Texto

Descrição gerada automaticamente

**Documentação de um**

**Produto de Software**

**TCC – Trabalho de Conclusão de Curso**

**Felipe Ribeiro de Souza**

**João Lucas Miclos de Negreiros**

**Lettícia de Sousa Tonon**

**Lucas Kaue Guimarães Mota**

**2024**

ÍNDICE DETALHADO

3. Requisitos do Sistema de Software 3

3.1. Requisitos Funcionais 4

3.2. Modelagem Funcional 4

3.3. Requisitos Não-Funcionais 6

3.4. Protótipo – FAZER POSTERIORMENTE 8

4. Análise 9

4.1. Diagrama de Classes de Análise (Visão de Negócio) 9

5. Projeto 10

5.1. Arquitetura do Sistema 10

5.2. Diagrama de Classes de Projeto por Caso de Uso 10

5.3. Diagrama de atividades 11

5.4. Diagrama de estados 12

6. Testes 13

6.1. Plano de Testes 13

6.2. Roteiro de Testes 13

1. Requisitos do Sistema de Software

Este capítulo visa especificar os requisitos funcionais e não funcionais, juntamente com as regras de negócio, além de fornecer o protótipo das telas e o cronograma de atividades para o desenvolvimento de software.

1. Requisitos Funcionais

Neste item serão descritos os requisitos funcionais que especificam ações que o sistema deve ser capaz de executar, ou seja, os objetivos do sistema, incluindo prioridade e regras de negócio.

**[RF001] – Cadastrar usuário**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade**: | n | Essencial | ¨ | Importante | ¨ | Desejável |

**Descrição**: Este requisito permite que o usuário crie uma conta para a utilização do sistema.

**[RF002] – Realizar login**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade**: | n | Essencial | ¨ | Importante | ¨ | Desejável |

**Descrição**: Este requisito permite que o usuário faça o login em sua conta.

**[RF003] – Abrir uma denúncia**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade**: | n | Essencial | ¨ | Importante | ¨ | Desejável |

**Descrição**: Este requisito permite que o usuário abra uma denúncia pública.

**[RF004] – Visualizar feed de denúncias**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade**: | n | Essencial | ¨ | Importante | ¨ | Desejável |

**Descrição**: Este requisito permite que o usuário veja as informações principais das denúncias em um feed.

**[RF005] – Visualizar detalhes de uma denúncia**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade**: | n | Essencial | ¨ | Importante | ¨ | Desejável |

**Descrição**: Este requisito permite que o usuário visualize todos os dados de uma denúncia, incluindo suas curtidas e comentários.

**[RF006] – Curtir uma denúncia**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade**: | n | Essencial | ¨ | Importante | ¨ | Desejável |

**Descrição**: Este requisito permite que o usuário adicione ou remova uma curtida em qualquer denúncia.

**[RF007] – Comentar uma denúncia**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade**: | n | Essencial | ¨ | Importante | ¨ | Desejável |

**Descrição**: Este requisito permite que o usuário adicione ou remova um comentário em qualquer denúncia.

**[RF008] – Alterar status de uma denúncia**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade**: | n | Essencial | ¨ | Importante | ¨ | Desejável |

**Descrição**: Este requisito permite que um usuário altere o status de uma denúncia.

**[RF009] – Alterar informações de uma denúncia**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade**: | n | Essencial | ¨ | Importante | ¨ | Desejável |

**Descrição**: Este requisito permite que o usuário altere os dados de uma denúncia feita anteriormente.

**[RF010] – Deletar uma denúncia**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade**: | n | Essencial | ¨ | Importante | ¨ | Desejável |

**Descrição**: Este requisito permite que o usuário delete uma denúncia feita anteriormente.

**[RF011] – Ver relatório de denúncias**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade**: | ¨ | Essencial | ¨ | Importante | n | Desejável |

**Descrição**: Este requisito permite que o usuário visualize um relatório de denúncias, contendo informações sobre quantidade de denúncias separadas por tipo e região.

**Regras de Negócio**

**[RN001] – Ações do usuário**

**Descrição:** O usuário pode visualizar o feed e os detalhes de denúncias sem estar logado, mas para realizar os demais casos de uso, o usuário deve estar logado.

