Nome: Vinicius Roberto Gonzales Arado – RGM: 11221100374

Nome: Guilherme da Silva Leon – RGM: 11221103758

Nome: Guilherme Marques Rezende – RGM: 11221104226

Nome: Lucas de Jesus Oliveira – RGM: 11221100971

Nome: João Gabriel Gomes Correia – RGM: 11222201815

**Documento de Requisitos e Arquitetura**

## **1. Introdução**

## A segurança da informação é um dos pilares essenciais no desenvolvimento de sistemas modernos, especialmente em contextos que envolvem dados pessoais e sensíveis. Neste cenário, autenticação robusta e mecanismos seguros de recuperação de senha se tornam indispensáveis para garantir a confidencialidade, integridade e disponibilidade das informações. A presente atividade tem como foco a implementação de dois componentes amplamente adotados pela indústria: autenticação baseada em **Access Token e Refresh Token**, e um sistema de **recuperação de senha via token temporário enviado por e-mail**.

## Enquanto os Access Tokens proporcionam agilidade e escalabilidade na comunicação entre cliente e servidor, os Refresh Tokens garantem sessões prolongadas com segurança, evitando o uso contínuo de credenciais do usuário. Já o mecanismo de recuperação de senha representa uma camada crítica de proteção e usabilidade, exigindo uma abordagem que assegure a validade temporária, a unicidade e a confidencialidade dos tokens utilizados.

## Este documento define os requisitos técnicos e operacionais para a implementação desses dois componentes, organizando-os em requisitos funcionais, não funcionais, de segurança e de dados. O objetivo é garantir uma base sólida para a construção de um sistema de autenticação seguro, escalável e em conformidade com a LGPD e com as melhores práticas da engenharia de software.

## **2. Especificação do Sistema**

### 2.1 Requisitos Funcionais

### Esta seção apresenta os comportamentos esperados do sistema relacionados às funcionalidades de sessão prolongada via Refresh Token e ao processo de recuperação de senha.

### **Autenticação com Access Token e Refresh Token:**

### O sistema deverá implementar autenticação por meio de dois tipos de token:

### **Access Token:** usado para autenticação em endpoints protegidos e com tempo de expiração curto (ex.: 15 minutos).

### **Refresh Token:** utilizado para renovação do Access Token, com expiração mais longa (ex.: 7 dias), mantendo a sessão ativa sem exigir novo login com credenciais.

### O fluxo da autenticação deverá obedecer às seguintes diretrizes:

### O endpoint de login (/auth/login) deverá retornar um Access Token e um Refresh Token.

### O endpoint /auth/refresh-token deverá permitir a renovação do Access Token utilizando um Refresh Token válido.

### O endpoint /auth/logout deverá invalidar o Refresh Token ativo.

### O sistema deverá armazenar informações dos Refresh Tokens emitidos para verificação e controle de uso indevido.

### **Recuperação de Senha via Token Temporário:**

### A funcionalidade de recuperação de senha deverá ser implementada com base em tokens temporários e seguros, seguindo os seguintes critérios:

### O endpoint /auth/request-password-reset deverá receber o e-mail do usuário e gerar um token único com tempo de expiração (ex.: 1 hora).

### O token deverá ser enviado ao e-mail informado, em um link que direcione o usuário ao processo de redefinição de senha.

### O endpoint /auth/reset-password deverá receber o token e a nova senha, validando sua integridade e expirando-o após uso único.

### 2.2 Requisitos Não Funcionais

### Para garantir um sistema eficiente, confiável e de fácil integração, os seguintes requisitos não funcionais deverão ser atendidos:

* **Desempenho:** As operações de autenticação e renovação de sessão devem apresentar baixa latência.
* **Assíncrono:** O envio de e-mails de recuperação de senha deverá ser realizado de forma assíncrona, evitando bloqueios na aplicação.
* **Documentação:** A API deverá ser documentada conforme o padrão OpenAPI/Swagger, permitindo fácil integração por desenvolvedores terceiros.
* **Disponibilidade:** O sistema deverá ser capaz de atender múltiplas requisições simultâneas, escalando horizontalmente se necessário.

