

Guía de estudio 1 - React Router I



¡Hola! Te damos la bienvenida a esta nueva guía de estudio.

Detalles importantes del módulo

Los ejercicios de este módulo fueron contruidos con:

- Node 15
- React 18
- React router 6

¿Qué aprenderemos en esta guía?

La siguiente guía de estudio tiene como objetivo recordar y repasar los contenidos que hemos visto en clase y dentro de los cuales se encuentran:

- ¿Qué es React Router y qué problemas resuelve?
- ¿Cómo instalar el paquete react-router-dom?
- ¿Cómo agregar navegación basada en rutas a una aplicación en React?
- ¿Cómo agregar una ruta por defecto a una aplicación?

¡Vamos con todo!



Tabla de contenidos

Guía de estudio 1 - React Router I	1
Detalles importantes del módulo	1
¿Qué aprenderemos en esta guía?	1
Tabla de contenidos	2
La navegación en una página estática	3
¿Qué es React Router?	4
Setup	4
Mi primer enrutador	5
¡Manos a la obra! - Navbar enrutado	7
Creación de vistas	7
¡Manos a la obra! - Vistas Registro y Login	9
Creación de vistas	9
¿Qué es una SPA?	10
Preguntas de cierre	11



¡Comencemos!

La navegación en una página estática

Cuando navegamos en la web, por ejemplo, haciendo clic en un link dentro de un sitio, podemos ver que la URL mostrada en el navegador cambia.



Imagen 1. Cambiando una URL
Fuente: Desafío Latam

Lo anterior sucede en muchos sitios web, especialmente en aquellos sitios web estáticos, como los que hemos hecho solo con HTML y CSS. En estos sitios cada uno de estos archivos tiene una dirección web que coincide con el nombre del archivo.

Por ejemplo:

- Si tenemos la página *blog.html* en la raíz de un proyecto con un dominio llamado *ejemplo.com* la URL de la página será: *ejemplo.com/blog.html*
- Si tenemos la página *blog.html* en la subcarpeta *paginas* la URL será: *ejemplo.com/paginas/blog.html*

Cuando trabajamos con React funciona de la misma forma, cuando entramos a *localhost:3000* (o el puerto en que corra el proyecto) estamos cargando el *index.html* donde están los componentes de nuestro proyecto.

Si quisiéramos una segunda dirección dentro del proyecto tendríamos que crear una segunda página similar a *index*. Pero de esta forma, al movernos de una página a la otra, tendría que volver a cargar todas las componentes, haciendo el sitio más lento. Hay otra opción mejor, usar un sistema de enrutamiento como *react-router*.

¿Qué es React Router?



React Router es un paquete disponible en NPM que nos permite dividir fácilmente una aplicación de React en distintas URLs.

Con React Router podemos gestionar diferentes rutas enlazadas a diferentes vistas, mejorando de esta manera la usabilidad de nuestra aplicación y la experiencia de nuestros usuarios.

Setup

En esta unidad trabajaremos sobre un proyecto base llamado Dogstel que consiste en una página web de un hotel para perros.

A continuación, veremos los pasos necesarios para inicializar el proyecto:

- Descarga el proyecto Dogstel
- Entra el proyecto desde el terminal e instala los paquetes con `npm install`
- Levanta el proyecto con `npm start`

Si el navegador no se levantó automáticamente, abre una página nueva en localhost:3000. Al cargar la página deberíamos ver un navbar con el texto Dogstel.

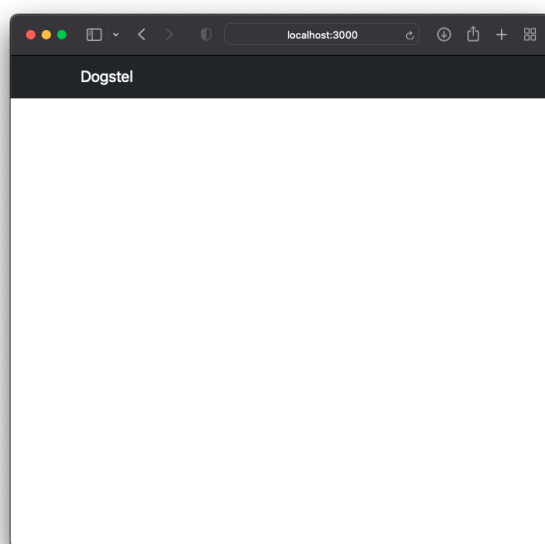


Imagen 2. El proyecto en estado inicial
Fuente: Desafío Latam

Al abrir el proyecto con el editor de código, encontraremos los siguientes archivos:

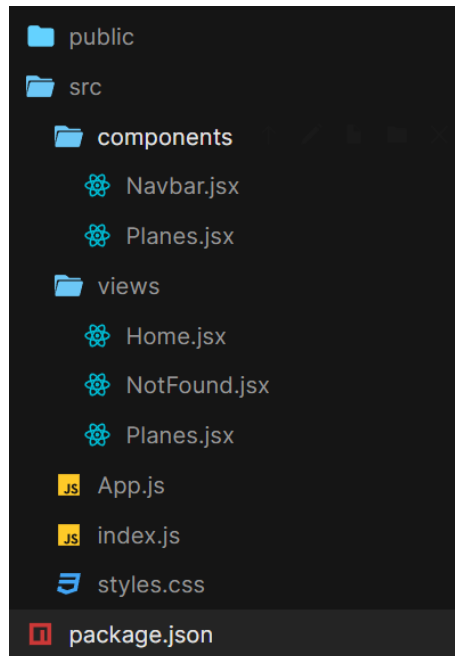


Imagen 3. Archivos del proyecto Dogstel
Fuente: Desafío Latam

El objetivo será agregar react router al proyecto con los componentes que ya tenemos a disposición.

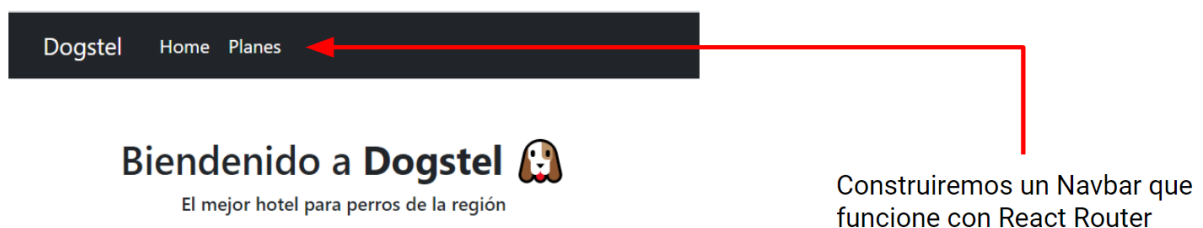


Imagen 3. Navbar a construir
Fuente: Desafío Latam

Mi primer enrutador

Una vez descargado el proyecto, instala por terminal el paquete de *react-router-dom*

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

Luego modifica el código del componente App para importar y agregar el componente **BrowserRouter** del paquete react-router-dom y el componente Navbar.

BrowserRouter es el componente encargado de mantener actualizado el UI en función de la URL, este debe envolver a toda la aplicación no solo la barra de navegación. Todos los componentes se agregan dentro de BrowserRouter.

```
/* App.js */

import 'bootstrap/dist/css/bootstrap.min.css'
import { BrowserRouter } from "react-router-dom"

function App() {
  return (
    <div className="App">
      <BrowserRouter>
        {/* Toda la app */}
      </BrowserRouter>
    </div>
  );
}

export default App;
```

Ahora dentro del componente *Navbar* incluido en el proyecto agrega 2 Links: **Home** y **Planes**

```
/* components/nNavbar.jsx */

import { Link } from "react-router-dom";
import { Navbar, Container } from "react-bootstrap";

export default function Navigation() {
  return (
    <>
      <Navbar bg="dark" variant="dark">
        <Container className="justify-content-start">
          <Navbar.Brand>Dogstel</Navbar.Brand>
          <Link to="/" className="text-white ms-3 text-decoration-none">
            Home
          </Link>
          <Link to="/planes" className="text-white ms-3 text-decoration-none">
            Planes
          </Link>
        </Container>
      </Navbar>
    </>
  );
}
```



El componente **Link** funciona similar a la etiqueta de hipervínculo(a) de HTML, la diferencia radica en que este componente produce el redireccionamiento con React Router y evita la recarga del navegador.

Ahora levantemos nuestra aplicación y probemos el Navbar presionando en la opción “Planes”

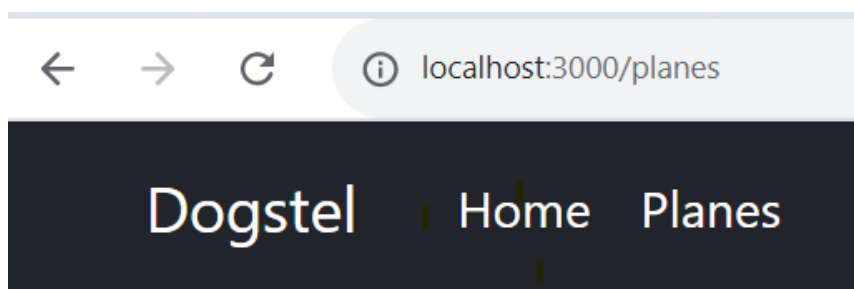


Imagen 4. URL cambiada al consultar una ruta
Fuente: Desafío Latam

Como se puede observar, la URL de la aplicación cambia en función a la ruta que accedemos.

