**Desenvolvimento FrontEnd com Angular**

Na aula de hoje, vamos começar a desenvolver uma aplicação FrontEnd com Angular para consumir a API de usuários desenvolvida na última aula.



**Setup do ambiente:**

Precisamos configurar as ferramentas necessárias para desenvolvimento do projeto Angular (FrontEnd)

**NodeJS**

É a plataforma para desenvolvimento de aplicações baseadas em JavaScript. O framework do Angular é executado a partir de um servidor NodeJS (JavaScript)

Tela de celular com publicação numa rede social

Descrição gerada automaticamente

**Visual Studio Code**

IDE para desenvolvimento dos projetos.

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

**GITBASH**

Terminal de linhas de comandos para o GITHUB.

<https://gitforwindows.org/>

Interface gráfica do usuário, Site

Descrição gerada automaticamente

\*\* Vamos criar uma pasta no raiz do

c:\\ para desenvolvedor os projetos:

**@angular/cli**

Conjunto de ferramentas para desenvolvimento de projetos baseados em Angular. Toolkit para criação e desenvolvimento com Angular.

<https://angular.io/cli>

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

**Verificando se o NODEJS está instalado:**

Samsung@DESKTOP-P9F6D9F MINGW64 /c/dev\_angular\_sqs

**$ node -v**

v18.14.2

**Instalando o @angular/cli**

Samsung@DESKTOP-P9F6D9F MINGW64 /c/dev\_angular\_sqs

**$ npm install -g @angular/cli**

**Verificando a versão do Angular:**

Samsung@DESKTOP-P9F6D9F MINGW64 /c/dev\_angular\_sqs

**$ ng version**

Texto

Descrição gerada automaticamente

**Criando o projeto:**

Desenvolvendo o sistema:

Samsung@DESKTOP-P9F6D9F MINGW64 /c/dev\_angular\_sqs

**$ ng new projeto\_angular**

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

**Acessando a pasta do projeto:**

Samsung@DESKTOP-P9F6D9F MINGW64 /c/dev\_angular\_sqs

**$ cd projeto\_angular**

**Abrindo o projeto no VisualStudio Code:**

Samsung@DESKTOP-P9F6D9F MINGW64 /c/dev\_angular\_sqs/projeto\_angular (master)

**$ code .**

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

**Configurando o Visual Studio Code**

**para trabalhar no modo AutoSave:**

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

**Instalando extensões no Visual Studio Code para trabalharmos com o framework Angular:**

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Tela de celular com aplicativo aberto

Descrição gerada automaticamente

**Executando o projeto:**

Os projetos criados em Angular já possuem um servidor web “embarcado” através do qual podemos executar as aplicações.

Samsung@DESKTOP-P9F6D9F MINGW64 /c/dev\_angular\_sqs/projeto\_angular (master)

**$ ng s**

Texto

Descrição gerada automaticamente

<http://localhost:4200/>

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Esta página está desenhada no arquivo:

/app.component.html

Tela de celular com aplicativo aberto

Descrição gerada automaticamente

Neste projeto, a primeira página que será exibida no navegador é um formulário para login do usuário.

Para isso, precisamos instalar o bootstrap no projeto.

**Bootstrap**

Toolkit para desenvolvimento CSS responsivo.

<https://getbootstrap.com/>

<https://getbootstrap.com/>

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Samsung@DESKTOP-P9F6D9F MINGW64 /c/dev\_angular\_sqs/projeto\_angular (master)

**$ npm install --save bootstrap**

Texto

Descrição gerada automaticamente

**Após a instalação do bootstrap, execute o projeto novamente:**

Samsung@DESKTOP-P9F6D9F MINGW64 /c/dev\_angular\_sqs/projeto\_angular (master)

**$ ng s**

Texto

Descrição gerada automaticamente

Configurando o uso do **CSS** e **JS** default do bootstrap no projeto:

/angular.json

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

"styles": [

              "src/styles.css",

              "node\_modules/bootstrap/dist/css/bootstrap.min.css"

            ],

            "scripts": [

              "node\_modules/bootstrap/dist/js/bootstrap.bundle.min.js"

            ]

\*\* Após a configuração acima, reinicie o servidor do projeto.

