



Missão Prática | Nível 4 | Mundo 2

Nome: Ednaldo Ribeiro de Oliveira Matrícula: 202310274175

Campus: Polo cidade universitária Juazeiro do norte CE

Nível 4: Conhecendo Outro Framework – Número da Turma – 2024.2

Repositorio GitHub: https://github.com/ednaldo1901/trabalho_estacio_nivel_4

OBS: resultado do Proje

Catálogo Novo Livro

Catálogo de Livros

Título	Resumo	Editora	Autores
Excluir Use a Cabeça: Java	Uma experiência completa de aprendizado em programação orientada a objetos (OO) e Java.	Alta Books	<ul style="list-style-type: none">Bert BatesKathy Sierra
Excluir Java, como Programar	Milhões de alunos e profissionais aprenderam programação e desenvolvimento de software com os livros Deitel.	Pearson	<ul style="list-style-type: none">Paul DeitelHarvey Deitel
Excluir Python, para Programar	Profissionais dominaram a programação e o desenvolvimento de software através dos livros sobre Python.	Pearson	<ul style="list-style-type: none">Edy RibeiroCicera Cunha

Dados do Livro

Título

Resumo

Editora

Autores (1 por linha)

Salvar Dados

Objetivo da Prática

Implementar serviços injetáveis para o Angular, na sintaxe Type Script

Implementar componentes, utilizando Type Script e modelos HTML

Utilizar a biblioteca para gerenciamento de formulários no Angular

Implementar a navegação interna do front-end com base em Angular

1º Procedimento | Projeto Angular para o Catálogo de Livros

Livro.lista,componet.HTML:

```
<main class="container">

  <h1>Catálogo de Livros</h1>

  <table class="table table-striped">

    <thead>

      <tr>

        <th>Título</th>

        <th>Resumo</th>

        <th>Editora</th>

        <th>Autores</th>

      </tr>

    </thead>
```

```

<tbody>

  <tr *ngFor="let livro of livros">

    <td>

      <button class="btn btn-danger btn-sm"
(click)="excluir(livro.codigo)">Excluir</button>

      {{ livro.titulo }}

    </td>

    <td>{{ livro.resumo }}</td>

    <td>{{ obterNome(livro.codEditora) }}</td>

    <td>

      <ul>

        <li *ngFor="let autor of livro.autores">{{ autor }}</li>

      </ul>

    </td>

  </tr>

</tbody>

</table>

</main>

```

livro.lista.component.ts:

```

import { Component, OnInit } from '@angular/core';
import { ControleEditoraService } from '../controle-editora.service';
import { ControleLivrosService } from '../controle-livros.service';
import { Editora } from '../editora';
import { Livro } from '../livro';
import { CommonModule } from '@angular/common';
import { RouterLink, RouterOutlet } from '@angular/router';

```

```
@Component({
  selector: 'app-livro-lista',
  standalone: true,
  imports: [CommonModule, RouterOutlet, RouterLink],
  templateUrl: './livro-lista.component.html',
  styleUrls: ['./livro-lista.component.css']
})
```

```
export class LivroListaComponent implements OnInit {
```

```
  public editoras: Array<Editora> = [];
```

```
  public livros: Array<Livro> = [];
```

```
  constructor(
```

```
    private servEditora: ControleEditoraService,
```

```
    private servLivros: ControleLivrosService
```

```
  ) {}
```

```
  ngOnInit(): void {
```

```
    this.editoras = this.servEditora.getEditoras();
```

```
    this.livros = this.servLivros.obterLivros();
```

```
  }
```

```
  excluir = (codigo: number) => {
```

```
    this.servLivros.excluir(codigo);
```

```
    this.livros = this.servLivros.obterLivros();
```

```
  }
```

```
  obterNome = (codEditora: number) => {
```

```
    return this.servEditora.getNomeEditora(codEditora);  
  }  
}
```

Classe livro.ts

```
export class Livro {  
  codigo:number;  
  codEditora:number;  
  titulo:string;  
  resumo:string;  
  autores:Array<string>
```

```
  constructor(codigo:number,codEditora:number,titulo:string,resumo:string,autor:Array  
<string>){  
    this.codigo =codigo;  
    this.codEditora = codEditora;  
    this.titulo = titulo;  
    this.resumo = resumo;  
    this.autores = autor  
  }  
}
```

controle-livros.servico.ts:

```
import { Injectable } from '@angular/core';  
import { Livro } from './livro';
```

```

@Injectable({
    providedIn: 'root'
})

export class ControleLivrosService {

    private livros: Array<Livro> = [

        { codigo: 1, codEditora: 1, titulo: 'Use a Cabeça: Java',
          resumo: 'Uma experiência completa de aprendizado em programação orientada a
          objetos (OO) e Java.',
          autores: ['Bert Bates', 'Kathy Sierra'] },

        { codigo: 2, codEditora: 2, titulo: 'Java, como Programar',
          resumo: 'Milhões de alunos e profissionais aprenderam programação e
          desenvolvimento de software com os livros Deitel.',
          autores: ['Paul Deitel', 'Harvey Deitel'] },

        { codigo: 2, codEditora: 2, titulo: 'Python, para Programar',
          resumo: 'Profissionais dominaram a programação e o desenvolvimento de software
          através dos livros sobre Python. ',
          autores: ['Edy Ribeiro', 'Cicera Cunha'] },

        // Adicione outros livros aqui

    ];

    obterLivros(): Array<Livro> {

        return this.livros;

    }

    incluir(livro: Livro): void {

        const novoCodigo = Math.max(...this.livros.map(l => l.codigo)) + 1;

        livro.codigo = novoCodigo;

        this.livros.push(livro);
    }
}

