



**UTN.BA**  
UNIVERSIDAD TECNOLÓGICA NACIONAL  
FACULTAD REGIONAL BUENOS AIRES

**Centro de  
e-Learning**

# Desarrollo en React JS

**Centro de e-Learning SCEU UTN - BA.**

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148

**[www.sceu.frba.utn.edu.ar/e-learning](http://www.sceu.frba.utn.edu.ar/e-learning)**



**UTN.BA**  
UNIVERSIDAD TECNOLÓGICA NACIONAL  
FACULTAD REGIONAL BUENOS AIRES

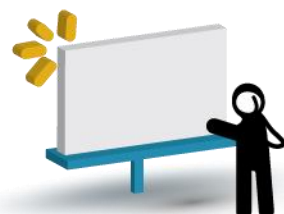
**Centro de  
e-Learning**

p. 2

## **Módulo II**

## **Componentes y Virtual DOM**

## **Unidad 3: Manejo de rutas**



**Centro de e-Learning SCEU UTN - BA.**

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148  
[www.sceu.frba.utn.edu.ar/e-learning](http://www.sceu.frba.utn.edu.ar/e-learning)

## Presentación:

En esta unidad aprenderemos como hacer que nuestra aplicación sea navegable, como configurar nuestro router y hacer los links entre páginas.

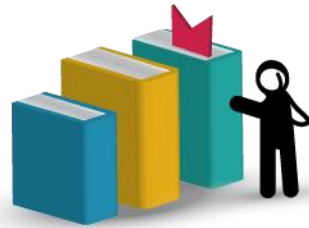
Veremos un ejemplo de cómo refactorizar nuestra app para que su estructura sea más legible y escalable.



## Objetivos:

### Que los participantes\*:

- Comprendan que es el router
- Aprendan cómo desarrollar el router en react
- Configuren su aplicación con una estructura más escalable y legible.



## Bloques temáticos\*:

- React router
- Configurar ruteo en app.js
- Configurar enlaces de dirección
- Recibir parámetros por URL
- useHistory
- NavLink
- Redirect
- Not Found



•

## Consignas para el aprendizaje colaborativo

En esta Unidad los participantes se encontrarán con diferentes tipos de actividades que, en el marco de los fundamentos del MEC\*, los referenciarán a tres comunidades de aprendizaje, que pondremos en funcionamiento en esta instancia de formación, a los efectos de aprovecharlas pedagógicamente:

- Los foros proactivos asociados a cada una de las unidades.
- La Web 2.0.
- Los contextos de desempeño de los participantes.

Es importante que todos los participantes realicen algunas de las actividades sugeridas y compartan en los foros los resultados obtenidos.

Además, también se propondrán reflexiones, notas especiales y vinculaciones a bibliografía y sitios web.

El carácter constructivista y colaborativo del MEC nos exige que todas las actividades realizadas por los participantes sean compartidas en los foros.



## Tomen nota\*

Las actividades son opcionales y pueden realizarse en forma individual, pero siempre es deseable que se las realice en equipo, con la finalidad de estimular y favorecer el trabajo colaborativo y el aprendizaje entre pares. Tenga en cuenta que, si bien las actividades son opcionales, su realización es de vital importancia para el logro de los objetivos de aprendizaje de esta instancia de formación. Si su tiempo no le permite realizar todas las actividades, por lo menos realice alguna, es fundamental que lo haga. Si cada uno de los participantes realiza alguna, el foro, que es una instancia clave en este tipo de cursos, tendrá una actividad muy enriquecedora.

Asimismo, también tengan en cuenta cuando trabajen en la Web, que en ella hay de todo, cosas excelentes, muy buenas, buenas, regulares, malas y muy malas. Por eso, es necesario aplicar filtros críticos para que las investigaciones y búsquedas se encaminen a la excelencia. Si tienen dudas con alguno de los datos recolectados, no dejen de consultar al profesor-tutor. También aprovechen en el foro proactivo las opiniones de sus compañeros de curso y colegas.



## **React router**

El componente router no es módulo oficial de react. Este nos permitirá hacer una web navegables, es decir poder redirigir entre distintas páginas.

## **Instalación**

Debemos ubicarnos dentro del directorio en el cual está nuestra aplicación y ejecutar:

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

**npm install react-router-dom**





## Configurar ruteo en app.js

En el archivo **app.js** debemos realizar lo siguiente:

```
import HomePage from "../Pages/HomePage"
import RegistroPage from "../Pages/RegistroPage";
import Menu from "../Components/Menu";
import {
  BrowserRouter as Router,
  Switch,
  Route
} from "react-router-dom";

function App() {
  return (
    <div className="App">
      <Router>
        <Menu />
        <Switch>
          <Route path="/registro" >
            <RegistroPage />
          </Route>
          <Route path="/" >
            <HomePage />
          </Route>
        </Switch>
      </Router>
    </div>
  );
}

export default App;
```

Como vemos debemos incluir los módulos **Route**, **Router (BrowserRouter)** y **Switch**

Si ejecutamos nuestra aplicación ingresara al contenido del componente **App** y si en la barra de direcciones colocamos **/** ingresara al contenido del componente **HomePage**

Si en la barra de direcciones colocamos **/registro** entonces se visualizara el contenido del componente **RegistroPage**

Vemos que el elemento **route** recibe 1 propiedad **path** (indica la url por la que se ingresara a ese componente) .



