



## Conteúdo deste Manual

- (1) Objetivo e Funcionalidades Disponíveis
- (2) UML do Banco Dados e Entidades de Modelo
- (3) Regras de Negócio
- (4) Exemplos de Transações

## 1- Objetivo e Funcionalidades Disponíveis

O Sistema possui as principais funcionalidade para efetuar operações da conta criar cliente e associar cliente (pessoa) realizar depósitos e saques de uma bancária e visualizar extrato. Existe também .

Funcionalidades Disponíveis:

**PUT** [/dock/conta/ativar/numeroConta/{idConta}](#)

Ativar uma conta por idConta

**PUT** [/dock/conta/atualizar/{idConta}](#)

Atualizar conta

**POST** [/dock/conta/criar](#)

Criar conta

**PUT** [/dock/conta/depositoValor/{valorCreditar}/numeroConta/{idConta}](#)

Realiza o deposito em uma conta por IdConta

**PUT** [/dock/conta/desativar/numeroConta/{idConta}](#)

Desativar uma conta por idConta

**GET** [/dock/conta/listar](#)

Visualizar a lista de contas

**GET** [/dock/conta/numeroConta/{idConta}](#)

Obter conta por Id - consulta limite de saque e Saldo

**GET** [/dock/conta/saldo/numeroConta/{idConta}](#)

Obter saldo da conta por Id - consulta limite de saque e Saldo

**PUT** [/dock/conta/saqueValor/{valorDebitar}/numeroConta/{idConta}](#)

Realiza o saque da conta por IdConta



**POST** </dock/pessoa/criar>

Cadastrar Cliente

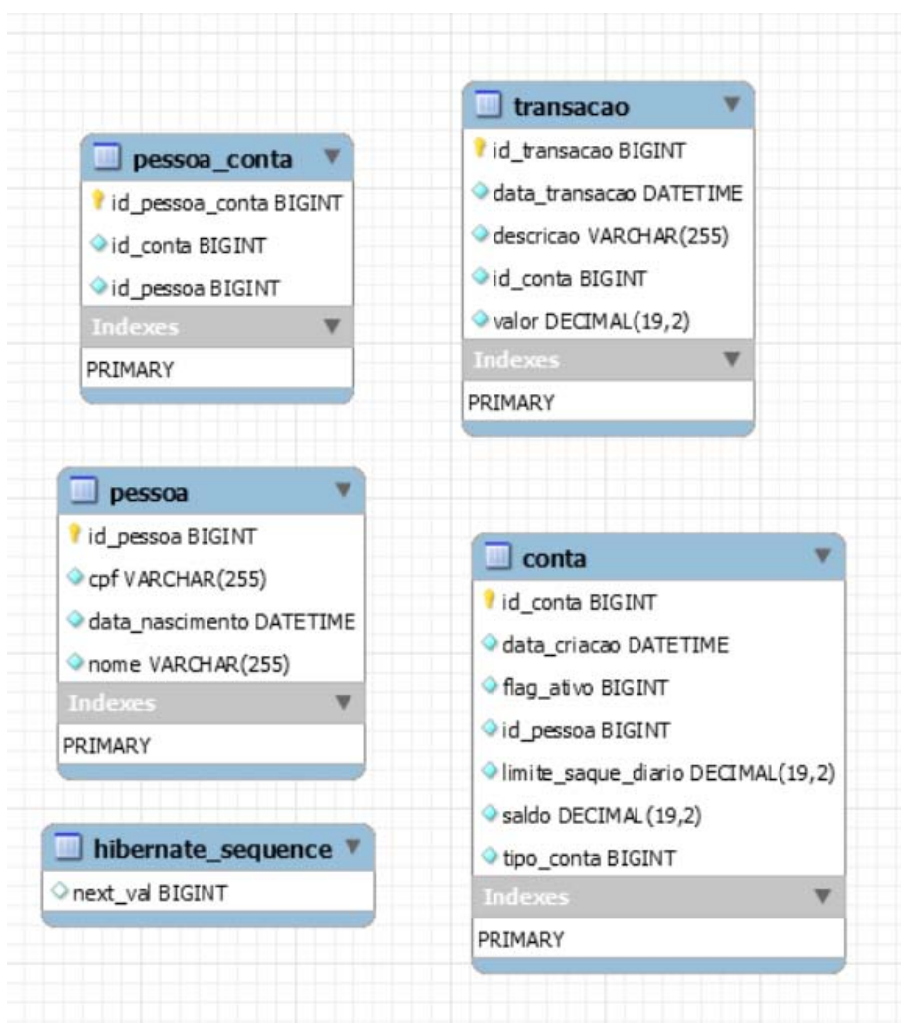
**GET** </dock/pessoa/idPessoa/{idPessoa}>

