CENTRO PAULA SOUZA

Centro Paula Souza

GOVERNO DO ESTADO SÃO PAULO

CalmaMente

Projeto Integrador

Laura Jane Antunes,
RA 3011392313034
Marcus Vinicyus Souza Barros,
RA 3011392313024
Yara Paes de Bona,
RA 3011392313039

Orientador: Rodrigo Diver

Centro Paula Souza

GOVERNO DO ESTADO SÃO PAULO

INTRODUÇÃO

O CalmaMente surgiu devido à necessidade de fornecer apoio e recursos abrangentes para pessoas que enfrentam ansiedade, depressão e transtorno de déficit de atenção e hiperatividade (TDAH).

O site foi desenvolvido para promover o bem-estar mental e melhorar a qualidade de vida dos usuários.

MOTIVAÇÃO

A criação do CalmaMente foi motivada pela crescente necessidade de recursos acessíveis e de alta qualidade para apoio à saúde mental.

Nos últimos anos, a conscientização sobre a importância da saúde mental tem aumentado, mas ainda existem lacunas significativas no acesso a informações e suporte adequados.

Não recebem tratamento:

Países de baixa e média renda: Entre 76% e 85%
Países de alta renda: Entre 35% e 50%

Fonte: organização Pan-Americana de saúde

DIAGRAMA DE CLASSE

O diagrama de classes do CalmaMente foi elaborado para ilustrar a estrutura e os relacionamentos entre os diferentes componentes do sistema, proporcionando uma visão clara e organizada da arquitetura do projeto.

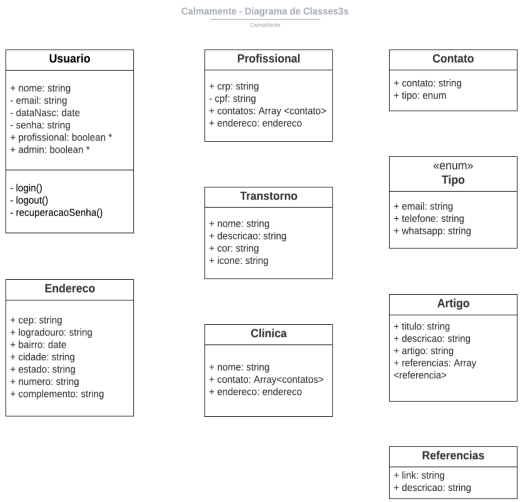
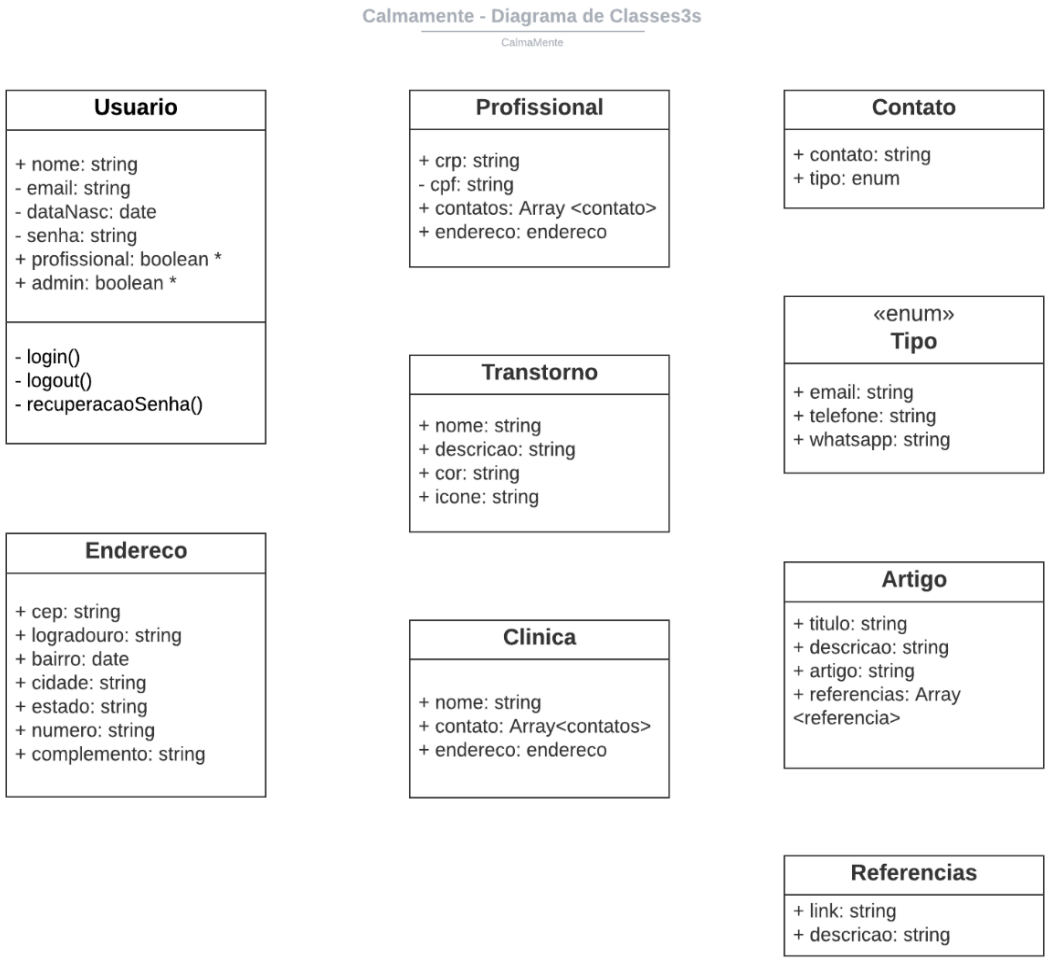


