

**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA
PAULA SOUZA**

**Faculdade de Tecnologia Baixada Santista
Rubens Lara**

**Curso Superior de Tecnologia em
Sistemas para Internet**

**Jackeline Ferreira Fidêncio
Leonardo Lima Gomes
Pedro Gregorio do Couto**

**Abraço Rosa
Sistema para combate de violência contra mulheres**

**Santos, SP
2023**

**Jackeline Ferreira Fidêncio
Leonardo Lima Gomes
Pedro Gregorio do Couto**

**Abraço Rosa
Sistema para combate de violência contra mulheres**

Trabalho de Conclusão de Curso apresentado à Faculdade de Tecnologia Rubens Lara, como exigência para a obtenção do Título de Tecnólogo em Sistemas para Internet.

Orientador: Profa. Dra. Vanina Carrara Sigrist

**Santos, SP
2023**

RESUMO

O desenvolvimento de um aplicativo móvel para ajudar na prevenção e combate à violência contra a mulher. O objetivo deste projeto é desenvolver um aplicativo para ajudar mulheres a enfrentar a violência e buscar apoio. Oferecerá recursos como orientações, informações legais, autodefesa, chat de suporte, o desenvolvimento de um aplicativo móvel para ajudar na prevenção e combate à violência contra a mulher buscando conscientização e motivação para as vítimas procurar por ajuda.

Palavras-chave: Aplicativo móvel, prevenção, combate, violência contra a mulher, recursos, orientações, informações legais, autodefesa, chat de suporte, conscientização, ajuda.

ABSTRACT

The development of a mobile application to assist in the prevention and combat of violence against women. The aim of this project is to create an app to help women face violence and seek support. It will provide resources such as guidance, legal information, self-defense, support chat, developing a mobile application to aid in the prevention and combat of violence against women, aiming for awareness and motivation for victims to seek help.

Keywords: Mobile app, prevention, combat, violence against women, resources, guidance, legal information, self-defense, support chat, awareness, help.

LISTA DE ABREVIATURAS E SIGLAS

SPA - SINGLE PAGE APPLICATIONS	12
MVC - MODEL, VIEW, CONTROLLER	12
IoT - INTERNET OF THINGS.....	13
API - APPLICATION PROGRAMMING INTERFACE	13
IoC - INVERSION OF CONTROL.....	14
DI - DEPENDENCY INJECTION.....	14
HTTP - HYPERTEXT TRANSFER PROTOCOL SECURE	19
DTO - DATA TRANSFER OBJECT.....	20
URL - UNIFORM RESOURCE LOCATOR	20
JSON - JAVASCRIPT OBJECT NOTATION	20
ID -IDENTITY	20

LISTA DE ILUSTRAÇÕES

ILUSTRAÇÃO 01 – PENHAS (LOGO)	10
ILUSTRAÇÃO 02 – SOS MULHER (LOGO).....	11
ILUSTRAÇÃO 03 – ANGULAR (LOGO)	12
ILUSTRAÇÃO 04 – TYPECRIPT (LOGO)	12
ILUSTRAÇÃO 05 – JAVA (LOGO).....	13
ILUSTRAÇÃO 06 – SPRING (LOGO)	13
ILUSTRAÇÃO 07 – IONIC (LOGO)	14
ILUSTRAÇÃO 08 – QUADRO DE REQUISITOS FUNCIONAIS	15
ILUSTRAÇÃO 09 – QUADRO DE REQUISITOS NÃO FUNCIONAIS	15
ILUSTRAÇÃO 10 – DIAGRAMA DE CASO DE USO	16
ILUSTRAÇÃO 11 – MODELO ENTIDADE RELACIONAMENTO.....	17
ILUSTRAÇÃO 12 – DIAGRAMA ENTIDADE-RELACIONAMENTO.....	17
ILUSTRAÇÃO 13 – CÓDIGO BACK-END.....	18
ILUSTRAÇÃO 14 – CÓDIGO BACK-END.....	19
ILUSTRAÇÃO 15 – TELA DE LOGIN.....	21
ILUSTRAÇÃO 16 – TELA DE CADASTRO DE GUARDIÃO.....	22
ILUSTRAÇÃO 17 – TELA DE GUARDIÕES	22
ILUSTRAÇÃO 18 – GRÁFICO DAS RESPOSTAS DA PERGUNTA 1	26
ILUSTRAÇÃO 19 – GRÁFICO DE RESPOSTAS DA PERGUNTA 2.....	26
ILUSTRAÇÃO 20 – RESPOSTAS DA PERGUNTA 3	27

SUMÁRIO

1 INTRODUÇÃO.....	8
1.1 OBJETIVO.....	9
1.1.1 OBJETIVO GERAL.....	9
1.1.2 OBJETIVOS ESPECÍFICOS.....	10
1.2 ESTADO DA ARTE.....	10
2.1 ANÁLISE DO SISTEMA.....	12
2.1.1 ANÁLISE DE REQUISITOS.....	14
2.1.2 DIAGRAMA DE CASO DE USO.....	16
2.1.3 FLUXO DE EVENTOS.....	16
2.2 BANCO DE DADOS.....	17
2.3 CAMADA DE NEGÓCIO.....	18
2.4 CAMADA DE APRESENTAÇÃO.....	21
3 RESULTADO.....	24
3.1 PERFIL DOS PARTICIPANTES DO TESTE E SUAS OBSERVAÇÕES.....	24
3.2 RESULTADOS DOS QUESTIONÁRIOS.....	25
3.3 CONCLUSÃO.....	27
REFERÊNCIAS.....	29
APENDICE A.....	30
APENDICE B.....	32