Desenhando uma página inicial para login do usuário:

**/app.component.html**

<div class="container">

  <div class="row justify-content-center mt-5">

    <div class="col-md-4">

      <div class="card">

        <div class="card-body">

          <div class="text-center">

            <h5 class="card-title text-center">

Autenticação de Usuários

</h5>

            <p>Entre com as suas credenciais de acesso</p>

          </div>

          <hr/>

          <form>

            <div class="mb-3">

              <label for="inputEmail" class="form-label">

Entre com seu email:</label>

              <input

                type="email"

                class="form-control"

                id="inputEmail"

                aria-describedby="emailHelp"

                placeholder="Digite o seu e-mail"

                required

              />

            </div>

            <div class="mb-3">

              <label for="inputPassword" class="form-label">

Entre com sua senha:</label>

              <input

                type="password"

                class="form-control"

                id="inputPassword"

                placeholder="Digite a sua senha"

                required

              />

            <div class="text-end mb-2">

              <a href="#">

                Esqueci minha senha?

              </a>

            </div>

            </div>

            <div class="d-grid gap-2">

              <button class="btn btn-primary" type="submit">

                Acessar Sistema

              </button>

            </div>

            <div class="d-grid gap-2 mt-2">

              <a href="#" class="btn btn-light">

                Não possui conta? <strong>Cadastre-se aqui!</strong>

              </a>

            </div>

          </form>

        </div>

      </div>

      <div class="mt-2 text-center">

        <small>

          Projeto Angular<br/>Java WebDeveloper COTI Informática

        </small>

      </div>

    </div>

  </div>

</div>

**Resultado:**

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Desenhando a página completa:

/app.component.html

<div class="container">

  <div class="row justify-content-center mt-5">

    <div class="col-md-4">

      <div class="card">

        <div class="card-body">

          <div class="text-center">

            <h5 class="card-title text-center">

Autenticação de Usuários</h5>

            <p>Entre com as suas credenciais de acesso</p>

          </div>

          <hr/>

          <form>

            <div class="mb-3">

              <label for="inputEmail" class="form-label">

Entre com seu email:</label>

              <input

                type="email"

                class="form-control"

                id="inputEmail"

                aria-describedby="emailHelp"

                placeholder="Digite o seu e-mail"

                required

              />

            </div>

            <div class="mb-3">

              <label for="inputPassword" class="form-label">

Entre com sua senha:</label>

              <input

                type="password"

                class="form-control"

                id="inputPassword"

                placeholder="Digite a sua senha"

                required

              />

            <div class="text-end mb-2">

              <a href="#" data-bs-toggle="modal" data-bs-

target="#janelaRecuperarSenha">

                Esqueci minha senha?

              </a>

            </div>

            </div>

            <div class="d-grid gap-2">

              <button class="btn btn-primary" type="submit">

                Acessar Sistema

              </button>

            </div>

            <div class="d-grid gap-2 mt-2">

              <a href="#" class="btn btn-light" data-bs-toggle="modal"

data-bs-target="#janelaCadastro">

                Não possui conta? <strong>Cadastre-se aqui!</strong>

              </a>

            </div>

          </form>

        </div>

      </div>

      <div class="mt-2 text-center">

        <small>

          Projeto Angular<br/>Java WebDeveloper COTI Informática

        </small>

      </div>

    </div>

  </div>

</div>

<!-- Modal para recuperação de senha do usuário -->

<div class="modal fade" id="janelaRecuperarSenha" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">

  <div class="modal-dialog modal-dialog-centered">

    <div class="modal-content">

      <div class="modal-header">

        <h1 class="modal-title fs-5" id="exampleModalLabel">

Esqueci minha senha</h1>

        <button type="button" class="btn-close" data-bs-dismiss="modal"

aria-label="Close"></button>

      </div>

      <div class="modal-body">

        <form>

          <div class="mb-3">

            <label for="inputEmail" class="form-label">

Entre com seu email:</label>

            <input

              type="email"

              class="form-control"

              id="inputEmail"

              aria-describedby="emailHelp"

              placeholder="Digite o seu e-mail"

              required

            />

          </div>

          <div class="d-grid gap-2">

            <button class="btn btn-primary" type="submit">

              Confirmar

            </button>

          </div>

        </form>

      </div>

    </div>

  </div>

</div>

<!-- Modal para cadastro do usuário -->

<div class="modal fade" id="janelaCadastro" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">