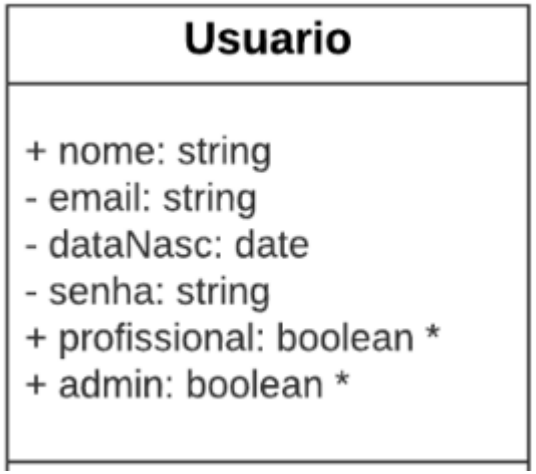
DIAGRAMA DE CLASSE



Fatec Votorantim

Centro Paula Souza

ATRIBUTOS



nome: uma string que armazena o nome do usuário.

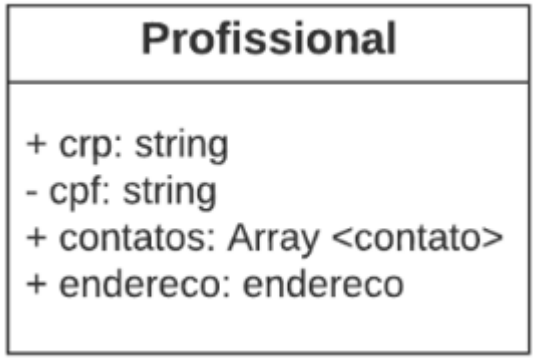
email: uma string que armazena o endereço de e-mail do usuário.

dataNasc: um date que representa a data de nascimento do usuário.

senha: uma string que armazena a senha do usuário.
profissional: um booleano que pode ou não ser preenchido.

admin: um booleano que pode ou não ser preenchido.

ATRIBUTOS



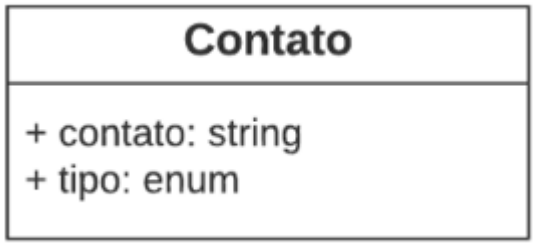
crp: uma string que armazena o registro profissional do profissional.

cpf: uma string que armazena o número de CPF do profissional.

contatos: um Array que armazena os dados de contato do profissional.

endereco: endereço.