**[RN002] – Cadastro**

**Descrição:** Para que o usuário realize o cadastro ele deve ter preenchido todos os seguintes campos corretamente: Nome completo; E-mail; Senha; Telefone; Sexo; Endereço; CEP; e Data de Nascimento.

**[RN003] – Login**

**Descrição:** Para que o usuário realize o login ele deve estar cadastrado e preencher os campos de e-mail e senha.

**[RN004] – Criar denúncia**

**Descrição:** Para que o usuário crie uma denúncia ele deve preencher os seguintes campos: Foto; Descrição; Categoria; e localização conforme dados obtidos pelo Google Maps dentro do próprio sistema. Após os campos serem preenchidos o sistema deve verificar se já existe uma denúncia com o mesmo tipo na mesma região (área de 2500m²), caso positivo, será exibido ao usuário uma lista com as denúncias encontras dando a opção de curti-las caso seu problema esteja dentro da lista, ou seguir com a criação de uma nova denúncia.

**[RN005] – Deletar uma denúncia**

**Descrição:** O usuário pode deletar uma denúncia feita por ele mesmo, já o moderador pode excluir qualquer denúncia dentro do sistema.

**[RN006] – Modificar status de uma denúncia**

**Descrição:** O status da denúncia pode ser alterado tanto pelo criador dela quanto por um moderador do sistema, podendo ser “Resolvida” ou “Não Resolvida”.

**[RN007] – Visualização de relatório de denúncias**

**Descrição:** O relatório pode ser gerado utilizando filtros por região ou por tipo de denúncia.

**[RN008] – Alterar denúncia**

**Descrição:** O usuário pode alterar informações de uma denúncia feita por ele mesmo, já o moderador pode alterar informações de qualquer denúncia dentro do sistema.

**[RN009] – Visualizar feed de denúncias**

**Descrição:** O feed deve ser organizado pelas denúncias mais novas ou mais curtidas.

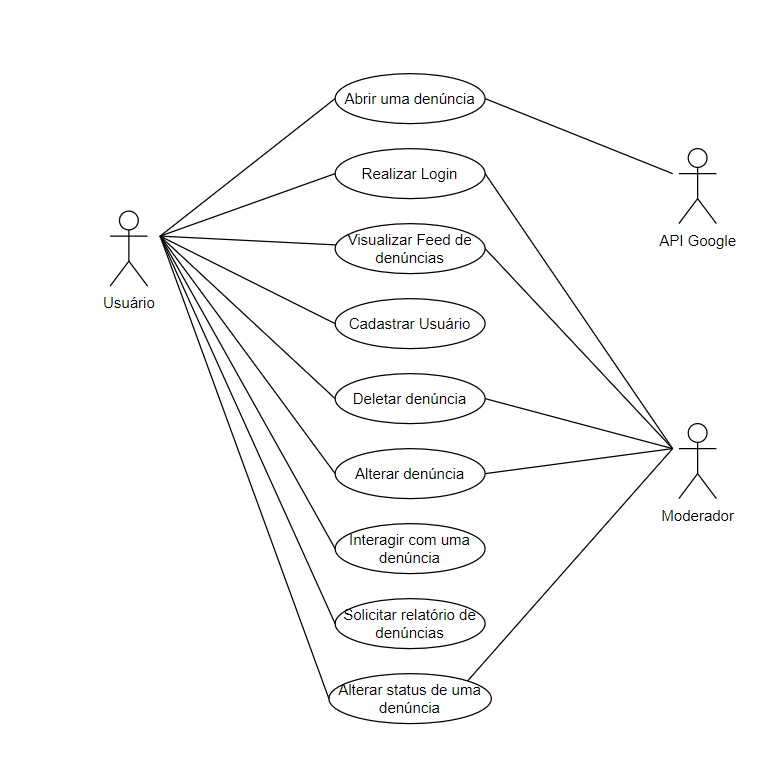
**[RN0010] – Criação de usuário moderador**

**Descrição:** A criação do usuário moderador será feita pelo administrador do sistema direto no banco de dados, visto que o sistema não necessita de interface gráfica para essa ação.

