**Capa**

**Histórico de Alterações**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
|  | 0.0 | Criação do documento de requisitos, identificação dos requisitos | Rafael de Castro Gomes |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Sumário**

[**1. INTRODUÇÃO** 3](#_Toc166087370)

[**1.1.** **Visão geral do documento** 4](#_Toc166087371)

[**1.2.** **Convenções, termos e abreviações** 4](#_Toc166087372)

[**2. DESCRIÇÃO GERAL DO SISTEMA** 4](#_Toc166087373)

[**3. REQUISITOS FUNCIONAIS (CASOS DE USO)** 4](#_Toc166087374)

[**4. REQUISITOS NÃO FUNCIONAIS** 6](#_Toc166087375)

[**5. ESPECIFICAÇÃO DE REQUISITOS DO SISTEMA** 7](#_Toc166087376)

[**5.1. Tabela Casos de Uso** 7](#_Toc166087377)

[**5.1.1 Usuário não Cadastrado** 7](#_Toc166087378)

[**6. MODELOS DO SISTEMA** 16](#_Toc166087379)

[**7. EVOLUÇÃO DO SISTEMA** 16](#_Toc166087380)

[**8. GLOSSÁRIO** 16](#_Toc166087381)

[**9. FERRAMENTAS UTILIZADAS** 16](#_Toc166087382)

[**10. BOAS PRÁTICAS** 16](#_Toc166087383)

# **1. INTRODUÇÃO**

Este documento detalha os requisitos do protótipo de uma rede social que funcionará em conjunto com uma inteligência artificial treinada em detecção de postagens tóxicas. O documento fornece as informações necessárias para o projeto e implementação do sistema, bem como para a realização dos testes e homologação dele.

## **Visão geral do documento**

Este documento está organizado da seguinte forma:

* **Seção 2 - Descrição Geral do Sistema:** Apresenta uma visão geral da rede social.
* **Seção 3 - Requisitos Funcionais (Casos de Uso):** Detalha brevemente os casos de uso do sistema.
* **Seção 4 - Requisitos Não-Funcionais:** Cita e explica os requisitos não funcionais do sistema.
* **Seção 5 - Arquitetura do Sistema:** Descreve de forma geral a arquitetura planejada para o sistema.
* **Seção 6 - Especificação de Requisitos do Sistema:** Apresenta uma descrição detalhada dos requisitos funcionais e não-funcionais.
* **Seção 7 - Modelos do Sistema:** Estabelece modelos que mostram as relações entre os componentes do sistema e seu ambiente.
* **Seção 8 - Evolução do Sistema:** Descreve as mudanças e melhorias previstas para o sistema.
* **Seção 9 - Glossário:** Define termos técnicos e abreviações usadas no documento.

## **Convenções, termos e abreviações**

A correta interpretação deste documento exige o conhecimento de algumas

convenções e termos específicos, que são descritos a seguir.

* **RF:** Requisito Funcional
* **RNF:** Requisito Não Funcional

# **2. DESCRIÇÃO GERAL DO SISTEMA**

O sistema é um protótipo de rede social que permitirá aos usuários registrados realizarem postagens, acessarem e gerenciarem seu próprio conteúdo. Além disso, usuários administradores terão a capacidade de monitorar as atividades e postagens dentro da rede. O protótipo também servirá como uma base para o funcionamento de uma inteligência artificial focada na detecção de postagens tóxicas ou ofensivas.

# **3. REQUISITOS FUNCIONAIS (CASOS DE USO)**

**RF001:** **Permitir que novos usuários se cadastrem na plataforma.**

* **Registrar:** O sistema deve oferecer uma funcionalidade de cadastro que permita aos novos usuários criarem suas contas, fornecendo informações básicas e passando por processos de verificação de identidade quando necessário.

**RF002: Permitir que usuários registrados criem e publiquem postagens.**

* **Postar:** Usuários devem ter a capacidade de criar conteúdo em formato de texto e publicá-los em seus perfis (que são públicos).

**RF003: Possibilitar que usuários registrados editem e atualizem seus perfis.**

* **Editar Perfil:** Os usuários devem poder alterar informações em seus perfis, como foto e nome, a fim de manter sua identidade digital atualizada e relevante.

**RF004: Permitir que usuários registrados visualizem seus registros de atividades na plataforma.**

* **Visualizar Log de Acesso:** Usuários podem acessar um histórico de suas atividades e interações dentro da plataforma, incluindo datas e horários de acesso, para monitoramento pessoal de segurança e uso.

