



<Link /> 🤪

React router - <Link />

Una vez que tenemos definidas las rutas de nuestra aplicación utilizando el componente **<Route />**. Si bien podemos navegar cambiando la url desde la barra de navegación de nuestro navegador, lo normal o ideal es hacerlo desde un menú de navegación. Ahí entra en juego el nuevo componente **<Link />**.

Este componente automáticamente nos crea un elemento a que nos cambia la ruta o path al hacerle click.

El componente acepta un atributo to, dónde podemos indicarle la ruta a la cual nos redirecciona una vez que se hace click.

useNavigate() 🤯

React router - useNavigate

No siempre deseamos navegar desde un `<Link />`, hay veces que deseamos navegar desde código, una vez que suceda algo en el flujo de nuestra aplicación.

Por ejemplo una vez que el usuario inició sesión, llevarlo a la ruta del dashboard.

Para esto debemos usar el **hook de react router** que es **useNavigate**, y nos permite navegar desde código a cualquier ruta definida.

React router - useNavigate



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

function Home() {
  const navigate = useNavigate();

  const handleGoDashboard = () => {
    navigate('/dashboard');
  }
  return (
    <div>
      <h1>Home</h1>
      <button onClick={handleGoDashboard}>Ir a Dashboard</button>
    </div>
  );
}
```

useParams() 🤖

React router - useParams

Existe la posibilidad de **definir parámetros en nuestras rutas**, por ejemplo al ampliar un artículo o un producto en nuestra web.

Esto se puede realizar mediante simplemente **poner en el path de nuestro <Route />, dos puntos (:) y el alias / nombre que va a tener ese parámetro.**

```
import { Route, Routes } from 'react-router-dom';

function App() {
  return (
    <div className="App">
      <Routes>
        <Route path="/" element={<Home />} />
        <Route path="details/:id" element={<Details />} />
      </Routes>
    </div>
  );
}
```


React router - useParams

Luego en el componente destino, en este ejemplo **<Details />** podemos acceder a esos parámetros.

Utilizando el **hook de react router useParams()**, que nos **retorna un objeto** con todos los parámetros actuales.

```
import { useParams } from 'react-router-dom';

function Details(){
  const params = useParams();

  return <h1>Details {params.id}</h1>
}
```

LINKS

LINKS

- **React**
 - <https://reactrouter.com/docs/en/v6/getting-started/overview#configuring-routes>



[gustavgueez](#)



[gustavgueez](#)



[gustavgueez](#)

GUSTAVO RODRIGUEZ

FULL STACK DEVELOPER
SOLCRE