

# 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

1. Cadastro e consulta das bebidas armazenadas em cada seção com suas respectivas queries.
2. Desejável consulta do volume total no estoque por cada tipo de bebida ou por seção.
3. Desejável consulta dos locais disponíveis de armazenamento de um determinado volume de bebida. (calcular via algoritmo).

## As seguintes regras devem ser respeitadas no fluxo de cadastro e cálculo

1. Uma seção não pode ter dois ou mais tipos diferentes de bebidas (como já fora dito).
2. Registro deve conter pelo menos os seguintes campos obrigatórios: horário, tipo, volume, seção e responsável pela entrada.
3. O endpoint de consulta das bebidas deve retornar os resultados ordenados por data e seção, podendo alterar a ordenação via parâmetros.
4. Para situações de erro, é necessário que a resposta da requisição seja coerente em exibir uma mensagem condizente com o erro.

Pontos relevantes:

1. Deve contém automação de testes(teste unitário).

2. Utilize a plataforma Java com Spring Boot com gradle para o desenvolvimento da solução do desafio.
3. Fique a vontade na escolha de frameworks e bibliotecas que quiser usar no projeto.
4. Fique a vontade com relação a parte de persistência.
  - a. Uma sugestão é a utilização de banco de dados em memória, como HSQLDB.
5. Todas as documentação deve estar em inglês.
6. Seria interessante informações e configurações do projeto no Readme da aplicação.
7. Desejável criação imagem docker.
8. O código fonte deve estar hospedado no Github ou Bitbucket. Enviar o link após a finalização do desafio.
9. Utilização do Swagger para auxílio da documentação dos endpoints rest .