

Trabalho de Conclusão da Matéria de Frameworks Web

Tópicos abordados no projeto

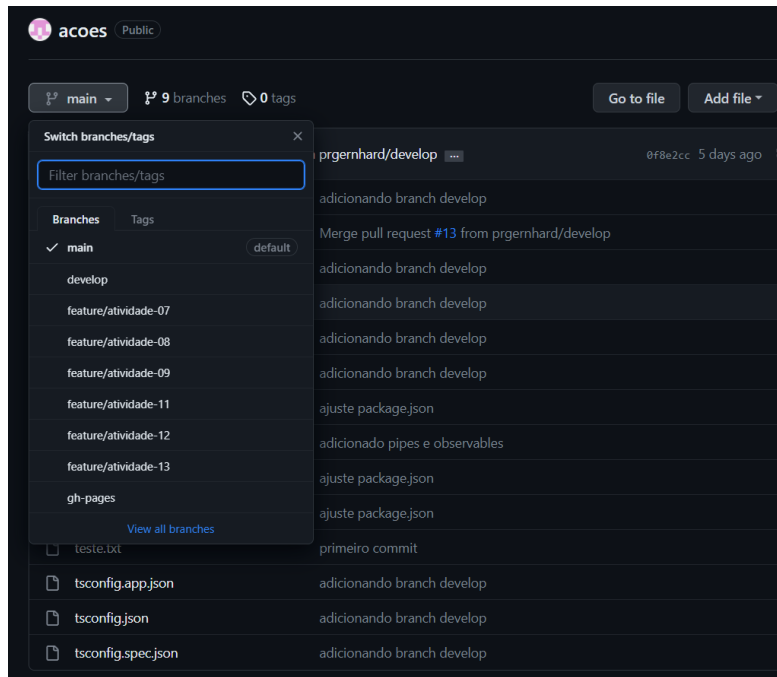
- CRIAR O REPOSITÓRIO NO GITHUB COM A ESTRUTURA DO GITFLOW, OU SEJA, BRANCHES MAIN E DEVELOP.
- USAR COMPONENTES DE ALGUM FRAMEWORK CSS (BOOTSTRAP, MATERIALIZE OU OUTRO).
- APRESENTAR AS TELAS COM LAYOUT RESPONSIVO USANDO OU NÃO ALGUM FRAMEWORK CSS.
- CONSTRUIR PÁGINAS WEB COM O CONCEITO DE COMPONENTES.
- CRIAR O LAYOUT DA APLICAÇÃO COM COMPONENTES, OU SEJA, O CABEÇALHO E RODAPÉ PRECISAM SER COMPONENTES.
- USAR PELO MENOS DOIS TIPOS DE DATA-BINDING (INTERPOLATION, PROPERTY BINDING, EVENT BINDING E TWO WAY DATA BINDING).
- PASSAR DADOS VIA HIERARQUIA DE COMPONENTES, OU SEJA, USANDO @INPUT OU @OUTPUT.
- MAPEAR COMPONENTES À ROTAS NO MÓDULO DE ROTAS.
- CRIAR NAVEGAÇÃO ENTRE PÁGINAS POR MEIO DE ROTAS.
- PASSAR DADOS ENTRE COMPONENTES QUE REPRESENTAM DIFERENTES TELAS VIA PARÂMETROS DE ROTAS.
- VALIDAR CAMPOS DO FORMULÁRIO COM REGEX E APRESENTAR OS ERROS.
- DESABILITAR O BOTÃO DE SUBMIT ENQUANTO O FORMULÁRIO ESTÁ INVÁLIDO.
- FAZER REQUISIÇÕES A API COM TRATAMENTO DA RESPOSTA COM PROMISES OU OBSERVABLES.
- CADASTRAR UMA ENTIDADE NO JSON SERVER.
- APRESENTAR UMA LISTA DE DADOS COM A DIRETIVA ESTRUTURAL NGFOR.
- USAR A DIRETIVA NGIF
- FORMATAR A APRESENTAÇÃO DE DADOS COM PIPES.
- BUILD E DEPLOY DA APLICAÇÃO

- Criar o repositório no GitHub com a estrutura do Gitflow, ou seja, branches main e develop.

