GerenciadorDeCon...

+ idGerenciadorContas: string

+ adicionarConta(conta: Co...
+ listarContas(): List<Conta...

+ idCliente: strin...
+ nome: string
+ cpf: string

+ adicionarCon...
+ listarContas(...

HistoricoDeTransacoes

+ idTransacao: string
+ tipoTransacao: int
+ valor: double
+ data: date
+ idConta: string

+ adicionarTransacao(transacao: Transacao): void
+ listarTransacoes(): List<Transacao>

Conta(Abstract)

+ idConta: string
+ saldo: double
+ dataAbertura: date
+ idCliente:string

+ consultarSaldo(): double
+ registrarTransacao(transacao: Transacao): void

Banco

+ idBanco: string
+ nome: string

+ adicionarGerenciadorDeContas(gerenciador: GerenciadorDeContas): void
+ adicionarGerenciadorDeClientes(gerenciador: GerenciadorDeClientes): void
+ configurarCalendario(calendario: Calendario): void

+ idCalendari
+ dataAtual: d

+ atualizarDa
+ obterDataA

...ntas

...g

...onta): void
...a>

GerenciadorDeClientes

+ idGerenciadorClientes: string

+ adicionarCliente(cliente: Cliente): void
+ listarClientes(): List<Cliente>

...Cliente

...g

...nta(conta: Conta): void
...): List<Conta>

GerenciadorDeTransacoes

+ idGerenciadorTransacoes: string

+ gerenciarTransacao(transacao: Transacao): void
+ listarTransacoes(): List<Transacao>

Relatorio

+ idRelatorio: string
+ tipoRelatorio: int
+ dataGeracao: date

+ gerarRelatorio(): void
+ listarRelatorios(): List<Relatorio>

<<interface>>
Transacao

+ executarTransacao(): void

Sacar

+ valor: double

Transferir

+ valor: double
+ contaDestino: string

Deposita...

+ valor: double

Calendario

o: string
date

ata(data: DateTime): void
Atual(): DateTime

ar

Extends

Extends

ContaCorrente
+ limite: double
+ definirLimite(limite: double): void

Poupanca
+ field: type
+ calcularJuros(): double

+ executarTransacao(): void

+ executarTransacao(): void
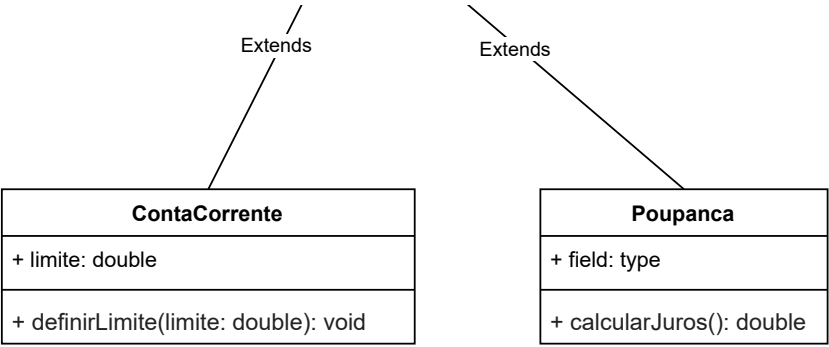
+ executarTransa

+ executarTransacao(): void

+ executarTransacao(): void

+ executarTransa

acao(): void