

Desafio Developer - Pagway

Nesse desafio você construirá uma versão super simplificada de um Payment Service Provider (PSP) como a Pagway e talvez aprender um pouco mais sobre como funcionam os pagamentos no Brasil.

Contexto

Em sua essência um PSP tem duas funções muito importantes:

1. Permitir que nossos clientes processam transações ("cash-in")
2. Efetuar os pagamentos dos recebíveis para os nossos clientes ("cash-out")

Na Pagway, nós temos duas entidades que representam essas informações:

- `checkouts`: que representam as informações da compra, dados do cartão, valor, etc
- `contas a pagar`: que representam os recebíveis que pagaremos ao cliente

Nota: quando um cliente passa uma transação de crédito, ele normalmente recebe o valor em média apenas 30 dias depois (o que chamamos de D+30), porque é assim que a cadeia financeira (bancos, bandeiras, adquirentes) funciona.

Porém é possível receber esse valor antes dos 30 dias através de um mecanismo chamado "antecipação".

Requisitos

Você deve criar um serviço com os seguintes requisitos:

1. O serviço deve processar transações, recebendo as seguintes informações:
 - Valor da transação
 - Descrição da transação. Ex: `'Smartband XYZ 3.0'`
 - Número do cartão
 - Nome do portador do cartão
 - Data de validade do cartão
 - Código de verificação do cartão (CVV)
2. O serviço deve retornar uma lista das transações já criadas
3. Como o número do cartão é uma informação sensível, o serviço só pode armazenar e retornar os 4 últimos dígitos do cartão.

4. O serviço deve criar os recebíveis do cliente (`contas a pagar`), com as seguintes regras:
 - O payable deve ser criado com `status = pendente` (indicando que o cliente vai receber esse dinheiro no futuro)
 - O payable deve ser criado com a data de pagamento (`payment_date`) = data da criação da transação + 30 dias (D+30).
5. No momento de criação dos payables também deve ser descontado a taxa de processamento (que chamamos de `custo`) do cliente. Ex: se a taxa for 5% e o cliente processar uma transação de R\$100,00, ele só receberá R\$95,00. Considere as seguintes taxas:
 - 5% para transações feitas com um cartão de crédito
6. O serviço deve prover um meio de consulta para que o cliente visualize seu saldo com as seguintes informações:
 - Saldo `disponível` (disponível): tudo que o cliente já recebeu (payables `liquidado`)
 - Saldo `previsto` (a receber): tudo que o cliente tem a receber (payables `pendente`)

Nota: neste desafio, você não precisa se preocupar com parcelamento.

Restrições

1. O serviço deve ser escrito em Node.js
2. O serviço deve armazenar informações em um banco de dados. Você pode escolher o banco que achar melhor. Aqui usamos o PostgreSQL
3. O projeto deve ter um README.md com todas as instruções sobre como executar e testar o projeto e os serviços disponibilizados.
4. O projeto deve conter testes automatizados (Opcional/Diferencial).

Avaliação

1. O desafio deve ser disponibilizado em um repositório do Github;
2. Iremos te avaliar pela arquitetura do serviço, qualidade do código, entendimento das regras de negócio, capricho com o desafio e o quanto preparado esse serviço estaria para ser rodado em produção;
3. Depois que corrigirmos o desafio, te chamamos para conversar, apresentar o desafio e discutir sobre as decisões que você tomou;
4. Achamos que 1 semana é um tempo ok para fazer o desafio, mas sabemos que nem todo mundo tem o mesmo nível de disponibilidade. Portanto, nos avise se precisar de mais tempo, ok?
5. Boa sorte :)