Requisitos Funcionais do Sistema

1. Cadastro de Usuários:
   - O sistema deve permitir o cadastro de usuários com dois tipos de permissões: ADMINISTRADOR e COMUM.
   - Administradores podem cadastrar, alterar e excluir usuários.
   - Usuários comuns podem se cadastrar, alterar seus dados e senha.

2. Autenticação e Autorização:
   - O sistema deve possuir um mecanismo de login e logout.
   - Somente usuários autenticados podem acessar o sistema.

3. Gerenciamento de Eventos:
   - Usuários podem realizar operações de CRUD (Create, Read, Update, Delete) em eventos.
   - Administradores podem gerenciar eventos de todos os usuários.
   - Usuários comuns só podem gerenciar eventos criados por eles mesmos.

4. Atributos dos Eventos:
   - Cada evento deve possuir os seguintes atributos:
     - Nome
     - Descrição
     - Responsável
     - Criador
     - Data limite
     - Data de conclusão
     - Observação
     - Recorrência
     - Prioridade
     - Categoria

5. Visualização e Filtragem de Eventos:
   - O sistema deve permitir a visualização dos eventos em um calendário.
   - Deve haver funcionalidade de filtragem de eventos por categoria, prioridade, data, e recorrência.

Requisitos Não Funcionais

1. Segurança:
   - O sistema deve garantir a segurança dos dados dos usuários, incluindo criptografia de senhas.
   - Deve haver controle de acesso baseado nas permissões dos usuários (ADMINISTRADOR e COMUM).

2. Usabilidade:
   - A interface deve ser intuitiva e fácil de usar, com feedbacks claros para as ações do usuário.

3. Desempenho:
   - O sistema deve ser capaz de lidar com um grande número de usuários e eventos sem perda significativa de desempenho.
   - Respostas às ações dos usuários devem ser rápidas, com tempo de resposta inferior a 2 segundos para a maioria das operações.

4. Disponibilidade:
   - O sistema deve estar disponível pelo menos 99% do tempo.
   - Deve haver backup regular dos dados para garantir a recuperação em caso de falhas.

5. Manutenibilidade:
   - O código do sistema deve ser bem documentado e seguir boas práticas de desenvolvimento para facilitar a manutenção e evolução do sistema.

6. Compatibilidade:
   - O sistema deve ser compatível com os principais navegadores (Chrome, Firefox, Safari, Edge).

7. Escalabilidade:
   - O sistema deve ser projetado de forma a permitir a adição de novas funcionalidades sem necessidade de grandes reestruturações.

Requisitos de Implementação

1. Linguagens de Programação e Frameworks:
   - Backend:
     - Linguagem: Java
     - Framework: Spring Boot
   - Frontend:
     - Linguagens: HTML, CSS, JavaScript
     - Frameworks/Libraries: React ou Vue.js (opcional, dependendo da complexidade da interface)

2. Banco de Dados:
   - Tipo: Relacional ou não relacional
   - SGBD: PostgreSQL, MySQL ou MongoDB

3. Controle de Versão:
   - Ferramenta: Git
   - Plataforma de Hospedagem de Repositório: GitHub, GitLab ou Bitbucket

4. Servidores e Deploy:
   - Servidor de Aplicação: Tomcat (embutido no Spring Boot) ou outro servidor compatível
   - Serviço de Hospedagem: Heroku, AWS, DigitalOcean, ou outra plataforma de hospedagem compatível.