



Desafio

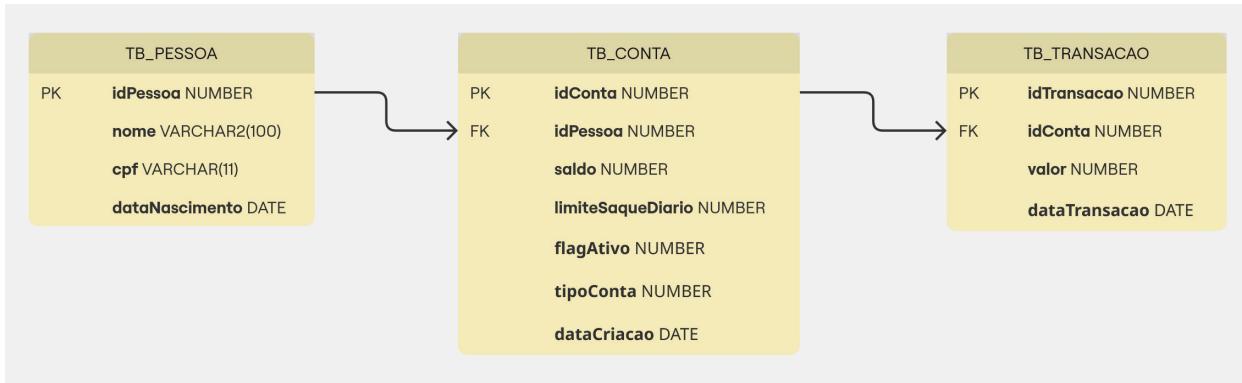
Requerimentos

- Oracle Database 23ai Free Release 23.0.0.0.0 - Version 23.7.0.25.01
- Java 24.0.1
- Node v22.12.0
- npm 10.9.0

Pacotes npm

- **(Material UI)** `npm install @mui/material @emotion/react @emotion/styled`
- **(Material UI Icons)** `npm i @mui/icons-material`

Banco de dados



Entidades

TB_PESSOA

Armazena informações sobre os clientes do banco.

Atributo	Descrição
idPessoa	Identificador único da pessoa (gerado por seq_pessoa)
nome	Nome completo da pessoa
cpf	CPF da pessoa (sem formatação)
dataNascimento	Data de nascimento da pessoa formato DD/MM/YYYY

TB_CONTA

Armazena informações sobre as contas bancárias dos clientes.

Atributo	Descrição
idConta	Identificador único da conta (gerado por seq_conta)
idPessoa	Referência à TB_PESSOA
saldo	Saldo atual da conta
limiteSaqueDiario	Limite diário para saques
flagAtivo	Indica o estado da conta (1 - Ativa 0 - Desativada)

Atributo	Descrição
tipoConta	Tipo da conta (1 - Corrente 2 - Salário 3 - Poupança)
dataCriacao	Data de criação da conta formato DD/MM/YYYY

TB_TRANSACAO

Registra todas as transações financeiras realizadas nas contas.

Atributo	Descrição
idTransacao	Identificador único da transação (gerado por seq_transacao)
idConta	Referência à TB_CONTA
valor	Valor da transação (positivo para depósito, negativo para saques)
dataTransacao	Data da transação formato DD/MM/YYYY

Sequências

`seq_pessoa` : Gera IDs autoincrementais para **TB_PESSOA**

`seq_conta` : Gera IDs autoincrementais para **TB_CONTA**

`seq_transacao` : Gera IDs autoincrementais para **TB_TRANSACAO**

Funcionalidades da API

Pessoa

Rota da api: <http://localhost:8080/api/pessoa/>

▼ Funcionalidade de criar pessoa (POST):

Rota: <http://localhost:8080/api/pessoa/add>

Recebe um JSON com o nome, cpf e data de nascimento da pessoa

- Valida se tem alguma conta com esse cpf

Exemplo de request:

<http://localhost:8080/api/pessoa/add>

JSON:

```
{ "nome": "Isabela",
  "cpf": "11232365590",
  "dataNascimento": "2000-03-20"
}
```

