

## Aula de Laboratório

### Banco com Contas e Usuários

O projeto desenvolvido deverá ser enviado para o e-mail [fmutz@inf.ufes.br](mailto:fmutz@inf.ufes.br) até o fim da aula, mesmo se estiver incompleto. Ele será contabilizado como um exercício feito em aula.

O que veremos:

- Criação de menus para interação com usuários.
- Atributos e métodos estáticos.
- Prática de uso de ArrayLists.

### Atividade

Implementar um sistema que seria usado por um caixa de banco e que permita realizar os casos de uso descritos no diagrama de casos de uso da Figura 1. Para isto, implementaremos as classes descritas no diagrama de classes apresentado na Figura 2. Os itens marcados em azul são novidades em relação à aula anterior.

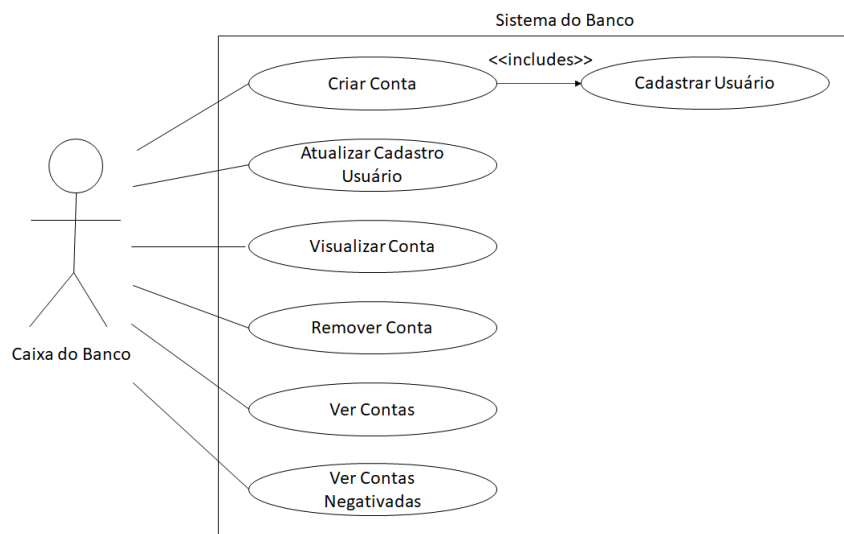


Figure 1. Diagrama de Casos de Uso

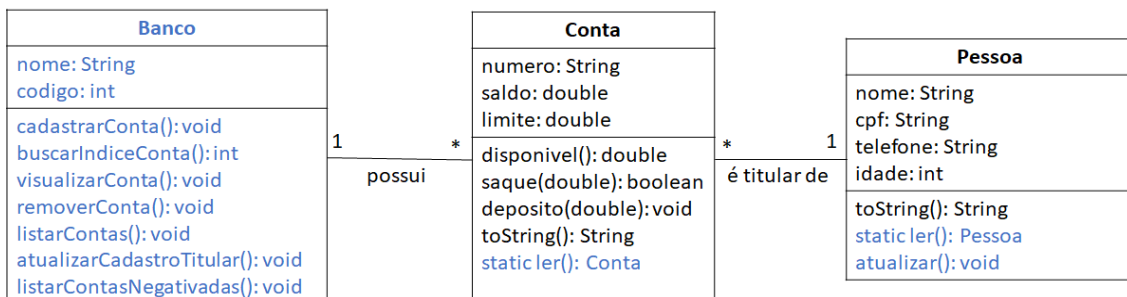


Figure 2. Diagrama de Classes

## 1) Classe Pessoa

Deve implementar os métodos:

- **ler:** é um método estático que lê os dados da pessoa usando um Scanner, usa estes dados para construir um objeto do tipo Pessoa e o retorna.
- **atualizar:** é um método que permite atualizar os atributos nome, telefone e idade (o cpf não pode ser atualizado). Para cada atributo, deve ser informado o valor atual do atributo e o novo valor deve ser solicitado usando um Scanner.

## 2) Classe Conta

Deve implementar o método:

- **ler:** é um método estático que lê o número da conta usando um Scanner e lê os dados do titular usando o método Pessoa.ler() e então usa estes dados para construir uma nova tipo Conta e a retorna.

## 3) Classe Banco

Tem como atributos o nome do banco, o número e um ArrayList<Conta> para armazenar todas as contas cadastradas no banco.

Métodos a serem implementados:

- **Construtor:** recebe o nome e o código do banco e inicializa o ArrayList vazio (nenhuma conta cadastrada).
- **cadaststrarConta:** deve usar o método Conta.ler() para ler os dados de uma nova conta e adicionar a conta na lista.
- **buscarIndiceConta():** deve solicitar que o usuário digite o número de uma conta e retornar o índice da conta no ArrayList cujo número é igual àquele digitado. Se não existir uma conta na lista com aquele número, deve ser retornado o valor -1.
- **visualizarConta():** deve usar o método buscarIndiceConta para recuperar a conta da lista e então imprimi-la na tela.
- **removerConta():** deve usar o método buscarIndiceConta para recuperar a posição da conta da lista e então usar o método remove dos ArrayLists para apagar o item da lista.
- **listarContas():** deve imprimir todas as contas na tela.
- **atualizarCadastroTitular():** deve usar o método buscarIndiceConta para recuperar uma conta da lista e então usar o método atualizar da classe Pessoa para atualizar os dados do titular.
- **listarContasNegativada():** deve imprimir na tela todas as contas com saldo negativo.

## 4) Programa Principal

Deve exibir um menu que permita ao usuário seleciona uma das ações apresentadas no diagrama de casos de uso e uma opção de encerrar o programa. Ao selecionar uma opção, os métodos da classe Banco devem ser usados para realizar o caso de uso. O menu deve exibido repetidamente, até que o usuário escolha encerrar o programa.

- Crie um método estático na classe App chamado menu() que exiba as opções na tela e retorne a opção escolhida pelo usuário. Use o método na função main.