**RF005: Permitir que administradores gerenciem contas de usuários.**

* **Gerenciar Usuários:** Administradores devem ter ferramentas para gerenciar contas de usuários, incluindo a habilidade de ativar, desativar, ou alterar os níveis de acesso dos usuários conforme necessário.

**RF006: Permitir que administradores modifiquem e gerenciem o conteúdo publicado na rede.**

* **Gerenciar Conteúdo:** Deve existir uma funcionalidade que permita aos administradores remover qualquer conteúdo publicado na rede social para garantir a conformidade com as diretrizes da plataforma.

**RF007: Habilitar o monitoramento de registros de acesso por administradores.**

* **Monitorar Log de Acesso:** Administradores devem poder acessar e analisar registros de acesso detalhados dos usuários para monitorar atividades suspeitas e manter a segurança da plataforma.

**RF008: Permitir que administradores analisem padrões de comportamento dos usuários com base em seus dados de acesso.**

* **Analisar Comportamento de Acesso:** Ferramentas analíticas devem estar disponíveis para administradores, permitindo-lhes compreender melhor as tendências de uso e comportamentos dos usuários para melhorar a gestão da rede social.

**RF009: Permitir que usuários registrados sigam outros usuários.**

* **Seguir Usuário:** O sistema deve oferecer a funcionalidade para que usuários possam seguir outros perfis na rede social. Ao seguir um usuário, os posts deste usuário começam a aparecer no feed do seguidor, aumentando a interação e o engajamento na plataforma.

**RF010: Permitir que usuários visualizem os perfis de outros usuários.**

* **Visualizar Perfil:** Os usuários devem ter a capacidade de acessar e visualizar os perfis de outros usuários na rede social. Isso inclui ver informações públicas como nome de usuário, biografia, número de seguidores e postagens, promovendo maior conectividade e interação social.

# **4. REQUISITOS NÃO FUNCIONAIS**

**RNF01:** **O sistema deve garantir a segurança das informações dos usuários.**

* **Segurança:** Para proteger dados sensíveis e garantir a privacidade, o sistema deve implementar medidas de segurança robustas, incluindo criptografia de dados, autenticação forte e políticas de controle de acesso.

**RNF02:** **O sistema deve ser escalável para acomodar um aumento no número de usuários.**

* **Escalabilidade:** À medida que a base de usuários cresce, o sistema deve manter sua performance e estabilidade. Isso envolve a utilização de arquiteturas escaláveis, balanceamento de carga e recursos de computação flexíveis.

**RNF03:** **A interface do usuário deve ser intuitiva e fácil de navegar.**

* **Intuitividade:** O design da interface deve ser claro e amigável, permitindo que os usuários de todos os níveis técnicos interajam com a plataforma sem dificuldades. Isso inclui uma navegação lógica, design responsivo e assistência ao usuário.

**RNF04:** **O sistema deve processar as requisições dos usuários em tempo aceitável.**

* **Desempenho:** Para proporcionar uma experiência de usuário fluida e responsiva, o sistema deve ser capaz de processar todas as requisições dentro de um tempo definido como aceitável, garantindo tempos de resposta rápidos mesmo sob carga pesada.

**RNF05:** **O sistema deve ser altamente disponível para garantir acesso contínuo aos usuários.**

* **Disponibilidade:** O sistema deve garantir uma operacionalidade contínua, com um tempo de disponibilidade de, pelo menos, 99%, reduzindo interrupções e maximizando o tempo de acesso para os usuários.

**RNF06:** **O sistema deve ser compatível com uma variedade de dispositivos e navegadores.**

* **Compatibilidade:** Para atender a uma base de usuários diversificada, o sistema deve funcionar de maneira eficaz em diferentes plataformas, incluindo computadores, smartphones e tablets, e ser compatível com os principais navegadores do mercado, como Chrome, Firefox, Safari e Edge.

**RNF07:** **O sistema deve ser fácil de manter e atualizar.**

* **Manutenibilidade:** Deve-se projetar o sistema de maneira que permita fácil manutenção e atualização. Isso inclui a utilização de código modular, documentação abrangente e padrões de projeto que facilitam futuras atualizações, correções de bugs e adições de novas funcionalidades sem interrupções significativas.

