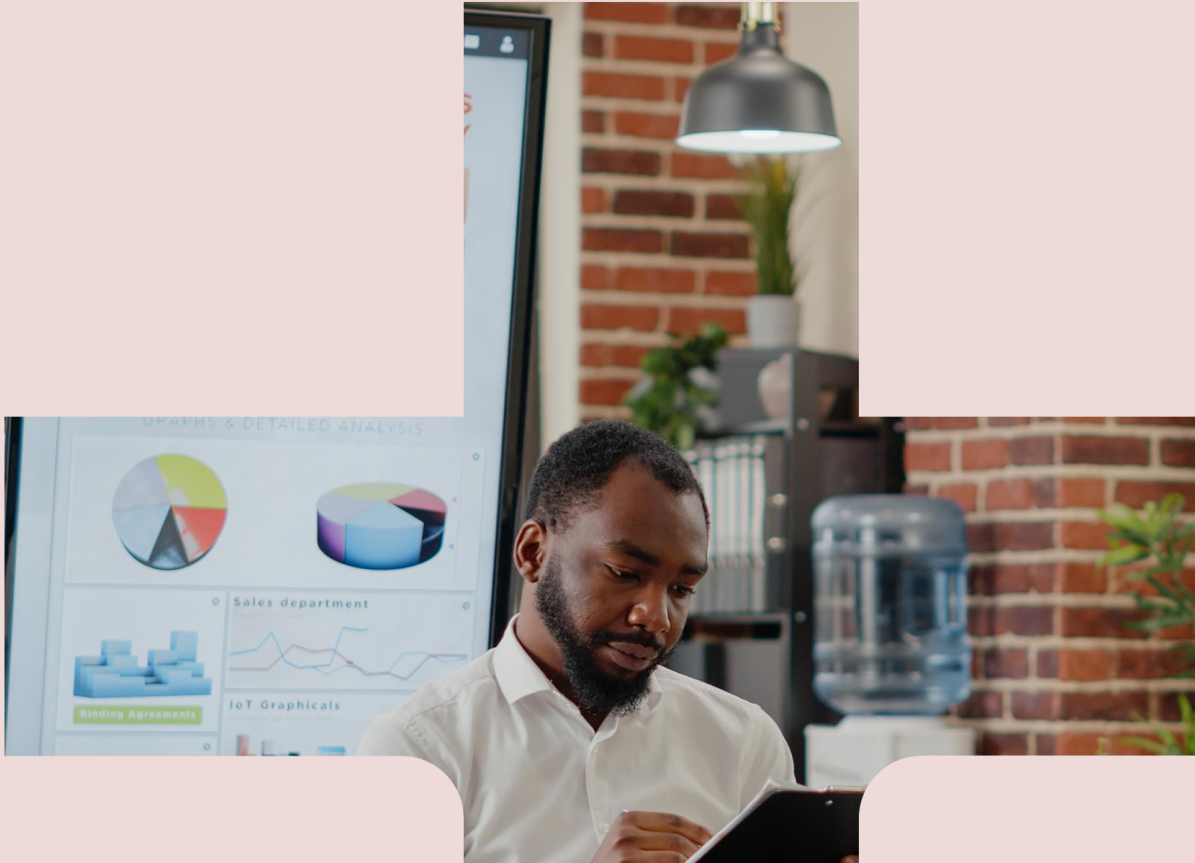
pulsati

Parâmetros

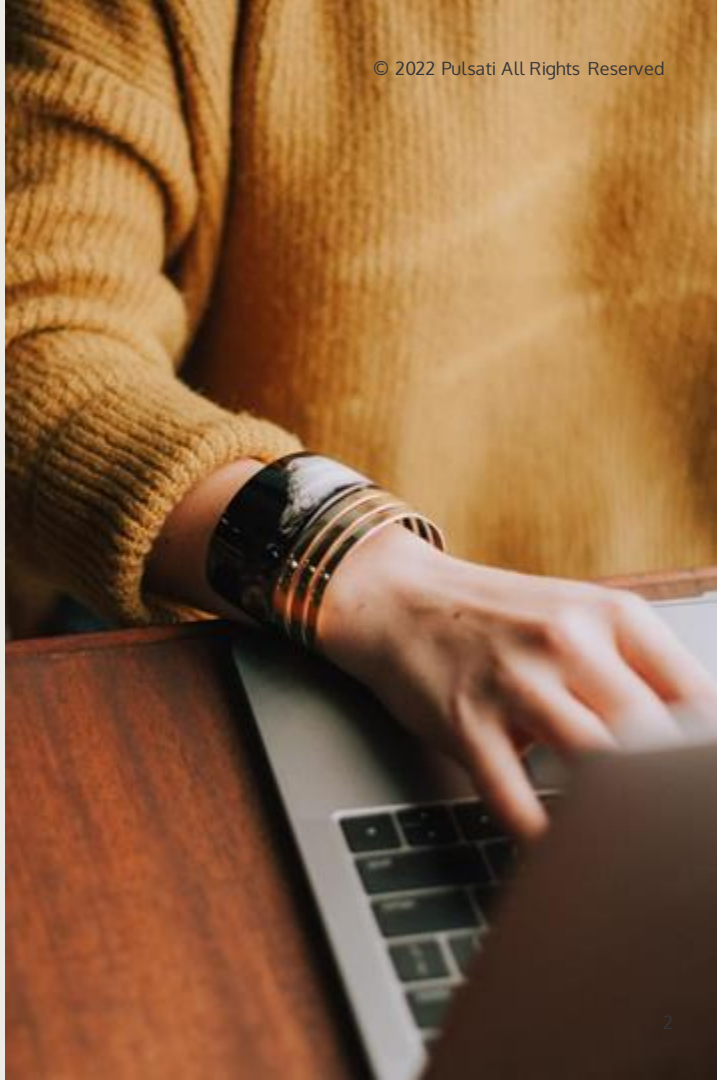


Obter parâmetros

Com o "react-router-dom" podemos trabalhar com as rotas para obter informações.

Utilizando o "useLocation" conseguimos obter dados em relação à url que estamos trabalhando no Browser.

Com o "useParams", podemos ter acesso aos parâmetros que definimos na url



Como definir parâmetros

Quando criamos um link, podemos criar uma URL dinâmica, de forma que possamos enviar parâmetros na URL para que possa ser trabalhado na definição de um contexto para a aplicação, através da URL

Na linha 8, estamos definindo que o link irá direcionar para a página de Usuários, porém é enviado um dado a mais. Nesse caso, estou definindo que na URL também irá conter o código do Usuário

```
1 usuarios?.map(usuario => {
2    return (
3        <tr>
4            <td>{usuario.codigo}</td>
5            <td>{usuario.nome}</td>
6            <td>{usuario.sobrenome}</td>
7            <td>
8                <a href={`/usuarios/${usuario.codigo}`}>
9                    Ver
10                </a>
11           </td>
12       </tr>
13    )
14 })
```

Como utilizar os parâmetros

Na linha 5, adicionamos uma nova rota na nossa aplicação, onde eu vou suportar a chamada a para a rota criada no passo anterior

Quando adiciona ":", estou transformando aquele em dado em uma variável de parâmetro para tratar dentro da tela que for implementar a visualização daquela rota

```
1 <Router>
2     <Routes>
3         <Route path='/home' element={<Home />} />
4         <Route path='/usuarios' element={<Usuarios />} />
5         <Route path='/usuarios/:id' element={<UsuarioVer />} />
6     </Routes>
7 </Router>
```

Como utilizar os parâmetros

O "react-router-dom" provê dois métodos para ler as informações do Browser, em relação a rota, importados na linha 1.

Com o método "useLocation", utilizado na linha 5, podemos obter a informação da rota completa do Browser.