1 INTRODUÇÃO

Segundo Moreno, 2022, um levantamento revelou que 45% das mulheres já tiveram seus corpos tocados sem consentimento em locais públicos, enquanto apenas 5% dos homens admitem ter feito isso. Os dados destacam a necessidade de incluir a noção de consentimento na educação dos meninos desde a infância para combater esse comportamento agressivo. Além disso, o estudo aponta que 31% das mulheres sofreram tentativa de abuso sexual e mais de 40% foram agredidas ou xingadas por rejeitarem investidas. Esses números mostram que tanto o ambiente privado quanto o público continuam sendo espaços difíceis para as mulheres, apesar dos avanços sociais. O levantamento também destaca o aumento no número de homicídios de mulheres ao longo da última década, com uma estimativa de 4.954 mortes violentas em 2015, representando uma média de 413 mortes por mês, 14 por dia e quase uma morte por hora. Essa violência de gênero configura o feminicídio, evidenciando a necessidade de leis e medidas efetivas para combater essa problemática, como citado na bibliografia “a violência contra as mulheres reduz o poder de decisão delas e promove a desigualdade econômica, social, política e cultural; acredita-se que a vida da mulher vale menos que a do homem; as leis de equilíbrio são necessárias, mas insuficientes (a questão é cultural); para ser considerado “macho” o homem precisa ser dominante, forte e nunca se render; a mulher precisa se preocupar com a aparência, submissão e estar a serviço do “seu homem”; o homem não perde nenhuma oportunidade para mostrar sua “força” contra o “objeto” que lhe pertence; a cultura machista está impregnada nos jovens latino-americanos (e também as jovens são as vítimas preferenciais); claro que o problema não é exclusivo da América Latina, mas é aí que ele se manifesta mais agudamente; uma de cada três mulheres do mundo já experimentaram violência física ou sexual; de todas as mulheres mortas no mundo quase metade o é pelas mãos de seus companheiros.” (Gomes, 2016).

O objetivo principal de um aplicativo relacionado à violência contra a mulher é ajudar a prevenir, denunciar e combater a violência de gênero. Esses aplicativos fornecem recursos úteis para mulheres em situações de perigo ou que procuram apoio. Eles podem incluir linhas diretas de apoio, informações legais, locais seguros,

serviços de emergência, orientações sobre como lidar com a violência e ferramentas para registrar incidentes e coletar evidências.

O propósito desses aplicativos é capacitar as mulheres, promover a igualdade de gênero e contribuir para a criação de uma sociedade mais segura e livre de violência. É crucial utilizar a tecnologia de forma responsável para combater problemas sociais e promover o bem-estar de todos os membros da sociedade, incluindo as mulheres.

Assim surgiu a ideia do "Abraço Rosa" que é uma plataforma que visa proteger e capacitar mulheres em relação à sua segurança e bem-estar. Ela oferece várias ferramentas e recursos abrangentes para lidar com desafios específicos que as mulheres enfrentam em sua vida diária. A plataforma é intuitiva e fácil de usar, tornando-a acessível para mulheres de todas as idades e níveis de familiaridade com a tecnologia. Uma das principais características é o rastreamento de localização em tempo real, permitindo que as mulheres compartilhem sua localização com contatos de confiança. Além disso, há um botão de pânico para solicitar ajuda em emergências, gravação de áudio como evidência e acesso a informações de contato e locais seguros. A plataforma também oferece uma comunidade de suporte onde as mulheres podem se conectar com outras usuárias para compartilhar experiências, obter conselhos e apoio emocional.

1.1 OBJETIVO

O aplicativo que tem como principal objetivo ajudar as mulheres a se protegerem da violência, fornecendo informações, apoio emocional, orientações legais e meios seguros de denúncia. Ele busca criar um ambiente seguro e promover a igualdade de gênero, contribuindo para uma sociedade livre de violência contra as mulheres.

1.1.1 OBJETIVO GERAL

O aplicativo Abraço Rosa irá ajudar mulheres a se protegerem contra a violência e fornecer recursos necessários para enfrentar situações de perigo. O

aplicativo visa criar um ambiente seguro, capacitando as mulheres e promovendo a igualdade de gênero.

Em resumo, o objetivo geral do aplicativo Abraço Rosa é proporcionar às mulheres as ferramentas e recursos necessários para se protegerem contra a violência. Ao oferecer informações, apoio emocional, orientações legais e meios seguros de denúncia, o aplicativo busca promover a segurança, a igualdade de gênero e contribuir para a construção de uma sociedade livre de violência contra as mulheres.

1.1.2 OBJETIVOS ESPECÍFICOS

Objetivos específicos do nosso projeto:

- Realizar um estudo e análise geral do tema
- Levantamento de funcionalidades e requisitos importantes para um aplicativo de violência contra mulher
- Avaliar a eficácia de usabilidade do sistema por meio de testes de usuários.

1.2 ESTADO DA ARTE

O Aplicativo tem como base as aplicações que tinham as funcionalidades mais parecidas com o que se deseja no nosso aplicativo. São elas:

Ilustração 1 – PenhaS (logo)



Fonte: <https://azmina.com.br/projetos/penhas/>, 2023

Segundo o texto “Criando conexões contra a violência”, o PenhaS, ilustração 1, é um aplicativo que oferece apoio e recursos para mulheres em relacionamentos abusivos. Ele proporciona um ambiente seguro onde as mulheres cadastradas, independentemente de estarem em situação de violência ou não, têm acesso a informações, diálogo sigiloso, rede de acolhimento e um botão de pânico. O aplicativo também auxilia na produção de provas através da gravação de áudio em momentos de violência, possibilitando o fortalecimento legal das vítimas. Seu

objetivo é fornecer suporte abrangente e facilitar o acesso a recursos que ajudem as mulheres a sair de relacionamentos abusivos e promover a conscientização sobre a violência doméstica.

Ilustração 2 – SOS Mulher (logo)



Fonte: <https://www.sosmulher.sp.gov.br/>, 2021

O SOS MULHER, conforme ilustração 2, é focado no apoio à mulher em situação de vulnerabilidade e tem como base três pilares: segurança, saúde e independência financeira.

A plataforma é uma ampliação do projeto que foi iniciado em março de 2019 pelo Governo do Estado com o lançamento do aplicativo que permite que as vítimas de violência doméstica peçam ajuda para polícia apertando apenas um botão do celular por cinco segundos.