# **5. ESPECIFICAÇÃO DE REQUISITOS DO SISTEMA**

## **5.1. Tabela Casos de Uso**

|  |  |
| --- | --- |
| **Caso de uso 001 – Registrar** | |
| Ator principal | Usuário |
| Resumo | Esse caso de uso, descreve a etapa em que o usuário faz seu cadastro em nosso banco de dados. |
| Pré-condições | O usuário possui um endereço de e-mail válido que ainda não está registrado no sistema; ter idade mínima de dezoito anos. |
|  |  |
| Pós-condições | O sistema confirma o sucesso do registro ao usuário, e ele pode fazer login no sistema usando os detalhes fornecidos. |
| **Cenário principal** | |
| **Ações do ator** | **Ações do sistema** |
| 1. O usuário clica no link descrito como Registre-se “Aqui”. | 1. Redireciona o usuário para a página de cadastro. |
| 1. O usuário pode preencher os campos solicitados e ou obrigatórios, como nome, e-mail, senha. Clica no link descrito “Cadastrar”. | 1. Um número de registro é criado para este usuário. O registro é único e exclusivo. |
|  | |
| **Restrições** | |
|  | **Ações do sistema** |
|  | Não podem existir dois ou mais registros iguais de usuários. |
| **Cenário alternativo** | |
| **Ações do ator** | **Ações do sistema** |
| 1. O usuário entra com informações que já existem no sistema (como um e-mail já registrado). | 1. O sistema verifica a existência de duplicatas e exibe uma mensagem de erro indicando que o e-mail já foi registrado. |
| 1. O usuário pode corrigir as informações e tentar registrar novamente. |  |
| **Cenário de exceção** | |
| **Ações do ator** | **Ações do sistema** |
| 1. O usuário tenta se registrar com informações inválidas (por exemplo, um formato de e-mail inválido). | 1. O sistema valida a entrada e exibe uma mensagem de erro solicitando que o usuário insira informações válidas. |
| **Regras de negócio** | |
| 1. O e-mail do usuário deve ser único. | |
| 1. O nome do usuário pode ser duplicado. | |
| 1. A senha do usuário deve atender aos requisitos mínimos de segurança (por exemplo, pelo menos 8 caracteres, incluindo números, letras maiúsculas e minúsculas, e símbolos especiais). | |

|  |  |
| --- | --- |
| **Caso de uso 002 – Postar** | |
| Ator principal | Usuário |
|  |  |
| Resumo | Este caso de uso descreve o processo que um usuário registrado segue para criar e publicar um post em sua linha do tempo. |
|  |  |
| Pré-condições | O usuário deve estar autenticado e ter uma sessão ativa. |
|  |  |
| Pós-condições | O post do usuário é visível na linha do tempo do usuário e pode ser visto por outros usuários que o seguem. |
|  |  |
| **Cenário principal** | |
| **Ações do ator** | **Ações do sistema** |
| 1. O usuário navega até a sua linha do tempo ou área de postagem. | 1. O sistema verifica o conteúdo do post contra o banco de dados de palavras proibidas. Se nenhuma palavra proibida for encontrada, o sistema publica o post na linha do tempo do usuário. |
| 1. O usuário clica no campo de entrada do post. |  |
| 1. O usuário insere o texto desejado no campo de entrada do post |  |
| 1. O usuário clica no botão "Postar". |  |
| **Restrições** | |
|  | **Ações do sistema** |
|  | O sistema deve monitorar e moderar o conteúdo postado para garantir que esteja em conformidade com as regras da comunidade e com as restrições de palavras específicas. |
| **Cenário alternativo** | |
| **Ações do ator** | **Ações do sistema** |
| 1. O usuário insere o conteúdo do post, mas decide não postar. | 3. O sistema descarta o post não publicado. |
| 1. O usuário clica no botão "Cancelar" ou navega para outra parte do site. |  |
| **Cenário de exceção** | |
| **Ações do ator** | **Ações do sistema** |
| 1. O usuário tenta postar conteúdo que contém palavras listadas no banco de dados de palavras proibidas. | 1. O sistema recusa a postagem e notifica o usuário sobre a restrição. |
| **Regras de negócio** | |
| 1. O conteúdo do post deve estar em conformidade com as regras da comunidade e não deve conter palavras proibidas. | |
| 1. O sistema deve limitar o conteúdo do post a um máximo de 250 caracteres. | |
| 1. As configurações de privacidade do usuário devem ser respeitadas em todas as postagens. | |

