



JSX en variables / constantes 🤯

Hay momentos que el JSX de nuestro componente **contiene muchas condiciones** y es difícil manejar que JSX retornar, ya que los **ifs** junto al JSX no nos resulta cómodo.

JSX nos permite **utilizar las variables y constantes** de JavaScript para **guardar JSX** y simplificar la lógica de nuestro componente. Luego simplemente **imprimimos la variable con {}** y el JSX funciona como siempre.

JSX en variables / constantes - Ejemplo

```
import { TaskList } from "../TaskList";

export function TaskContainer() {
  const tareas = [];
  let contenido;

  if (tareas.length > 0) {
    contenido = <TaskList />;
  } else {
    contenido = <p>No hay tareas</p>;
  }

  return (
    <div className="container container-tasks">
      <h2 className="subtitle">Listado de tareas</h2>
      {contenido}
    </div>
  );
}
```

Listas



Primero, vamos a revisar **cómo transformas listas en Javascript.**

Dado el siguiente código, usamos la función **map()** para tomar un array de numbers y duplicar sus valores. Asignamos el nuevo array devuelto por map() a la variable doubled y la mostramos:

```
const numbers = [1, 2, 3, 4, 5];  
const doubled = numbers.map((number) => {  
  return number * 2;  
});  
console.log(doubled);  
//[2, 4, 6, 8, 10]
```

Listas - JSX

En React, **transformar arrays en listas** de elementos es casi idéntico.

Recorreremos el array `numbers` usando la función **`map()`** de Javascript. Devolvemos **un elemento ``** por cada ítem . Finalmente, asignamos el array de elementos resultante a `listItems`

```
function Lista(){
  const numbers = [1, 2, 3, 4, 5];
  const listItems = numbers.map((number) => {
    return <li>{number}</li>;
  });

  return (
    <ul>
      {listItems}
    </ul>
  )
}
```

Librerías externas ⚡

Si deseamos **incluir funcionalidades y componentes de terceros**, lo podemos realizar utilizando el **manejador de paquetes npm** y/o yarn.

Por ejemplo podríamos instalar una librería que se encargue del ruteo de mi app, sin la necesidad de implementar esta funcionalidad. **Solo instalamos la librería y la utilizamos.**

Las librerías son piezas de código realizadas por terceros que podemos traer a nuestra aplicación para usar.

Librerías externas - Pasos

1. Abrir la terminal o la línea de comandos de nuestro SO.
2. **Posicionarnos en la carpeta del proyecto** donde queremos instalar la librería.
3. Ejecutamos el comando **npm install nombre-librería** (ejemplo `npm install react-router-dom@6`).
4. El comando instala la librería y **modifica el archivo package.json**, guardando una referencia de que nuestro proyecto tiene como **dependencia tal librería externa**.

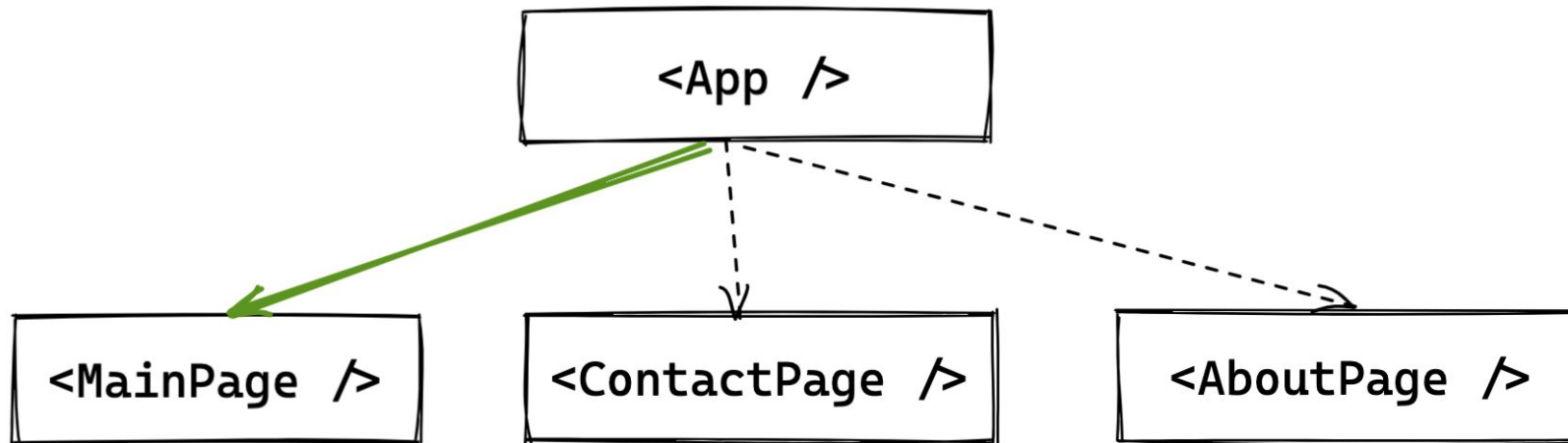
Navegación

En React podemos pensar en la navegación en términos simples: lo único que debemos hacer es cambiar un componente por otro. Podríamos tener un componente por “página” de nuestro sitio, por ejemplo:

- `<MainPage />`
- `<ContactPage />`
- `<AboutPage />`

El cambio de componentes en pantalla se puede lograr de varias maneras, por ejemplo **utilizando el estado de los componentes** (useState) y el **dibujado condicional**.

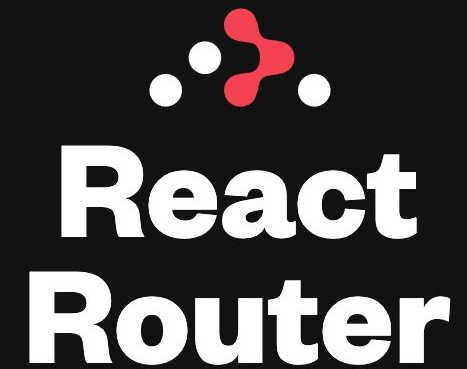
Navegación - Intro



Hay muchas formas de implementar una navegación en nuestra app, desde utilizando el estado como vimos recién o otra forma de hacerlo es utilizando la **librería externa react-router**, la cual nos facilita bastante el proceso.

Esta librería no es parte de React, pero es la más usada para resolver este problema.

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



Navegación - React router

```
1  import * as React from "react";
2  import * as ReactDOM from "react-dom";
+  import { BrowserRouter } from "react-router-dom";
4  import "./index.css";
5  import App from "./App";
6  import * as serviceWorker from "./serviceWorker";
7
8  ReactDOM.render(
+    <BrowserRouter>
+      <App />
+    </BrowserRouter>,
12  document.getElementById("root")
13  );
```

```
1  import * as React from "react";
+  import { Routes, Route, Link } from "react-router-dom";
3  import "./App.css";
4
5  function App() {
6    return (
7      <div className="App">
+        <h1>Welcome to React Router!</h1>
+        <Routes>
+          <Route path="/" element={<Home />} />
+          <Route path="about" element={<About />} />
+        </Routes>
13     </div>
14   );
15 }
```

LINKS



LINKS

- **React**

- <https://es.reactjs.org/docs/lists-and-keys.html>
- <https://reactrouter.com/>
- <https://www.npmjs.com/>
- <https://reactrouter.com/docs/en/v6/getting-started/installation#basic-installation>



[gustavgueez](#)



[gustavgueez](#)



[gustavgueez](#)

GUSTAVO RODRIGUEZ

FULL STACK DEVELOPER
SOLCRE