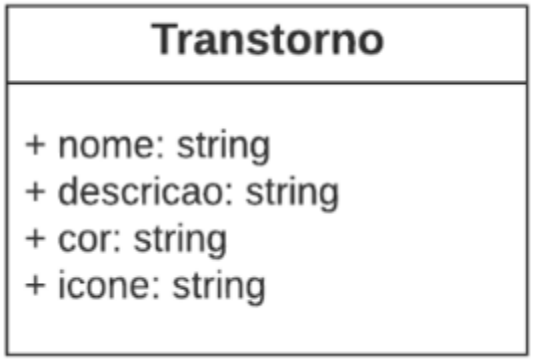
ATRIBUTOS



contato: uma string que contém o dados para contato.

tipo: chama um Enum que fornece um tipo de contato.

ATRIBUTOS



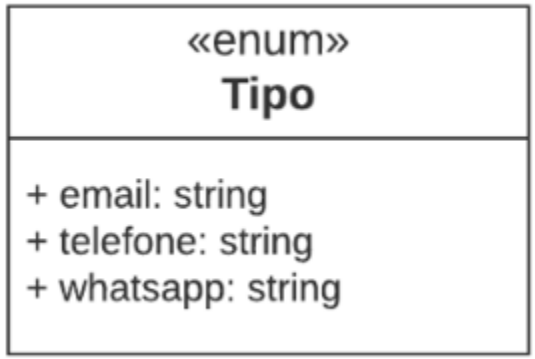
nome: uma string que descreve o nome do transtorno.

descricao: uma string que fornece uma descrição mais detalhada do transtorno, incluindo características, sintomas ou informações relevantes relacionadas a ele.

cor: uma string que determina a cor do layout que será usado.

icone: uma string que o determina o endereço da imagem que será usada no layout.

ATRIBUTOS



email: uma string que armazena o endereço de e-mail.

telefone: uma string que armazena o telefone.

whatsapp: uma string que armazena o contato de whatsapp.

ATRIBUTOS

cep: uma string que armazena o código postal (CEP) do endereço.

logradouro: uma string que armazena o nome da rua do endereço.

bairro: uma string que armazena o nome do bairro.

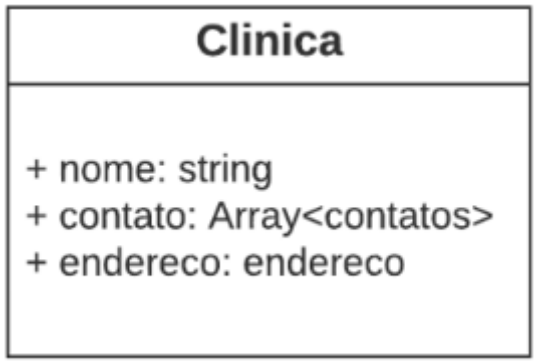
cidade: uma string que armazena o nome da cidade.

estado: uma string que armazena o nome do estado.

numero: uma string que armazena o número da residência ou prédio, ou s/n.

complemento: uma string que armazena informações adicionais do endereço, como apartamento, bloco, etc.

ATRIBUTOS

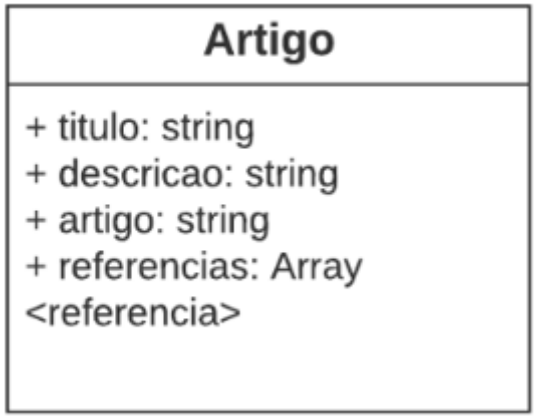


nome: uma string que armazena o nome da clínica.

contatos: um Array que armazena os dados de contato da clínica.

endereco: endereço.

ATRIBUTOS



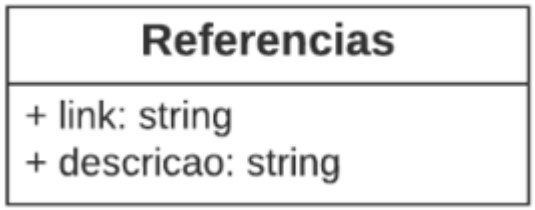
titulo: uma string que descreve o título do artigo.

descricao: uma string que fornece uma breve descrição do artigo.

artigo: uma string que contém o conteúdo completo do artigo.

referencias: um Array que armazena a autoria do artigo.

