

REACT: Router

DEVinHouse

Parcerias para desenvolver a sua carreira

SENAI

<LAB365>

AGENDA

- Router

ROUTER: O que são rotas?

Quando acessamos uma aplicação esperamos que ela tenha **rotas** para acessar cada página.

Por exemplo, uma rede social, com rotas para cada usuário:



ROUTER: “react-router-dom”

Numa aplicação React podemos criar rotas utilizando a biblioteca **react-router-dom**, que deve ser instalada no projeto.

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

Com essa biblioteca é possível criar funcionalidades na navegação entre páginas, como:

- rotas públicas e privadas
(ex: usuário logado ou não)
- redirecionamento em caso de erro
(ex: página não encontrada)
- redirecionamento para página diferente da digitada originalmente
(ex: usuário digita **"/home"** e é redirecionado para **"/"**)

ROUTER: Recursos

Para que tudo isso seja possível, importaremos 4 recursos da biblioteca:

- **BrowserRouter**: informa a nossa aplicação que teremos roteamento de componentes
- **Routes**: irá receber todas as rotas
- **Route**: **associa** a rota a cada componente
- **Navigate**: permite o redirecionamento de rotas

```
import { BrowserRouter, Routes, Route, Navigate } from "react-router-dom";
```

ROUTER: Estrutura

Essa é a estrutura do arquivo de rotas:

```
export const Rotas = () => {  
  return (  
    <BrowserRouter>  
      <Routes>  
        <Route path="/" element={<Home />} />  
        <Route path="/home" element={<Navigate replace to="/" />} />  
      </Routes>  
    </BrowserRouter>  
  );  
}
```

ROUTER: <Route>

Cada rota (Route) recebe 2 atributos:

- **path:** é o caminho na URL que precisa ser acessado para exibir o componente
- **element:** recebe o componente que será exibido ao acessar a rota

```
<Route path="/" element={<Home />} />
```


ROUTER: <Route>

Mas é possível que o usuário digite o caminho errado na URL
(ex: queria acessar **"/home"** e digita **"/hame"**)

Seria extremamente difícil prever todos esses casos
e redirecionar para a página correta,
por isso podemos usar no path o parâmetro *****
que redireciona o usuário para um componente específico.

```
<Route path="/" element={<NaoEncontrado />} />
```

ROUTER: <Route>

Em alguns casos precisamos passar **parâmetros** para uma rota.

Exemplos:

- ver detalhes de um produto
- ver a página de um determinado usuário

Para isso, adicionamos no path "**:nomeDoParametro**"

```
<Route path="/pago/:userName" element={<Usuario />} />
```

ROUTER: Link

Criadas as rotas, precisamos **acessá-las** na nossa aplicação e para isso utilizamos mais um recurso da biblioteca: **Link**. Ele permitirá a alteração da URL para que a nova página seja carregada.

Utilizaremos nele o parâmetro **to="rotaQueSeraAcessada"**

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

export const NaoEncontrado = () => {
  return (
    <div>
      <h1>Página não encontrada</h1>
      <p>
        <Link to="/">Voltar para Home</Link>
      </p>
    </div>
  );
};
```

ROUTER: Link

Também podemos usar o **Link** em outras situações, como voltar para a página anterior.

```
import { Link } from "react-router-dom";
import "./Cabecalho.css";
import logo from "../../assets/clamedfy.png";
import { useUsuario } from "../../contexts/usuarioContext";

export const Cabecalho = () => {
  const { nome } = useUsuario();

  return (
    <header>
      <Link to="/">
        <img src={logo} alt="Clamedfy" />
      </Link>
      <div>
        <h1>{nome}</h1>
        <Link to={-1}>
          <button type="button">Voltar</button>
        </Link>
      </div>
    </header>
  );
};
```

ROUTER: useNavigate

Mas em determinadas situações não é possível usar o Link (ex: quando fazemos o redirecionamento dentro de uma função)

Nesse caso, usamos um hook da biblioteca chamado `useNavigate`

```
import { useNavigate } from 'react-router-dom'
import { useUsuario } from "../contexts/usuarioContext";

export const Gratuito = () => {
  const navigate = useNavigate();
  const { alteraTipoDePlano } = useUsuario();

  const mudarParaPremium = () => {
    alteraTipoDePlano("premium");
    navigate("/pago");
  }

  return (
    <div>
      <p className="TipoDePlano">Você está usando o plano gratuito.</p>
      <button className="BotaoAssine" onClick={mudarParaPremium}>
        APROVEITE 3 MESES GRÁTIS
      </button>
    </div>
  );
};
```

ROUTER: useNavigate

Também podemos usar o `useNavigate` em outras situações, como voltar para a página anterior.