## Configurar enlaces de dirección

Cuando queramos armar los enlaces (lo que sería `<a href` ) lo debemos hacer usando el componente Link:

```
import React from "react"
import {
  Link
} from "react-router-dom";
function Menu () {
  return (
    <div>
      <ul>
        <li>
          <Link to="/">Home</Link>
        </li>
        <li>
          <Link to="/registro">Registro</Link>
        </li>
      </ul>
    </div>
  )
}
export default Menu;
```

En este ejemplo vemos el componente Link aplicado al menú, pero podemos aplicarlo a cualquier componente.

En la propiedad **to** colocamos la url destino (sería el href del componente Link)



## Recibir parámetros por URL

En caso de querer recibir parámetros por URL vamos a indicar en el componente route lo siguiente:

```
<Route path="/detalle/:id" >
  <DetallePage />
</Route>
```

En caso de ser un componente de tipo clase este elemento **:id** se enviará al componente **DetallePage** como una propiedad, de la siguiente manera:

```
constructor(props){
  super(props)
  console.log(this.props.params.id)
}
```

En caso de ser un componente tipo función podemos utilizar el hook **useParams()** de la siguiente manera:

```
import React from "react"
import {
  useParams
} from "react-router-dom";
function DetallePage(){
  const {id} = useParams()
  console.log(id)
  return(
    <div>DetallePage</div>
  )
}
export default DetallePage;
```

En ambos casos se hace referencia al elemento **id** porque se identifico el parámetro de esa manera en el route (:id)



## useHistory

Hook que permite acceder al histórico de navegación. Podemos utilizar para realizar una redirección (push) desde la lógica del componente.

Por ejemplo:

```
import React from "react"
import {
  useHistory
} from "react-router-dom";

function RegistroPage () {
  let history = useHistory();

  function handleClick() {
    history.push("/home");
  }

  return(
    <div>
      Registro
      <button type="button" onClick={handleClick}>
        Go home
      </button>
    </div>
  )
}

export default RegistroPage;
```

En este caso al hacer click en **Go To Home** llamamos a la función **handleClick** y en dicha función hacemos un `history.push("/home")` el cual nos redirigirá a la pagina de inicio.



## NavLink

Este componente tiene una serie de propiedades utilizadas a la hora de armar un navbar en nuestra aplicación. Ejemplo de utilización en el Menu:

```
function Menu () {  
  return(  
    <div>  
      <NavLink to="/" exact >Home</NavLink>  
      <NavLink to="/registro">Registro</NavLink>  
    </div>  
  )  
}  
  
export default Menu;
```

Se puede ver la totalidad de las propiedades en

<https://reactrouter.com/web/api/NavLink>

## Redirect

Permite realizar redirecciones en la lógica de ruteo. Lo aplicamos en el Switch en el cual se realiza la comparación de las direcciones:

```
<Redirect from="/home" to="/" />  
<Route path="/">  
  <HomePage />  
</Route>  
<Route path="/" >  
  <HomePage />  
</Route>
```

En el ejemplo cuando recibamos **/home** redirigirá a **/**

<https://reactrouter.com/web/api/Redirect>



## Not Found

Para hacer la lógica de ruteo de nuestra pagina 404 Not Found debemos declarar la siguiente regla:

```
<Router>
<Menu />
  <Switch>
    <Route path="/" exact >
      <HomePage />
    </Route>
    <Route path="/registro" exact >
      <RegistroPage />
    </Route>
    <Route path="/detalle/:id" exact >
      <DetallePage />
    </Route>
    <Redirect from="/home" to="/" exact />

    <Route path="*">
      <NotFoundPage />
    </Route>
  </Switch>
</Router>
```

De esta manera estaremos accediendo a cualquier url no identificada en reglas anteriores.



## Bibliografía y Webgrafía utilizada y sugerida

Fedosejev, A. (2015). React.js Essentials (1 ed.). EEUU, Packt.

Amler, . (2016). ReactJS by Example (1 ed.). EEUU, Packt.

Stein, J. (2016). ReactJS Cookbook (1 ed.). EEUU, Packt.

<https://medium.com/@pshrmn/a-simple-react-router-v4-tutorial-7f23ff27adf>

<https://reactrouter.com/web/guides/quick-start>



## Lo que vimos:

En esta unidad aprendimos como hacer que nuestra aplicación sea navegable, como configurar nuestro router y hacer los links entre páginas.



## Lo que viene:

En la próxima unidad aprenderemos cómo manejar eventos y vistas con react js

