



POSTECH

ARQUITETURA E
DESENVOLVIMENTO JAVA

TECH CHALLENGE

FASE 03

Boas-vindas ao Tech Challenge da fase 3! Agora é hora de colocar em prática os conhecimentos que você está adquirindo para criar uma solução de software robusta e segura, projetada para ambientes dinâmicos e de comunicação assíncrona. Este desafio será fundamental para reforçar seu aprendizado.

Problema

Em um ambiente hospitalar, é essencial contar com sistemas que garantam o agendamento eficaz de consultas, o gerenciamento do histórico de pacientes e o envio de lembretes automáticos para garantir a presença dos pacientes nas consultas. Este sistema deve ser acessível a diferentes tipos de usuários (médicos, enfermeiros e pacientes), com acesso controlado e funcionalidades específicas para cada perfil.

Objetivo

O objetivo é desenvolver um backend simplificado e modular, com foco em segurança e comunicação assíncrona, garantindo que o sistema seja escalável, seguro e que utilize boas práticas de autenticação, autorização e comunicação entre serviços.

Requisitos do Sistema

1. Segurança em Aplicações Java

- **Autenticação com Spring Security:** implementar autenticação básica para garantir que cada tipo de usuário tenha acesso controlado às funcionalidades.
- **Níveis de Acesso:**
 - **Médicos:** podem visualizar e editar o histórico de consultas.
 - **Enfermeiros:** podem registrar consultas e acessar o histórico.
 - **Pacientes:** podem visualizar apenas as suas consultas.

2. Consultas e Histórico do Paciente com GraphQL

- **Implementação de GraphQL:** permitir consultas flexíveis sobre o histórico médico, como listar todos os atendimentos de um paciente ou apenas as futuras.
- **Serviço de Agendamento:** médicos e enfermeiros poderão registrar novas consultas e modificar consultas existentes.

3. Separação em mais de um serviço

- **Serviço de Agendamento:** responsável pela criação e edição das consultas.
- **Serviço de notificações:** envia lembretes automáticos aos pacientes sobre consultas futuras.
- **Serviço de histórico (opcional):** armazena o histórico de consultas e disponibiliza dados via GraphQL.

4. Comunicação Assíncrona com RabbitMQ ou Kafka

- **RabbitMQ ou Kafka:** utilizar uma dessas ferramentas para gerenciar a comunicação assíncrona entre os serviços.
 - O Serviço de agendamento deve enviar uma mensagem ao serviço de notificações quando uma consulta for criada ou editada.
 - O serviço de notificações processa essa mensagem e envia um lembrete ao paciente.

Entregáveis e Fatores de Avaliação da Fase 3

1. Funcionalidade:

- O backend deve atender a todos os requisitos especificados.
- Os endpoints devem estar funcionando conforme descrito e permitir acesso seguro e correto às funcionalidades.

2. Qualidade do Código:

- Utilização adequada das práticas de desenvolvimento com Spring Boot.
- Código organizado, seguindo padrões de nomenclatura, modularização e documentação adequados para facilitar a manutenção.

3. Documentação do Projeto:

- Uma descrição detalhada do projeto, incluindo arquitetura, endpoints da API, e instruções de configuração e execução, deve ser incluída para facilitar a compreensão do funcionamento do sistema.

4. Collections para Teste:

- Incluir collections do Postman (ou similar) para testar os endpoints da API e facilitar a validação das funcionalidades.

5. Repositório de Código:

- Repositório aberto (GitHub, GitLab etc.) com o código-fonte do projeto, onde professores poderão acessar e revisar o trabalho entregue.

Tem alguma dúvida? Participe das nossas lives e acesse nossos grupos de estudo para falar com o(a) professor(a) responsável, que poderá auxiliá-los nesta fase. Alternativamente, você pode nos procurar no Discord para obter suporte adicional. Boa atividade!



POSTECH