```

```

}

excluir(codigo: number): void {
  const indice = this.livros.findIndex(l => l.codigo === codigo);
  if (indice !== -1) {
    this.livros.splice(indice, 1);
  }
}
}
}

```

Análise e Conclusão:

códigos atendendo a todos os requisitos da missão e em perfeito funcionamento.

2º Procedimento | Página de Cadastro e Sistema de Navegação

Componente Livro-Dados:

Livro.dados.comonet.HTML.

```

<main class="container mt-4">
  <h1>Dados do Livro</h1>
  <form (submit)="incluir()" ngNativeValidate>
    <div class="mb-3">
      <label for="titulo" class="form-label">Título</label>
      <input type="text" id="titulo" class="form-control" [(ngModel)]="livro.titulo"
name="titulo" required>
    </div>
    <div class="mb-3">

```

```

    <label for="resumo" class="form-label">Resumo</label>

    <textarea id="resumo" class="form-control" [(ngModel)]="livro.resumo"
name="resumo"></textarea>

</div>

<div class="mb-3">

    <label for="editora" class="form-label mr-3">Editora</label>

    <select id="editora" class="form-select" [(ngModel)]="livro.codEditora"
name="editora" required>

        <option *ngFor="let editora of editoras" [value]="editora.codEditora">{{
editora.nome }}</option>

    </select>

</div>

<div class="mb-3">

    <label for="autores" class="form-label">Autores (1 por linha)</label>

    <textarea id="autores" class="form-control" [(ngModel)]="autoresForm"
name="autores"></textarea>

</div>

<button type="submit" class="btn btn-primary">Salvar Dados</button>

</form>

</main>

```

Livro.dados.component.ts

```

import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { Livro } from '../livro';
import { ControleEditoraService } from '../controle-editora.service';
import { ControleLivrosService } from '../controle-livros.service';
import { Editora } from '../editora';

```



```
import { CommonModule } from '@angular/common';
import { FormsModule } from '@angular/forms';
```

```
@Component({
  selector: 'app-livro-dados',
  standalone: true,
  imports: [CommonModule, FormsModule],
  templateUrl: './livro-dados.component.html',
  styleUrls: ['./livro-dados.component.css']
})
```

```
export class LivroDadosComponent implements OnInit {
  public livro: Livro = new Livro(0,0,"",[]);
  public autoresForm: string = "";
  public editoras: Array<Editora> = [];
```

```
  constructor(
    private servEditora: ControleEditoraService,
    private servLivros: ControleLivrosService,
    public router: Router
  ) {}
```

```
  ngOnInit(): void {
    this.editoras = this.servEditora.getEditoras();
  }
```

```
  incluir = () => {
    this.livro.autores = this.autoresForm.split("\n");
    this.servLivros.incluir(this.livro);
```

```
    this.router.navigateByUrl('/lista');  
  };  
}
```

controle-editora.service.ts:

```
import { Injectable } from '@angular/core';  
import { Editora } from './editora';
```

```
@Injectable({  
  providedIn: 'root'  
})
```

```
export class ControleEditoraService {  
  private editoras: Array<Editora> = [  
    { codEditora: 1, nome: 'Alta Books' },  
    { codEditora: 2, nome: 'Pearson' },  
    { codEditora: 3, nome: 'O Reilly' }  
  ];
```

```
  getEditoras(): Array<Editora> {  
    return this.editoras;  
  }
```

```
  getNomeEditora(codEditora: number): string | undefined {  
    return this.editoras.find(e => e.codEditora === codEditora)?.nome;  
  }  
}
```

análise e conclusão:

todos os códigos atendendo aos requisitos da missão e em perfeito funcionamento.

classe editora

```
export class Editora {  
  codEditora: number;  
  nome: string;  
  
  constructor(codEditora: number, nome: string) {  
    this.codEditora = codEditora;  
    this.nome = nome;  
  }  
}
```

app.routes.ts:

```
import { Routes } from '@angular/router';  
  
import { LivroListaComponent } from './livro-lista/livro-lista.component';  
import { LivroDadosComponent } from './livro-dados/livro-dados.component';  
  
export const routes: Routes = [  
  { path: 'lista', component: LivroListaComponent },  
  { path: 'dados', component: LivroDadosComponent },  
  { path: '**', redirectTo: '/lista', pathMatch: 'full' }  
];
```

Análise da Missão:

Missão onde é necessário conhecimentos e práticas com linguagem de programação typescript, HTML, CSS e frameworks Angula e Bootstrap, confesso que tive que estudar

bastante pois não dominava algumas dessas tecnologias, mais ao final o resultado foi satisfatório

além de adquirir novos conhecimentos ao me desafiar em concluir esse projeto