Nesta etapa serão disponibilizadas orientações importantes sobre assuntos que visam garantir proteção em relação às principais formas de violência listadas na Lei Maria da Penha: física, psicológica, moral, sexual e patrimonial.

2 DESENVOLVIMENTO

Nesse capítulo será abordado as fases do desenvolvimento do sistema proposto. Para tanto, esse capítulo será subdividido nos seguintes itens; análise do sistema, a estrutura do Banco de Dados, as ferramentas utilizadas para o desenvolvimento do *front-end* e as principais rotinas implementadas no *back-end*.

2.1 ANÁLISE DO SISTEMA

As linguagens utilizadas no aplicativo serão:

Ilustração 3 – Angular (logo)



Fonte: https://more.ufsc.br/homepage/inserir_homepage, 2023

Angular, ilustração 3, é um *framework* de desenvolvimento *web* mantido pelo Google. Ele é usado para criar aplicativos *web* de página única (*SPAs - Single Page Applications*) com recursos avançados e uma arquitetura sólida. O Angular utiliza TypeScript e componentes reutilizáveis para criar interfaces de usuário dinâmicas e interativas.

Ele oferece recursos como *binding* de dados bidirecional, injeção de dependência e roteamento. O Angular segue o padrão *MVC* (Modelo, visão e controle) e é amplamente utilizado na indústria de desenvolvimento *web*.

Ilustração 4 – TypeScript (logo)



Fonte: <https://www.dataside.com.br/dataside-community/linguagem-de-programacao/o-que-e-o-typescript>, 2020

TypeScript, ilustração 4, é uma linguagem de programação desenvolvida pela Microsoft que adiciona recursos e funcionalidades extras ao JavaScript. Ela oferece a opção de tipagem estática, o que ajuda a identificar erros de tipo durante a compilação e a melhorar a qualidade do código. O TypeScript suporta as características do JavaScript moderno, como classes, interfaces e módulos, e é compilado para JavaScript padrão. É comumente utilizado no desenvolvimento *web*, especialmente com *frameworks* como Angular, React e Vue.js. No entanto, pode ser usado em qualquer projeto JavaScript.

Ilustração 5 – Java (logo)



Fonte: https://www.alura.com.br/artigos/java?gclid=CjwKCAjwv8qkBhAnEiwAkY-ahrhpcZnqeAwzdQhAQw0b7FIHwq40f_XDUXqjfw1s0oLAOL6ImpV-2hoCkuYQAvD_BwE,

2023

Java é uma linguagem de programação de alto nível, conforme ilustração 5, orientada a objetos e multiplataforma. Ela foi desenvolvida pela Sun Microsystems e é amplamente utilizada para o desenvolvimento de aplicativos corporativos, aplicativos móveis, sistemas embarcados, jogos e aplicações de Internet das Coisas (*IoT*). Java possui uma sintaxe simples e amigável, e sua plataforma oferece uma vasta biblioteca de classes e *APIs*. A linguagem é conhecida por sua robustez, segurança e portabilidade, além de contar com uma grande comunidade de desenvolvedores.

Ilustração 6 – Spring (logo)



Fonte: <https://www.treinaweb.com.br/blog/o-que-e-o-spring-boot>, 2021

O Spring Framework, ilustração 6, é um *framework* de desenvolvimento de aplicativos Java. Ele utiliza o padrão de inversão de controle (*IoC*) e injeção de dependência (*DI*) para facilitar o desenvolvimento e a integração de componentes de *software*. O Spring oferece diversos módulos e recursos para abordar diferentes aspectos do desenvolvimento, como segurança, persistência de dados, *web services* e escalabilidade. O Spring *MVC* é um módulo popular que permite o desenvolvimento de aplicativos *web* baseados no padrão *MVC*. O Spring Framework é amplamente adotado na comunidade Java devido à sua facilidade de uso e suporte ativo.

Ilustração 7 – Ionic (logo)



Fonte: <https://blog.wkm.com.br/como-configurar-ambiente-ionic-ef0a6d2de803>, 2022

Ionic, ilustração 7, é um *Framework* para criação de aplicações móveis e *desktop* de alto desempenho e qualidade, que utiliza as tecnologias *web HTML*, *CSS* e *JavaScript*. Como diz em seu site, o Ionic ajuda as equipes a criarem e distribuírem belos aplicativos híbridos em várias plataformas, possuindo foco na experiência do usuário ou interação do usuário com o aplicativo (controles, interação, gestos, animações).

Projetado para funcionar e ser exibido em diferentes plataformas, possui um *design* limpo, simples e funcional, com componentes padrões, tipografia, paradigmas interativos e diversos *templates* (formato gráfico pronto que serve para facilitar o processo de criação de layouts de sites).

2.1.1 ANÁLISE DE REQUISITOS

A análise de requisitos é uma etapa fundamental no desenvolvimento de *software*. Para isso, costumam ser criadas duas tabelas: uma para os requisitos funcionais, que detalham as funções específicas que o sistema deve executar, e outra para os requisitos não funcionais, que definem características gerais como

desempenho, segurança e usabilidade. Enquanto os requisitos funcionais se concentram nas ações e comportamentos do *software*, os requisitos não funcionais enfatizam atributos e qualidades gerais, contribuindo para a qualidade do sistema além de suas funcionalidades específicas.

Ilustração 8 – Quadro de Requisitos Funcionais

RNF01	O sistema conta com um processamento de até 1 segundo para o envio da mensagem de alerta ao protetor.
RNF02	O sistema integra com outras aplicações: Android a partir da versão 8.1 e IOS a partir da versão 11.
RNF03	O sistema irá tratar os dados dos usuários com confidencialidade e segurança.

Fonte: Elaborado pelos autores, 2023

Requisitos funcionais servem para ter uma descrição sobre uma função específica que o *software* precisa executar. Tendo como definição as ações que o *software* deve realizar para atender às necessidades do usuário ou dos objetivos finais do sistema.

