

Objetivo

Seu objetivo é desenvolver uma API RESTful simples para o gerenciamento de baldes de frutas. De modo resumido, baldes podem armazenar frutas, e cada fruta possui preço e instante de expiração. A aplicação deve consistir de um conjunto de endpoints que permita a criação e manipulação de instâncias das entidades **Balde** e **Fruta**, de acordo com as regras abaixo.

Cada **Balde** é criado com um valor de capacidade máxima, referente à quantidade máxima de frutas que podem ser depositadas dentro dele. Após sua criação, um **Balde** pode ser excluído, desde que esteja vazio. Cada instância de **Fruta** é criada com as informações de *nome*, *preço* e *expiração*. O campo *nome* se refere à qualquer *string* simples que o usuário da API deseje utilizar para nomear a fruta, enquanto *preço* se refere ao custo da mesma. O atributo *expiração* representa o instante específico onde uma fruta irá expirar.

Frutas podem ser depositadas e removidas entre os diferentes baldes disponíveis. Após expirar, a fruta deixa de existir automaticamente dentro do contexto da aplicação e, caso estivesse dentro de um balde, o espaço que ela ocupava dentro dele é liberado.

Exemplo: Em determinado momento é criado o balde *B*, e as frutas *A*, *L* e *U* são criadas e depositadas dentro de *B*:

```
B = {capacidade: 4}
A = {nome: "Abacaxi", preço: 4.95, expiração: 30s}
L = {nome: "Laranja", preço: 0.80, expiração: 70s}
U = {nome: "Uva", preço: 3.95, expiração: 130s}
```

Com o passar do tempo, as características do balde *B* mudam de acordo com a expiração dos itens:

- No instante 0, o balde *B* possui 3 frutas, ocupação de 75% e o valor total da soma dos itens depositados dentro dele é igual a 9.70;
- Após 1 minuto, o balde *B* possui 2 frutas, ocupação de 50% e o valor total da soma dos itens depositados é igual a 4.75;
- Após 2 minutos, o balde *B* possui 1 fruta, ocupação de 25% e o valor total da soma dos itens depositados é igual a 3.95;
- Após 3 minutos, o balde *B* possui 0 frutas, ocupação de 0% e o valor total da soma dos itens depositados é igual a 0;

Os seguintes casos de uso devem estar cobertos pela sua implementação:

- Criar/excluir baldes;
- Criar/excluir frutas;
- Depositar/remover frutas de um balde;
- Obter a lista completa de baldes, com o valor total da soma dos preços das frutas em cada um, ordenados de modo decrescente pela sua ocupação (% da capacidade ocupada).

Observações

- Em nenhum momento a capacidade máxima de um balde deve ser violada;

- Um balde deve estar vazio para que possa ser excluído;
- Uma fruta não precisa necessariamente estar dentro de um balde no momento de sua criação, ela pode ser criada fora de um balde e ser depositada em um balde posteriormente. Do mesmo modo, ela pode ser removida de um balde e continuar existindo, até o momento de sua expiração;
- Frutas devem deixar de existir após o seu instante de expiração, independente de estarem em um balde ou não;
- Os dados de baldes/frutas devem ser persistidos (ou seja, caso a aplicação seja reiniciada, os dados devem se manter inalterados). Você pode utilizar o mecanismo de armazenamento que preferir (banco de dados, arquivo, etc);
- Fique à vontade para utilizar quaisquer linguagens/frameworks/bibliotecas que preferir, assim como para pensar o design da API;
- Embora não obrigatório, adição de testes automatizados no teste é bem-vinda, especialmente para os casos de uso que você julgar mais relevantes;
- Inclua instruções de como executar a sua aplicação junto do código (pode ser um arquivo *markdown* ou *.txt* simples).

Entrega

O prazo para conclusão do desafio é de 1 (uma) semana. A aplicação deve ser versionada com Git e GitHub, e o link do repositório deve ser enviado por email (jobs@planne.com.br). O repositório pode ser público ou privado (no segundo caso, forneça permissão de acesso ao usuário [PlanneBot](#)).