  <div class="modal-dialog modal-dialog-centered">

    <div class="modal-content">

      <div class="modal-header">

        <h1 class="modal-title fs-5" id="exampleModalLabel">Criar Conta

de Usuário</h1>

        <button type="button" class="btn-close" data-bs-dismiss="modal"

aria-label="Close"></button>

      </div>

      <div class="modal-body">

        <form>

          <div class="mb-3">

            <label for="inputNome" class="form-label">Entre com seu

nome:</label>

            <input

              type="text"

              class="form-control"

              id="inputNome"

              placeholder="Digite o seu nome"

              required

            />

          </div>

          <div class="mb-3">

            <label for="inputEmail" class="form-label">Entre com seu

email:</label>

            <input

              type="email"

              class="form-control"

              id="inputEmail"

              aria-describedby="emailHelp"

              placeholder="Digite o seu e-mail"

              required

            />

          </div>

          <div class="mb-3">

            <label for="inputSenha" class="form-label">Entre com sua

senha:</label>

            <input

              type="password"

              class="form-control"

              id="inputSenha"

              placeholder="Digite a sua senha"

              required

            />

          </div>

          <div class="mb-3">

            <label for="inputSenhaConfirmacao"

class="form-label">Confirme a sua senha:</label>

            <input

              type="password"

              class="form-control"

              id="inputSenhaConfirmacao"

              placeholder="Confirme a sua senha"

              required

            />

          </div>

          <div class="d-grid gap-2">

            <button class="btn btn-primary" type="submit">

              Realizar Cadastro

            </button>

          </div>

        </form>

      </div>

    </div>

  </div>

</div>

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

**Abrindo o eclipse (projeto Java):**

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

**Executando a API:**

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Tabela, Excel

Descrição gerada automaticamente

<http://localhost:8083/swagger-ui/index.html>

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

A primeira integração que vamos construir será do formulário para **cadastro de conta de usuário**.

FRONTEND:

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

BACKEND:

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Para isso, precisamos fazer com que o angular seja capaz de capturar todos os dados preenchidos no formulário. E para isso vamos usar uma biblioteca nativa do angular chamada de: **@angular/forms**

Esta biblioteca permite a criação de formulários reativos. Ou seja, formulários capazes de capturar validar informações no FRONTEND em tempo real.

Ativando e configurando a biblioteca para desenvolvimento de formulários:

/app.module.ts

\*\* Os arquivos de extensão .ts são programados em linguagem **TypeScript**

import { NgModule } from '@angular/core';

import { BrowserModule } from '@angular/platform-browser';

**import { FormsModule, ReactiveFormsModule } from '@angular/forms';**

import { AppComponent } from './app.component';

@NgModule({

  declarations: [

    AppComponent

  ],

  imports: [

    BrowserModule,

**FormsModule, //formulários reativos**

**ReactiveFormsModule //formulários reativos**

  ],

  providers: [],

  bootstrap: [AppComponent]

})

export class AppModule { }

Para desenvolvermos a captura dos formulários, vamos utilizar um arquivo de extensão **TypeScript** voltado para programação do “Código por de trás” da página principal:

Tela de celular com aplicativo aberto

Descrição gerada automaticamente

/app.component.ts

Criando a estrutura para capturar o formulário

import { Component } from '@angular/core';

import { FormGroup, FormControl } from '@angular/forms';

@Component({

  selector: 'app-root',

  templateUrl: './app.component.html',

  styleUrls: ['./app.component.css']

})

export class AppComponent {

**//criando um objeto para capturar os**

**//campos do formulário**

**formCriarConta = new FormGroup({**

**nome: new FormControl('', []), //campo 'nome'**

**email: new FormControl('', []), //campo 'email'**

**senha: new FormControl('', []) //campo 'senha'**

**});**

**//função para capturar o SUBMIT do formulário**

**criarConta(): void {**

**//TODO**

**}**

}

/app.component.html

Integrando o conteúdo HTML da página com a programação feita para capturar os campos do formulário.