ATRIBUTOS



link: uma string que contém o link para acessar o artigo.

descricao: uma string que fornece uma breve descrição do artigo.

DEFINIÇÃO DOS MÉTODOS

Os métodos foram desenvolvidos e definidos de forma a proporcionar organização, reutilização e abstração do código. Eles desempenham um papel crucial na criação de sites eficientes, responsivos e fáceis de manter.

DEFINIÇÃO DAS FUNÇÕES

- login()

Processo de autenticação em que o usuário fornece suas credenciais, nome de usuário e senha, para acessar uma área restrita do sistema.

Este processo verifica a identidade do usuário e garante que apenas indivíduos autorizados tenham acesso a determinados recursos ou informações.

DEFINIÇÃO DAS FUNÇÕES

- logout()

Processo de encerrar uma sessão autenticada no site. Ao fazer logout, o usuário desconecta sua conta, encerrando o acesso a áreas restritas e protegendo suas informações pessoais e dados sensíveis de acessos não autorizados.

DEFINIÇÃO DAS FUNÇÕES

- recuperacaoSenha()

Recuperação de senha é o processo que permite ao usuário redefinir sua senha esquecida ou perdida. Envolve a verificação da identidade do usuário através de um e-mail, seguido por um link ou código para criar uma nova senha.

DESCRIÇÃO DOS MÉTODOS

```
// ########################################## FUNÇÃO PARA LOGOUT ##################

document.getElementById('logout').addEventListener('click', function() {
    // Redirecionar para a página de login ou executar outras ações de logout
    window.location.href = 'index.html';
});
```

O código seleciona o elemento HTML com o ID logout, adiciona um ouvinte de evento para o clique nesse elemento. Quando o elemento é clicado, a função redireciona o usuário para a página index.html, efetivamente fazendo o logout.

“Mostra apenas a ação de redirecionamento para a página de login."

DESCRIÇÃO DOS MÉTODOS

```
//Captura o botão logout pelo ID
const btnLogout = document.getElementById('logout')
//Adicionamos o listener no click
btnLogout.addEventListener('click', function(){
    //Removemos o localStorage
    localStorage.removeItem('token')
    //Redirecionamos para o login
    window.location.href = 'index.html'
})
```

Captura do botão pelo ID: Acessa o botão de logout no DOM (Document Object Model) usando getElementById.

Adição de listener de clique: Adiciona um evento de clique ao botão, para que quando o botão for clicado, a função anônima definida seja executada.

Remoção do token do localStorage: Remove o token de autenticação, o que efetivamente desloga o usuário do ponto de vista do cliente (navegador).

Redirecionamento para a página de login: Após remover o token, o usuário é redirecionado para a página de login, indicando que ele foi deslogado.

DESCRIÇÃO DOS MÉTODOS

```
import { Component } from '@angular/core';
import { footerColors } from 'src/app/components/footer/footer.component';
import { btnColors, h1HeaderColors, headerColors, textColors } from 'src/app/components/header/header.component';
import { h2Colors } from 'src/app/components/card-info/card-info.component';
import { DisorderResponse } from 'src/app/types';
import { HttpClient } from '@angular/common/http';
import { TranstornoService } from 'src/app/services/transtorno-service.service';

@Component({
  selector: 'app-tdah-page',
  templateUrl: './tdah-page.component.html',
  styleUrls: ['./tdah-page.component.css']
})
export class TdahPageComponent {
  public headerColors = headerColors;
  public btnColors = btnColors;
  public textColors = textColors;
  public footerColors = footerColors;
  public h2Colors = h2Colors;
  public h1HeaderColors = h1HeaderColors;
  public apiResponse?: DisorderResponse;

  constructor(private httpClient: HttpClient) {}

  ngOnInit() {
    this.getTranstornos()
  }

  async getTranstornos() {
    const service = new TranstornoService(this.httpClient);

    service.getTranstorno(3).subscribe(
      data => this.apiResponse = data
    );
  }
}
```

Em resumo, este componente TdahPageComponent é responsável por exibir informações sobre um transtorno específico, estilizado com várias cores importadas de outros componentes e fazendo uso do serviço TranstornoService para obter os dados necessários.