- Link do Repositório

<https://github.com/prgernhard/acoes>

- Definição de Branches



- Usar componentes de algum framework CSS (Bootstrap, Materialize ou outro).

- Uso do Bootstrap na aplicação

```

EXPLORER
PROJECTO-ACOES-ANGULAR-PAULO
├── angular
├── vscode
├── dist
├── node_modules
├── src
├── editorconfig
├── gitignore
├── angular.json
├── db.json
├── package-lock.json
├── package.json
├── README_old.md
├── README.md
├── tsconfig.app.json
├── tsconfig.json
└── tsconfig.spec.json

package.json
16 "dependencies": {
17   "@angular/animations": "^16.0.0",
18   "@angular/common": "^16.0.0",
19   "@angular/compiler": "^16.0.0",
20   "@angular/core": "^16.0.0",
21   "@angular/forms": "^16.0.0",
22   "@angular/platform-browser": "^16.0.0",
23   "@angular/platform-browser-dynamic": "^16.0.0",
24   "@angular/router": "^16.0.0",
25   "@ng-bootstrap/ng-bootstrap": "^15.0.1",
26   "angular-cli-ghpages": "^1.0.6",
27   "bootstrap": "^5.2.3",
28   "jquery": "^3.7.0",
29   "json-server": "^0.17.3",
30   "material-icons": "^1.13.7",
31   "materialize-css": "^1.0.0",
32   "ng-bootstrap": "^16.3",
33   "ngx-bootstrap": "^10.2.0",
34   "or": "^0.2.0",
35   "rxjs": "^7.8.0",
36   "tslib": "^2.3.0",
37   "zone.js": "^0.13.0"
38 },

```

```

"styles": [
  "src/styles.css",
  "node_modules/bootstrap/dist/css/bootstrap.min.css"
],
"scripts": [
  "./node_modules/jquery/dist/jquery.min.js",
  "./node_modules/bootstrap/dist/js/bootstrap.bundle.js"
]

```

- Apresentar as telas com layout responsivo usando ou não algum framework CSS.
 - App no celular esconde a barra de menus e o formulário se adapta

- Já no navegador do computador com a tela maior o menu sidebar aparece e os campos se adequam na tela.

- Construir páginas web com o conceito de componentes.

```

> node_modules
  > src
    > app
      > acoes
        > acoes-form
          # acoes-form.component.css
          <> acoes-form.component.html
          TS acoes-form.component.ts
        > acoes-list
          # acoes-list.component.css
          <> acoes-list.component.html
          TS acoes-list.component.ts
        TS acoes-routing.module.ts
        TS acoes.module.ts
        TS acoes.ts
      > components
      > empresas
    > OUTLINE

```

- Criar o layout da aplicação com componentes, ou seja, o cabeçalho e rodapé precisam ser componentes.

```

> empresas
> home
  layout
    # layout.component.css
    layout.component.html
    TS layout.component.ts
  template
    footer
    navbar
    sidebar
    TS template-routing.module.ts
    TS template.module.ts
  
```

```

1 <app-navbar></app-navbar>
2 <div id="layoutSidenav">
3   <app-sidebar></app-sidebar>
4   <div id="layoutSidenav_content">
5     <main>
6       <div class="container">
7         <router-outlet></router-outlet>
8       </div>
9     </main>
10  </div>
11 </div>
12
13
  
```

- Usar pelo menos dois tipos de data-binding (Interpolation, Property Binding, Event Binding e Two Way Data Binding).