Ilustração 9 – Quadro de Requisitos não Funcionais

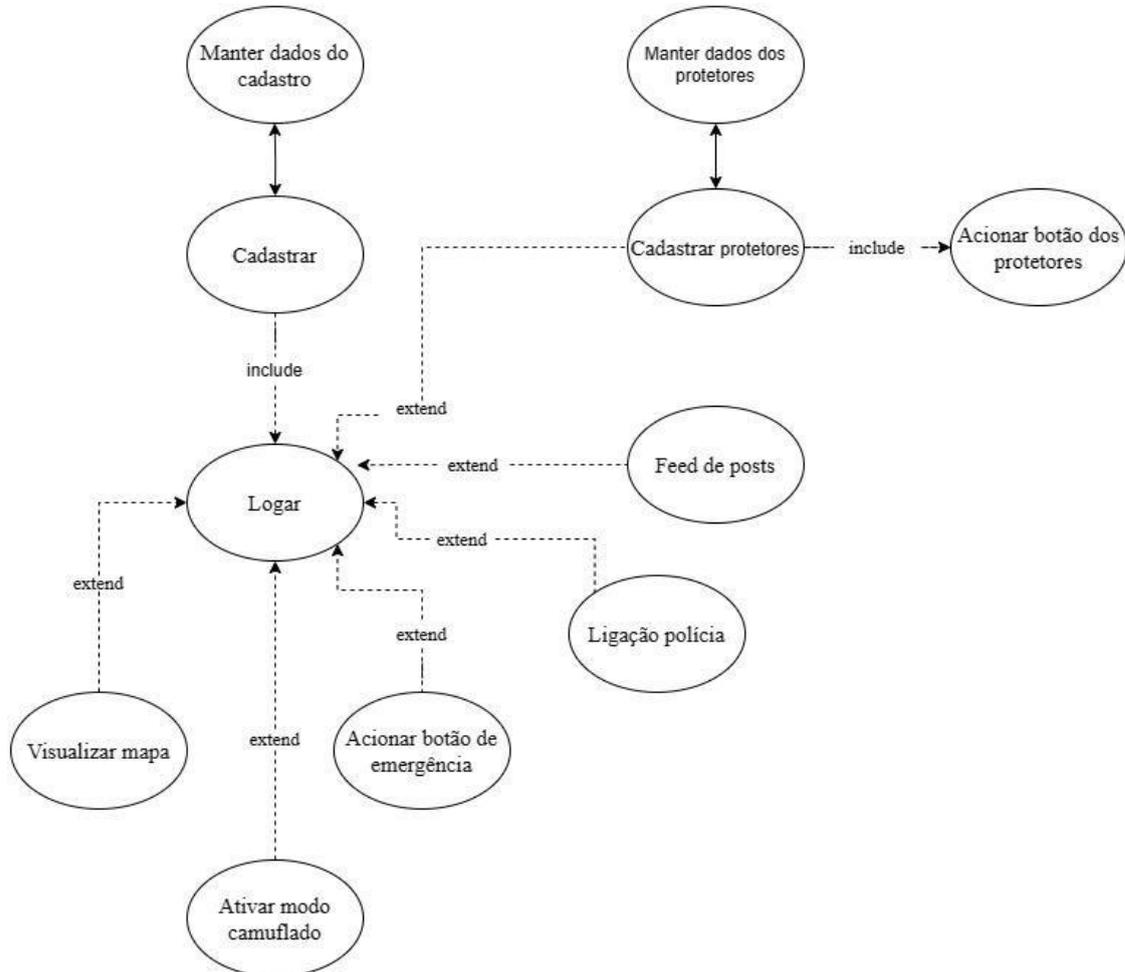
RF01	O sistema irá permitir que o usuário se cadastre.
RF02	O sistema deve permitir que o usuário acione um botão que realiza uma chamada telefônica para a polícia local.
RF03	O sistema irá possuir mapa indicando locais com baixa segurança, além de ongs e delegacia da mulher; localização do tempo real do usuário.
RF04	O sistema irá permitir ao usuário adicionar um ou mais “protetores”, que serão pessoas de confiança que serão notificadas com um pedido de ajuda assim que o botão for acionado.
RF05	O sistema deverá permitir que o usuário faça uma denúncia anônima, além de gravações de áudio, que ficará salvo no aplicativo.
RF06	O sistema irá permitir conversa por chat privativo.
RF07	O sistema terá um modo camuflado, onde mudará a interface do aplicativo para ocultar informações confidenciais.

Fonte: Elaborado pelos autores, 2023

Requisitos não funcionais em programação referem-se a características gerais, como desempenho, segurança e usabilidade, que não estão diretamente relacionadas às funções específicas de um *software*. Eles afetam a qualidade e a experiência do usuário.

2.1.2 DIAGRAMA DE CASO DE USO

Ilustração 10 – Diagrama de caso de uso



Fonte: Autores, 2023

Um diagrama de caso de uso é uma representação visual que descreve as interações entre atores e um sistema em desenvolvimento. Ele ajuda a identificar e comunicar os requisitos funcionais do sistema, mostrando os atores (usuários, sistemas externos), os casos de uso (funcionalidades) e as relações entre eles. É uma ferramenta importante para entender o comportamento esperado do sistema e orientar o desenvolvimento de *software*.

2.1.3 FLUXO DE EVENTOS

Um fluxo de eventos é uma representação detalhada que descreve a sequência de passos que orienta a execução de um caso de uso em um sistema.

Ele proporciona uma visão estruturada das interações entre o usuário e o sistema, apresentando em ordem lógica as ações que são realizadas e as respostas que se espera obter. Essa representação é uma ferramenta fundamental para a compreensão minuciosa e a correta implementação das funcionalidades do sistema, tornando mais claro como o *software* se comportará em resposta às ações do usuário.

A descrição dos fluxos de eventos desse sistema pode ser consultada no Apêndice A.

