

Desafio Técnico: Desenvolvedor Backend Kotlin/Spring Boot

Objetivo

Desenvolver uma aplicação backend simples utilizando Kotlin e Spring Boot, que inclua funcionalidades de comunicação assíncrona, testes automatizados, e práticas de Clean Code. A aplicação deve ser capaz de registrar e consultar informações de usuários, além de enviar notificações assíncronas.

Requisitos do Desafio

1. **Funcionalidades da Aplicação:**

- **Cadastro de Usuários:** A aplicação deve permitir o cadastro de novos usuários com os campos: ID, nome, email e data de criação.
- **Consulta de Usuários:** A aplicação deve permitir a consulta de usuários cadastrados.
- **Notificações Assíncronas:** Ao cadastrar um novo usuário, a aplicação deve enviar uma notificação assíncrona utilizando Kafka ou RabbitMQ.

2. **Requisitos Técnicos:**

- **Kotlin e Spring Boot:** A aplicação deve ser desenvolvida utilizando Kotlin e Spring Boot.
- **Banco de Dados:** Utilize um banco de dados relacional (por exemplo, PostgreSQL) para armazenar os dados dos usuários.
- **SQL:** Crie e mantenha queries SQL para as operações de cadastro e consulta de usuários.
- **Testes Automatizados:** Crie testes automatizados para as funcionalidades de cadastro e consulta de usuários utilizando JavaScript (por exemplo, com Jest).
- **Clean Code:** Aplique princípios de Clean Code e utilize design patterns apropriados.
- **Monitoramento:** Implemente monitoramento básico utilizando DataDog ou uma ferramenta similar.

3. **Entrega:**

- **Código Fonte:** O código fonte deve ser entregue em um repositório Git (por exemplo, GitHub ou GitLab).
- **Documentação:** Inclua um arquivo README.md com instruções claras sobre como configurar e executar a aplicação, bem como como rodar os testes automatizados.
- **Docker:** Opcionalmente, forneça um arquivo Dockerfile e um docker-compose.yml para facilitar a configuração e execução da aplicação.

Critérios de Avaliação

1. **Funcionalidade:**

- A aplicação atende aos requisitos funcionais especificados?
- As operações de cadastro e consulta de usuários funcionam corretamente?

2. **Qualidade do Código:**

- O código segue os princípios de Clean Code?
- Foram utilizados design patterns apropriados?

3. **Testes Automatizados:**

- Existem testes automatizados cobrindo as funcionalidades principais?
- Os testes são claros e bem estruturados?

4. **Comunicação Assíncrona:**

- A notificação assíncrona é enviada corretamente ao cadastrar um novo usuário?
- Foi utilizado Kafka ou RabbitMQ de forma adequada?

5. **Monitoramento:**

- O monitoramento básico foi implementado corretamente?
- A aplicação fornece métricas úteis para monitoramento?

6. **Documentação:**

- A documentação é clara e fornece todas as informações necessárias para configurar e executar a aplicação?
- O uso de Docker (se fornecido) facilita a configuração e execução da aplicação?

Dicas para o Candidato

- ****Planejamento:**** Antes de começar a codificar, planeje a arquitetura da aplicação e como você vai estruturar o código.
- ****Testes:**** Escreva testes automatizados desde o início para garantir que sua aplicação funcione conforme esperado.
- ****Documentação:**** Mantenha a documentação atualizada e clara para facilitar a avaliação do seu trabalho.
- ****Clean Code:**** Preste atenção aos detalhes e siga as melhores práticas de desenvolvimento para escrever um código limpo e manutenível.

Esse desafio técnico deve fornecer uma boa visão das habilidades do candidato em relação aos requisitos da vaga. Boa sorte na avaliação!