



**INSTITUTO FEDERAL**  
Rio Grande do Sul  
Campus Erechim

---

# **AULA 03 - PILARES DA POO - ENCAPSULAMENTO**

Linguagem de Programação II  
Dário L. Beutler  
[dario.beutler@erechim.ifrs.edu.br](mailto:dario.beutler@erechim.ifrs.edu.br)



# Objetivo da aula

- Entender os principais conceitos do encapsulamento.

## Programa da aula

- 1) Os três pilares da POO
- 2) Encapsulamento

# 1) Os três pilares da POO

## ■ Importante!!!!

- Para adotar uma estratégia de OO para o desenvolvimento de software, você não pode simplesmente sair programando.
- Você deve ter um planejamento cuidadoso e uma **BOA BASE NAS IMPORTANTES TEORIAS EXISTENTES POR TRÁS DA Programação Orientada a Objetos (POO)**.
- O domínio destas teorias não virá rapidamente, assim como qualquer habilidade, seus conhecimentos de POO melhorarão e aumentarão apenas com o estudo e prática.

# 1) Os três pilares da POO

- **Os três conceitos que precisam estar presentes para que uma linguagem seja considerada realmente OO:**

**1) Encapsulamento**

**2) Herança**

**3) Polimorfismo**



## 2) Encapsulamento

- Em vez de pensar um programa como uma única entidade grande e monolítica, o encapsulamento permite que você divida em várias partes menores e independentes.
- Cada parte possui implementação e realiza seu trabalho independente de outras partes.
- O encapsulamento mantém essa independência, ocultando os detalhes internos (a implementação de cada parte), através de uma interface externa.

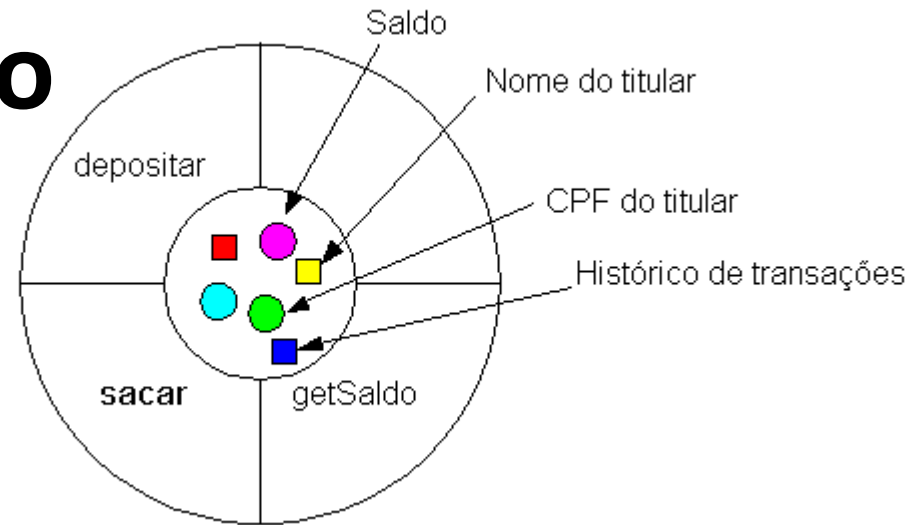
## 2) Encapsulamento

### CONCEITO:

- É a característica da OO de ocultar partes independentes da implementação. O encapsulamento permite que você construa partes ocultas da implementação do software, que atinjam uma funcionalidade e ocultam os detalhes da implementação do mundo exterior.

**Observação:** Ao invés de falar “software encapsulado” , usam-se como sinônimos do mesmo os termos **módulo, componente ou *bean***.

## 2) Encapsulamento



### TERMOS IMPORTANTES:

- **Interface** – é a lista dos serviços fornecidos por um componente. A interface é um contrato com o mundo exterior, que define exatamente o que uma entidade externa pode fazer com o objeto. É o painel de controle do objeto (**o nome dos métodos**). A interface é semelhante a API (Interface de Programa Aplicativo) para um objeto.
- **Implementação** – define como um componente realmente fornece um serviço. Define os detalhes internos do componente (**é o código dentro de cada método**).

## 2) Encapsulamento

- Um software encapsulado é como uma caixa preta.
- Você sabe o que a caixa faz, pois conhece a sua interface externa, mas não sabe como ela é por dentro.
- Você simplesmente envia mensagens para a caixa preta.
- Você não se preocupa com o que acontece dentro da caixa, você só se preocupa em solicitar o serviço e esperar que a caixa o forneça.



## 2) Encapsulamento

A maioria das linguagens OO suporta três níveis de acesso:

- **Público** (public) – garante o acesso a todos os objetos
- **Protegido** (protected) – garante o acesso à instância, ou seja, para aquele objeto, e para todas as subclasses
- **Privado** (private) – garante o acesso apenas para a instância, ou seja para aquele objeto

# Exercícios

1) Crie na classe Conta um método chamado *calculaRendimento* que não recebe parâmetro algum e devolve o valor do saldo multiplicado por 0.1.

Use o método criado para imprimir na tela o valor do rendimento de cada um dos objetos Conta instanciados.