**Desafio Backend | Java**

Olá, candidato! Se você chegou até aqui, é porque manifestou interesse em fazer parte do nosso tempo. Preparamos um desafio para entendermos um pouco mais sobre suas habilidades como desenvolvedor backend em Java.

**🚀 Objetivo:**

Desenvolver uma API em Java Spring Boot que implementa um sistema de gerenciamento de pedidos para um restaurante.

**📜 Requisitos:**

**1. Tecnologias:**

* Java 11+
* Bota Primavera
* Maven ou Gradle
* Banco de dados de sua escolha (H2, MySQL, PostgreSQL, etc.)
* Documentação da API com Swagger ou similar

**2. Gerenciamento de Produtos:**

* Criar, listar, atualizar e excluir produtos.
* Cada produto deve ter um nome, preço e uma categoria (bebida, entrada, prato principal, sobremesa).

**3. Gerenciamento de Pedidos:**

* Criar e listar pedidos.
* Cada pedido deve conter um ou mais produtos e o valor total do pedido.
* Implemente a funcionalidade de adicionar produtos a um pedido.

**4. Validações e Erros:**

* Implemente validações para garantir a integridade dos dados.
* Responder com mensagens de erro claras e status HTTP protegido.

**5. Documentação**

* Documente todos os endpoints da API com Swagger ou ferramenta semelhante.

**🥇 Diferenciais:**

* Paginação nos endpoints.
* Registro de logs.
* Dockerização da aplicação.
* Implemente autenticação básica ou JWT para proteger as rotas da API.

**🗳️ Instruções de Submissão:**

1. Faça um fork deste repositório para sua conta pessoal do GitHub.
2. Comprometa-se e empurre suas mudanças para seu fork.
3. Envie um e-mail para [ [pedro.miotti@khipo.com.br](mailto:pedro.miotti@khipo.com.br) ] com o link do repositório.

**🧪 Avaliação:**

* Estrutura do código e organização.
* Uso adequado das ferramentas e tecnologias.
* Implementação dos requisitos e regras de negócio.
* Design e usabilidade.
* Funcionalidades extras (diferenciais).

Boa sorte com o desafio! Estamos ansiosos para ver sua solução.

Etapas de abordagem:

1. **Compreensão dos Requisitos**: Leia cuidadosamente todos os requisitos do desafio para garantir que você entendeu completamente o que precisa ser feito.
2. **Configuração do Ambiente**: Configure seu ambiente de desenvolvimento com Java 11+, Spring Boot, Maven ou Gradle e o banco de dados de sua escolha. Você também precisará configurar o Swagger ou uma ferramenta similar para documentar a API.
3. **Modelagem dos Dados**: Projete o modelo de dados para produtos e pedidos, considerando as informações necessárias para cada um deles, como nome, preço, categoria, etc.
4. **Desenvolvimento da API**: Comece a implementar os endpoints para gerenciamento de produtos e pedidos. Isso inclui criar, listar, atualizar e excluir produtos, além de criar e listar pedidos. Certifique-se de implementar a lógica de negócios necessária para cada operação.
5. **Validações e Erros**: Implemente validações para garantir a integridade dos dados e responda com mensagens de erro claras e status HTTP apropriados quando necessário.
6. **Documentação da API**: Documente todos os endpoints da API usando Swagger ou uma ferramenta similar. Isso ajudará os usuários a entenderem como usar sua API.
7. **Diferenciais**: Considere implementar funcionalidades extras listadas como diferenciais, como paginação nos endpoints, registro de logs, dockerização da aplicação e autenticação básica ou JWT para proteger as rotas da API. Isso pode destacar ainda mais sua solução.
8. **Testes**: Não se esqueça de testar sua aplicação para garantir que ela funcione conforme o esperado. Você pode usar testes unitários, de integração e de ponta a ponta, conforme necessário.
9. **Submissão**: Após concluir o desafio, faça um fork do repositório fornecido, comprometa suas mudanças e envie um e-mail para o endereço especificado com o link do seu repositório.
10. **Avaliação Pessoal**: Antes de enviar sua solução, revise cuidadosamente seu código, verificando a estrutura, organização, uso adequado de ferramentas e tecnologias, implementação dos requisitos e regras de negócio, design e usabilidade, bem como funcionalidades extras implementadas.