2.2 BANCO DE DADOS

O modelo entidade-relacionamento é uma ferramenta de modelagem de dados usada para representar a estrutura e os relacionamentos de um banco de dados. Ele ajuda a definir as entidades, seus atributos e como elas se relacionam entre si. É útil para projetar e comunicar a estrutura de um banco de dados de forma clara e concisa.

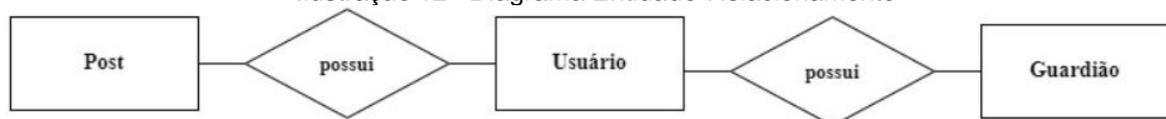
Ilustração 11 – Modelo Entidade Relacionamento



Fonte: Autores, 2023

Representação visual que descreve as relações entre entidades em um sistema, destacando suas interações e atributos.

Ilustração 12 - Diagrama Entidade-Relacionamento



Fonte: Autores, 2023

Diagrama de Entidade-Relacionamento (*DER*) é uma representação gráfica usada para modelar entidades e seus relacionamentos em um banco de dados. Ele

usa símbolos simples para ilustrar como os dados estão organizados e interconectados no sistema. É uma ferramenta essencial para o design de bancos de dados.

2.3 CAMADA DE NEGÓCIO

Ilustração 13 – Código Back-end

```

application.properties  LoginController.java
guardia-rosa  src/main/java  br.com.tcc.guardia.rosa.controller  LoginController  login(@RequestBody @Valid LoginDTO) : ResponseEntity<LoginResponseDTO>
1 package br.com.tcc.guardia.rosa.controller;
2
3 import javax.validation.Valid;
22
23 @RestController
24 @RequestMapping("/api/auth")
25 public class LoginController {
26
27     @Autowired
28     private AuthenticationManager authenticationManager;
29     @Autowired
30     private UsuarioRepository repository;
31     @Autowired
32     private TokenService tokenService;
33
34     @PostMapping("/login")
35     public ResponseEntity<LoginResponseDTO> login(@RequestBody @Valid LoginDTO login) {
36         UsernamePasswordAuthenticationToken usernamePasswordAuthenticationToken = new UsernamePasswordAuthenticationToken(login.getEmail(), login.getPassword());
37         Authentication auth = authenticationManager.authenticate(usernamePasswordAuthenticationToken);
38         String token = tokenService.generateToken(auth);
39         Usuario usuario = (Usuario) auth.getPrincipal();
40
41         return ResponseEntity.ok(new LoginResponseDTO(usuario.getId(), usuario.getNome(), usuario.getEmail(), token, usuario.getCpf()));
42     }
43
44     @PostMapping("/register")
45     public ResponseEntity<?> register(@RequestBody @Valid RegisterDTO register) {
46         if (repository.findByEmail(register.getEmail()) != null) {
47             return ResponseEntity.badRequest().build();
48         }
49
50         String encryptedPassword = new BCryptPasswordEncoder().encode(register.getPassword());
51         Usuario usuario = new Usuario(register.getName(), register.getEmail(), encryptedPassword, register.getCpf());
52         repository.save(usuario);
53
54         return ResponseEntity.ok().build();
55     }
56
57 }

```

Fonte: Autores, 2023

Esse código é um controlador em Java para um aplicativo *web* construído com o *framework Spring Boot*. Ele lida com operações de autenticação e registro de usuários. Aqui está um resumo simplificado do que cada parte do código faz:

Importações: Importa várias classes do *Spring Framework* e outras classes necessárias para o funcionamento do controlador.

Declaração de Classe: Define uma classe chamada *LoginController* que é um controlador *Spring Rest*, responsável por lidar com as solicitações relacionadas à autenticação e registro de usuários.

Injeção de Dependências: Usa anotações `@Autowired` para injetar dependências nas propriedades da classe, como `AuthenticationManager`, `UsuarioRepository` e `TokenService`.

Método `login`: Este método é mapeado para a rota `/api/auth/login` e lida com as solicitações de `login`. Ele recebe um objeto `LoginDTO` como entrada, que contém informações de `login` do usuário. O método autentica o usuário, gera um `token` de autenticação e retorna um objeto `LoginResponseDTO` contendo informações do usuário autenticado e o `token`.

Método `register`: Este método é mapeado para a rota `/api/auth/register` e lida com as solicitações de registro de novos usuários. Ele recebe um objeto `RegisterDTO` como entrada, que contém informações do usuário a ser registrado. O método verifica se o e-mail já está registrado, criptografa a senha do usuário usando `BCrypt`, criando um objeto `Usuario`, salva-o no banco de dados e retorna uma resposta `HTTP` apropriada.

Ilustração 14 – Código Back-end

```

application.properties  LoginController.java  GuardiaoController.java  X
guardia-rosa  src/main/java  br.com.tcc.guardia.rosa.controller  GuardiaoController
1 package br.com.tcc.guardia.rosa.controller;
2
3 import java.net.URI;
26
27 @RestController
28 @RequestMapping("/api/guardian")
29 public class GuardiaoController {
30
31     private final GuardiaoBusiness business;
32
33     @Autowired
34     public GuardiaoController(GuardiaoBusiness business, UsuarioBusiness usuarioBusiness) {
35         this.business = business;
36     }
37
38     @GetMapping("/{id}")
39     public List<GuardiaoDTO> getGuardioesByUsuario(@PathVariable Long id) {
40         List<Guardiao> guardioes = business.getGuardioesByUsuario(id);
41         List<GuardiaoDTO> guardioesDTO = GuardiaoDTO.toGuardioesDTO(guardioes);
42
43         return guardioesDTO;
44     }
45
46     @PostMapping
47     public ResponseEntity<GuardiaoDTO> addGuardiao(@RequestBody @Valid GuardiaoForm guardiaoForm, UriComponentsBuilder uriBuilder) {
48         Guardiao guardiao = business.addGuardiao(guardiaoForm);
49         URI uri = uriBuilder.path("/api/guardiao/{id}").buildAndExpand(guardiao.getId()).toUri();
50
51         return ResponseEntity.created(uri).body(new GuardiaoDTO(guardiao));
52     }
53
54     @PutMapping("/{id}")
55     public ResponseEntity<GuardiaoDTO> updateGuardiao(@RequestBody @Valid UpdateGuardiaoForm guardiaoForm, @PathVariable Long id) {
56         if (business.findById(id) != null) {
57             Guardiao guardiao = business.updateGuardiao(id, guardiaoForm);
58             return ResponseEntity.ok(new GuardiaoDTO(guardiao));
59         }
60         return ResponseEntity.notFound().build();
61     }

