



TECNÓLOGO EM ANÁLISE E DESENVOLVIMENTO DE  
SISTEMAS

EMILLY DA SILVA DA LUZ

# **Programação Orientada a Objeto II.**

**Professor:** Silvio Souza e Diego Madureira

JOINVILLE – SC  
2025

# Sumário

|                                |   |
|--------------------------------|---|
| Encapsulamento .....           | 2 |
| Public.....                    | 2 |
| Private.....                   | 2 |
| Protected.....                 | 3 |
| Métodos Getters e Setters..... | 3 |
| Getters .....                  | 3 |
| Setters.....                   | 3 |
| \$This .....                   | 3 |
| REFERÊNCIA.....                | 4 |

## **Encapsulamento**

Ao falarmos sobre Encapsulamento, estamos falando sobre modificadores de acesso, onde protegemos e efetuamos um controle de acesso dos nossos atributos ou métodos.

### **Public**

Define que o atributo ou método é público, ou seja, pode ser acessado por qualquer classe ou parte do código, por exemplo nosso objeto.

### **Private**

Define que o atributo ou método é privado, diferente do público, aqui apenas a classe na qual ele foi definido pode acessá-lo, restringindo o acesso das outras classes ou objetos.

### **Protected**

Define que o atributo ou método é protegido, ou seja, apenas a classe na qual ele foi definido ou as subclasses (herança) pode acessá-lo, restringindo o acesso das outras partes do código que não possuem ligação com a classe principal.

## **Métodos Getters e Setters**

Esses métodos servem para acessar e manipular nossos atributos que estão protegidos ou privados, tendo como uma vantagem a segurança do acesso aos dados e da organização do código. Podemos utilizá-los quando precisamos definir o terá ou não a necessidade de privar ou proteger, ou seja, quando vamos mexer com dados sensíveis que precisam ter um cuidado ao seu acesso.

### **Getters**

Este tipo de método nos permite retornar um valor de um atributo encapsulado informado pelo objeto.

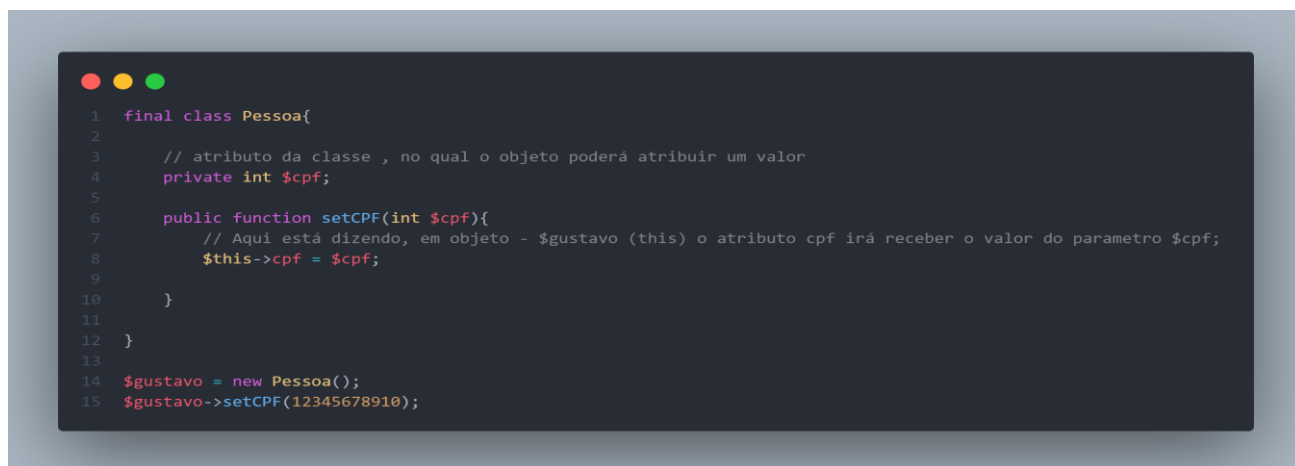
### **Setters**

Este outro tipo de método anda em conjunto na maioria das vezes com o Getters, com eles podemos manipular um valor e atribuí-los a um atributo encapsulado

## \$This

O uso da palavra-chave **this** em POO, é para criar uma auto-referência do objeto no método que está sendo executado na classe, permitindo acessar os atributos e métodos da instância atual.

Figura 1 – Uso do \$this



(Autor - 2025)

## REFERÊNCIA

DEVMedia. *PHP – Modificadores de acesso*. Disponível em: <https://www.devmedia.com.br/php-modificadores-de-acesso/38438>. Acesso em: 29 set. 2025.

GOMES, Emanuel. **Programação Orientada a Objetos com PHP: Métodos Getters e Setters**. Medium, 2023. Disponível em: <https://medium.com/emanuelg-blog/programa%C3%A7%C3%A3o-orientada-a-objetos-com-php-m%C3%A9todos-getters-e-setters-352790a1f60>. Acesso em: 12 out. 2025.

DIO. **Getters e Setters: Entenda a importância e como usar de forma básica esses métodos**. DIO.me, 2023. Disponível em: <https://www.dio.me/articles/getters-e-setters-entenda-a-importancia-e-como-usar-de-forma-basica-esses-metodos-20417db375b6>. Acesso em: 12 out. 2025.

TECNOPODE. **POO F-11**. Tecnopode Blog, 2018. Disponível em: <https://tecnopode.blogspot.com/2018/02/poo-f-11.html>. Acesso em: 13 out. 2025.

SILVA, João et al. **AULA 3: PROGRAMACAO ORIENTADA A OBJETOS – UNIDADE II**. Scribd, 2023. Disponível em:

<https://pt.scribd.com/document/717742469/AULA-3-PROGRAMACAO-ORIENTADA-A-OBJETOS-UNIDADE-II>. Acesso em: 13 out. 2025.