

## Anton Henrique bresolin

Última modificação: 20/07/2024 - 00:00

### Requisitos elaborados:

#### Requisitos Funcionais

##### Cadastro de Usuários (Cooperados):

- **Descrição:** O sistema deve permitir o cadastro de cooperados (usuários) com os seguintes dados: CPF, senha e papel (role).
- **Campos:**
  - **idUser:** UUID (gerado automaticamente).
  - **cpf:** String (11 caracteres, único e não nulo).
  - **password:** String (não nulo).
  - **role:** ENUM (USER, ADMIN).
- **Validações:**
  - O CPF deve ser único.
  - A senha deve ser armazenada de forma segura (hash).

##### Autenticação de Usuários:

- **Descrição:** Implementar um sistema de login que autentica usuários usando CPF e senha, gerando um token JWT para acesso às rotas protegidas.
- **Fluxo:**
  - O usuário envia CPF e senha.
  - O sistema valida as credenciais e, se corretas, retorna um token JWT.
  - O token JWT é usado para autenticação em todas as rotas protegidas.

##### Criação de Pautas:

- **Descrição:** Permitir que administradores (usuários com papel ADMIN) criem novas pautas de votação.
- **Campos:**
  - **idSchedule:** Long (gerado automaticamente).
  - **title:** String (45 caracteres, não nulo).
  - **description:** String (500 caracteres, não nulo).
  - **createdAt:** Timestamp (não nulo).
  - **timeLimit:** Integer (tempo limite para votação em segundos, padrão 60 segundos).
- **Validações:**
  - Somente usuários com papel ADMIN podem criar pautas.
  - Campos title, description, createdAt e timeLimit são obrigatórios.

### Registro de Votos:

- **Descrição:** Permitir que cooperados registrem seus votos com base no tempo limite definido na pauta.
- **Campos:**
  - **idVote:** Long (gerado automaticamente).
  - **createdAt:** Timestamp (não nulo).
  - **voteTime:** Timestamp (não nulo).
  - **value:** ENUM (YES, NO, INVALID).
  - **idUser:** UUID (referência ao usuário que votou).
  - **idSchedule:** Long (referência à pauta).
- **Validações:**
  - O cooperado pode votar apenas uma vez por pauta.
  - O voto deve ser registrado dentro do tempo limite definido na pauta.

### Contabilização de Votos e Resultado:

- **Descrição:** Contabilizar os votos ao término do período de votação definido na pauta e disponibilizar o resultado.
- **Campos:**
  - **idSchedule:** Long (referência à pauta).
  - **totalVotes:** Integer (total de votos registrados).
  - **votesInFavor:** Integer (total de votos "Sim").
  - **votesAgainst:** Integer (total de votos "Não").
  - **votesInvalid:** Integer (total de votos "Invalid").
- **Validações:**
  - A contabilização deve ocorrer após o término do tempo de votação definido na pauta.
  - O resultado deve ser armazenado e estar disponível para consulta.

## Requisitos Não Funcionais

### Persistência de Dados:

- **Descrição:** Garantir que as pautas e votos sejam persistidos em um banco de dados e não sejam perdidos com o reinício da aplicação.
- **Tecnologias:** Utilizar banco de dados relacional (e.g., MySQL).

### Desempenho:

- **Descrição:** A aplicação deve ser capaz de lidar com cenários de centenas de milhares de votos de forma performática.
- **Implementação:** Otimizar consultas SQL.

### Segurança:

- **Descrição:** Implementar autenticação e autorização robustas utilizando JWT e Spring Security.
- **Validações:** Proteger todas as rotas, exceto as de autenticação e registro.

#### **Usabilidade:**

- **Descrição:** A interface do usuário deve ser intuitiva e fácil de usar.
- **Implementação:** Utilizar React com MaterialUI para criar uma interface responsiva e amigável.

#### **Confiabilidade:**

- **Descrição:** Garantir a integridade dos dados e robustez da aplicação.
- **Implementação:** Implementar testes unitários e de integração para garantir a qualidade do código.

○