**<form [formGroup]="formCriarConta" (ngSubmit)="criarConta()">**

          <div class="mb-3">

            <label for="inputNome" class="form-label">

Entre com seu nome:

</label>

            <input

              type="text"

              class="form-control"

              id="inputNome"

              placeholder="Digite o seu nome"

              required

              formControlName="nome"

            />

          </div>

          <div class="mb-3">

            <label for="inputEmail" class="form-label">

Entre com seu email:

</label>

            <input

              type="email"

              class="form-control"

              id="inputEmail"

              aria-describedby="emailHelp"

              placeholder="Digite o seu e-mail"

              required

              formControlName="email"

            />

          </div>

          <div class="mb-3">

            <label for="inputSenha" class="form-label">

Entre com sua senha:

</label>

            <input

              type="password"

              class="form-control"

              id="inputSenha"

              placeholder="Digite a sua senha"

              required

              formControlName="senha"

            />

          </div>

          <div class="d-grid gap-2">

            <button class="btn btn-primary" type="submit">

              Realizar Cadastro

            </button>

          </div>

        </form>

**Exibindo os dados capturados:**

import { Component } from '@angular/core';

import { FormGroup, FormControl } from '@angular/forms';

@Component({

  selector: 'app-root',

  templateUrl: './app.component.html',

  styleUrls: ['./app.component.css']

})

export class AppComponent {

  //criando um objeto para capturar os

  //campos do formulário

  formCriarConta = new FormGroup({

    nome: new FormControl('', []), //campo 'nome'

    email: new FormControl('', []), //campo 'email'

    senha: new FormControl('', []) //campo 'senha'

  });

  //função para capturar o SUBMIT do formulário

  criarConta(): void {

    //capturando os valores preenchidos no formulário

    var dados = this.formCriarConta.value;

    //Teste!

    alert(JSON.stringify(dados));

  }

}

**Ao executar, vemos que os dados preenchidos no formulário são capturados com sucesso:**

Interface gráfica do usuário, Aplicativo, Word

Descrição gerada automaticamente

Para que possamos integrar outras aplicações com uma API JAVA, precisamos habilitar na API quais projetos poderão fazer requisições (chamadas) para os seus ENDPOINTS.

O nome desta configuração chama-se

**CORS (CROSS ORIGIN RESOURCE SHARING)**

package br.com.cotiinformatica.application.controllers;

import javax.validation.Valid;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.CrossOrigin;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RestController;

import br.com.cotiinformatica.application.dtos.PostAutenticarDTO;

import br.com.cotiinformatica.application.dtos.ResponseAutenticarDTO;

import br.com.cotiinformatica.domain.interfaces.IUsuarioDomainService;

import io.swagger.annotations.Api;

import io.swagger.annotations.ApiOperation;

@Api(tags = "Autenticação de usuários")

@RestController

public class AutenticarController {

@Autowired // injeção de dependência

private IUsuarioDomainService usuarioDomainService;

**@CrossOrigin(origins = "\*")**

@ApiOperation("ENDPOINT para autenticação de usuários e obtenção de Token.")

@PostMapping("/api/autenticar")

public ResponseEntity<ResponseAutenticarDTO> post

(@Valid @RequestBody PostAutenticarDTO dto) {

ResponseAutenticarDTO response = null;

try {

response = usuarioDomainService.autenticar(dto);

return ResponseEntity.status(HttpStatus.OK).body(response);

} catch (IllegalArgumentException e) {

response = new ResponseAutenticarDTO();

response.setStatus(401); // UNAUTHORIZED (CLIENT ERROR)

response.setMensagem(e.getMessage());

return ResponseEntity.status

(HttpStatus.BAD\_REQUEST).body(response);

} catch (Exception e) {

response = new ResponseAutenticarDTO();

response.setStatus(500); // INTERNAL SERVER ERROR

response.setMensagem(e.getMessage());

return ResponseEntity.status

(HttpStatus.INTERNAL\_SERVER\_ERROR).body(response);

}

}

}

**@angular/common/http**

Biblioteca utilizada no Angular para realizar as chamadas dos serviços das APIs integradas ao frontend.

