

## DESAFIO

### CÉLULA FINANCEIRO E CONTROLADORIA

Criar um projeto JAVA, utilizando o banco de dados ORACLE que deverá versionado no GitHub.

Projeto bem simples para controlar as receitas e as despesas de vários clientes, a empresa (XPTO) precisa controlar as receitas e as despesas de diversos clientes (X, Y, Z) que podem ser PJ ou PF.

Além de manter os clientes (Y, X, Z) é preciso fazer o controle da receita da empresa (XPTO) com relação as movimentações dos clientes (X, Y, Z).

O sistema deve permitir manter (CRUD) os clientes, no cadastro de um novo cliente é preciso ter uma movimentação inicial, que será o ponto de partida para o controle das despesas e receitas, além dos dados dos clientes, como:

Nome;

CPF;

CNPJ;

Endereço;

Contas;

Se é Pessoa Física ou Pessoa Jurídica;

Telefone;

e mais...

**Observação:** Existem alguns dados que não são comuns entre os clientes PJ e PF. Defina a melhor forma normalizar essa situação.

Um cliente (X) pode ter uma ou várias contas bancárias (C1, C2, C3) cadastradas, de onde são feitas as movimentações, todos os pagamentos e recebimentos são feitos por meio dessas contas. Os dados são enviados via integração a cada movimentação:

**Exemplo:**

Instituição Financeira: envia movimentação --->

XPTO: Recebe a movimentação --->

XPTO: identifica o cliente --->

XPTO: efetuar o cadastro da movimentação;

Essa simulação de integração pode ser feita com uma carga de dados no método principal da sua aplicação;

O cálculo da receita empresa (XPTO) vem da quantidade de operações (entradas e saídas).

Exemplo, a cada período de **30 dias**, a partir da data de cadastro o cliente (Y) efetuar:

Até 10 movimentações o cliente irá pagar R\$ 1,00 por movimentação;

De 10 a 20 movimentações o cliente irá pagar R\$ 0,75 por movimentação;

Acima de 20 movimentações o cliente irá pagar R\$ 0,50 por movimentação;

O sistema deve funções para:

**CRUD de clientes;**

Efetuar a manutenção dos clientes, ficar atento para dados que não podem sofrer alterações para manter histórico de movimentação;

**CRUD de endereços;**

Efetuar a manutenção dos endereços dos clientes;

**CRUD de contas;**

Efetuar a manutenção de contas dos clientes, caso exista alguma movimentação associada, não permitir alteração, permitir apenas exclusão lógica;

**Relatório de saldo do cliente X:**

Cliente: X - Cliente desde: DD/MM/YYYY;

Endereço: Rua, número, complemento, bairro, cidade, UF, CEP;

Movimentações de crédito: 00;

Movimentações de débito: 0;

Total de movimentações: 00;

Valor pago pelas movimentações: 00,00

Saldo inicial: 0.000,00

**Saldo atual: 00.000,00**

**Relatório de saldo do cliente X e período:**

Período: DD/MM/YYYY a DD/MM/YYYY

Cliente: X - Cliente desde: DD/MM/YYYY;

Endereço: Rua, número, complemento, bairro, cidade, UF, CEP;

Movimentações de crédito: 00;

Movimentações de débito: 0;

Total de movimentações: 00;

Valor pago pelas movimentações: 00,00

Saldo inicial: 0.000,00

**Saldo atual: 00.000,00**

**Relatório de saldo de todos os clientes;**

Cliente: X - Cliente desde: DD/MM/YYYY – Saldo em DD/MM/YYYY: 0.000,00

Cliente: Y - Cliente desde: DD/MM/YYYY - Saldo em DD/MM/YYYY: 000,00

Cliente: Z - Cliente desde: DD/MM/YYYY - Saldo em DD/MM/YYYY: 00,00

**Relatório de receita da empresa (XPTO) por período:**

Período: DD/MM/YYYY a DD/MM/YYYY

Cliente X - Quantidade de movimentações: 80 - Valor das movimentações: R\$ 0.000,00

Cliente Y - Quantidade de movimentações: 120 - Valor das movimentações: R\$ 00.000,00

Cliente z - Quantidade de movimentações: 100 - Valor das movimentações: R\$ 00.000,00

**Total de receitas: R\$ 00.000,00**

**Seja Criativo!**

Aproveite suas qualidades para desenvolver recursos adicionais caso necessário.

Em caso de dúvidas, tente analisar e resolver da melhor forma possível no seu ponto de vista.

Crie um arquivo README no seu projeto e coloque todas as suas observações;

É obrigatório a criação pelo menos um objeto PL-SQL (procedure, trigger, function) para que possa ser utilizado e chamado a partir do Java;

Criar a classe de testes;

As impressões dos relatórios podem ser feitas utilizando qualquer forma de saída de dados (Report, txt, console da ide...);

Tratar de exceções;

No arquivo README, descreva as boas práticas de desenvolvimento utilizadas, e um tópico sobre os padrões de projetos utilizados no desenvolvimento;

**Observação:**

**Não é obrigatório o envio do projeto completo, deve ser criado pelo menos um CRUD e um Relatório. Iremos analisar apenas a sua forma de desenvolvimento e o seu conhecimento em Java e Oracle. Nos envie o que você conseguiu evoluir com o prazo dado.**

Você tem 96 horas (4 dias) para concluir esta tarefa. Após finalizar enviar o link do projeto no GitHub.