- Exemplo de data-binding (Interpolation)

```

<div class="row">
  <div class="col-md-12">
    <div class="alert alert-success" role="alert" *ngIf=" success == true">
      Ação Salva/Atualizado com sucesso!!
    </div>
    <div class="alert alert-danger" role="alert" *ngFor=" let erro of errors">
      {{ erro }}
    </div>
  </div>
</div>
  
```

- Exemplo de data-binding (Two Way)

```

<div class="form-check">
  <input class="form-check-input" type="radio" [value]="false" name="dividento" id="nao"
    FormControlName="dividento" [(ngModel)]= "acao.dividento" checked>
  <label class="form-check-label" for="nao">
    Não
  </label>
</div>
  
```

- Passar dados via hierarquia de componentes, ou seja, usando @Input ou @Output.

```

8
9 export class InformacaoEmpresaComponent implements OnInit {
10   @Input() public emp: any;
11   @Output() public onAlert = new EventEmitter();
12
13   ngOnInit(): void {
14     throw new Error('Method not implemented.');

```

Go to component
<h3>Lista Empresas</h3>
<p>
 Nome: {{ emp.nome }}

 CNPJ: {{ emp.cnpj }}

 Data: {{ emp.data }}
</p>
<button (click)="alertEvent($event)">Exibir Saudação</button>

```


```

- Mapear componentes à rotas no módulo de rotas.

```
ts acoes-routing.module.ts X
src > app > acoes > TS acoes-routing.module.ts > (0) routes
1 import { NgModule } from '@angular/core';
2 import { RouterModule, Routes } from '@angular/router';
3 import { AcoesFormComponent } from './acoes-form/acoes-form.component';
4 import { AcoesListComponent } from './acoes-list/acoes-list.component';
5 import { LayoutComponent } from '../layout/layout.component';
6
7 const routes: Routes = [
8   {
9     path: 'acoes', component: LayoutComponent, children: [
10      { path: 'form', component: AcoesFormComponent },
11      { path: 'form/:id', component: AcoesFormComponent },
12      { path: 'lista', component: AcoesListComponent },
13      { path: '', redirectTo: '/acoes/lista', pathMatch: 'full' }
14    ]
15  }
16 ];
17
```

```
rc > app > empresas > TS empresas-routing.module.ts > (0) routes
1 import { NgModule } from '@angular/core';
2 import { RouterModule, Routes } from '@angular/router';
3 import { EmpresasFormComponent } from './empresas-form/empresas-form.component';
4 import { EmpresasListComponent } from './empresas-list/empresas-list.component';
5 import { LayoutComponent } from '../layout/layout.component';
6
7 const routes: Routes = [
8   {
9     path: 'empresas', component: LayoutComponent, children: [
10      { path: 'form', component: EmpresasFormComponent },
11      { path: 'form/:id', component: EmpresasFormComponent },
12      { path: 'lista', component: EmpresasListComponent },
13      { path: '', redirectTo: '/empresas/lista', pathMatch: 'full' }
14    ]
15  }
16 ];
17
```

- Criar navegação entre páginas por meio de rotas.

```
4 import { AppRoutingModule } from './app-routing.module';
5 import { AppComponent } from './app.component';
6 import { HomeComponent } from './home/home.component';
7 import { TemplateModule } from './template/template.module';
8 import { EmpresasModule } from './empresas/empresas.module';
9 import { AcoesModule } from './acoes/acoes.module';
10 import { HttpClientModule } from '@angular/common/http';
11 import { PipesExemplosComponent } from './components/pipes-exemplos/pipes-exemplos.component';
12 import { ObservableExemploComponent } from './components/observable-exemplo/observable-exemplo.component';
13 import { InformacaoEmpresaComponent } from './components/informacao-empresa/informacao-empresa.component';
14
15 @NgModule({
16   declarations: [
17     AppComponent,
18     HomeComponent,
19     TemplateModule,
20     PipesExemplosComponent,
21     ObservableExemploComponent,
22     InformacaoEmpresaComponent
23   ],
24   imports: [
25     BrowserModule,
26     AppRoutingModule,
27     TemplateModule,
28     EmpresasModule,
29     AcoesModule,
30     HttpClientModule
31   ],
32 })
33 export class AppModule {}
```

```
etc > index.html > (0) html > body > app-root
1 <doctype html>
2 <html lang="en">
3
4 <head>
5   <meta charset="utf-8">
6   <title>Acoes</title>
7   <base href="/">
8   <meta name="viewport" content="width=device-width, initial-scale=1">
9   <link rel="icon" type="image/x-icon" href="/favicon.ico">
10  <link href="https://unpkg.com/ngx-bootstrap/datepicker/bs-datepicker.css" rel="stylesheet">
11 </head>
12 <body>
13   <app-root></app-root>
14 </body>
15 </html>
```