/app.module.ts

import { NgModule } from '@angular/core';

import { BrowserModule } from '@angular/platform-browser';

import { FormsModule, ReactiveFormsModule } from '@angular/forms';

import { HttpClientModule } from '@angular/common/http';

import { AppComponent } from './app.component';

@NgModule({

  declarations: [

    AppComponent

  ],

  imports: [

    BrowserModule,

    FormsModule, //formulários reativos

    ReactiveFormsModule, //formulários reativos

    HttpClientModule //executar chamadas para APIs

  ],

  providers: [],

  bootstrap: [AppComponent]

})

export class AppModule { }

Voltando no componente:

/app.component.ts

import { Component } from '@angular/core';

import { FormGroup, FormControl } from '@angular/forms';

import { HttpClient } from '@angular/common/http';

@Component({

  selector: 'app-root',

  templateUrl: './app.component.html',

  styleUrls: ['./app.component.css']

})

export class AppComponent {

  constructor(

    private httpClient: HttpClient //injeção de dependência

  ) {

  }

  //criando um objeto para capturar os

  //campos do formulário

  formCriarConta = new FormGroup({

    nome: new FormControl('', []), //campo 'nome'

    email: new FormControl('', []), //campo 'email'

    senha: new FormControl('', []) //campo 'senha'

  });

**//função para capturar o SUBMIT do formulário**

**criarConta(): void {**

**//capturando os valores preenchidos no formulário**

**var dados = this.formCriarConta.value;**

**//executando a chamada para a API**

**this.httpClient.post('http://localhost:8083/api/criar-conta', dados)**

**.subscribe({ //capturando a resposta da API**

**next: (data) => { //sucesso!**

**alert(JSON.stringify(data));**

**},**

**error: (e) => { //erro!**

**alert(JSON.stringify(e.error));**

**}**

**});**

**}**

}

Interface gráfica do usuário, Texto, Aplicativo, Word

Descrição gerada automaticamente

Interface gráfica do usuário, Aplicativo, Word

Descrição gerada automaticamente

Interface gráfica do usuário, Aplicativo, Word

Descrição gerada automaticamente

/app.component.ts

Capturando os demais formulários do sistema.

import { Component } from '@angular/core';

import { FormGroup, FormControl } from '@angular/forms';

import { HttpClient } from '@angular/common/http';

@Component({

  selector: 'app-root',

  templateUrl: './app.component.html',

  styleUrls: ['./app.component.css']

})

export class AppComponent {

  apiUrl: string = 'http://localhost:8083';

  constructor(

    private httpClient: HttpClient //injeção de dependência

  ) {

  }

  //criando um objeto para capturar os

  //campos do formulário de criação de usuário

  formCriarConta = new FormGroup({

    nome: new FormControl('', []), //campo 'nome'

    email: new FormControl('', []), //campo 'email'

    senha: new FormControl('', []) //campo 'senha'

  });

  //criando um objeto para capturar os

  //campos do formulário de autenticação

  formAutenticar = new FormGroup({

    email: new FormControl('', []), //campo 'email'

    senha: new FormControl('', []) //campo 'senha'

  });

  //criando um objeto para capturar os

  //campos do formulário de recuperação de senha

  formRecuperarSenha = new FormGroup({

    email: new FormControl('', []), //campo 'email'

  });

  //função para capturar o SUBMIT do formulário

  criarConta(): void {

    //capturando os valores preenchidos no formulário

    var dados = this.formCriarConta.value;

    //executando a chamada para a API

    this.httpClient.post(this.apiUrl + '/api/criar-conta', dados)

      .subscribe({ //capturando a resposta da API

        next: (data) => { //sucesso!

          alert(JSON.stringify(data));

        },

        error: (e) => { //erro!

          alert(JSON.stringify(e.error));

        }

      });

  }

  //função para capturar o SUBMIT do formulário

  autenticar(): void {

    //capturando os valores preenchidos no formulário

    var dados = this.formAutenticar.value;

    //executando a chamada para a API

    this.httpClient.post(this.apiUrl + '/api/autenticar', dados)