Con este redireccionamiento evitamos tener que volver a consumir todos los recursos de nuestra aplicación al evitar que se recargue el navegador. React Router se encargará de actualizar parcialmente el DOM en función a la ruta que consultemos.



¡Manos a la obra! - Navbar enrutado

Crea una nueva aplicación de React y:

1. Modifica la estructura de carpetas y archivos según el ejemplo anterior
2. Instala el paquete *react-router-dom*
3. Crea un componente Navbar con 2 componentes Links que dirijan a Registro y Login
4. En el componente App, importa e incluye en el template el BrowserRouter y el Navbar
5. Prueba tu enrutador confirmando que cambia la URL en función a la ruta que consultas

Creación de vistas

Ahora que nuestra aplicación puede redireccionar al usuario a diferentes rutas, es momento de definir en nuestro enrutador cuáles son los componentes que utilizará como vistas.

Para esto modifica el archivo App nuevamente agregando los componentes Routes, Route y las 2 vistas incluidas en el proyecto

```
import "bootstrap/dist/css/bootstrap.min.css";
import { BrowserRouter, Routes, Route } from "react-router-dom";

import Home from "../views/Home";
import Planes from "../views/Planes";

import Navbar from "../components/Navbar";

function App() {
  return (
    <div className="App">
      <BrowserRouter>
        <Navbar />

        <Routes>
          <Route path="/" element={<Home />} />
          <Route path="/planes" element={<Planes />} />
        </Routes>
      </BrowserRouter>
    </div>
  );
}
export default App;
```

El componente **Routes** funciona como un envoltorio de rutas.

El componente **Route** define una ruta dentro de nuestro enrutador y tiene la siguiente estructura:

```
<Route path="/<ruta>" element={< Componente(vista) />} />
```

En donde el atributo **path** define la ruta y el atributo **element** el componente que se mostrará cuando coincida el **path** con la URL del navegador.

Aunque se escriben varios componentes **Route**, solo se renderiza 1.

Ahora probemos nuevamente nuestra aplicación y comprobemos que se muestran ambas vistas en ambas rutas:

Home

Dogstel Home Planes

Biendenido a **Dogstel** 🐕

El mejor hotel para perros de la región

Planes

Dogstel Home Planes

Tenemos planes especiales para ti y su mascota

Plan - Básico



Plan - Estándar



Plan - Guau!



Imagen 5. Vistas: Home y Planes

Fuente: Desafío Latam



¡Manos a la obra! - Vistas Registro y Login

En la aplicación creada en el ejercicio anterior:

1. Crea 2 vistas correspondientes al Registro y al Login.
2. Agrega las rutas correspondientes en el enrutador
3. Comprueba en el navegador que ambas vistas se muestran correctamente según las rutas consultadas

Creación de vistas

La ruta por defecto nos ayudará a poder personalizar y devolver una vista en caso de que el usuario consulte nuestra página web con una URL que no coincida con ninguna de nuestras rutas.

Para agregar la ruta por defecto incluye la vista **NotFound** que sea utilizado por un **Route** nuevo que tenga como path un asterisco: *

```
import "bootstrap/dist/css/bootstrap.min.css";
import { BrowserRouter, Routes, Route } from
"react-router-dom";

import Home from "./views/Home";
import Planes from "./views/Planes";
import NotFound from "./views/NotFound";
import Navbar from "./components/Navbar";

function App() {
  return (
    <div className="App">
```

```
<BrowserRouter>
  <Navbar />

  <Routes>
    <Route path="/" element={<Home />} />
    <Route path="/planes" element={<Planes />} />
    <Route path="*" element={<NotFound />} />
  </Routes>
</BrowserRouter>
</div>
);
}

export default App;
```

Ahora si consultamos nuestra aplicación con una ruta extraña o que simplemente no coincida con ningún componente **Route**, obtenemos la vista **NotFound**



La ruta que intentas consultar no existe :/

Imagen 6. Ruta por defecto
Fuente: Desafío Latam

¿Qué es una SPA?

Una Single Page Application (SPA) es una aplicación que ofrece la navegación de su contenido basada en rutas dentro de una misma página web sin que ésta se tenga que recargar.

Este tipo de aplicaciones simula el comportamiento y fluidez de una aplicación de escritorio al renderizar parcialmente los componentes interpretados como vistas, en función a la ruta que el usuario consulte.

Se dice que es de “Una sola página” puesto que toda la aplicación se carga dentro de un HTML final.

Por lo tanto, podemos construir fácilmente aplicaciones SPA con React.

Preguntas de cierre

Reflexiona:

- ¿Qué problema resuelve React Router?
- Menciona 3 diferentes componentes de React Router Dom
- ¿Cuál es el rol del componente Browser Route?
- ¿Cuál es la diferencia entre el componente Link y una etiqueta de hipervínculo?
- ¿Cómo configuramos una ruta por defecto?
- ¿Qué es una SPA?

