

Senhoras e senhores, esse será nosso projeto a ser desenvolvido dentro dos nossos módulos. Peço encarecidamente que, não saiam feito loucos querendo fazer logo, já colocar banco de dados, colocar frameworks (isso nem pensar ainda), fazer front, enfim, controlem a ansiedade, tudo ao seu tempo, no tempo devido.

Outro ponto, esse é nossa proposta inicial, tudo aqui pode e irá mudar a medida em que vamos avançando no curso, por isso tem/existem coisas aqui que preciso sim de bem mais explicações, então mais uma vez não queiram adivinhar e sair cogitando situações e enviando mensagem com dúvidas, as dúvidas viram na hora certa. Prometo que iremos trabalhar todos os pontos descritos no projeto.

Porém, no entanto, todavia. Quem se sentir confortável de fazer, pode ficar à vontade de seguir, só não vou poder atender/antecipar as coisas em detrimento de talvez alguns que queiram já ir metendo a mão na massa, ok? Fechou! Vamos ao texto.

## **Desafio**

Sobre o desafio, nos iremos trabalhar com um banco fictício, o ACME-PAY, e vamos poder realizar além das operações básicas que um banco fornece aos seus clientes, também sermos capazes de manipular algumas capacidades que iremos explorar para construir nosso banco.

## **Requisitos Funcionais:**

- Ao realizar uma transação, a conta de origem deve passar por algumas validações:
  - Verificar se existe saldo suficiente na conta para que possa ser realizada a transação;
    - Se houver saldo na conta, maior ou igual ao valor da transação, essa operação poderá ser realizada
    - Se não houver saldo na conta, ou seja, a conta estiver zerada, deverá ser emitida uma mensagem informando que a conta não possui saldo
    - Se houver saldo na conta, porém seja menor do que o valor da operação, deverá ser emitida uma mensagem informando que o saldo da conta é insuficiente para realizar a operação
    - Transações entre bancos diferentes, deverá ser cobrada uma taxa de transferência do banco de origem de 1% do valor da operação, sendo esse valor deduzido do saldo da conta, e não do valor transferido.
- Dentro da proposta do sistema, do fluxo em si, teremos apenas dois tipos de usuários, conforme informação no início desse documento, que podem e devem ser cadastrados. São os usuários comuns e usuários lojistas, sendo assim devemos ter as seguintes regras para ambos:
  - Dados cadastrais dos usuários
    - Nome Completo

- CPF,
  - e-mail
  - Senha
- O cadastro do usuário deve ser único, ou seja, o sistema não deve permitir que, dois usuários sejam cadastrados com o mesmo e-mail ou CPF, evitando ambiguidade e inconsistência no sistema
- O processo de transação / transferência entre usuário só poderá ocorrer entre usuários comuns e lojistas, ou seja, lojistas não poderão efetuar transferências, nem para outros lojistas nem para usuários comuns
- Todo processo ocorrido, ou seja, todas as transações realizadas entre os usuários, devem ser armazenadas em uma tabela no banco de dados, as informações geradas pela operação serão:
  - Identificador da transação,
  - Data da transação
  - Conta de origem
  - Conta de destino
  - Valor da transação
- O sistema deverá ser capaz de permitir o cadastro de uma conta, ou seja, os usuários poderão abrir suas contas para poderem transacionar entre si. Porém alguns pontos devem ser considerados para a criação de uma conta no nosso banco. Os dados solicitados para a criação de conta são:
  - Nome do usuário
  - Nome do banco
  - Documento do usuário – CPF/CNPJ
  - E-mail do usuário
    - Após a criação da conta, o usuário receberá como resposta informações sobre a conta aberta por ele:
      - Número da agência
      - Número da conta
      - Tipo da conta
  - O usuário que está em processo de abertura de conta, deverá ter seus dados validados, ou seja, o sistema deverá verificar se há alguma restrição no CPF/CNPJ do usuário.
  - Caso haja alguma restrição para o usuário que está solicitando abertura de conta, o sistema deverá emitir uma mensagem para o usuário, informando-o da não abertura da conta, solicitando que ele verifique restrições em seu nome.
  - Se não houver restrições, o processo de abertura de conta deverá fluir normalmente.
  - Com a conta aberta o usuário poderá usufruir dos seguintes serviços:
    - Realizar depósito
    - Realizar saque
    - Realizar transferência entre contas
    - Realizar consulta de extrato de transações

Antes de finalizar a transferência, deve-se consultar um serviço autorizador externo, use este mock para simular (<https://run.mocky.io/v3/5794d450-d2e2-4412-8131-73d0293ac1cc>).

A operação de transferência deve ser uma transação (ou seja, revertida em qualquer caso de inconsistência) e o dinheiro deve voltar para a carteira do usuário que envia.

No recebimento de pagamento, o usuário ou lojista precisa receber notificação (envio de e-mail, sms) enviada por um serviço de terceiro e eventualmente este serviço pode estar indisponível/instável. Use este mock para simular o envio (<https://run.mocky.io/v3/54dc2cf1-3add-45b5-b5a9-6bf7e7f1f4a6>).

Este serviço deve ser RESTFul.

### Payload

Faça uma **proposta** de payload, se preferir, temos um exemplo aqui:

POST /transaction

```
{
  "value": 100.0,
  "payer": 4,
  "payee": 15
}
```

Requisito: Solicitação de cartão de crédito no banco acmepay

- O cliente que possui conta no nosso banco, poderá fazer solicitação de cartões de crédito nessa instituição. Para que ele possa solicitar o seu cartão ele deverá enviar os dados necessários: CPF, renda.
- Ao receber a solicitação de cartão, o sistema deverá:
  - Enviar uma mensagem ao cliente: “Solicitação de cartão em análise, por favor, aguarde!”
  - Enviar o CPF do cliente para um verificador antifraude, que poderá retornar um dessas duas informações:
    - String - Cliente possui restrições, Boolean false
    - String - Cliente não possui restrições, Boolean true
  - Enviar e-mail e mensagem para o solicitante contendo o aceite da solicitação com uma lista de possíveis cartões que podem ser adquiridos:
    - Bandeira do cartão – Enum (MASTER, VISA, HIPER, ELLO, ACMEPAY)
    - Limite do cartão – BigDecimal
    - Anuidade do cartão - BigDecimal
  - Se a devolutiva for negativa, o cliente só receberá a seguinte mensagem:
    - String - Cliente possui restrições

1. OBS: Se alguém por acaso achar alguma inconsistência no texto, seja de lógica e ou escrita só me avisar que ajusto daqui.

2. OBS: Esse desafio foi em sua essência retirado do PicPay, conforme havia falado, porém sofreu adaptações minhas para atender nossos propósitos dentro das nossas aulas.

## Start Project – Simple Maven Project

### Necessário:

Instalação do Maven - [Link](#) Garanta que o Maven está devidamente instalado em sua máquina, garanta está com isso configurado (quem for levar sua máquina pessoal) os demais deverão seguir esses passos nas máquinas do laboratório, caso o Maven não esteja instalado nas mesmas.

**Criando Projeto** - [Link](#) Ou execute o comando abaixo realizando as seguintes substituições.

1. com.mycompany.app -> br.com.acmepay
2. my-app -> acmepay

```
mvn archetype:generate -DgroupId=com.mycompany.app -DartifactId=my-app  
-DarchetypeArtifactId=maven-archetype-quickstart -DarchetypeVersion=1.  
4 -DinteractiveMode=false
```

### Seu comando deverá ficar dessa forma:

```
mvn archetype:generate -DgroupId=br.com.acmepay -DartifactId=acme-pay -  
DarchetypeArtifactId=maven-archetype-quickstart -DarchetypeVersion=1.4  
-DinteractiveMode=false
```

Após a execução do comando acima, o resultado deverá ser o mesmo apresentado abaixo. SUGIRO fortemente que, usem (porque iremos usar) apenas o terminal, seja no Linux para quem tem e é feliz, ou quem usa o Rwindows, usem o bash, vai dá na mesma.

```
[INFO] Scanning for projects...
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-antrun-plugin/3.1.0/maven-antrun-plugin-3.1.0.pom (9.1 kB at 9.6 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-antrun-plugin/3.1.0/maven-antrun-plugin-3.1.0.pom
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-antrun-plugin/3.1.0/maven-antrun-plugin-3.1.0.jar (41 kB at 299 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-antrun-plugin/3.1.0/maven-antrun-plugin-3.1.0.jar
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-assembly-plugin/3.6.0/maven-assembly-plugin-3.6.0.pom (15 kB at 185 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-assembly-plugin/3.6.0/maven-assembly-plugin-3.6.0.pom
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-assembly-plugin/3.6.0/maven-assembly-plugin-3.6.0.jar (236 kB at 1.2 MB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-assembly-plugin/3.6.0/maven-assembly-plugin-3.6.0.jar
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-dependency-plugin/3.6.0/maven-dependency-plugin-3.6.0.pom (18 kB at 222 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-dependency-plugin/3.6.0/maven-dependency-plugin-3.6.0.pom
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-dependency-plugin/3.6.0/maven-dependency-plugin-3.6.0.jar (190 kB at 1.7 MB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-dependency-plugin/3.6.0/maven-dependency-plugin-3.6.0.jar
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-release-plugin/3.0.1/maven-release-plugin-3.0.1.pom (9.8 kB at 133 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-release-plugin/3.0.1/maven-release-plugin-3.0.1.pom
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-release-plugin/3.0.1/maven-release-plugin-3.0.1.jar (11 kB at 151 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-release-plugin/3.0.1/maven-release-plugin-3.0.1.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-release-plugin/3.0.1/maven-release-plugin-3.0.1.jar (61 kB at 777 kB/s)
[INFO] -----< org.apache.maven:standalone-pom >-----
[INFO] Building Maven Stub Project (No POM) 1
[INFO] -----[ pom ]-----
[INFO]
[INFO] >>> archetype:3.2.1:generate (default-cli) > generate-sources @ standalone-pom >>>
[INFO]
[INFO] <<< archetype:3.2.1:generate (default-cli) < generate-sources @ standalone-pom <<<
[INFO]
[INFO] --- archetype:3.2.1:generate (default-cli) @ standalone-pom ---
[INFO] Generating project in Batch mode
[INFO] -----
[INFO] Using following parameters for creating project from Archetype: maven-archetype-quickstart:1.4
[INFO] -----
[INFO] Parameter: groupId, Value: br.com.acmepay
[INFO] Parameter: artifactId, Value: acme-pay
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Parameter: package, Value: br.com.acmepay
[INFO] Parameter: packageInPathFormat, Value: br/com/acmepay
[INFO] Parameter: package, Value: br.com.acmepay
[INFO] Parameter: groupId, Value: br.com.acmepay
[INFO] Parameter: artifactId, Value: acme-pay
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Project created from Archetype in dir: /Users/cbgames/Projetos/acme-pay
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 7.166 s
[INFO] Finished at: 2024-04-23T14:44:54-03:00
[INFO] -----
[14:44:54] [cost 12.412s] mvn archetype:generate -DgroupId=br.com.acmepay -DartifactId=acme-pay -DarchetypeArtifactId=maven-archetype-quickstart -DarchetypeVersion=1.4 -DinteractiveMode=false
```

Em seguida, entre na pasta do projeto, ou seja, na acme-pay, se estiverem no shell/prompt/terminal com o comando tree instalado, basta rodar “tree” e a árvore de diretórios irá aparecer dessa maneira. Se vc chegou até aqui, parabéns está no nível Jedi 0.

[14:47:47] [~/Projetos/acme-pay] >>> tree

```
.
├── pom.xml
└── src
    ├── main
    │   ├── java
    │   │   └── br
    │   │       └── com
    │   │           └── acmepay
    │   │               └── App.java
    └── test
        ├── java
        │   └── br
        │       └── com
        │           └── acmepay
        │               └── AppTest.java
```

12 directories, 3 files

Próxima etapa é abrir seu projeto no IntelliJ e aguardar o tio Carlos no sábado.

# Modelo Entidade / Relacionamento

## Atenção para os comentários!!