      .subscribe({ //capturando a resposta da API

        next: (data) => { //sucesso!

          alert(JSON.stringify(data));

        },

        error: (e) => { //erro!

          alert(JSON.stringify(e.error));

        }

      });

  }

  //função para capturar o SUBMIT do formulário

  recuperarSenha(): void {

    //capturando os valores preenchidos no formulário

    var dados = this.formRecuperarSenha.value;

    //executando a chamada para a API

    this.httpClient.post(this.apiUrl + '/api/recuperar-senha', dados)

      .subscribe({ //capturando a resposta da API

        next: (data) => { //sucesso!

          alert(JSON.stringify(data));

        },

        error: (e) => { //erro!

          alert(JSON.stringify(e.error));

        }

      });

  }

}

/app.component.html

Capturando os demais formulários do sistema.

<div class="container">

  <div class="row justify-content-center mt-5">

    <div class="col-md-4">

      <div class="card">

        <div class="card-body">

          <div class="text-center">

            <h5 class="card-title text-center">

Autenticação de Usuários</h5>

            <p>Entre com as suas credenciais de acesso</p>

          </div>

          <hr/>

          <form [formGroup]="formAutenticar" (ngSubmit)="autenticar()">

            <div class="mb-3">

              <label for="inputEmail" class="form-label">

Entre com seu email:</label>

              <input

                type="email"

                class="form-control"

                id="inputEmail"

                aria-describedby="emailHelp"

                placeholder="Digite o seu e-mail"

                required

                formControlName="email"

              />

            </div>

            <div class="mb-3">

              <label for="inputPassword" class="form-label">

Entre com sua senha:</label>

              <input

                type="password"

                class="form-control"

                id="inputPassword"

                placeholder="Digite a sua senha"

                required

                formControlName="senha"

              />

            <div class="text-end mb-2">

              <a href="#" data-bs-toggle="modal" data-bs-

target="#janelaRecuperarSenha">

                Esqueci minha senha?

              </a>

            </div>

            </div>

            <div class="d-grid gap-2">

              <button class="btn btn-primary" type="submit">

                Acessar Sistema

              </button>

            </div>

            <div class="d-grid gap-2 mt-2">

              <a href="#" class="btn btn-light" data-bs-toggle="modal"

data-bs-target="#janelaCadastro">

                Não possui conta? <strong>Cadastre-se aqui!</strong>

              </a>

            </div>

          </form>

        </div>

      </div>

      <div class="mt-2 text-center">

        <small>

          Projeto Angular<br/>Java WebDeveloper COTI Informática

        </small>

      </div>

    </div>

  </div>

</div>

<!-- Modal para recuperação de senha do usuário -->

<div class="modal fade" id="janelaRecuperarSenha" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">

  <div class="modal-dialog modal-dialog-centered">

    <div class="modal-content">

      <div class="modal-header">

        <h1 class="modal-title fs-5" id="exampleModalLabel">

Esqueci minha senha</h1>

        <button type="button" class="btn-close" data-bs-dismiss="modal"

aria-label="Close"></button>

      </div>

      <div class="modal-body">

        <form [formGroup]="formRecuperarSenha"

(ngSubmit)="recuperarSenha()">

          <div class="mb-3">

            <label for="inputEmail" class="form-label">

Entre com seu email:</label>

            <input

              type="email"

              class="form-control"

              id="inputEmail"

              aria-describedby="emailHelp"

              placeholder="Digite o seu e-mail"

              required

              formControlName="email"

            />

          </div>

          <div class="d-grid gap-2">

            <button class="btn btn-primary" type="submit">

              Confirmar

            </button>

          </div>

        </form>

      </div>

    </div>

  </div>

</div>

<!-- Modal para cadastro do usuário -->

<div class="modal fade" id="janelaCadastro" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">

  <div class="modal-dialog modal-dialog-centered">

    <div class="modal-content">

      <div class="modal-header">

        <h1 class="modal-title fs-5" id="exampleModalLabel">

Criar Conta de Usuário</h1>

        <button type="button" class="btn-close" data-bs-dismiss="modal"

aria-label="Close"></button>

      </div>

      <div class="modal-body">

        <form [formGroup]="formCriarConta" (ngSubmit)="criarConta()">

          <div class="mb-3">

            <label for="inputNome" class="form-label">

Entre com seu nome:</label>

            <input

              type="text"

              class="form-control"

              id="inputNome"

              placeholder="Digite o seu nome"

              required

              formControlName="nome"

