

## Desafio DEV Java MG5 [HO]

### Apresentação do desafio

A proposta do desafio é a criação de uma API RESTful, para gerir dados de armazenamento e estoque de um depósito de bebidas. Atualmente o estoque é responsável por armazenar dois tipos de bebidas (alcoólicas e não alcoólicas), contudo, isto pode mudar no futuro. O estoque possui 5 seções e cada seção só pode armazenar um tipo de bebida, isto é, não é possível armazenar ou manter bebidas alcoólicas e não alcoólicas juntas.

Cada seção possui capacidade de armazenamento de 500 litros de bebidas alcoólicas e 400 de não alcoólicas.

### A API deve ser responsável por gerenciar:

- Cadastro e consulta das bebidas armazenadas em cada seção com suas respectivas queries.
- Consulta do volume total no estoque por cada tipo de bebida.
- Consulta dos locais disponíveis de armazenamento de um determinado volume de bebida. (calcular via algoritmo).
- Consulta das seções disponíveis para venda de determinado tipo de bebida (calcular via algoritmo).
- Cadastro de histórico de entrada e saída de bebidas em caso de venda e recebimento.
- Consulta do histórico de entradas e saídas por tipo de bebida e seção. As

seguintes regras devem ser respeitadas no fluxo de cadastro e cálculo: ➤ Uma seção não pode ter dois ou mais tipos diferentes de bebidas ➤ Não há entrada ou saída de estoque sem respectivo registro no histórico. ➤ Registro deve conter horário, tipo, volume, seção e responsável pela entrada. ➤ Uma seção

não pode receber bebidas não alcoólicas se recebeu alcoólicas no

mesmo dia. Ex: Seção 2 começou o dia com 50 litros de bebidas alcoólicas que foram consumidas do estoque, só poderá receber não alcoólicas no dia seguinte.

➤ O endpoint de consulta de histórico de entrada e saída de estoque, deve retornar os resultados ordenados por data e seção, podendo alterar a ordenação via parâmetros.

➤ Para situações de erro, é necessário que a resposta da requisição seja coerente em exibir uma mensagem condizente com o erro.

### **Pontos relevantes:**

➤ Utilize a plataforma Java para o desenvolvimento da solução do desafio ➤ Fique à vontade na escolha de frameworks e bibliotecas que quiser usar no projeto

➤ Fique à vontade com relação a parte de persistência. Uma sugestão é a utilização de banco de dados em memória, como HSQLDB ou até mesmo Mysql (Hoje é o banco de dados que utilizamos na Magis5).

➤ Seria bacana ler sobre suas motivações no Readme da aplicação, bem como nos comentários. Além disso, se puder acrescentar nos comentários, também gostaríamos de saber:

1 - O que achou do teste? Grau de dificuldade, desafios encontrados, etc.

2 - Alteraria algo no teste para analisar alguma outra habilidade?

➤ Entende que testes unitários são necessários para garantia da qualidade do código entregue? Se sim, seria interessante escrever os testes unitários

➤ Para enviar o código do desafio, utilize sua conta pessoal do Bitbucket ou GitLab, e compartilhe conosco de forma privada (o repositório NÃO pode ser público).

### **Endereços de e-mail que devem ter acesso ao repositório:**

➤ vitor.lima@magis5.com.br

➤ jose.fonseca@magis5.com.br

➤ jose.fernando@magis5.com.br

➤ marcelo.silva@magis5.com.br

➤ [ronaldo.araujo@magis5.com.br](mailto:ronaldo.araujo@magis5.com.br)