### 2.3 Requisitos de Segurança

### A segurança é o ponto central dessas funcionalidades. As seguintes práticas e controles deverão ser obrigatoriamente implementados:

* **Hash seguro para tokens:** Os Refresh Tokens e tokens de recuperação de senha devem ser armazenados em formato hash para evitar vazamentos.
* **Assinatura criptográfica:** Todos os JWTs deverão ser assinados com algoritmos seguros como HS256 ou RS256.
* **Expiração e revogação:** Todos os tokens devem conter data de expiração e permitir revogação.
* **Validação contextual:** Os Refresh Tokens poderão ser validados com base em informações como IP, User-Agent e dispositivo.
* **Proteção contra ataques:** Implementar rate limiting em endpoints de login e recuperação de senha, bem como mecanismos antifraude.

### 2.4 Requisitos de Dados

### A Para sustentar tecnicamente as funcionalidades descritas, será necessário estender a modelagem do banco de dados com as seguintes tabelas:

### **Tabela refresh\_tokens:** Armazena os Refresh Tokens ativos e suas características.

### id: Identificador único do token.

### usuario\_id: Referência ao usuário proprietário.

### token\_hash: Hash seguro do Refresh Token.

### data\_criacao: Data e hora da emissão.

### data\_expiracao: Data de expiração do token.

### user\_agent: Informações do navegador/dispositivo.

### ip: Endereço IP da solicitação.

### revogado: Indicador booleano de revogação.

### **Tabela tokens\_recuperacao\_senha:** Registra os tokens temporários gerados para redefinição de senha.

### id: Identificador único do token.

### usuario\_id: Usuário associado ao token.

### token\_hash: Hash do token enviado por e-mail.

### data\_criacao: Data de emissão.

### data\_expiracao: Tempo limite de validade.

### utilizado: Indicador de uso único (booleano).

## **3. Fluxo de Funcionamento**

## 3.1 Fluxo de Autenticação com Refresh Token

## O usuário envia suas credenciais (e-mail e senha) para o endpoint /auth/login.

## Se válidas, o sistema gera e retorna um Access Token e um Refresh Token.

## O Access Token é utilizado em chamadas protegidas e expira rapidamente.

## Quando expira, o cliente pode solicitar um novo par de tokens enviando o Refresh Token para /auth/refresh-token.

## O sistema valida o Refresh Token (hash, validade, revogação) e retorna novos tokens.

## Ao realizar logout, o usuário revoga o Refresh Token, invalidando o uso futuro.

## 3.2 Fluxo de Recuperação de Senha

## O usuário acessa o endpoint /auth/request-password-reset e informa seu e-mail.

## O sistema verifica se o e-mail está cadastrado e gera um token temporário único.

## O token é armazenado com hash e enviado ao usuário por e-mail em um link.

## O usuário clica no link e é direcionado ao formulário de redefinição de senha.

## O token e a nova senha são enviados ao endpoint /auth/reset-password.

## O sistema valida o token e, se válido, atualiza a senha e invalida o token.

## **4. Exemplos de Endpoints**

Os seguintes endpoints deverão ser adicionados à API para suportar as novas funcionalidades:

* POST /auth/login — Autenticação de usuário (retorna Access e Refresh Tokens).
* POST /auth/refresh-token — Renovação de Access Token utilizando um Refresh Token válido.
* POST /auth/logout — Revogação de Refresh Token e encerramento de sessão.
* POST /auth/request-password-reset — Solicitação de redefinição de senha via e-mail.
* POST /auth/reset-password — Redefinição de senha com base em token temporário.

## **5. Considerações Finais**

A implementação de mecanismos avançados como o **Refresh Token** e a **recuperação segura de senha por e-mail** é indispensável para aplicações que visam oferecer sessões prolongadas, experiência fluida ao usuário e conformidade com normas de segurança e privacidade. Este documento especifica os requisitos necessários para garantir o desenvolvimento eficaz e seguro dessas funcionalidades, reforçando o compromisso com a **proteção de dados pessoais** e a **resiliência contra ataques cibernéticos**.

As diretrizes aqui propostas devem servir como referência técnica para a equipe de desenvolvimento na construção de um sistema robusto, confiável e compatível com padrões de mercado.