Obter pessoa por Id

**GET** </dock/transacao/listarByIdConta/{idConta}>

Visualizar a lista de transacoes de uma conta

## 2- UML do Banco de Dados e Entidades do Modelo





```
Conta ▾ {  
  description:      Todos os detalhes da conta bancária  
  
  dataCriacao      string($date-time)  
                   allowEmptyValue: false  
                   Data de criação da conta  
  
  flagAtivo        integer($int64)  
  idConta          integer($int64)  
                   allowEmptyValue: false  
                   ID com autoincrement  
  
  idPessoa         integer($int64)  
                   allowEmptyValue: false  
                   ID do usuario  
  
  limiteSaqueDiario number  
                   allowEmptyValue: false  
                   Limite da conta  
  
  saldo            number  
                   allowEmptyValue: false  
                   Saldo da conta  
  
  tipoConta        integer($int64)  
                   allowEmptyValue: false  
                   Tipo da Conta  
  
}
```

## Entidades de Cadastro

- Conta
- Pessoa
- Transação

```
Pessoa ▾ {  
  description:      Todos os detalhes do cliente da conta  
  
  cpf              string  
                   allowEmptyValue: false  
                   Cpf da pessoa  
  
  dataNascimento   string($date-time)  
                   allowEmptyValue: false  
                   Data de Nascimento da pessoa  
  
  idPessoa         integer($int64)  
                   allowEmptyValue: false  
                   ID do usuario  
  
  nome             string  
                   allowEmptyValue: false  
                   Nome da pessoa  
  
}
```

```
Transacao ▾ {  
  description:      Todos os detalhes da transacao bancária  
  
  dataTransacao    string($date-time)  
                   allowEmptyValue: false  
                   Data da transacao  
  
  descricao        string  
                   allowEmptyValue: false  
                   Descrição da operacao  
  
  idConta          integer($int64)  
                   allowEmptyValue: false  
                   ID do conta  
  
  idTransacao      integer($int64)  
                   allowEmptyValue: false  
                   ID da transacao  
  
  valor            number  
                   allowEmptyValue: false  
                   Valor da transacao  
  
}
```



### 3- Regras de Negócio

Código	Estímulo	Resposta
RN1	Efetuar Saque	Não é possível efetuar saque de conta inativa
RN2	Efetuar Saque	Não é possível efetuar Saque Limite diário insuficiente
RN3	Efetuar Saque	Não é possível efetuar Saque Saldo Insuficiente
RN4	Efetuar Deposito	Não é possível efetuar deposito a conta de destino está inativa
RN5	Cadastrar Pessoa	Não é possível cadastrar esta pessoa. O CPF informado já existe.

### 4- Exemplo de Operações

#### 4.1 Criar Conta

← → ↺ ⌂ localhost:8080/swagger-ui.html#/conta-controller

**POST** /dock/conta/criar Criar conta

Parameters

Name	Description
<b>conta</b> * required (body)	Armazenar uma nova conta

Example Value | Model

```
{  "dataCriacao": "2020-05-21T13:45:45.315",  "flagAtivo": 1,  "idConta": 0,  "idPessoa": 2,  "limiteSaqueDiario": 14910,  "saldo": 10,  "tipoConta": 1}
```

Cancel

Parameter content type  
application/json

Execute

Deve informar os dados no corpo do JSON ficar atento quando ao idPessoa. É necessário o id já está previamente cadastrado.



## Retorno em Caso de Sucesso:

Server response	
Code	Details
200	<div><p>Response body</p><pre>{   "idConta": 25,   "idPessoa": 2,   "saldo": 10,   "limiteSaqueDiario": 14510,   "tipoConta": 1,   "dataCriacao": "2020-05-21T13:45:45.315+0000" }</pre></div> <div><p>Response headers</p><pre>content-type: application/json;charset=UTF-8 date: Thu, 21 May 2020 14:58:26 GMT transfer-encoding: chunked</pre></div>
Responses	

## Retorno em Caso de Erro:

Request URL	
<code>http://localhost:8080/dock/conta/criar</code>	
Server response	
Code	Details
404	<div><p>Error:</p><p>Response body</p><pre>{   "timestamp": "2020-05-21T14:59:06.794+0000",   "message": "Pessoa não encontrada para o id: 222",   "details": "uri=/dock/conta/criar" }</pre></div> <div><p>Response headers</p><pre>content-type: application/json;charset=UTF-8 date: Thu, 21 May 2020 14:59:06 GMT transfer-encoding: chunked</pre></div>

Exemplo Erro: Formato do Saldo incorreto:

# Manual do Sistema

Desafio Dock - Paulo Victor Sobrinho

Criação: 21/05/2020



## Request URL

```
http://localhost:8080/dock/conta/criar
```

## Server response

### Code

400

*Undocumented*

### Details

Error:

### Response headers

```
connection: close
content-length: 0
date: Thu, 21 May 2020 14:59:40 GMT
```