```

```

54● @PutMapping("/{id}")
55 public ResponseEntity<GuardiaoDTO> updateGuardiao(@RequestBody @Valid UpdateGuardiaoForm guardiaoForm, @PathVariable Long id) {
56     if (business.findById(id) != null) {
57         Guardiao guardiao = business.updateGuardiao(id, guardiaoForm);
58         return ResponseEntity.ok(new GuardiaoDTO(guardiao));
59     }
60     return ResponseEntity.notFound().build();
61 }
62
63● @DeleteMapping("/{id}")
64 public ResponseEntity<GuardiaoDTO> deleteGuardiao(@PathVariable Long id) {
65     if (business.findById(id) != null) {
66         business.deleteGuardiao(id);
67         return ResponseEntity.ok(new GuardiaoDTO());
68     }
69     return ResponseEntity.notFound().build();
70 }
71
72 }
73

```

Fonte: Autores, 2023

Este código é um exemplo de um controlador de *API* em Java, usando o *framework Spring Boot*. Ele lida com solicitações relacionadas a "guardiões" em uma aplicação web ou móvel. Simplificando o que acontece neste código:

Importações: No início, o código faz várias importações de bibliotecas necessárias para o funcionamento, como bibliotecas do *Spring* e classes personalizadas relacionadas aos guardiões.

Definição de Controlador: A classe é definida como um controlador usando a anotação `@RestController`. Isso significa que ela trata solicitações *HTTP* e retorna respostas.

Construtor: O controlador possui um construtor que recebe duas instâncias de classes de negócios (`GuardiaoBusiness` e `UsuarioBusiness`) como argumentos.

Mapeamento de Endpoints: Os métodos no controlador são anotados com várias anotações, como `@GetMapping`, `@PostMapping`, `@PutMapping`, e `@DeleteMapping`, que mapeiam os *URLs* das solicitações *HTTP* para métodos específicos.

Manipulação de Solicitações: Cada método no controlador executa uma ação específica relacionada aos "guardiões". Por exemplo, o método `getGuardioesByUsuario` recupera uma lista de guardiões com base no *ID* do usuário fornecido na *URL*.

Validação de Dados: Há anotações como `@Valid` que indicam que os dados fornecidos nas solicitações devem ser validados antes de serem processados.

Resposta: Cada método retorna uma resposta *HTTP*, como `ResponseEntity`. Isso pode incluir dados no corpo da resposta, como objetos *DTO* (*Data Transfer Object*) convertidos em *JSON*.

Em resumo, este código define um controlador que lida com operações relacionadas a guardiões em uma aplicação *web* ou móvel, como criar, atualizar, recuperar e excluir guardiões. Ele mapeia *URLs* específicos para métodos Java que executam essas operações e retorna respostas *HTTP* apropriadas. Isso é parte do desenvolvimento de uma *API* para gerenciar guardiões em um sistema maior.

2.4 CAMADA DE APRESENTAÇÃO

Ilustração 15 – Tela de Login



Fonte: Autores, 2023

Ao abrir o aplicativo, os usuários deparam-se com a tela de *login*, onde inserem suas credenciais, geralmente nome de usuário ou e-mail, junto com uma senha. Essa etapa é crucial para garantir a segurança, autenticando os usuários e prevenindo acesso não autorizado. A interface busca equilibrar segurança e facilidade de uso, proporcionando uma experiência intuitiva. Além da autenticação, a tela de *login* pode oferecer recursos como redefinição de senha, garantindo uma interação eficiente e segura. Em resumo, essa tela desempenha um papel fundamental na entrada do usuário no sistema, assegurando uma experiência positiva desde o início.

Ilustração 16 – Tela de Cadastro de Guardião

Nome

Telefone

Email

Cadastrar ✓

Home Localização Alertar Polícia Guardião

Fonte: Autores, 2023

Após o cadastro da usuária, ela poderá cadastrar o contato das pessoas que ela confia ser parte da lista de guardiões, para o cadastro será necessário apenas o nome, telefone e e-mail da pessoa para confirmação.

Ilustração 17 – Tela de Guardiões

Guardiões

Leonardo ★ Alertar ⋮

Pedro Gregorio Alertar ⋮

Home Localização Alertar Polícia Guardião

Fonte: Autores, 2023

Após o cadastro dos guardiões, a usuária poderá ver e adicionar mais guardiões o momento que ela achar necessário, podendo alertá-los como um pedido de ajuda ou excluindo caso não queira mais ter a pessoa na lista e podendo escolher seu guardião “favorito”, que será sempre o primeiro a ser chamado.

Mais informações sobre as telas encontram-se na Apêndice B

3 RESULTADO

Foram realizados os testes de usabilidade durante as primeiras duas semanas de junho, convidando cuidadosamente cinco pessoas para participar. Esses testes foram conduzidos em um ambiente local, na máquina em que a aplicação foi desenvolvida.

As tarefas designadas foram as seguintes:

1. Efetuar o cadastro no sistema.
2. Criar uma publicação.
3. Escolher uma postagem e expressar aprovação, curtindo-a.
4. Alterar a senha da conta.
5. Adicionar um contato de confiança.
6. Simular uma ligação para as autoridades policiais.
7. Encerrar a sessão na conta.