1. Modelagem Funcional

Neste item será descrito os requisitos a serem atendidos funcionalmente pelo sistema, contendo atores e o diagrama de caso de uso.

**Diagrama de caso de uso**



**Atores**

Detalhamento dos atores que aparecem no diagrama do caso de uso.

|  |  |
| --- | --- |
| **Nome** | **Descrição** |
| Moderador | Usuário do sistema responsável por visualizar, modificar, alterar o status de uma denúncia e deletar denúncias de outros usuários. |
| API Maps | Sistema externo responsável por obter dados de localização |
| Usuário | Usuário do sistema responsável por abrir, visualizar, modificar, alterar o status e deletar denúncias. Além de poder interagir com denúncias de outros usuários e visualizar relatórios de denúncias. |

**Especificação do caso de uso**

A seguir é apresentado um exemplo da especificação de casos de uso.

|  |  |
| --- | --- |
| **CSU001 – Pesquisar Livro** | |
| Sumário: | Este caso de uso tem como objetivo pesquisar livros vendidos pela Web. |
| Pré-condição:O livro deve estar cadastrado. | |
| **Fluxo Principal** Este caso de uso se inicia quando o ator cliente escolhe a opção de “Pesquisar Livros”.O ator cliente informa nome do livro. [RN1]O nome do livro é enviado para o Sistema de Estoque que retorna os dados do livro: Código, Nome do livro, Preço e Disponibilidade/Indisponibilidade. [FE1]O sistema exibe os dados do livro que são: Código, Nome do Livro, Preço e Disponibilidade/Indisponibilidade. | |
| Fluxos AlternativosNão se aplica. | |
| Fluxos de Exceção[FE1] Fluxo Exceção 1: Livro não encontradoEste fluxo alternativo ocorre quando o livro não é encontrado no Sistema de Estoque.O sistema exibe uma mensagem que o livro pesquisado não foi encontrado.O fluxo retorna ao passo 1 do fluxo básico. | |
| Pós-condições:Não se aplica. | |
| Regras de Negócio:RN1 | |

1. Requisitos Não-Funcionais

**[RNF001] - Portabilidade**

**Descrição:** O sistema deve funcionar tanto em computadores, tablets e celulares de maneira responsiva.

**[RNF002] - Conformidade**

**Descrição:** O sistema deve seguir a LGPD.

**[RNF003] - Segurança**

**Descrição:** O sistema deve estar protegido de acessos de usuários não permitidos, por meio de login, senha criptografada e controle de sessões.

**[RNF004] - Usabilidade**

**Descrição:** A interface do sistema deve ser simples e amigável para o usuário, permitindo que qualquer ação dentro dele seja feita em uma média de 10 minutos.

**[RNF005] - Arquitetura**

**Descrição:** A arquitetura utilizada seguira o modelo MVC junto com o padrão DAO.

**[RNF006] - Tecnologia**

**Descrição:** A tecnologia utilizada para o back-end será Python com o micro framework flask; para o front-end será utilizado a biblioteca REACT; e para o banco de banco de dados, MySQL. As senhas serão criptografadas utilizando o método AES256 e o controle de sessão via tokens JWT.

1. Protótipo – FAZER POSTERIORMENTE

Neste item deve ser apresentado o protótipo do sistema que consiste na interface preliminar contendo um subconjunto de funcionalidades e telas. O protótipo deve ser incrementalmente evoluído até a concordância completa dos requisitos previstos para o sistema, de comum acordo com o usuário. O protótipo é um recurso que deve ser adotado como estratégia para levantamento, detalhamento, validação de requisitos e modelagem de interface com o usuário (usabilidade).

Sugere-se que cada tela possua uma descrição detalhada do seu funcionamento. Alguns itens importantes na descrição são:

* Objetivo da tela;
* De onde é chamada e que outras telas podem chamar;
* Regras:
* Domínio (tamanho de campo, tipo de dados que aceita valor default);
* Tipo de usuários que podem acessar;
* Lógica de negócio (campos obrigatórios, validade entre datas, preenchimento anterior de um campo para efetuar uma operação, etc).

A descrição detalhada das telas deve registrar informações que possam ser consultadas na implementação do sistema, facilitando, agilizando e minimizando erros de implementação e na execução de testes.

