Desafio Will

Olá! Muito obrigada por seu interesse em participar do nosso processo seletivo.

Para desenvolvimento do desafio que será proposto fique a vontade para utilizar a linguagem ou framework onde tem mais domínio. O intuito deste desafio é tentar identificar o seu entendimento para desenvolvimento de uma solução seguindo os requisitos passados e como é sua organização para compartilhamento de código.

Este desafio possui 3 cenários propostos, idealmente esperamos que os 3 sejam atendidos porém entendemos que caso não possua tempo hábil para executar todos os cenários fique à vontade para entregar o desafio com os cenários que conseguir executar. Apenas tente, nos cenários que for atuar, utilizar o máximo de técnica e ferramentas conhecidas para solucionar o problema para conseguirmos fazer uma boa avaliação da sua entrega.

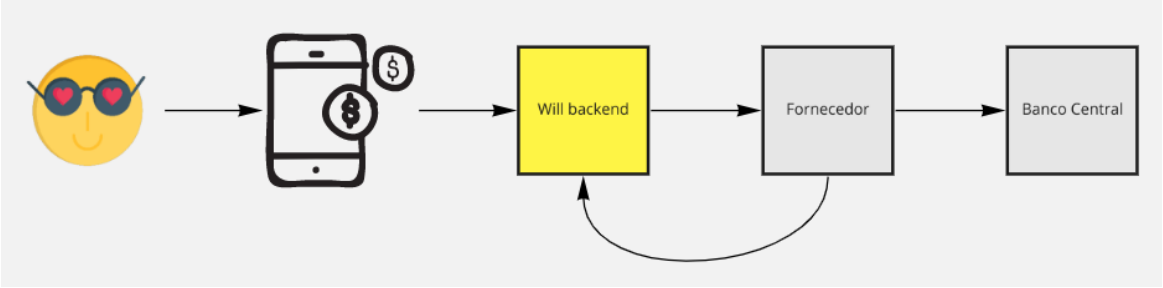
Contextualização

O cliente acessa o APP do Will para realizar o pagamento do boleto, com isso o frontend realiza uma requisição para o Will backend processar o pagamento.

Para esse pagamento ser processado, um dos passos realizados pelo Will backend é a utilização da API de um fornecedor externo responsável acionar o Banco Central e efetivar o pagamento.

Esta API externa apenas disponibiliza informações sobre o boleto e realiza o pagamento de um boleto.

A solução proposta por você, fará parte do ecossistema Will backend e ela não consegue se comunicar diretamente com o Banco Central sem ser através da API do fornecedor.



Desafio Pagamentos

Nosso objetivo é construir uma solução dentro do ecossistema Will backend para processar pagamentos e para o cliente conseguir efetuar o pagamento de seus boletos utilizando o Will Bank. Nosso cliente pode utilizar o serviço de pagamento para pagar seus boletos de **consumo**, exemplo, conta de luz, água, gás e etc. Ou boletos **bancários**, exemplo, boleto gerado por uma compra online.

Sabemos que o time de marketing deseja realizar algumas campanhas que exigirão da solução o processamento de grandes volumes de requisições. Precisaremos também ser capazes de metrificar a campanha, pensando em alguns dados como usuários novos, quantidade de pagamentos concluídos, possíveis casos de falhas e etc.

Por fim, o Will deseja também beneficiar seus clientes com a possibilidade de receber *dindim de volta* através do pagamento de boletos. Assim deve-se estruturar e pensar uma solução que abranja esta funcionalidade.

o que é dindim de volta?
O dindim de volta, também conhecido como cashback, é um benefício exclusivo para clientes Will. Com ele, o cliente recebe de volta na sua conta parte do valor pago nas compras feitas.

Requisitos:

1. O usuário deve ser capaz de realizar o pagamento de um boleto, utilizando código de barras.
2. A aplicação deve persistir as informações de cada pagamento em um banco de dados de sua preferência. Lembre-se apenas de justificar sua escolha, utilize o readme para isso.
3. Espera-se também que a aplicação desenvolvida seja capaz de se comunicar com um serviço externo para conclusão do pagamento. Use este mock para simular a requisição: *https://run.mocky.io/v3/0bca48f0-16db-4726-96a8-d4206306f698*.

   Este é um exemplo de requisição esperada pelo fornecedor:

```
curl --location --request POST
'https://run.mocky.io/v3/ba9815d4-bb7e-440e-b2b2-b1bd832d4581' \

--header 'Content-Type: application/json' \
--data-raw '{
   "billet": "826500000011323116990009002022153320476101001040",
   "amount": "70.24"
}'
```

4. Testes automatizados a fim de verificar que a funcionalidade está de acordo com o pedido.
5. A aplicação deve ser tolerante à falhas, ao passo que o cliente deve conseguir realizar o pagamento mesmo em meio a falhas junto ao fornecedor.

Rotas:

Uma rota deve ser desenvolvida na API com o intuito de permitir que nossos clientes realizem pagamentos de boletos. Faça uma proposta de payload. Lembre-se de documentar a rota criada. No entanto, temos um exemplo logo abaixo:

```
{
   "billet": "826500000011323116990009002022153320476101001040",
   "amount": "70.24"
}
```

Informações sobre a entrega

Github ou Bitbucket - para compartilhamento de código;

Docker - para execução da aplicação localmente;

Critérios para avaliação

A avaliação da sua solução será realizada seguindo os seguintes critérios:

- Documentação;
- *Readme* detalhado;
- Organização de código;
- Argumentação em suas escolhas técnicas;
- Arquitetura da solução;
- Utilização de padrões de desenvolvimento;
- Tratamento de exceções;
- Desacoplamento de componentes;
- Qualidade do software e testes automatizado