- Passar dados entre componentes que representam diferentes telas via parâmetros de rotas.

```
8 <a href="#" class="nav-link text-white" routerLink="/acoes"> <svg xmlns="http://www.w3.org/2000/svg" width="24"
9   <path fill="currentColor" d="M17 3H7C4.79 3 3 4.79 3 7v10c0 2.21 1.79 4 4 4h12c1.11 0 2-.89 2-2V9a2 2 0 0
10 </svg><span class="ms-2">Ação</span> </a>
11 </li>
12 <li><br /><a href="#" class="nav-link text-white" routerLink="/empresas"> <svg width="24" height="24"
13   viewBox="0 0 43 42" fill="currentColor" xmlns="http://www.w3.org/2000/svg">
14   <path
15     d="M34.0419 5.25V36.75H23.5419V30.625H20.0419V36.75H9.5419V30.625H34.0419ZM27.0419 12.25H30.5419V8.75
16     fill="white" />
17 </svg><span class="ms-2">Empresa</span> </a> </li>
18 </ul>
19 <hr>
```

- Validar campos do formulário com REGEX e apresentar os erros.

```
56 createForm(empresa: Empresa) {
57   this.formEmpresa = this.formBuilder.group({
58     id: [empresa.id],
59     nome: [empresa.nome, [Validators.required, Validators.minLength(3)]],
60     cnpj: [empresa.cnpj, [Validators.required, Validators.maxLength(18), Validators.pattern(/^\d{2}\.\d{3}\.\d{3}\d{4}\-\d{2}$)/]],
61     tipoEmpresa: [empresa.tipoEmpresa],
62     data: [empresa.data]
63   });
64 }
65
66
```

- Desabilitar o botão de submit enquanto o formulário está inválido.

[illegible]

- Fazer requisições a API com tratamento da resposta com Promises ou Observables.

```

1 import { Injectable } from '@angular/core';
2 import { Empresa } from '../empresas/empresa';
3 import { HttpClient } from '@angular/common/http';
4 import { Observable } from 'rxjs';
5 import { environment } from '../environments/environment';
6
7 @Injectable({
8   providedIn: 'root'
9 })
10 export class EmpresasService {
11
12   apiUrl: string = environment.apiUrl;
13
14   constructor(private http: HttpClient) {}
15
16   salvar(empresa: Empresa): Observable<Empresa> {
17     return this.http.post<Empresa>(`${this.apiUrl}/empresas`, empresa);
18   }
19
20   atualizar(empresa: Empresa): Observable<any> {
21     return this.http.put<Empresa>(`${this.apiUrl}/empresas/${empresa.id}`, empresa);
22   }
23
24   getEmpresas(): Observable<Empresa[]> {
25
26     return this.http.get<Empresa[]>(`${this.apiUrl}/empresas`);
27   }
28
29   getEmpresaById(id: number): Observable<Empresa> {
30
31     return this.http.get<any>(`${this.apiUrl}/empresas/${id}`);
32   }
33
34 }