3.1 PERFIL DOS PARTICIPANTES DO TESTE E SUAS OBSERVAÇÕES

Neste tópico será apresentado o perfil das participantes e observações sobre a participação delas.

1ª Interação:

A primeira interação foi com Andrea, uma mulher de 50 anos, cuja interação durou por 4 minutos e 26 segundos. Mesmo sendo uma pessoa de idade mais avançada, ela não enfrentou grandes dificuldades para localizar informações, graças à sua interação diária com as redes sociais. Sua principal dificuldade residia na compreensão do significado dos ícones.

2ª Interação:

A segunda interação foi realizada por Giovana, uma jovem de 21 anos, e durou 3 minutos e 41 segundos. Ela demonstrou ser a usuária mais proficientemente habilidosa com o aplicativo entre todas as participantes do teste, sugerindo que sua destreza pode derivar do uso diário das redes sociais.

3ª Interação:

A terceira interação foi conduzida por Ana, 57 anos, e a duração da interação foi de 6 minutos e 48 segundos, tornando-se a mais demorada entre todos os testes realizados. Sendo uma pessoa com mais idade e não familiarizada com as redes sociais, Ana enfrentou consideráveis desafios na interpretação dos ícones e nas interações devido à sua falta de costume com tais tecnologias.

4ª Interação:

A quarta interação foi conduzida por Leticia, uma jovem de 19 anos, com uma duração de 2 minutos e 11 segundos. Sua principal dificuldade residia na compreensão da distinção entre a seção 'Meu Perfil' e 'Configurações', especialmente quando se tratava de alterar a senha, além de algumas outras interações de natureza mais sutil.

5ª Interação:

A quinta interação foi realizada por Luisa, uma jovem de 22 anos, e durou 3 minutos e 5 segundos. Luisa enfrentou consideráveis desafios durante o processo de cadastro, bem como na compreensão dos ícones, como o de adicionar um guardião. Além disso, assim como Leticia, ela compartilhou a mesma dúvida, encontrando dificuldades para localizar o botão de alteração de senha.

3.2 RESULTADOS DOS QUESTIONÁRIOS

Aqui serão apresentados os resultados dos questionários aplicados aos participantes dos testes. Essa etapa é essencial para avaliar o desempenho do projeto, abrangendo aspectos como usabilidade e *feedback* sobre funcionalidades. As respostas dos participantes oferecem insights valiosos, influenciando decisões futuras e contribuindo para aprimoramentos contínuos. A qualidade das perguntas e a interpretação dos resultados são fundamentais para uma avaliação precisa. Esta seção proporciona uma visão detalhada das percepções dos usuários, orientando o desenvolvimento do projeto com base em dados concretos.

Ilustração 18 - Gráfico das respostas da pergunta 1

Achou o sistema intuitivo?

5 respostas

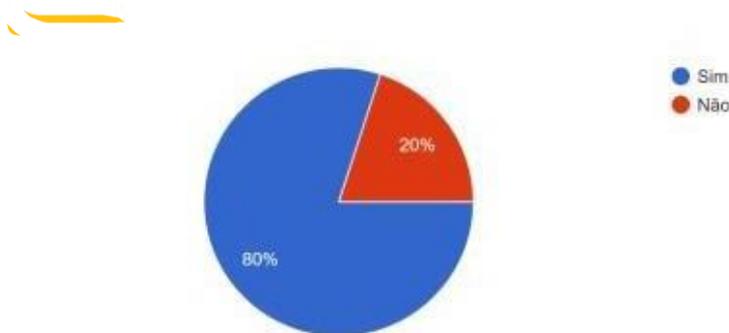


Fonte: Autores, 2023

Na figura 18 é apresentado o gráfico com o resultado da pergunta 1 respondida pelos participantes. A maioria das participantes concordaram que é um sistema intuitivo.

Ilustração 19 – Gráfico de respostas da pergunta 2

Você acha que um aplicativo que aciona diretamente a polícia te ajudaria nessa situação?



Fonte: Autores, 2023

Na Figura 19, o gráfico revela que 80% das participantes consideram que um aplicativo capaz de acionar a polícia de forma mais rápida e fácil seria útil em situações críticas. Essa percepção destaca a importância atribuída à agilidade e facilidade de acionar as autoridades em momentos de urgência. A interpretação desses resultados sugere uma demanda clara por soluções tecnológicas que possam oferecer assistência imediata em casos de emergência. Essa constatação pode influenciar estratégias futuras para o desenvolvimento ou aprimoramento de aplicativos voltados para a segurança.

Ilustração 20 – Respostas da pergunta 3

Os rótulos e ícones utilizados no aplicativo são claros e compreensíveis? Se não, quais melhorias você sugere?

5 respostas

Sim, mas acho que se o perfil fosse no canto direito inferior seria mais intuitivo ainda
Sim, achei bem intuitivo. O que eu achei menos intuitivo foi para adicionar o guardião, mas fuçando um pouquinho o APP, eu acredito que encontraria essa função.
Pra quem não conhece nada fica difícil
trocar a imagem visual do "adicionar guardião " para melhor identificação
Facilitar a parte de achar o guardião

Fonte: Autores, 2023

Na ilustração 20 foram falados melhorias que poderiam ter dentro do aplicativo para melhor usabilidade do mesmo.

3.3 CONCLUSÃO

Após a conclusão de todos os testes realizados, identificou-se que a alteração de senha e os ícones de interação foram os aspectos que apresentaram maior dificuldade de compreensão por parte dos usuários. Para abordar essa questão, serão tomadas medidas para melhorar a experiência do usuário.

Primeiramente, está planejada uma alteração no botão de cadastro, tornando-o mais visível e intuitivo. Observou-se que muitos participantes optaram pelo login direto em vez de realizar o cadastro durante os testes. Ao tornar o botão de cadastro mais proeminente, a expectativa é direcionar os usuários de forma mais clara e evitar essa confusão.

