Este desafio avaliará diversos aspectos do candidato, tornando-se então a entrega deste obrigatória para conclusão do processo. Caso não consiga entregar dentro do

prazo ou completá-lo por algum motivo, favor nos reportar.

O objetivo desse desafio é demonstrar sua experiência e conhecimento como desenvolvedor, assim saberemos como você pensa e resolve os problemas do cotidiano.

O processo de avaliação levará em consideração os seguintes pontos:

* Organização e clareza;
* Lógica de programação;
* Boas práticas;
* Design Patterns;
* Habilidade para resolução de problemas.

**Regras gerais:**

O candidato terá prazo de **7 dias** para completar a prova.

O projeto deverá ser disponibilizado em um repositório público no GitHub.

**Desafio:**

Um e-commerce de discos de vinil resolveu implementar um programa de fidelidade baseado em cashback\* para aumentar o volume de vendas e conquistar novos clientes. Após algumas reuniões, o time comercial definiu uma tabela de porcentagem de cashback por gênero musical e dia da semana:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Gênero** | **Domingo** | **Segunda** | **Terça** | **Quarta** | **Quinta** | **Sexta** | **Sábado** |
| **POP** | 25% | 7% | 6% | 2% | 10% | 15% | 20% |
| **MPB** | 30% | 5% | 10% | 15% | 20% | 25% | 30% |
| **CLASSIC** | 35% | 3% | 5% | 8% | 13% | 18% | 25% |
| **ROCK** | 40% | 10% | 15% | 15% | 15% | 20% | 40% |

Desenvolva um serviço de back-end que será responsável por efetuar vendas de discos de vinil e calcular o valor de cashback para o cliente respeitando as regras da tabela acima.

Considerar os seguintes critérios de aceite:

* O serviço deverá disponibilizar  APIs REST contendo as seguintes operações:

* Consultar o catálogo de discos de forma paginada, filtrando por gênero e ordenando de forma crescente pelo nome do disco;
* Consultar o disco pelo seu identificador;
* Consultar todas as vendas efetuadas de forma paginada, filtrando pelo range de datas (inicial e final) da venda e ordenando de forma decrescente pela data da venda;
* Consultar uma venda pelo seu identificador;
* Registrar uma nova venda de discos calculando o valor total de cashback considerando a tabela.

* Para alimentar o catálogo de discos, o seu serviço deverá consumir a [API do Spotify](http://email.mg.kenoby.com/c/eJwdjkEOgyAURE-jOwn4QcuChS68B8JHSRWoYpvevthkFjMvmclY1THeOl571VImWcuASRBACSOyBzHyCbpxACHlUHG6L-SJIc5fYuJer8rZWaPQCE64ngsDrOs7ao3EB0Uuab2pNed0VjBU7VRk8Y1bTHiQM8Xs3X_o5tFcO4ass4-h5A_OjU6-uNflzbM5sz5yfagDgw7OrOXLsmu_3fUfc7I9Bg) e retornar os 50 primeiros discos de cada gênero;

* Atribua um identificador único e um preço qualquer para cada disco;

* Cada venda poderá ter 1 ou mais discos selecionados, o cashback deverá ser calculado e armazenado individualmente para cada disco bem como o cashback total da venda.

**Requisitos não funcionais:**

* De preferência deve estar preparado para ser resiliente e tolerante a falhas;
* Fique à vontade para escolher quaisquer frameworks, bibliotecas e ferramentas para desenvolver o seu serviço, bem como princípios e padrões de arquitetura;
* Descreva brevemente no README do projeto os passos para buildar e executar a aplicação.
* Facilite a execução de seu(s) serviço(s) localmente. (Dê preferência para bancos em memória)
* Testes unitários e cobertura.

**Diferenciais:**

* Docker;
* Ferramentade CI (Circle, Travis, etc);
* Eventos
* Database Migration
* SOLID
* Imutabilidade

\* **Cashback**: Dinheiro de volta, é o valor calculado a partir de um determinado percentual para calcular o valor recebido de volta na conta.