

Olá!

Chegamos na etapa do teste prático!

Esta etapa ocorrerá no dia e horário marcados e vamos desenvolver juntos (um dev da Pismo e você) uma solução pré-definida, que deve demorar no máximo 1h.

Para este dia, você precisa trazer uma pequena parte da solução já feita e vamos apenas adicionar novas features à aplicação.

Ah, teremos um computador disponível para você, mas caso prefira pode trazer o seu (assim garante seus atalhos e IDE favoritos).

Na página seguinte descrevemos o que deve **estar pronto** para o dia e horário marcados.

Observações técnicas importantes:

- Desenvolver utilizando **Java**, **Groovy** ou **Go**;
- A solução deve ser publicada no github e deve conter um readme com instruções para execução.

Critérios de avaliação:

1. Manutenibilidade;
2. Simplicidade;
3. Testabilidade.

Bônus:

- Gostamos de docker;
- Fácil execução é um ponto favorável;
- Uma boa documentação facilita muito.

Se tiver qualquer dúvida é só perguntar, ligar, entrar em contato ou enviar sinais de fumaça - estaremos à disposição.

Um abraço e sucesso,
Time Pismo



Rotina de transações

Cada portador de cartão (cliente) possui uma conta com seus dados.

A cada operação realizada pelo cliente uma transação é criada e associada à sua respectiva conta.

Cada transação possui um tipo (compra a vista, compra parcelada, saque ou pagamento), um valor e uma data de criação.

Transações de tipo **compra e saque** são registradas com **valor negativo**, enquanto transações de **pagamento** são registradas com **valor positivo**.

Estrutura de dados

Segue abaixo uma estrutura de dados **sugerida** (*fique a vontade para criar seu próprio modelo*):

Accounts

Account_ID Document_Number

1 12345678900

OperationsTypes

OperationType_ID Description

1 COMPRA A VISTA

2 COMPRA PARCELADA

3 SAQUE

4 PAGAMENTO

Transactions

Transaction_ID Account_ID OperationType_ID Amount EventDate

1 1 1 -50.0 2020-01-01T10:32:07.7199222 2 1 1 -23.5 2020-01-01T10:48:12.2135875 3 1

1 -18.7 2020-01-02T19:01:23.1458543 4 1 4 60.0 2020-01-05T09:34:18.5893223

Na tabela de **Transactions**, a coluna **Amount** guarda o valor da transação e a coluna **EventDate** guarda o momento em que ocorreu a transação.



Endpoints

Desenvolva os endpoints abaixo considerando as regras de negócio mencionadas anteriormente:

POST /accounts *(criação de uma conta)*

Request Body:

```
{  
  "document_number": "12345678900"  
}
```

GET /accounts/:accountId *(consulta de informações de uma conta)*

Response Body:

```
{  
  "account_id": 1,  
  "document_number": "12345678900"  
}
```

POST /transactions *(criação de uma transação)*

Request Body:

```
{  
  "account_id": 1,  
  "operation_type_id": 4,  
  "amount": 123.45  
}
```