|  |  |
| --- | --- |
| **Caso de uso 003 – Editar perfil** | |
| Ator principal | Usuário |
|  |  |
| Resumo | Este caso de uso descreve o processo em que um usuário cadastrado pode editar as informações em seu próprio perfil. |
|  |  |
| Pré-condições | O usuário deve estar autenticado e ter uma sessão ativa. |
|  |  |
| Pós-condições | O perfil do usuário é atualizado conforme as alterações feitas. |
|  |  |
| **Cenário principal** | |
| **Ações do ator** | **Ações do sistema** |
| 1. O usuário acessa seu perfil. | 1. O sistema exibe as informações atuais do perfil. |
| 1. O usuário clica em um botão ou link para editar o perfil. | 1. O sistema exibe uma página ou formulário de edição. |
| 1. O usuário realiza as alterações desejadas, como trocar a imagem de perfil, editar o nome ou alterar a senha. Depois clica em "Salvar". | 1. O sistema valida as informações inseridas e atualiza o perfil. |
| **Restrições** | |
|  | **Ações do sistema** |
|  | Todas as informações devem ser validadas antes de atualizar o perfil. |
| **Cenário alternativo** | |
| **Ações do ator** | **Ações do sistema** |
| **Trocar imagem do perfil:** |  |
| 1. O usuário clica para trocar a imagem de perfil. | 1. O sistema abre um diálogo para o upload de uma nova imagem. |
| 1. O usuário seleciona uma nova imagem e confirma. | 1. O sistema atualiza a imagem de perfil. |
| **Editar senha:** |  |
| 1. O usuário clica para alterar a senha. | 1. O sistema pede a senha atual e a nova senha. |
| 1. O usuário insere as informações requeridas. | 1. O sistema valida e atualiza a senha. |
| **Cenário de exceção** | |
| **Ações do ator** | **Ações do sistema** |
| 1. O usuário insere informações inválidas, como um formato de imagem não suportado ou uma senha fraca. | 1. O sistema exibe uma mensagem de erro e solicita que o usuário corrija as informações. |
| **Regras de negócio** | |
| 1. Todas as informações inseridas devem passar por uma validação rigorosa para garantir a integridade e segurança dos dados. | |
| 1. O sistema deve confirmar a identidade do usuário antes de permitir alterações sensíveis, como a mudança de senha. | |

|  |  |
| --- | --- |
| **Caso de uso 004 – Visualizar log de acesso** | |
| Ator principal | Usuário |
|  |  |
| Resumo | Este caso de uso descreve o processo pelo qual um usuário registrado pode visualizar seu próprio log de acessos, mostrando os horários e talvez até locais de login anteriores. |
|  |  |
| Pré-condições | O usuário deve estar autenticado e ter uma sessão ativa. |
|  |  |
| Pós-condições | O usuário tem mais informações sobre suas atividades de login, o que pode ser útil para monitorar a segurança de sua conta. |
|  |  |
| **Cenário principal** | |
| **Ações do ator** | **Ações do sistema** |
| 1. O usuário navega até a seção de segurança ou configurações de sua conta. | 1. O sistema exibe as opções disponíveis relacionadas à segurança e configurações. |
| 1. O usuário seleciona a opção para visualizar o log de acesso. | 1. O sistema recupera os dados do log de acesso do banco de dados e os exibe ao usuário, mostrando os horários de login anteriores. |
| **Restrições** | |
|  | **Ações do sistema** |
|  | Apenas o próprio usuário pode visualizar seu log de acesso; terceiros ou usuários não autenticados não têm essa permissão. |
| **Cenário alternativo** | |
|  | **Ações do sistema** |
|  | Se não houver dados de log disponíveis, o sistema informa ao usuário que não existem logs de acesso para serem visualizados no momento. |
|  |  |
| **Cenário de exceção** | |
| **Ações do ator** | **Ações do sistema** |
| 1. O usuário tenta acessar o log de acesso de outro usuário. | 1. O sistema nega o acesso e exibe uma mensagem de erro, informando que o usuário não tem permissão para acessar esses dados. |
| **Regras de negócio** | |
| 1. O sistema deve garantir que apenas o proprietário da conta possa acessar o log de acesso. | |
| 1. O log de acesso deve ser mantido de forma segura para garantir que informações sensíveis não sejam expostas. | |

|  |  |
| --- | --- |
| **Caso de uso 005 – Gerenciar usuários** | |
| Ator principal | Usuário administrador |
|  |  |
| Resumo | Este caso de uso permite que o usuário administrador gerencie contas de usuários registrados, podendo ativá-las, desativá-las ou modificá-las. |
|  |  |
| Pré-condições | O usuário administrador deve estar autenticado. |
|  |  |
| Pós-condições | O usuário administrador deve receber um feedback sobre o sucesso ou falha das ações realizadas. |
|  |  |
| **Cenário principal** | |
| **Ações do ator** | **Ações do sistema** |
| 1. O usuário administrador acessa a página de gerenciamento de usuários. | 1. O sistema exibe uma lista de usuários registrados. |
| 1. O administrador seleciona um usuário para gerenciar. | 1. O sistema exibe as opções disponíveis para gerenciamento desse usuário. |
| 1. O administrador realiza a ação desejada (ativar, desativar, modificar). | 1. O sistema atualiza o status do usuário selecionado. |
| **Restrições** | |
|  | **Ações do sistema** |
|  | Todas as alterações devem ser registradas para auditoria futura. |
| **Cenário alternativo** | |
|  | **Ações do sistema** |
|  | **Usuário não encontrado:** |
|  | O sistema informa ao administrador que o usuário não foi encontrado. |
|  |  |
| **Cenário de exceção** | |
| **Ações do ator** | **Ações do sistema** |
| 1. O administrador tenta realizar uma ação que ele não tem permissão para fazer (por exemplo, apagar um outro usuário administrador). | 1. O sistema nega o pedido e informa ao administrador que ele não tem permissão para executar essa ação. |
| **Regras de negócio** | |
| 1. Somente usuários com privilégios de administrador podem acessar a página de gerenciamento de usuários. | |
| 1. As ações de gerenciamento devem ser registradas em um log de auditoria. | |

|  |  |
| --- | --- |
| **Caso de uso 006 – Gerenciar conteúdo** | |
| Ator principal | Usuário administrador |
|  |  |
| Resumo | Este caso de uso permite ao usuário administrador gerenciar o conteúdo postado no sistema. Ele pode visualizar, editar, aprovar ou remover postagens de acordo com os critérios da plataforma. |
|  |  |
| Pré-condições | O usuário administrador deve estar autenticado. |
|  |  |
| Pós-condições | O usuário administrador deve receber um feedback sobre o sucesso ou falha das ações realizadas. |
|  |  |
| **Cenário principal** | |
| **Ações do ator** | **Ações do sistema** |
| 1. O usuário administrador acessa a página de gerenciamento de conteúdo. | 1. O sistema exibe uma lista de postagens recentes, pendentes de aprovação ou sinalizadas para revisão. |
| 1. O administrador seleciona uma postagem para gerenciar. | 1. O sistema exibe as opções de edição, aprovação, e remoção para essa postagem. |
| 1. O administrador realiza a ação desejada (editar, aprovar, remover). | 1. O sistema atualiza o status da postagem e a apresenta conforme a ação realizada pelo administrador. |
| **Restrições** | |
|  | **Ações do sistema** |
|  | Todas as alterações devem ser registradas para auditoria futura. |
| **Cenário alternativo** | |
| **Ações do ator** | **Ações do sistema** |
| **Pesquisa avançada:** |  |
| 1. O administrador usa filtros de pesquisa para encontrar postagens específicas. | 1. O sistema atualiza a lista de postagens de acordo com os critérios selecionados. |
| **Cenário de exceção** | |
| **Ações do ator** | **Ações do sistema** |
| 1. O administrador tenta editar ou remover uma postagem que ele não tem permissão para modificar. | 1. O sistema nega o pedido e informa ao administrador que ele não tem permissão para executar essa ação. |
| **Regras de negócio** | |
| 1. Somente usuários com privilégios de administrador podem acessar a página de gerenciamento de conteúdo. | |
| 1. O administrador deve seguir as diretrizes e políticas da plataforma ao tomar decisões de moderação. | |

|  |  |
| --- | --- |
| **Caso de uso 007 – Monitorar log de acesso** | |
| Ator principal | Usuário administrador |
|  |  |
| Resumo | Este caso de uso permite ao usuário administrador monitorar os logs de acesso de todos os usuários registrados. Isso inclui informações como data, hora. |
|  |  |
| Pré-condições | O usuário administrador deve estar autenticado. |
|  |  |
| Pós-condições | O sistema deve registrar a atividade do administrador para fins de auditoria. |
|  |  |
| **Cenário principal** | |
| **Ações do ator** | **Ações do sistema** |
| 1. O usuário administrador acessa a página de monitoramento de logs de acesso. | 1. O sistema exibe uma lista dos logs de acesso mais recentes para todos os usuários registrados. |
| 1. O administrador analisa os registros para identificar qualquer atividade suspeita ou não autorizada. | 1. N/A (Neste cenário, assumimos que o administrador apenas visualiza os dados). |
| **Restrições** | |
|  | **Ações do sistema** |
|  | Todas as atividades de monitoramento realizadas pelo administrador devem ser registradas em um log de auditoria. |
| **Cenário alternativo** | |
| **Ações do ator** | **Ações do sistema** |
| **Filtrar registros:** |  |
| 1. O administrador usa filtros para restringir os registros exibidos. | 1. O sistema atualiza a lista de registros de acordo com os critérios selecionados. |
| **Cenário de exceção** | |
| **Ações do ator** | **Ações do sistema** |
| 1. O administrador tenta acessar logs de acesso que ele não tem permissão para ver. | 1. O sistema nega o acesso e informa ao administrador que ele não tem permissão para visualizar esses registros. |
| **Regras de negócio** | |
| 1. Somente usuários com privilégios de administrador podem acessar a página de monitoramento de logs de acesso. | |
| 1. O acesso a logs sensíveis deve ser restrito e estar em conformidade com políticas de privacidade e regulamentações aplicáveis. | |

|  |  |
| --- | --- |
| **Caso de uso 008 – Analisar comportamento de acesso** | |
| Ator principal | Usuário administrador |
|  |  |
| Resumo | Este caso de uso permite ao usuário administrador analisar o comportamento de acesso dos usuários registrados para identificar padrões, otimizar a experiência do usuário e reforçar a segurança. |
|  |  |
| Pré-condições | O usuário administrador deve estar autenticado. |
|  |  |
| Pós-condições | O sistema deve registrar a atividade do administrador para fins de auditoria. |
|  |  |
| **Cenário principal** | |
| **Ações do ator** | **Ações do sistema** |
| 1. O usuário administrador acessa a página de análise de comportamento de acesso. | 1. O sistema apresenta várias métricas e gráficos relacionados ao comportamento de acesso dos usuários. |
| 1. O administrador analisa as informações apresentadas, como frequência de login e duração da sessão. | 1. N/A (Neste cenário, assumimos que o administrador apenas visualiza os dados). |
| **Restrições** | |
|  | **Ações do sistema** |
|  | Todas as atividades de análise realizadas pelo administrador devem ser registradas em um log de auditoria. |
| **Cenário alternativo** | |
| **Ações do ator** | **Ações do sistema** |
| **Pesquisa avançada:** |  |
| 1. O administrador utiliza ferramentas avançadas de análise, como segmentação de dados ou análise de tendências. | 1. O sistema atualiza as métricas e gráficos com base nas análises avançadas selecionadas. |
| **Cenário de exceção** | |
| **Ações do ator** | **Ações do sistema** |
| 1. O administrador tenta acessar métricas que ele não tem permissão para ver. | 1. O sistema nega o acesso e informa ao administrador que ele não tem permissão para visualizar essas métricas. |
| **Regras de negócio** | |
| 1. Somente usuários com privilégios de administrador podem acessar a página de análise de comportamento de acesso. | |
| 1. Os dados analisados devem ser armazenados e tratados de acordo com as políticas de privacidade e regulamentações aplicáveis. | |

|  |  |
| --- | --- |
| **Caso de uso 009 – Seguir Usuário** | |
| Ator principal | Usuário |
|  |  |
| Resumo | Este caso de uso descreve a funcionalidade onde um usuário segue outro usuário para começar a receber atualizações deste em seu feed. |
|  |  |
| Pré-condições | O usuário está logado na plataforma. |
|  |  |
| Pós-condições | O usuário seguido é adicionado à lista de seguindo do usuário que realizou a ação, e suas futuras postagens aparecerão no feed do seguidor. |
|  |  |
| **Cenário principal** | |
| **Ações do ator** | **Ações do sistema** |
| 1. O usuário navega pelo perfil de outro usuário. | 1. O sistema exibe o perfil, incluindo um botão para seguir. |
| 1. O usuário clica no botão "Seguir". | 1. O sistema registra essa ação, adicionando o perfil visitado à lista de usuários seguidos do ator. |
| **Restrições** | |
|  | **Ações do sistema** |
|  | O sistema deve verificar se o usuário já segue o outro antes de adicionar à lista de seguindo. |
| **Cenário alternativo** | |
| Nenhum cenário alternativo relevante. | |
| **Cenário de exceção** | |
| **Ações do ator** | **Ações do sistema** |
| 1. O usuário tenta seguir um perfil que ele já segue. | 1. O sistema não realiza a ação e pode exibir uma mensagem indicando que o perfil já é seguido. |
| **Regras de negócio** | |
| 1. Um usuário não pode seguir a si mesmo. | |