```

- Cadastrar uma entidade no JSON Server.

```
1 db.json > [ { "acoes": > } ] > empresa
2
3   {
4     "acoes": [
5       {
6         "id": 590,
7         "nome": "Itau SA",
8         "valor": "25.50",
9         "data": "Sat Jun 17 2023 16:39:02 GMT-0300 (Horário Padrão de Brasília)",
10        "dividendo": true,
11        "empresa": "Itau SA"
12      }
13    ],
14    "empresas": [
15      {
16        "id": 597,
17        "nome": "Banco do Brasil",
18        "cnpj": "00.000.000/0001-91",
19        "tipoEmpresa": "publica",
20        "data": "2023-06-17T19:41:25.929Z"
21      }
22    ]
23  }
```

```

}
    "jquery": "^3.7.0",
    "json-server": "^0.17.3",
    "material-icons": "^1.13.7",



```

- Apresentar uma lista de dados com a diretiva estrutural ngFor.

```
<tbody>
<tr *ngFor="let empresa of empresas">
<td>{{ empresa.nome | uppercase }}</td>
<td>{{ empresa.cnpj }}</td>
<td>{{ empresa.tipoEmpresa | uppercase }}</td>
<td>{{ empresa.data | date: 'dd/MM/yyyy' }}</td>
<td>
<svg routerLink="/empresas/form/{{ empresa.id }}" width="20" height="20"
viewBox="0 0 31 31" fill="none" xmlns="http://www.w3.org/2000/svg">
<path
d="M27.4195 10.3869L20.4332 3.40283L22.7346 1.10146C23.3647 0.471319 24.139 0.15625 25.1
fill="#A55307" />
</svg>
<svg data-bs-toggle="modal" data-bs-target="#modalDelecao"
(click)="preparaDelecao(empresa)" width="20" height="20" viewBox="0 0 32 35"
fill="none" xmlns="http://www.w3.org/2000/svg">
<path
d="M27.8934 11.7228L28.2423 11.72 28.5588 11.8722 28.809 12.1295C29.0422 12.4042 29.1596
fill="#990050" />
</svg>
</td>
</tr>
</tbody>
```

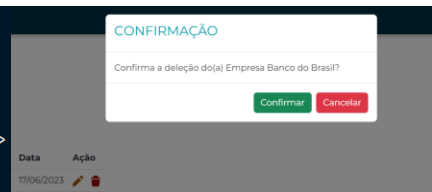
Empresa

Adicionar

Nome	CNPJ	Tipo Empresa	Data	Ação
BANCO DO BRASIL	00.000.000/0001-91	PUBLICA	17/06/2023	 

- Usar a diretiva ngIf

```
<div class="modal-body" *ngIf="empresaSelecioneada">
  Confirma a deleção do(a) Empresa {{empresaSelecioneada.nome}}?
</div>
<div class="modal-footer">
<button class="btn btn-success" (click)="deletarEmpresa()"
data-bs-dismiss="modal">Confirmar</button>
<button class="btn btn-danger" data-bs-dismiss="modal">Cancelar</button>
</div>
</div>
```



- Formatar a apresentação de dados com Pipes.

```
<td>{{ empresa.nome | uppercase }}</td>
<td>{{ empresa.cnpj }}</td>
<td>{{ empresa.tipoEmpresa | uppercase }}</td>
<td>{{ empresa.data | date: 'dd/MM/yyyy' }}</td>
```

Nome	CNPJ	Tipo Empresa	Data	Ação
BANCO DO BRASIL	00.000.000/0001-91	PUBLICA	17/06/2023	 

- Build e deploy da aplicação

```
dist \ acoes
> assets
3rdpartylicenses.txt
404.html
favicon.ico
index.html
main.b3676745ac2a1df9.js
polyfills.69a29a8139f83952.js
runtime.1e1f0365c0d75a0d.js
scripts.6515f3ba09796e1c.js
# styles.b6f61a18913b00a2.css
```

- Link Aplicação GitHub Pages

<https://prgernhard.github.io/acoes/home>