

Bootcamp de Full Stack

Bienvenidos a la clase N°42

- React Router
 - Concepto
 - BrowserRouter
 - useLocation
 - useNavigate
 - useSearchParams
 - useParams
- Desafío Semanal F2-N14

React Router

Concepto

React Router es una biblioteca de enrutamiento para aplicaciones React que permite la navegación entre vistas o componentes sin recargar la página. Implementa el concepto de Single Page Application (SPA) al permitir el cambio de rutas en el navegador de forma dinámica y eficiente, manteniendo el estado de la aplicación.

React Router gestiona la correspondencia entre URLs y componentes, y permite funcionalidades como navegación condicional, rutas anidadas, redirecciones, rutas protegidas y paso de parámetros a través de la URL.

El paquete **react-router-dom** es la implementación de React Router específicamente para aplicaciones web. Este paquete extiende el núcleo de react-router, que es independiente del entorno.

Instalación: `npm install react-router-dom`

Documentación oficial: <https://reactrouter.com/home>



React Router

BrowserRouter

El componente **BrowserRouter** proporciona un enrutador basado en la API history del navegador, ideal para aplicaciones SPA con URLs limpias (sin hash).

Envuelve toda la aplicación para habilitar navegación y enrutamiento dinámico entre componentes.

Usos comunes:

- Configurar enrutamiento principal en una app React.
- Permitir navegación con Link, useNavigate o useLocation.

Sintaxis

```
<BrowserRouter>  
  <App />  
</BrowserRouter>
```



Hooks

useLocation

El hook **useLocation** permite acceder al objeto de ubicación actual (location) que contiene detalles sobre la URL activa (como pathname, search y hash).

Devuelve un objeto inmutable que representa la ubicación actual.

Usos comunes:

- Leer el pathname actual.
- Obtener o reaccionar a los parámetros de búsqueda (?query=valor).
- Controlar cambios de ruta.

Sintaxis

```
const location = useLocation();
```



Hooks

useNavigate

El hook **useNavigate** permite realizar navegación programática entre rutas. Reemplaza el uso de useHistory en versiones anteriores.

Devuelve una función que se puede usar para redirigir a otras rutas de forma imperativa.

Usos comunes:

- Redirigir luego de una acción (login, submit).
- Volver atrás o avanzar en el historial.

Sintaxis

```
const navigate = useNavigate();
```



BREAK

Descansemos 10 minutos



Hooks

useSearchParams

El hook **useSearchParams** permite leer y modificar los parámetros de búsqueda (?clave=valor) en la URL.

Características:

1. Un objeto URLSearchParams (para leer valores).
2. Una función para actualizar los parámetros.

Usos comunes:

- Leer filtros, búsquedas u opciones desde la URL.
- Actualizar parámetros sin recargar la página.

Sintaxis

```
const [ searchParams, setSearchParams ] = useSearchParams();
```



Hooks

useParams

El hook **useParams** permite acceder a los parámetros dinámicos definidos en las rutas (/usuario/:id).

Devuelve un objeto con los valores extraídos de la URL.

Usos comunes:

- Obtener el id u otros parámetros desde la ruta.
- Cargar datos específicos según la URL.

Sintaxis

```
const { id } = useParams();
```



Desafío Semanal

F2-N14



- **Aclaración:** Esta actividad es de **carácter opcional** y se lleva a cabo de forma **individual** o en **parejas de estudiantes**, fuera del horario de clase.
- **Evaluación:** La realiza el **tutor**.
- **Fecha límite:** La **entrega** se debe realizar antes de iniciar la **siguiente clase**.
- **Consigna y recursos:** El siguiente enlace contiene todo lo necesario para el desarrollo de la actividad:
https://drive.google.com/drive/folders/1CreCtHoCIq5MLURpPuS7wRvloyAx0n3v?usp=drive_link
- **Entrega:** Se hace de forma individual en formato ZIP y se sube a la carpeta personal de Google Drive.
- **Registro de entrega:** Se debe registrar en este formulario: <https://forms.gle/VeRUKYcfjCtv4afXA>

CIERRE DE CLASE

Continuaremos en la próxima clase

