VISÃO DO SISTEMA

# Riscos do Projeto

Tabela 10 – Riscos do Projeto

|  |  |  |
| --- | --- | --- |
| **Nº** | **Descrição do Risco** | **Classificação** |
| 1 | A integração com a API Edamam pode apresentar instabilidades ou custos não previstos. | M |
| 2 | A complexidade na implementação do sistema de recomendação de receitas pode atrasar o desenvolvimento. | M |
| 3 | Possível sobrecarga de usuários sem login, limitando funcionalidades. | B |
| 4 | A base de dados de receitas pode não ser ampla ou detalhada o suficiente, resultando em recomendações pouco variadas. | B |
| 5 | O sistema de comentários e os formulários de cadastro/login podem ser vulneráveis a ataques de injeção de SQL se não forem devidamente protegidos. | A |
| 6 | O projeto por ser acadêmico possui prazo delimitado por semestre letivo, afetando o tempo total para o desenvolvimento do sistema. | M |

# Restrições do Projeto

Tabela 11 – Restrições do Projeto

|  |  |  |
| --- | --- | --- |
| **Nº** | **Descrição da Restrição** | **Tipo** |
| 1 | O desenvolvimento do front-end está restrito ao uso de HTML/CSS/JS. | Temporal |
| 2 | O back-end deve ser desenvolvido utilizando a linguagem Java. | Tecnológica |
| 3 | Disponibilidade da API Edamam sujeita a limites de uso diante pagamento. | Tecnológica |

# Regras do Negócio

Tabela 12 – Regras de Negócio.

|  |  |  |
| --- | --- | --- |
| **Código** | **Nome** | **Descrição** |
| **RN001** | Acesso Anônimo | Usuários sem conta podem visualizar receitas pré- recomendadas e buscar receitas pelos ingredientes que possuem. |

|  |  |  |
| --- | --- | --- |
| **RN002** | Acesso com Conta | Para ser possível salvar receitas, comentar, avaliar e classificar, o usuário precisa estar logado em uma conta. |
| **RN003** | Restrição de Funcionalidades | Caso um usuário anônimo tente acessar funcionalidades exclusivas de um usuário logado, o sistema solicitará o login para prosseguir. |
| **RN004** | Filtro de Receitas | O sistema deve permitir a filtragem de receitas por características específicas como diet, sem glúten, etc. |
| **RN005** | Avaliação única por usuário | Cada usuário pode avaliar (classificar) uma receita apenas uma vez, podendo atualizar sua nota posteriormente. |

# Requisitos Funcionais

Tabela 13 – Requisitos Funcionais.

|  |  |  |
| --- | --- | --- |
| **Código** | **Nome** | **Descrição** |
| **RF001** | Manter Cadastro e Login de Usuários | O sistema deve permitir que os usuários se cadastrem e façam login, com a senha sendo criptografada. |
| **RF002** | Manter Entidades do Sistema (CRUD) | O sistema deve possuir um CRUD (Criar, Ler, Atualizar, Deletar) completo para usuários, receitas e ingredientes. |
| **RF003** | Gerenciar Comentários | O sistema deve permitir que usuários logados comentem nas receitas. |
| **RF004** | Gerenciar Classificação de Receitas | Usuários logados devem poder classificar as receitas, e o sistema deve permitir a filtragem por essa classificação. |
| **RF005** | Sugerir Receitas | O sistema deve sugerir receitas com base em características pré-selecionadas pelo usuário. |
| **RF006** | Exibir Informações Nutricionais | O sistema deve exibir as informações nutricionais das receitas, obtidas através da API Edamam. |
| **RF007** | Recomendar Receitas por Ingredientes | O sistema deve permitir que o usuário selecione os ingredientes que possui para receber recomendações de receitas possíveis. |
| **RF008** | Favoritar Receitas | Usuários logados devem ter a funcionalidade de salvar receitas para visualização posterior. |

# Requisitos Não Funcionais

Tabela 14 – Requisitos Não Funcionais.

|  |  |  |
| --- | --- | --- |
| **Código** | **Nome** | **Descrição** |
| **RNF001** | Usabilidade | A interface do sistema deve ser objetiva e de fácil utilização, visando a economia de tempo do usuário e a transparência das ações tomadas durante o uso das funcionalidades. |
| **RNF002** | Segurança | As senhas dos usuários cadastrados devem ser armazenadas de forma segura no banco de dados utilizando criptografia. |
| **RNF003** | Desempenho | O sistema deve responder rapidamente às solicitações do usuário, especialmente na recomendação de receitas. |
| **RNF004** | Manutenibilidade | O código-fonte, tanto do back-end em Java quanto do front-end, deve seguir padrões de projeto e ser devidamente comentado para facilitar futuras correções de erros e implementações de novas funcionalidades. |