▼ Listar pessoas cadastradas (GET):

Rota: <http://localhost:8080/api/all>

Retorna uma lista JSON com todos os cadastros da tabela pessoa

▼ Buscar pessoa pelo cpf (GET):

Rota: <http://localhost:8080/api/pessoa/{cpf}>

Retorna o registro da pessoa com o cpf correspondente

Exemplo de request:

<http://localhost:8080/api/pessoa/32404384121>

▼ Buscar pessoa pelo id (GET):

Rota: <http://localhost:8080/api/pessoa/{id}>

Retorna o registro da pessoa com o id correspondente

Exemplo de request:

<http://localhost:8080/api/pessoa/1>

Conta

Rota da api: <http://localhost:8080/api/conta/>

▼ Funcionalidade de criar conta (POST):

Rota: <http://localhost:8080/api/conta/add>

Recebe um JSON com o id do titular da conta, saldo inicial, limite de saque diário e tipo da conta

- Valida se existe a pessoa

Exemplo de request:

<http://localhost:8080/api/conta/add>

JSON:

```
{ "pessoa": {  
    "idPessoa": 21  
},  
  "saldo": 1000.50,  
  "limiteSaqueDiario": 500.00,
```

```
"tipoConta": 3  
}
```

▼ Listar contas cadastradas (GET):

Rota: <http://localhost:8080/api/conta/all>

Retorna uma lista JSON com todos os cadastros da tabela conta, traz o cadastro da pessoa vinculada aquela conta

▼ Buscar pessoa pelo id da pessoa (GET):

Rota: <http://localhost:8080/api/conta/pessoa/{id}>

Retorna uma lista contendo todos os dados de todas as contas vinculadas a essa pessoa

Exemplo de request:

<http://localhost:8080/api/conta/pessoa/2>

▼ Buscar pessoa pelo id (GET):

Rota: <http://localhost:8080/api/conta/{id}>

Recebe um id e retorna os dados da conta que tem esse identificador

Exemplo de request:

<http://localhost:8080/api/conta/2>

▼ Consulta de saldo pelo id da conta (GET):

Rota: <http://localhost:8080/api/conta/saldo/{id}>

Recebe um id e retorna o valor no campo saldo da conta

Exemplo de request:

<http://localhost:8080/api/conta/saldo/2>

▼ Bloquear conta pelo id (PUT):

Rota: <http://localhost:8080/api/conta/bloquear/{id}>

Troca o status da conta para 0

Exemplo de request:

<http://localhost:8080/api/conta/bloquear/2>

Transação

Rota da api: <http://localhost:8080/api/transacao/>

▼ Funcionalidade de extrato (GET):

Rota: <http://localhost:8080/api/transacao/extrato/{id}>

Exemplo de request:

<http://localhost:8080/api/transacao/extratoPeriodo/1>

▼ Funcionalidade de extrato por período (GET):

Rota: <http://localhost:8080/api/transacao/extratoPeriodo/{id}>

Recebe como parâmetro duas datas no formato 'yyyy-mmdd'

Exemplo de request:

```
http://localhost:8080/api/transacao/extratoPeriodo/1?inicio=2023-01-10&fim=2023-01-11
```

▼ Funcionalidade de depósito (POST):

Rota: <http://localhost:8080/api/transacao/deposito/{id}>

Recebe um valor BigDecimal correspondente ao valor do depósito

- Valida se a conta está ativa

Exemplo de request:

```
http://localhost:8080/api/transacao/deposito/1
```

JSON: 100

▼ Funcionalidade de saque (POST):

Rota: <http://localhost:8080/api/transacao/saque/{id}>

Recebe um valor BigDecimal correspondente ao valor do saque

- Valida se a conta está ativa
- Valida se o valor do saque não atinge o limite diário
(Soma as transações do dia e compara com o valor do saque)
- Valida se o valor não ultrapassa o que tem na conta

Exemplo de request:

```
http://localhost:8080/api/transacao/saque/1
```

JSON: 100