DESCRIÇÃO DOS MÉTODOS

```
import { Component } from '@angular/core';
import { footerColors } from 'src/app/components/footer/footer.component';
import { btnColors, h1HeaderColors, headerColors, textColors } from 'src/app/components/header/header.component';
import { h2Colors } from 'src/app/components/card-info/card-info.component';
import { DisorderResponse } from 'src/app/types';
import { HttpClient } from '@angular/common/http';
import { TranstornoService } from 'src/app/services/transtorno-service.service';
```

Estas são as importações necessárias para o funcionamento do componente:
Component é o decorador usado para definir um componente Angular.

Várias constantes de cores (footerColors, btnColors, etc.) são importadas de outros componentes para estilizar o componente atual.

DisorderResponse é um tipo de dados.

HttpClient é usado para fazer requisições HTTP.
TranstornoService é um serviço personalizado que lida com operações relacionadas a "transtornos".

DESCRIÇÃO DOS MÉTODOS

```
@Component({
  selector: 'app-tdah-page',
  templateUrl: './tdah-page.component.html',
  styleUrls: ['./tdah-page.component.css']
})
```

selector: Define o seletor CSS usado para incluir este componente em um template.

templateUrl: O caminho para o arquivo de template HTML deste componente.

styleUrls: O caminho para o arquivo CSS deste componente.

DESCRIÇÃO DOS MÉTODOS

```
export class TdahPageComponent {
  public headerColors = headerColors;
  public btnColors = btnColors;
  public textColors = textColors;
  public footerColors = footerColors;
  public h2Colors = h2Colors;
  public h1HeaderColors = h1HeaderColors;
  public apiResponse?: DisorderResponse;
```

Estas são propriedades públicas do componente que serão usadas no template para estilização e para armazenar a resposta da API.

DESCRIÇÃO DOS MÉTODOS

```
constructor(private httpClient: HttpClient) {}
```

O construtor injeta uma dependência HttpClient que será usada para fazer chamadas HTTP.

```
ngOnInit() {
  this.getTranstornos()
}
```

O método ngOnInit é um ciclo de vida do Angular chamado assim que o componente é inicializado. Aqui, ele chama o método getTranstornos..

DESCRIÇÃO DOS MÉTODOS

```
async getTranstornos() {
  const service = new TranstornoService(this.httpClient);

  service.getTranstorno(3).subscribe(
    data => this.apiResponse = data
  );

}
```

Este método cria uma instância do TranstornoService usando o HttpClient injetado e chama o método getTranstorno passando o ID 3.

A resposta da API é então armazenada na propriedade apiResponse.

getTranstorno(3) é um método assíncrono do serviço que faz uma requisição HTTP para obter informações sobre um transtorno específico (neste caso, o ID 3).

DESCRIÇÃO DOS MÉTODOS

Método recuperarSenha(). **Atualmente, o método está em desenvolvimento**.

Este método será uma parte crucial da nossa API REST desenvolvida em TypeScript, e terá como objetivo permitir que os usuários recuperem suas senhas de maneira segura e eficiente.

Funcionamento Planejado: O usuário solicitará a recuperação da senha informando seu endereço de e-mail cadastrado.

Geração de Token: O sistema gerará um token temporário e seguro, que será enviado para o e-mail do usuário.Envio do E-mail: O usuário receberá um e-mail com um link contendo o token. Esse link redirecionará para uma página de redefinição de senha.

Validação do Token: Ao acessar o link, o sistema validará o token para garantir que ele ainda esteja válido e não tenha sido utilizado anteriormente.

Redefinição de Senha: Uma vez validado o token, o usuário poderá definir uma nova senha. A nova senha será então criptografada e armazenada de forma segura no banco de dados.

Tecnologias Utilizadas:

- **TypeScript**: Para garantir a tipagem estática e a robustez do código.
- **Node.js**: Para a criação do servidor backend.
- **Express.js**: Para facilitar a construção da API REST.JWT (JSON Web Tokens): Para a geração e validação de tokens de recuperação de senha.
- **Nodemailer**: Para o envio de e-mails aos usuários.

Este método será desenvolvido com foco na segurança e na usabilidade, proporcionando uma experiência fluida para os usuários que precisarem recuperar suas senhas.

CONCLUSÃO

Conforme visto anteriormente a idéia de desenvolver o site para este segmento é exatamente de oferecer suporte a quem precisa.

Oferecendo uma plataforma dinâmica, intuitiva e principalmente informativa, não somente para pessoas que sofrem com estes transtornos mas também para toda a comunidade que deseja obter maior conhecimento sobre o tema.