

Desafio de Desenvolvimento - Gerenciamento de Login em Ramais

Introdução

Bem-vindos ao desafio de desenvolvimento! Este projeto tem como objetivo aprimorar suas habilidades em backend e frontend, com um caso de uso baseado em um projeto real. O desafio consiste na construção de um sistema para o gerenciamento de login em ramais, permitindo uma experiência prática no desenvolvimento de APIs, banco de dados e interfaces de usuário.

Observação: Este documento apresenta um escopo de exemplo para guiar o desenvolvimento. Os participantes têm liberdade para implementar as funcionalidades de maneiras diferentes ou mais otimizadas, desde que os requisitos principais sejam atendidos. Criatividade e melhorias são bem-vindas!

Objetivo do Projeto

Desenvolver um sistema que permita o gerenciamento eficiente de login em ramais, composto por:

- **Backend em Java:** responsável por fornecer APIs REST para login e logout de ramais.
- **Banco de Dados:** armazenamento das informações de ramais disponíveis e usuários logados.
- **Frontend:** interface gráfica para exibição dos ramais conectados, busca e configuração.

Requisitos do Projeto

Backend (Java + Banco de Dados)

O backend deve fornecer os seguintes endpoints REST:

1. Solicitar um ramal disponível

- a. **Endpoint:** GET /extensions/available
- b. **Resposta esperada:** { "extension": "101" } (ou erro se não houver ramais disponíveis)

2. Registrar login em um ramal

- a. **Endpoint:** POST /extensions/login
- b. **Corpo da requisição:** { "usuario": "user123", "extension": "101" }
- c. **Ação esperada:** Registrar no banco que o ramal "101" está em uso pelo "user123".

3. Deslogar um ramal

- a. **Endpoint:** DELETE /extensions/logout/{extension}
- b. **Ação esperada:** Remover a relação do ramal com o usuário no banco de dados.

4. Configurar range de Ramais disponíveis

.....

Além disso, o backend deve permitir configurar um intervalo de ramais disponíveis e garantir que um mesmo ramal não seja utilizado por múltiplos usuários simultaneamente.

Banco de Dados

O banco de dados deve conter a tabela `extensions` com os seguintes campos:

- `id` (chave primária)
- `extension_number`
- `logged_user` (NULL quando não estiver em uso)

Frontend (Framework de sua escolha)

O frontend deve oferecer as seguintes funcionalidades:

- **Visualizar os ramais logados:** Exibir uma lista com os ramais em uso e os usuários associados.
- **Buscar ramais:** Permitir busca por número do ramal ou nome do usuário logado.
- **Deslogar um ramal:** Ao clicar em um botão "Deslogar", enviar requisição ao backend para remover a associação do ramal.
- **Configuração do intervalo de ramais:** Interface para definir um range de ramais disponíveis no sistema.

Critérios de Avaliação

Os projetos serão avaliados com base nos seguintes critérios:

- Código bem estruturado e boas práticas.
- Uso adequado de padrões REST.
- Qualidade e organização do frontend.
- Persistência correta dos dados no banco.
- Documentação clara do projeto e instruções de execução.

Entrega

Os participantes devem entregar o projeto em um repositório Git contendo:

- Código-fonte completo do backend e frontend.

- Script SQL para criação das tabelas.
- README com instruções de instalação e uso.

Bônus Principal: Aplicação Desktop

Desenvolver uma aplicação desktop para simular o login de um ramal.

A aplicação deve:

- Solicitar os ramais disponíveis ao backend.
- Permitir ao usuário escolher um ramal e enviar a requisição de login.
- Enviar automaticamente a requisição de logout ao ser fechada.
- Utilizar o usuário do Windows como identificador.

Bônus Extras

- Implementar autenticação de usuários no sistema.
- Criar testes automatizados para o backend.
- Melhorar a interface com feedbacks visuais e experiência do usuário.