elementos de navegación son componentes de React, por lo cual se respeta el paradigma declarativo de desarrollo de aplicaciones.

Para poder utilizarlo, se necesita instalarlo en el proyecto por `npm` o `yarn`:

- `npm i react-router-dom`
- `yarn add react-router-dom`

# React Router v5 (2/10)

Para poder empezar a crear rutas en la aplicación se necesita importar el componente `BrowserRouter` y envolver a todos los componentes que necesiten manejar rutas dentro de él.

También se debe importar el componente `Route`, necesario para definir las rutas y los componentes que se mostrarán en cada ruta.

Cuando un `Route` haga *match* con la ruta definida, se renderizará el componente definido en `component`.

👉 Ver demo.

```
import { BrowserRouter, Route }
  from "react-router-dom";

const BasicExample = () => {
  return (
    <BrowserRouter>
      <Route exact path="/" component={Home} />
      <Route path="/about" component={About} />
    </BrowserRouter>
  );
}
```

La keyword `exact` se utiliza para que la ruta “/about” no haga match con “/”.

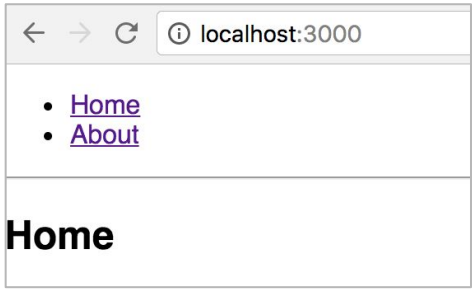
# React Router v5 (3/10)

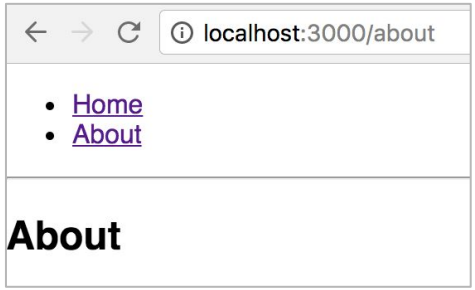
El componente `Link` sirve para generar links a otras rutas. Es muy similar al *anchor tag* de HTML. De hecho, `Link` termina renderizando un elemento `<a>` en el DOM.

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

const BasicExample = () =>
  <BrowserRouter>
    <ul>
      <li><Link to="/">Home</Link></li>
      <li><Link to="/about">About</Link></li>
    </ul>
    <Route exact path="/" component={Home} />
    <Route path="/about" component={About} />
  </BrowserRouter>;
```

# React Router v5 (4/10)

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

const Home = () => <h2>Home</h2>;
const About = () => <h2>About</h2>;

const BasicExample = () =>
  <Router>
    <ul>
      <li><Link to="/">Home</Link></li>
      <li><Link to="/about">About</Link></li>
    </ul>
    <Route exact path="/" component={Home} />
    <Route path="/about" component={About} />
  </Router>;
```



# React Router v5 (5/10)

El componente `NavLink` es muy similar a `Link` pero permite darle además una propiedad `activeClassName` para poder darle una clase CSS al enlace activo.

```
import { BrowserRouter, Route, NavLink } from "react-router-dom";

const BasicExample = () =>
  <BrowserRouter>
    <ul>
      <li><NavLink exact activeClassName="active" to="/">Home</NavLink></li>
      <li><NavLink activeClassName="active" to="/about">About</NavLink></li>
    </ul>
    <Route exact path="/" component={Home} />
    <Route path="/about" component={About} />
  </BrowserRouter>;
```

# React Router v5 (6/10)

Todos los componentes que son *rendereados* por `Route` recibirán en sus *props* una propiedad llamada `match`, que tiene acceso a varios datos de la ruta (ej: la URL).

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

const BasicExample = () =>
  <BrowserRouter>
    <ul>
      <li><Link to="/">Home</Link></li>
      <li><Link to="/about">About</Link></li>
    </ul>
    <Route exact path="/" component={Home} />
    <Route path="/about" component={About} />
  </BrowserRouter>;
```

```
const About = ({ match }) =>
  <div>
    <h2>About</h2>
    <h3>La url es: {match.url}</h3>
  </div>;
```

# React Router v5 (7/10)

A veces, puede ser necesario crear un ruta dentro de un sub-componente.
Además, se pueden crear rutas dinámicas usando la sintaxis `:parametro`. 👉 Ver demo.

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

const BasicExample = () =>
  <BrowserRouter>
    <ul>
      <li><Link to="/">Home</Link></li>
      <li><Link to="/about">About</Link></li>
    </ul>
    <Route exact path="/" component={Home} />
    <Route path="/users" component={Users} />
  </BrowserRouter>;
```

```
const Users = ({ match }) =>
  <div>
    <h2>Componente usuarios</h2>
    <Route path={`${match.url}/:userId`} component={User} />
    <Route exact path={match.url} render={() =>
        <p>No se seleccionó ningún usuario</p>
      }
    />
  </div>;

const User = ({ match }) =>
  <h3>Usuario: {match.params.userId}</h3>;
```

# React Router v5 (8/10)

En el caso de tener dos `Route` que puedan hacer *“match”* para la misma URL, se deberá envolver a los mismos en un `Switch`. De este modo, sólo se ejecutará la primer coincidencia.

En este caso, cuando se navegue a /about o /contact, si bien el tercer Route haría “match” con la URL, sólo se ejecutará la primer coincidencia.

```
const BasicExample = () => <BrowserRouter>
  <div>
    <ul>
      <li><Link to="/about">Sobre Nosotros (Ruta estática)</Link></li>
      <li><Link to="/contact">Contacto (Ruta estática)</Link></li>
      <li><Link to="/maria">María (Ruta dinámica)</Link></li>
    </ul>
    <Switch>
      <Route path="/about" component={About} />
      <Route path="/contact" component={Contact} />
      <Route path="/:user" component={User} />
    </Switch>
  </div>
</BrowserRouter>;
```

# React Router v5 (9/10)

El componente `Redirect` sirve para indicar que una ruta redirija a otra.

Por otro lado, notar que un `Route` sin `path` siempre hará *“match”* con cualquier URL. Por lo tanto, si se lo coloca en el último lugar dentro de un `Switch`, se puede usar para renderizar un componente de “Error 404 – Página no encontrada”.

```
<Switch>
    <Route exact path="/" component={Home} />
    <Redirect from="/estudiantes" to="/alumnos" />
    <Route path="/alumnos" component={Students} />
    <Route component={NoMatch} />
</Switch>
```

# React Router v5 (10/10)

En caso de precisar hacer un redireccionamiento del navegante sin que el mismo haga click sobre un `Link` o un `NavLink`, será necesario usar el hook `useHistory`.

Por ejemplo, esto podría ser útil para redireccionar al navegante a una página de gracias luego de realizar cierta acción.

```
const history = useHistory();

//...

history.push("/gracias")
```