**O protótipo de telas deve estar associado ao caso de uso correspondente, podendo ser apresentado conjuntamente com o caso de uso correspondente. A seguir é apresentado um exemplo de protótipo de tela.**

1. Análise

Este capítulo tem como objetivo analisar, detalhar e propor uma solução geral do sistema, sob o ponto de vista de negócio, de acordo com os requisitos levantados e validados no capítulo 3.

* 1. Diagrama de Classes de Análise (Visão de Negócio)

Neste item deve ser apresentado o modelo do domínio, visão de negócio, que representa um primeiro modelo conceitual do diagrama de classes. Posteriormente, esse diagrama deve ser validado, refinado e complementado para compor o diagrama de classes de projeto.

O diagrama de classes deve possuir todas as classes identificadas do sistema, deve conter os atributos e métodos de cada classe, e os relacionamentos entre elas.

A seguir é apresentada a notação básica de um diagrama de classes



Notação básica do diagrama de classes.

1. Projeto

Este capítulo tem como objetivo refinar a proposta de solução geral do sistema, apresentando a solução técnica, incluindo a visão de projeto e implementação, a arquitetura e a tecnologia utilizada.

* 1. Arquitetura do Sistema

Neste item deve ser apresentada a arquitetura de infraestrutura do sistema, demonstrando o tipo de arquitetura que será utilizada (por exemplo, cliente/servidor de n-camadas, MVC, ...), a configuração de hardware, de rede, de software, padrões de projeto, componentes específicos (dll, jar, ...) e componentes externos a serem utilizados, bem como o dimensionamento mínimo de conexões.

Para a representação da arquitetura de infraestrutura pode-se utilizar uma figura ilustrativa ou o diagrama de implantação. A seguir é apresentada a notação básica de um diagrama de implantação.

Notação básica do diagrama de implantação.

* 1. Diagrama de Classes de Projeto por Caso de Uso

Este item tem como objetivo apresentar a realização de caso de uso para cada caso de uso.

A realização de caso de uso deve conter:

* + diagrama de classes para cada caso de uso. Notação básica do diagrama de classes já foi apresentada no item 4.1.
  + diagrama de sequência (para cada cenário do fluxo principal/básico e fluxos alternativos). Neste item devem ser apresentados os diagramas de sequência essenciais ao sistema, identificados através dos casos de uso.

Um diagrama de sequência representa interações de objetos organizadas em uma sequência temporal, apresentando os objetos que participam da interação e a sequência das mensagens trocadas. O diagrama de sequência deve validar o diagrama de classes e vice-versa.

**Obs: Para os casos de uso que representam <<crud>> básico não há necessidade de realização do caso de uso.**

* 1. Diagrama de atividades

O diagrama de atividades representa o detalhamento de tarefas e o fluxo de uma atividade para outra de um sistema, geralmente utilizado para os métodos que contém regras de negócio. A seguir é apresentada a notação básica de um diagrama de atividades. **Esse diagrama somente deverá ser elaborado se houver necessidade e agregar valor ao projeto.**



Notação básica do diagrama de atividades.

* 1. Diagrama de estados

O diagrama de estados especifica as sequências de estados pelas quais o objeto pode passar durante seu ciclo de vida em resposta a eventos. A seguir é apresentada a notação básica de um diagrama de estados. **Esse diagrama somente deverá ser elaborado se houver necessidade e agregar valor ao projeto.**



Notação básica do diagrama de estados

6. Testes

Este capítulo tem como objetivo identificar erros no sistema, validar as funções do sistema, verificar se os requisitos foram implementados de forma adequada. Sugere-se a criação de um plano de testes e um roteiro de testes baseando nos casos de uso. Também pode-se utilizar a técnica de TDD (*Test Driven Development*), neste caso os testes também devem ser registrados.

6.1. Plano de Testes

Neste item deve ser criado o plano de testes do sistema, permitindo a validação do sistema por parte do desenvolvedor, através da verificação dos requisitos do sistema desenvolvido.

6.2. Roteiro de Testes

Neste item devem ser registrados os testes realizados no sistema tendo como base o Plano de Testes do Sistema. O roteiro de testes deve ser elaborado com base nos casos de uso.