

Desafio Backend Java

A Sogo, empresa do Grupo Alfa, é uma GovTech que provê soluções inovadoras para a transformação digital de órgãos governamentais, gerando soluções e processos inteligentes, levando eficiência, transparência e inovação para que a administração pública e privada obtenham os melhores resultados e uma melhor qualidade de serviços oferecidos. Possuímos um ambiente moderno e agradável de trabalho que permite liberar todo seu potencial.

O objetivo deste desafio é entender como você soluciona problemas e responde a desafios temporais, bem como conhecer seu estilo de programação e seus conhecimentos sobre padrões arquiteturais e de projeto de software.

Recomendamos que mantenha o projeto mais legível e manutenível que conseguir sem deixar de mostrar onde suas habilidades que mais se destacam.

Sobre o Projeto-Desafio

Criar um sistema controle de contratos tendo como back-end uma REST API em Java utilizando o banco de dados Postgresql conforme requisitos de usuário abaixo:

História	Informações Adicionais
Como gestor preciso cadastrar pessoas para que eu possa contratá-los como prestadores de serviços.	A entidade pessoa deve ter pelo menos nome, cpf e endereço completo.
Como gestor preciso cadastrar um contrato individual por pessoa para que possa ter um prestador de serviço.	A entidade contrato deve conter o número do contrato, a data de registro, a data de validade e os dados de uma pessoa.
Como gestor eu desejo visualizar uma lista de contratos vencidos para que eu possa imprimi-lo.	Os filtros de listagem podem ser: 1) Vencem em 30 dias, 2) Vencem em 15 dias, 3)Vencem em 7 dias" e 4) Vencem hoje.
Como gestor desejo visualizar um dashboard para entender as estatísticas dos meus contratos	Estatísticas desejadas: 1) número de contratos, 2) número de contratos a vencer, 3) número de tempo médio de prestação de serviços

Requisitos de qualidade e auditoria:

- 1) A aplicação deve utilizar http e a api ser exposta como um RESTful;
- 2) Deve ser possível visualizar no log as operações de criação e atualização;
- 3) A busca por uma entidade que não existe no banco de dados deve refletir em uma mensagem de "{entidade} não encontrado" e devolvida para o cliente como http-status 404;

Bônus

É esperado que desenvolvedores **plenos** realizem ao menos 3 dos itens abaixo relacionados ou 1 (um) item se for **júnior**.

1. Utilização do docker;
2. Implementação de testes;
3. Permitir pesquisa avançada (combinação entre os filtros - operadores “e”, “ou”, “maior que” e “menor que”);
4. Adicionar algum mecanismo de autenticação usando JWT;
5. API com swagger;
6. Utilização do framework Spring.

Requisitos do desafio

- Seguir padrões de codificação e nomeação estabelecida para a linguagem;
- Criar um banco normalizado (**somente pleno**);
- Ter documentação (README com instruções breves de configuração da aplicação backend e uma visão geral do aplicativo implementado);
- Disponibilizar código no github (enviar o link do projeto quando finalizar).

Avaliação ocorrerá sobre

- Atendimento das histórias de usuários;
- Organização do código;
- Aderência aos padrões e convenções;
- Normalização do banco (**somente pleno**);
- Demonstração do nível de conhecimentos sobre frameworks e ferramentas (**somente pleno**).

Prazo de entrega

5 dias (a partir do recebimento deste desafio).

Sucesso e até breve.