Com o método "useParams", utilizado na linha 6, conseguimos pegar todos os parâmetros da URL

```
1 import { useLocation, useParams }
2     from "react-router-dom"
3
4 export function UsuarioVer() {
5     const location = useLocation();
6     const params = useParams();
7     return (
8         <span>ver</span>
9     )
10 }
```

Prática

Alterar a aplicação de Usuários para que ela suporte uma rota para ver dados de um usuário.

Utilizando formulários

No HTML, temos o componente "form" para trabalhar com utilização de formulários, que auxilia tanto na apresentação de dados, quanto para enviar dados para uma api.

No quadro à direita, podemos visualizar um exemplo, da criação de um formulário, na linha 5.

Na linha 6, eu defino que terá um campo para inserir dados ("input") e envio duas propriedades: type e value

```
1 import { useParams } from "react-router-dom"
2 export function UsuarioVer() {
3     const params = useParams();
4     return (
5         <form>
6             <input type="text" value={params.id} />
7         </form>
8     )
9 }
```

Quais campos tenho em um "form"

<input type="text">

Define que um campo deve ser do tipo texto livre

<input type="radio">

Define que o campo vai ser do tipo seleção única

<input type="checkbox">

Define um campo que aceita estar marcado ao desmacador, como um item de um checklist

```
1 import { useParams } from "react-router-dom"
2 export function UsuarioVer() {
3     const params = useParams();
4     return (
5         <form>
6             <input type="text" value={params.id} />
7         </form>
8     )
9 }
```

Prática

Implementar a tela de visualização do usuário utilizando um "form"

O/
Exercícios

+

Cadastro de passageiros

Utilizando a aplicação de passageiros, fazer as alterações:

- Adicionar um botão 'Ver' para cada passageiro
- Ao clicar no botão 'Ver' deve ser redirecionado para a página 'passageiros/:id'
- Criar uma tela que carregue os dados do passageiro em um formulário

Obrigado!

pulsati

pulsati.com.br