            />

          </div>

          <div class="mb-3">

            <label for="inputEmail" class="form-label">

Entre com seu email:</label>

            <input

              type="email"

              class="form-control"

              id="inputEmail"

              aria-describedby="emailHelp"

              placeholder="Digite o seu e-mail"

              required

              formControlName="email"

            />

          </div>

          <div class="mb-3">

            <label for="inputSenha" class="form-label">

Entre com sua senha:</label>

            <input

              type="password"

              class="form-control"

              id="inputSenha"

              placeholder="Digite a sua senha"

              required

              formControlName="senha"

            />

          </div>

          <div class="d-grid gap-2">

            <button class="btn btn-primary" type="submit">

              Realizar Cadastro

            </button>

          </div>

        </form>

      </div>

    </div>

  </div>

</div>

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

Descrição gerada automaticamente

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

**Exibindo as mensagens do cadastro do usuário na página:**

import { Component } from '@angular/core';

import { FormGroup, FormControl } from '@angular/forms';

import { HttpClient } from '@angular/common/http';

@Component({

  selector: 'app-root',

  templateUrl: './app.component.html',

  styleUrls: ['./app.component.css']

})

export class AppComponent {

  apiUrl: string = 'http://localhost:8083';

  mensagem\_criarconta: string = '';

  constructor(

    private httpClient: HttpClient //injeção de dependência

  ) {

  }

  //criando um objeto para capturar os

  //campos do formulário de criação de usuário

  formCriarConta = new FormGroup({

    nome: new FormControl('', []), //campo 'nome'

    email: new FormControl('', []), //campo 'email'

    senha: new FormControl('', []) //campo 'senha'

  });

  //criando um objeto para capturar os

  //campos do formulário de autenticação

  formAutenticar = new FormGroup({

    email: new FormControl('', []), //campo 'email'

    senha: new FormControl('', []) //campo 'senha'

  });

  //criando um objeto para capturar os

  //campos do formulário de recuperação de senha

  formRecuperarSenha = new FormGroup({

    email: new FormControl('', []), //campo 'email'

  });

  //função para capturar o SUBMIT do formulário

  criarConta(): void {

    //capturando os valores preenchidos no formulário

    var dados = this.formCriarConta.value;

    //executando a chamada para a API

    this.httpClient.post(this.apiUrl + '/api/criar-conta', dados)

      .subscribe({ //capturando a resposta da API

        next: (data: any) => { //sucesso!

          this.mensagem\_criarconta = data.mensagem;

          this.formCriarConta.reset(); //limpar o formulário

        },

        error: (e) => { //erro!

          if(e.error.mensagem){

            this.mensagem\_criarconta = e.error.mensagem;

          }

          else if(e.error.errors){

            this.mensagem\_criarconta = e.error.errors;

          }

        }

      });

  }

  //função para capturar o SUBMIT do formulário

  autenticar(): void {

    //capturando os valores preenchidos no formulário

    var dados = this.formAutenticar.value;

    //executando a chamada para a API

    this.httpClient.post(this.apiUrl + '/api/autenticar', dados)

      .subscribe({ //capturando a resposta da API

        next: (data) => { //sucesso!

          alert(JSON.stringify(data));

        },

        error: (e) => { //erro!

          alert(JSON.stringify(e.error));

        }

      });

  }

  //função para capturar o SUBMIT do formulário

  recuperarSenha(): void {

    //capturando os valores preenchidos no formulário

    var dados = this.formRecuperarSenha.value;

    //executando a chamada para a API

    this.httpClient.post(this.apiUrl + '/api/recuperar-senha', dados)

      .subscribe({ //capturando a resposta da API

        next: (data) => { //sucesso!

          alert(JSON.stringify(data));

        },

        error: (e) => { //erro!

          alert(JSON.stringify(e.error));

        }

      });

  }

}

**Na página:**

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

<http://localhost:4200/>

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente