**1 INTRODUÇÃO**

Este documento apresenta os requisitos de uma aplicação web construída em Flask para gerenciamento de usuários, projetos e atividades. A aplicação oferece recursos de registro e login de usuários, além de permitir o cadastro e a gestão de projetos e atividades. Abaixo, são descritos os requisitos funcionais e não funcionais do sistema.

**2 OBJETIVOS**

O objetivo principal da aplicação é facilitar o gerenciamento de projetos e atividades, permitindo que usuários autenticados possam cadastrar, listar e excluir projetos e atividades relacionadas.

**3 REQUISITOS FUNCIONAIS**

Os requisitos funcionais (RF) descrevem as funcionalidades que a aplicação deve cumprir.

**3.1 GERENCIAMENTO DE USUÁRIOS**

* **RF01 - Cadastro de Usuário**
  + Descrição: Permitir o cadastro de novos usuários através da rota /register.
  + Requisitos: Campos obrigatórios: email e password.
  + Observação: O sistema deve verificar a existência prévia de um usuário com o e-mail informado e armazenar as senhas com hash.
* **RF02 - Login de Usuário**
  + Descrição: Permitir que um usuário realize login através da rota /login.
  + Requisitos: Campos obrigatórios: email e password.
  + Observação: A senha deve ser verificada utilizando a função check\_password\_hash.

**3.2 GERENCIAMENTO DE PROJETOS**

* **RF03 - Cadastro de Projeto**
  + Descrição: Permitir o cadastro de novos projetos através da rota /cadastrarprojeto.
  + Requisitos: Campos obrigatórios: projectName, projectDescription, technologies e startDate.
  + Observação: A data de início (startDate) deve estar no formato YYYY-MM-DD.
* **RF04 - Listagem de Projetos e Atividades Associadas**
  + Descrição: Permitir a listagem de todos os projetos, incluindo as atividades relacionadas, através da rota /projetos.
  + Observação: A resposta deve conter as informações do projeto como id, nome, descricao, tecnologias, data\_inicio, bem como uma lista de atividades (atividades) associadas.
* **RF05 - Exclusão de Projetos**
  + Descrição: Permitir a exclusão de projetos específicos através da rota /projetos/<project\_id>, onde <project\_id> é o ID do projeto.
  + Observação: Ao excluir um projeto, todas as atividades associadas a ele também devem ser removidas.

**3.3 GERENCIAMENTO DE ATIVIDADES**

* **RF06 - Cadastro de Atividade**
  + Descrição: Permitir o cadastro de novas atividades associadas a um projeto específico através da rota /cadastraratividade.
  + Requisitos: Campos obrigatórios: projeto\_id, descricao e due\_date.
  + Observação: A data de vencimento (due\_date) deve estar no formato YYYY-MM-DD. Caso omitida, a data atual é usada.
* **RF07 - Listagem de Atividades**
  + Descrição: Permitir a listagem de todas as atividades cadastradas, através da rota /atividades.
  + Observação: A resposta deve incluir as informações id, descricao, due\_date e projeto\_id.

**3.4 GERENCIAMENTO DE ARQUIVOS ESTÁTICOS**

* **RF08 - Exibição de Páginas Estáticas**
  + Descrição: Permitir o acesso às páginas estáticas de cadastro de projetos (/cadastrarprojeto), cadastro de atividades (/cadastraratividade), e configuração (/config).
  + Observação: A página principal (/) deve exibir o arquivo index.html localizado na pasta static.

**4 REQUISITOS NÃO FUNCIONAIS**

Os requisitos não funcionais (RNF) determinam características de qualidade que a aplicação deve cumprir.

* **RNF01 - Persistência de Dados**
  + Descrição: A aplicação deve utilizar um banco de dados SQLite para persistência de dados.
  + Observação: O banco de dados deve ser criado automaticamente ao iniciar a aplicação, caso as tabelas não existam.
* **RNF02 - Segurança**
  + Descrição: Senhas de usuários devem ser armazenadas de forma segura, utilizando hashing.
  + Observação: O sistema deve retornar mensagens de erro apropriadas para dados inválidos ou operações não permitidas.
* **RNF03 - Tratamento de Erros**
  + Descrição: Em caso de erro ao cadastrar um projeto ou atividade, o sistema deve retornar uma mensagem informando o erro específico.
* **RNF04 - Formato de Dados**
  + Descrição: Todas as respostas da API devem estar em formato JSON para facilitar o consumo em diferentes front-ends.

**5 CASOS DE USO**

A seguir, são descritos os principais casos de uso (CU) do sistema.

* **CU01 - Registrar Usuário**
  + Ator: Usuário
  + Descrição: Permitir que um novo usuário se registre no sistema.
* **CU02 - Realizar Login**
  + Ator: Usuário
  + Descrição: Permitir que um usuário existente realize login no sistema.
* **CU03 - Cadastrar Projeto**
  + Ator: Usuário autenticado
  + Descrição: Permitir o cadastro de um novo projeto, incluindo nome, descrição, tecnologias e data de início.
* **CU04 - Listar Projetos com Atividades**
  + Ator: Usuário autenticado
  + Descrição: Permitir que o usuário visualize a lista de projetos com suas atividades associadas.
* **CU05 - Excluir Projeto e Atividades**
  + Ator: Usuário autenticado
  + Descrição: Permitir que o usuário exclua um projeto específico e todas as suas atividades associadas.
* **CU06 - Cadastrar Atividade**
  + Ator: Usuário autenticado
  + Descrição: Permitir o cadastro de uma nova atividade associada a um projeto.
* **CU07 - Listar Atividades**
  + Ator: Usuário autenticado
  + Descrição: Permitir a visualização da lista de atividades cadastradas no sistema.

**6 CONSIDERAÇÕES FINAIS**

Este documento descreve as principais funcionalidades e especificações de qualidade para o desenvolvimento da aplicação. O cumprimento dos requisitos e casos de uso definidos é essencial para assegurar o funcionamento correto do sistema.