

# APS - Sistema Bancário.

André Zambroni Riedel, Letícia Siguinolfi de Lima, Lucas Guska Lopes

*Fundamentos de Orientação a Objetos*

*Engenharia de Computação, UTFPR Toledo/PR*

12 de dezembro de 2022

## Resumo

O objetivo deste trabalho foi desenvolver uma aplicação onde o usuário consiga interagir com o sistema, salvar o estado do programa em um arquivo, carregar esse arquivo de estado posteriormente e que trate eventuais erros e exceções, alertando o usuário quando alguma ação inválida é realizada. O programa escolhido pela equipe para ser desenvolvido foi um Sistema Bancário, que possui cadastro de contas, titulares, chave PIX, entre outras funcionalidades.

## 1 Desenvolvimento

Primeiramente, criou-se o diagrama de classes para facilitar a compreensão e desenvolvimento do projeto, em seguida, iniciamos a criação do software utilizando a linguagem de programação C++ e seguindo o paradigma de orientação à objetos.

O Sistema Bancário implementado conta com a possibilidade de cadastro de três tipos de contas:

- \* Conta simples;
- \* Conta especial;
- \* Conta poupança.

Se a conta cadastrada for especial, o titular possuirá um limite, onde poderá efetuar saques mesmo que não possua saldo suficiente, diferente das contas simples, que não possuem a possibilidade de limite.

Além disso, o programa possui a opção de transferência utilizando chave PIX (CPF, telefone ou e-mail) que é cadastrada e vinculada a uma determinada conta, permitindo que valores sejam transferidos entre diferentes contas. É importante ressaltar que foram implementados filtros de validação de dados, verificando se os dados inseridos estão corretos. Alguns exemplos são:

- \* Verificação de CPF;

- \* Verificação de data;
- \* Verificação de chave PIX (CPF, telefone ou e-mail).

Um outro ponto importante, é o cálculo de rendimento mensal na conta poupança, onde a taxa de juros pode ser atualizada quando necessário.

Ademais, vale ressaltar que os dados dos titulares e suas respectivas contas são salvos em um arquivo de texto (.txt), fazendo com que os estados possam ser carregados posteriormente.

Na figura 1, temos o diagrama de classes do programa para melhor entendimento do sistema:

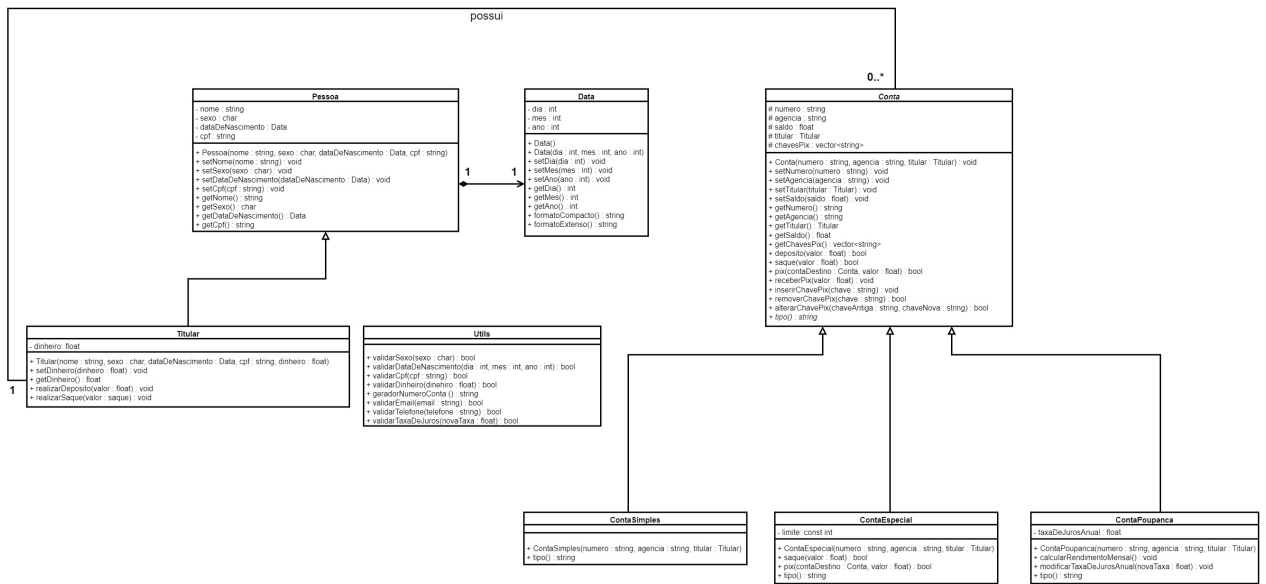


Figura 1: Diagrama de Classes.