Além disso, estão sendo exploradas alternativas para tornar o recurso de adicionar um guardião mais intuitivo. Considera-se a substituição do ícone atual por um botão de fácil identificação, simplificando o processo de adicionar contatos de confiança. Dessa forma, pretende-se facilitar o entendimento e a utilização dessa funcionalidade.

Por fim, reconhece-se a necessidade de realizar um estudo mais aprofundado em relação à alteração de senha. Para garantir uma abordagem adequada e eficiente, será dedicado tempo e esforço para examinar cuidadosamente as opções disponíveis. A decisão a ser tomada será baseada em uma análise criteriosa, visando garantir a melhor experiência para os usuários.

Com essas medidas em mente, a equipe está comprometida em aprimorar a usabilidade e a acessibilidade da aplicação, oferecendo aos usuários uma experiência mais intuitiva e satisfatória.

REFERÊNCIAS

CRIANDO conexões contra a violência: Informar, criar redes de proteção e agir -. Informar, criar redes de proteção e agir -. 2023. Coordenação Thais Folego. Disponível em: <https://azmina.com.br/projetos/penhas/>. Acesso em: 21 jun. 2023.

LIMA, Cleyson. **O que é Spring Boot?**: neste artigo veremos o que é o spring boot e como ele pode nos auxiliar no desenvolvimento de aplicações spring.. Neste artigo veremos o que é o Spring Boot e como ele pode nos auxiliar no desenvolvimento de aplicações Spring.. 2021. Disponível em: <https://www.treinaweb.com.br/blog/o-que-e-o-spring-boot>. Acesso em: 14 abr. 2021.

GOMES, Luiz Flávio. Femicídio: O Que Não Tem Nome Nem Identidade Não Existe. [s.l.], jan. - mar. 2016.

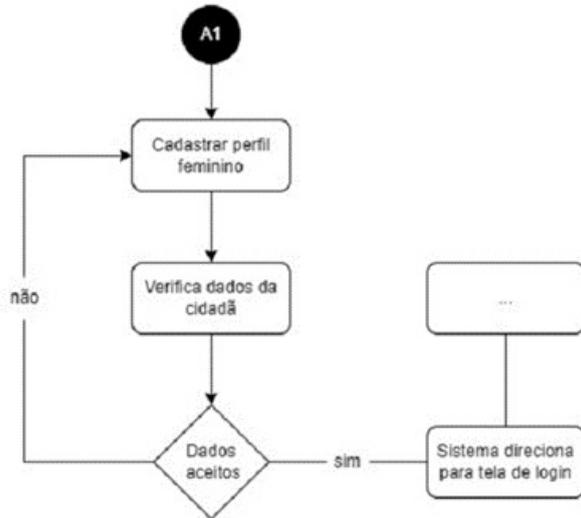
Disponível em:

https://bdjur.stj.jus.br/jspui/bitstream/2011/100618/feminicidio_nome_identidade_gomes.pdf

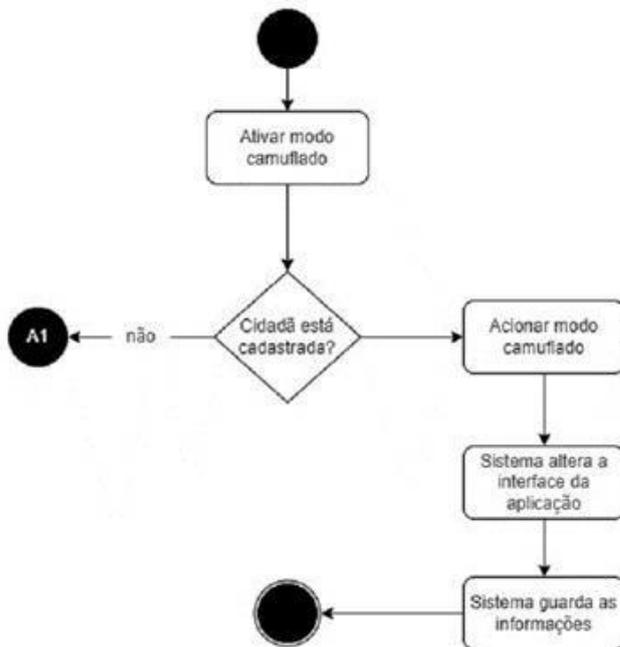
Acesso em: 20 de maio de 2023.

APENDICE A

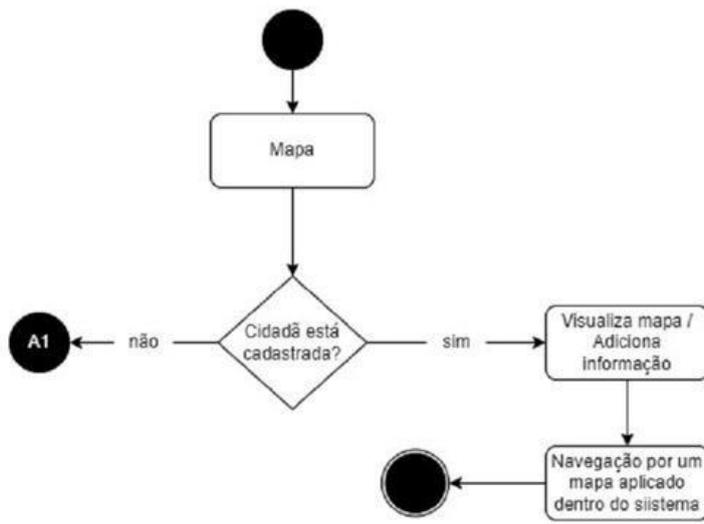
Cadastro



Modo camuflado



Mapa



APENDICE B

Feed de notícia (modo noturno)



Informações do Perfil



Menu



Sobre



Cadastrar

←

Cadastre-se

Nome

CPF

E-mail

Senha

Confirmar senha

Cadastrar