```
import { Link, useNavigate } from 'react-router-dom'
import './Cabecalho.css'
import logo from '../../assets/clamedfy.png'
import { useUsuario } from '../../contexts/usuarioContext'

export const Cabecalho = () => {
  const { nome } = useUsuario()
  const navigate = useNavigate ()

  return (
    <header>
      <Link to="/">
        <img src={logo} alt="Clamedfy" />
      </Link>
      <div>
        <h1>{nome}</h1>
        <button type="button" onClick={() => navigate(-1) || navigate("/")}>Voltar</button>
      </div>
    </header>
  )
}
```

ROUTER: useParams

Nós vimos antes que é possível passar parâmetros para uma rota.



Para **acessar uma rota com parâmetro** usamos o hook **useParams**

```
export const Musica = ({ musica }) => {
  const { id, nome, artista } = musica;
  const navigate = useNavigate();

  const irParaDetalhesDaMusica = () => {
    navigate(`./musica/${id}`);
  };

  return (
    <div className="Musica">
      /* com useNavigate */
      /* <button onClick={irParaDetalhesDaMusica}>{nome}</button> - {artista} */

      /* com Link */
      <button>
        <Link to={`./musica/${id}`}>
          {nome}
        </Link>
      </button>
    </div>
  );
};
```

```
DetalhesDaMusica.js pages
05-router > src > pages > DetalhesDaMusica.js > ...
...
1 import { useParams } from 'react-router-dom';
2 import { useMusicas } from '../contexts/musicasContext';
3
4 export const DetalhesDaMusica = () => {
5   const { idDaMusica } = useParams();
6   const { musicas } = useMusicas();
7   const musica = musicas?.find((musica) => musica.id === Number(idDaMusica));
8
9   return (
10     <div className="Detalhes">
11       <h2>{musica.nome}</h2>
12       <p>{musica.artista}</p>
13     </div>
14   );
15 }
```

ROUTER: Rotas privadas

É possível **criar rotas privadas**, que só poderão ser acessadas em determinadas condições.
(ex: se o usuário estiver logado)

Nesse caso precisamos usar 2 recursos da biblioteca: **Outlet** e **Navigate**

```
import { Outlet, Navigate } from "react-router-dom";
import { useUsuario } from '../contexts/usuarioContext'

export const RotaPrivada = () => {
  const { tipoDePlano } = useUsuario();

  return tipoDePlano === "premium" ? <Outlet/> : <Navigate to="/" replace />;
};
```

E para indicar que uma ou mais rotas são privada:

```
<Route element={<RotaPrivada />}>
  <Route path="/pago" element={<Pago />} />
  <Route path="/pago/:userName" element={<Usuario />} />
</Route>
```


INTERVALO DE AULA

DEV!

Finalizamos o nosso primeiro período de hoje. Que tal descansar um pouco?!

Nos vemos em 20 minutos.

Início: 20:20

Retorno: 20:40



EXERCÍCIO 09 - Criando rotas

Essencialmente nossa aplicação não necessitaria de rotas, já que na semana 7 implementou-se apenas a página Home.

No entanto, para praticarmos o que foi estudado na semana criaremos 3 páginas extras (que serão inseridas na pasta pages): ****NotFound****, ****Login**** e ****Details****

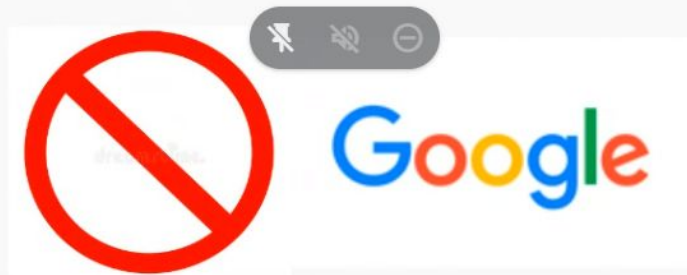
- Fica a critério de vocês criar conteúdo para cada página.

Mas há critérios adicionais:

- **Login** tem como path `"/login"`, mas se o usuário digitar `"/entrar"` será redirecionado para Login
- **Details** será uma rota protegida, que só pode ser acessada por quem está logado. A forma de acessá-la é clicando no card. Dentro dela serão exibidas todas as informações do card um "preview" do vídeo.

- [React Router](#)
- [ReactJs: Manipulando rotas com react-router](#)
- [A guide to using React Router v6 in React apps - LogRocket Blog](#)

Hey! Conta pra gente, qual é o seu fato não
“googleavel”...



AVALIAÇÃO DOCENTE

O que você está achando das minhas aulas neste conteúdo?

[Clique aqui](#) ou escaneie o QRCode ao lado para avaliar minha aula.

Sinta-se à vontade para fornecer uma avaliação sempre que achar necessário.





DEVinHouse

Parcerias para desenvolver a sua carreira

OBRIGADO!



<LAB365>