LABORATÓRIO 3

CRIANDO COMPONENTES REUTILIZÁVEIS

Vamos criar um componente reutilizável, bem simples, que mostre como um mesmo template e uma mesma visualização para o usuário pode ser utilizada para diferentes perfis, usuários e dados.

Iniciamos, da mesma maneira, criando uma nova aplicação:

```
$ ng new lab3
$ cd lab3
$ ng serve
```

1. Vamos adicionar alguns dados diretamente no AppComponent:

```
$ data = [{
            name : 'Sam Johnson',
            dept : 'Electrical',
            cpf: ‘25668515060’
          },{
            name : 'Roy Thomas',
            dept : 'Mechanical',
            cpf: ‘80125320000’
          },{
            name : 'Jim Lasker',
            dept : 'Medical',
            cpf: ‘60782914080’
          }]
```

2. Vamos renderizar os dados que definimos no HTML, adicionando o seguinte código:

```
$ <div class="container margin-30">
            <div class="card">
              <div class="card-body">
                <ol class="list-group list-group-numbered">
                  <li *ngFor="let item of data" class="list-group-item d-flex justify-content-between
          align-items-start">
```

```
                    <div class="ms-2 me-auto">
                      <div class="fw-bold font-fam">{{item.dept}}</div>
                      <div class="fw-bold font-fam">{{item.cpf}}</div>
                      {{item.name}}
                    </div>
                    <span class="badge bg-primary rounded-pill">14</span>
                  </li>
                </ol>
              </div>
            </div>
          </div>
```

3. Vamos também adicionar o seguinte código de estilo:

```
$       .margin-30{
            margin: 30px;
        }

        .font-fam{
            font-family: serif;
        }
        .list-group-item{
          display: flex;
          justify-content: space-between;
          align-items: center;
          padding-bottom: 5px
        }
```

A partir de agora, vamos criar o componente reutilizável

Como exemplo, vejam que no aplicativo criado, estamos mostrando o nome e um CPF desses nomes. Podem existir cenários em que queremos esconder esse dado sensível, dependendo do perfil de acesso.

Com esse estudo de caso, vamos transformar o código de renderização da lista reutilizável, criando um componente separado. O CPF vai ser configurado para ser mostrado ou escondido, dependendo da necessidade.

1. Criar um novo componente dentro da pasta app

$ ng g component dataList

2. Vamos mover os códigos que estavam no html e scss do app para esse novo componente.
3. Por enquanto, vamos mover também os dados que estavam no typescript do app para esse novo componente. Eventualmente vamos passar essa informação pelo decorator @Input.
4. Lembre-se de colocar o selector do novo componente no html do app, assim será possível visualizar a lista.

Por enquanto, o componente não é reutilizável. Vamos resolver essa questão:

1. Vamos utilizar o @Input para receber duas informações. Tanto os dados como uma variável showCpf:

$ @Input() data;

@Input() showCount = false;

2. No html do DataList, vamos modificar o código para receber essas informações dos Inputs e renderizar corretamente.

$ *ngIf="showCpf"

3. Agora, vamos modificar o template do app para passar essas informações de Input em suas chamadas:

$ <app-data-list [data]="data1" [showCpf]="true"></app-data-list>

<app-data-list [data]="data2"></app-data-list>

Finalmente, vamos definir data1 e data2 no app component para demonstrar dados diferentes em duas chamadas diferentes.

É um teste simples, mas mostra exatamente o que nós queríamos. O mesmo template pode mostrar diferentes dados de diferentes formas, podendo ser reutilizável para vários perfis e situações.