

Desafio Técnico - Backend

Gerenciamento de Tokens

Descrição

Este desafio tem como objetivo avaliar suas habilidades em **desenvolvimento backend com Elixir**, incluindo **qualidade do código, resolução de problemas, estruturação de commits, testes, design de software e arquitetura**.

Você deverá construir um sistema de gerenciamento de tokens, respeitando um limite máximo de **100 tokens ativos**, garantindo sua correta alocação e liberação automática conforme as regras estabelecidas.

Regras do Sistema

1. Tokens e Limites

- O sistema deve ter **100 tokens pré-gerados**.
- Cada token deve ser **único**, representado por um **UUID**.
- O sistema pode ter, no máximo, **100 tokens ativos** ao mesmo tempo.
- Um token pode ser **ativo por até 2 minutos**.
- Se um token **atingir 2 minutos de uso**, ele deve ser **liberado automaticamente** e voltar para a lista de tokens disponíveis.
- Se o sistema atingir **100 tokens ativos**, o **token mais antigo em uso deve ser liberado** para manter a disponibilidade.
- **Novos tokens não podem ser criados nem descartados**, apenas alternados entre **disponível** e **ativo**.

2. Uso do Token

- Apenas **um utilizador ativo por vez** pode estar associado a um token.
- Um identificador (**UUID**) deve ser armazenado para rastrear **quem está utilizando** o token.
- Um token pode ter **vários utilizadores no histórico**, mas apenas um ativo por vez.

3. Liberação de Tokens

- Se um token **atingir 2 minutos de uso**, ele deve ser **automaticamente liberado** e voltar para a lista de disponíveis.
 - Se um **novo token for solicitado** e o **limite de 100 tokens ativos já tiver sido atingido**, o sistema deve liberar o **token ativo mais antigo** antes de conceder um novo.
-

Requisitos Técnicos

1. Tecnologias

- **Elixir** (utilizando Phoenix para criação da API)
- **Banco de dados PostgreSQL** (utilizando **Ecto** para ORM e persistência dos tokens)

2. Implementação da Lógica de Liberação Automática

A lógica para **liberar automaticamente tokens após 2 minutos** pode ser implementada de diferentes formas. O candidato pode escolher a melhor abordagem, desde que atenda ao requisito.

Sugestões de implementação:

- Um processo supervisionado mantém a lista de tokens e libera automaticamente após 2 minutos.
- Cada token utilizado agenda um job que libera o token após 2 minutos.
- Um job periódico que verifica e libera tokens expirados.

3. Funcionalidades e Endpoints da API

A aplicação deve expor uma API que permita as seguintes operações:

- **Registrar a utilização de um token**
 - Retorno: ID do token e ID do utilizador
- **Listar todos os tokens disponíveis e utilizados**
 - Retorno: Lista de tokens
- **Consultar um token específico e seu utilizador**
 - Retorno: Informações do token e do utilizador ativo ou histórico de uso
- **Consultar o histórico de utilização de um token específico**
 - Retorno: Lista de utilizadores que já usaram o token
- **Limpar todos os tokens ativos**
 - Retorno: Confirmação da remoção dos tokens ativos

Instruções para Entrega

1. Prazo de 5 dias para realização do teste.
2. Crie um repositório público no **GitHub**.
3. Inclua um **README** explicando:
 - Como rodar o projeto.
 - Como utilizar a API.
 - Quais tecnologias foram utilizadas.

4. Envie o link do repositório para avaliação.
-

Para o dia da entrevista técnica

5. Na data marcada pelo recrutador tenha sua aplicação rodando na sua máquina local para execução dos testes e para nos mostrar os pontos desenvolvidos e possíveis questionamentos.
6. Faremos um code review juntos, você poderá explicar o que você pensou, como arquitetou e como pode evoluir o projeto.