Teste de Software

Sistema Bancário

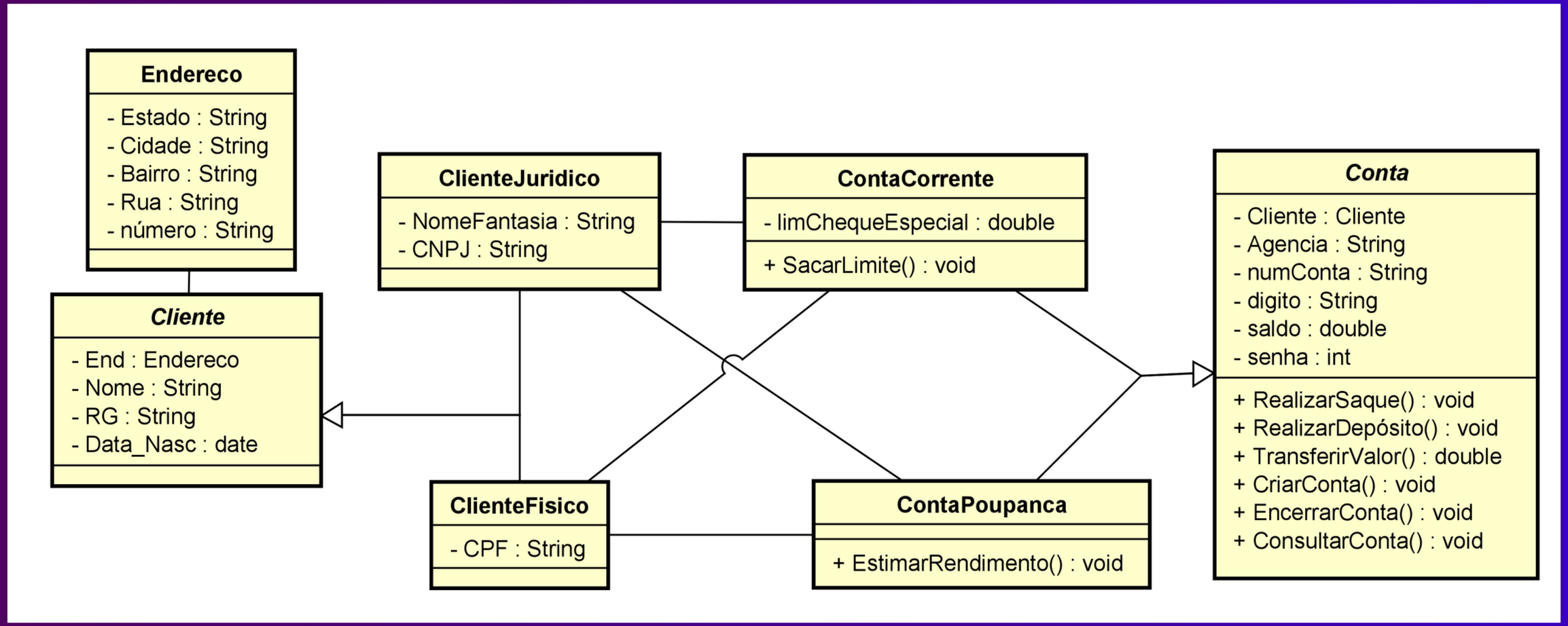
UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
Autores: Alan Almeida. Bruno Victo e Tiago Amaro
Orientador: Alysson Filgueira Milanez

Sobre o Projeto:

Relembrando o objetivo:

O projeto tem como objetivo criar uma aplicação robusta e segura para a gestão de atividades bancárias (utilizando do paradigma de orientação a objetos na linguagem JAVA).

Diagrama de Classes atualizado:



Requisitos atualizados:

Requisito Funcional (RF)	Descrição
RF001 - Criar conta bancária	Clientes podem criar conta (corrente ou poupança), para pessoa física ou jurídica, com dados pessoais.
RF002 - Acessar conta	Login com credenciais para acessar conta bancária.
RF003 - Visualizar informações	Ver saldo, número da conta e histórico.
RF004 - Realizar depósito	Adicionar fundos à conta.
RF005 - Realizar saque	Retirar dinheiro ou transferir para outras contas.
RF006 - Transferir entre contas	Entre contas próprias ou do mesmo banco.
RF007 - Exibir extrato	Ver transações recentes.
RF008 - Ver histórico	Visualizar histórico completo de transações.
RF009 - Cheque especial	Estabelecer um limite e ver informações do cheque especial.
RF010 - Encerrar conta	Solicitar encerramento e transferência de saldo.

Requisitos atualizados:

Requisito Não Funcional (RNF)	Descrição
RNF001 - Validar Idade	O usuário só pode criar uma conta bancária se possuir idade igual ou superior a 18 anos.
RNF002 - Controle de acesso	garantir que apenas pessoas autorizadas possam visualizar ou modificar informações financeiras.
RNF003 - Validar transação	Usuário só realizará uma transação após confirmar sua senha.
RNF004 - Limite de saque	Cada usuário tem direito a 2 saques por acesso.
RNF005 - Restrição cliente jurídico	Uma pessoa jurídica não deve conseguir abrir conta poupança.

Sobre os testes:

Teste de unidade: testar unidades do código fonte do programa como classes, funções, métodos ou até mesmo partes menores do código;

Teste de integração: executar as partes em conjunto, e verificar se o resultado foi aquele esperado;

Teste de validação: verificar se o produto final antende aos requisitos da modelagem;

Teste de aceitação: simular operações de rotina do sistema e conferir se o comportamento ocorreu como o esperado;

Teste de sistema: executar o sistema sob ponto de vista do usuário final, verificando as funcionalidades e requisitos.

Obrigado pela atenção!