|  |  |
| --- | --- |
| **Caso de uso 010 – Visualizar Perfil** | |
| Ator principal | Usuário |
|  |  |
| Resumo | Este caso de uso permite que um usuário visualize o perfil de outro usuário, incluindo informações públicas como biografia, postagens e número de seguidores e seguidos. |
|  |  |
| Pré-condições | O usuário está logado na plataforma. |
|  |  |
| Pós-condições | O perfil do outro usuário é exibido. |
|  |  |
| **Cenário principal** | |
| **Ações do ator** | **Ações do sistema** |
| 1. O usuário insere o nome de usuário na barra de pesquisa ou clica em um nome de usuário em qualquer parte da plataforma. | 1. O sistema processa a solicitação e redireciona para a página de perfil do usuário buscado. |
| 1. O usuário visualiza o perfil, incluindo postagens, seguidores e perfis seguidos. |  |
| **Restrições** | |
|  | **Ações do sistema** |
|  | O sistema deve garantir que apenas informações públicas sejam exibidas. |
| **Cenário alternativo** | |
| Nenhum cenário alternativo relevante. | |
| **Cenário de exceção** | |
| **Ações do ator** | **Ações do sistema** |
| 1. O usuário tenta acessar um perfil inexistente. | 1. O sistema exibe uma mensagem de erro indicando que o perfil não foi encontrado. |
| **Regras de negócio** | |
| 1. Informações de perfil são públicas por padrão, mas, as postagens do usuário não são públicas. | |

# **6. MODELOS DO SISTEMA**

... (modelos UML, ER, etc.)

# **7. EVOLUÇÃO DO SISTEMA**

... (possíveis atualizações, adições de novas funcionalidades, etc.)

# **8. GLOSSÁRIO**

**Usuário não registrado:** Indivíduo que ainda não completou o processo de registro.

**Usuário registrado:** Indivíduo que completou o processo de registro.

**Usuário administrador:** Usuário com permissões elevadas para gerenciar o sistema.

# **9. FERRAMENTAS UTILIZADAS**

Este capítulo descreve as ferramentas escolhidas para o desenvolvimento, implementação e manutenção do protótipo da rede social Bem-te-vi, delineando como cada uma contribui para a realização dos objetivos específicos do projeto. A seleção dessas ferramentas visa estabelecer um ambiente de desenvolvimento robusto, eficiente e adaptado às necessidades específicas do sistema.

***Laravel*:** é um *Framework* PHP para desenvolvimento *web* que segue o padrão arquitetural MVC (*Model-View-Controller*). Ele será utilizado para construir a estrutura básica da rede social, oferecendo uma rica biblioteca de funcionalidades que facilitam tarefas comuns como autenticação de usuários, roteamento, sessões e *caching*. A escolha do *Laravel* permite um desenvolvimento ágil e uma manutenção simplificada do código, o que é crucial para a adaptação rápida às necessidades de moderação de conteúdo e segurança.

***Laragon*:** é um ambiente de desenvolvimento portátil e isolado que facilita a gestão de serviços como *Apache*, *My*SQL e PHP no desenvolvimento local. Será utilizado para configurar rapidamente um ambiente de desenvolvimento consistente entre todos os membros da equipe, minimizando possíveis inconsistências entre diferentes ambientes de desenvolvimento.

***Maria*DB e *Heidi*SQL:** *Maria*DB, um sistema de gerenciamento de banco de dados derivado do *MySQL*, será usado para armazenar e gerenciar todas as informações da rede social. *Heidi*SQL, uma interface gráfica para gestão de bases de dados, será usada para facilitar a visualização e manipulação dos dados. Esta combinação permite um gerenciamento eficiente dos dados com alta performance, segurança e escalabilidade.

***Visual Studio Code*:** é um editor de código fonte leve, mas poderoso, que suporta uma variedade de linguagens de programação e *Frameworks*, incluindo PHP e *Laravel*. Ele é usado por programadores para escrever e revisar o código de forma eficiente, com suporte a extensões que melhoram a produtividade e a colaboração, como correção de código e controle de versão integrado.

***Bulma* *Framework*:** é um *Framework* CSS baseado em *Flexbox* que será utilizado para a construção das interfaces do usuário. Com sua abordagem modular e fácil de usar, o *Bulma* permite um desenvolvimento rápido e responsivo do *front-end*, assegurando que a rede social seja acessível em diversos dispositivos e tamanhos de tela.

***Visual Paradigm Online*:** é uma ferramenta utilizada para a criação de diagramas de casos de uso e classes, facilitando a visualização das estruturas de dados e fluxos de trabalho do sistema. *Visual Paradigm Online* ajuda a manter a documentação do projeto clara e atualizada, essencial para o alinhamento da equipe e para futuras revisões do projeto.

***Trello*:** é uma ferramenta de gerenciamento de projetos baseada em *Kanban*, será usada para organizar as tarefas de desenvolvimento, monitorar o progresso e priorizar atividades. Facilita a comunicação e colaboração entre os membros da equipe, garantindo que todos estejam alinhados com os objetivos do projeto.

***Git* e *GitHub*:** esteserá usado para controle de versão, permitindo que múltiplos desenvolvedores trabalhem simultaneamente no código sem conflitos. *GitHub*, uma plataforma de hospedagem de código-fonte, será usada para armazenar o repositório do projeto, facilitar revisões de código, rastrear problemas e gerenciar atualizações. Essas ferramentas são essenciais para a colaboração efetiva e para manter um histórico detalhado das mudanças no código.

Estas ferramentas foram selecionadas não apenas por suas capacidades individuais, mas também pela forma como se integram para criar um ambiente de desenvolvimento coeso e eficaz, diretamente alinhado com os objetivos específicos do projeto Bem-te-vi, especialmente no que tange à criação de um ambiente virtual seguro e amigável para compartilhamento de mensagens.

# **10. BOAS PRÁTICAS**

A adoção de boas práticas de desenvolvimento é crucial para garantir a qualidade, a manutenção e a escalabilidade do código no projeto da rede social Bem-te-vi. Este capítulo detalha as práticas adotadas pelos autores para promover um código limpo, bem documentado e fácil de manter.

**Guia de Estilo de Codificação (PSR-1/PSR-2 para PHP):** a adesão aos padrões PSR-1 e PSR-2 é fundamental para garantir a uniformidade e legibilidade do código PHP utilizado no projeto. O PSR-1 estabelece normas básicas de codificação, como o uso de *tags* de fechamento PHP, a codificação de caracteres UTF-8 sem BOM para arquivos PHP, e nomes de classes em *StudlyCaps*. Por sua vez, o PSR-2 trata de aspectos de formatação, como a indentação por quatro espaços (sem uso de tabulações), linhas com no máximo 120 caracteres, e colocação de chaves seguindo o estilo K&R.A implementação desses guias de estilo ajuda a manter o código organizado e consistente, facilitando a revisão por outros desenvolvedores e contribuindo para um ambiente de trabalho colaborativo mais eficiente. Além disso, a consistência promovida pelos padrões PSR facilita a detecção e correção de erros, melhorando a qualidade geral do *software*.

***DocBlock* para PHP:** O uso de *DocBlock* é essencial para a documentação no código PHP. *DocBlock* é um formato de comentário de múltiplas linhas que é colocado antes das declarações de classes, métodos e funções para descrever o comportamento, os parâmetros, os retornos e as exceções desses componentes de *software*. A prática de utilizar *DocBlock* permite que ferramentas de geração de documentação automática, como phpDocumentor, criem uma documentação de API detalhada e acessível. A documentação de código com *DocBlock* é especialmente importante em um projeto colaborativo como o Bem-te-vi, onde os desenvolvedores podem trabalhar em diferentes partes do sistema. Com uma documentação clara e acessível, os membros da equipe podem entender rapidamente a funcionalidade e o propósito do código existente, facilitando a integração e a continuidade do desenvolvimento. Além disso, uma boa documentação é crucial para a manutenção a longo prazo do sistema, permitindo que alterações e expansões sejam realizadas com segurança e eficácia.

Adotar essas práticas de desenvolvimento não apenas melhora a qualidade do código, mas também fomenta uma cultura de desenvolvimento disciplinado e comprometido com a qualidade entre os programadores, elementos essenciais para o sucesso do projeto da rede social Bem-te-vi.