



TECNÓLOGO EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS

EMILLY DA SILVA DA LUZ

INTERFACE

Professor: Silvio Souza e Diego Madureira

JOINVILLE – SC
2025

Sumário

O que é uma Interface em POO?	2
Diferença entre classe Abstrata x Interface	5
Por que usar interfaces pode melhorar a manutenção e a extensibilidade do código? ..	5
REFERÊNCIA	6

O que é uma Interface em POO?

É uma forma de contrato que define o conjunto de métodos que são obrigados a ser utilizados nas classes que foram implementadas. Sua vantagem é a organização da estrutura de uma classe e padronização de nomenclatura ou assinatura dos métodos, e com a utilização da interface se cria uma regra onde necessita ser seguida.

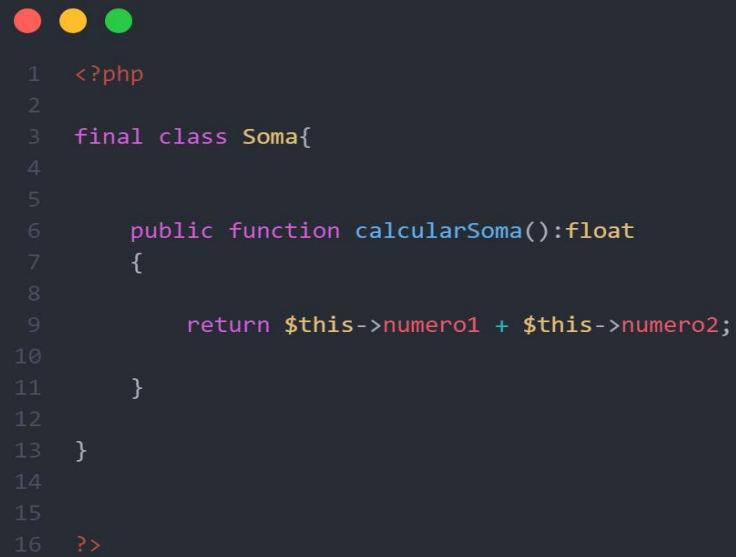
Figura 1 – Classe Dividir sem implementar uma Interface.

A screenshot of a code editor with a dark background and light-colored text. The code is in PHP and defines a final class named 'Dividir'. The class has a single public method named 'calcularDividir' which returns a float value calculated as '\$this->numero1 / \$this->numero2'. The code is numbered from 1 to 15 on the left side of the editor.

```
1  <?php
2
3  final class Dividir{
4
5
6      public function calcularDividir():float
7      {
8
9          return $this->numero1 / $this->numero2;
10
11     }
12
13 }
14
15 ?>
```

(Autor – 2025)

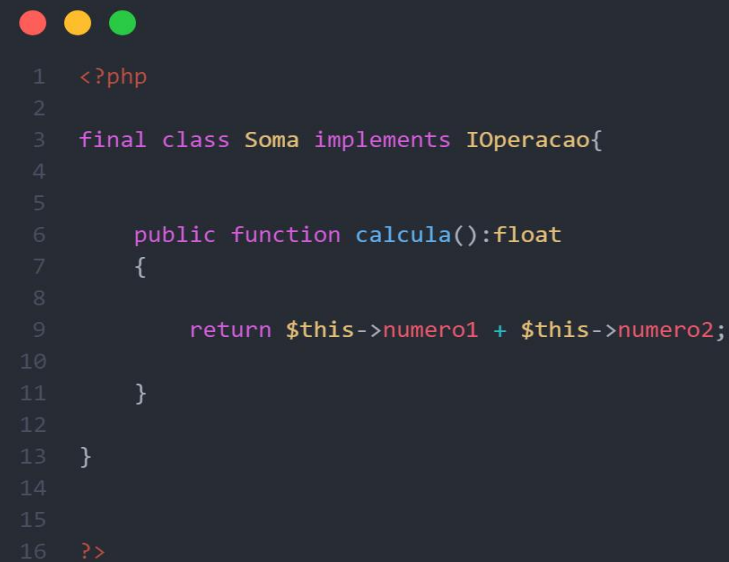
Figura 2 – Classe Soma sem a implementação da Interface



```
1  <?php
2
3  final class Soma{
4
5
6      public function calcularSoma():float
7      {
8
9          return $this->numero1 + $this->numero2;
10
11      }
12
13  }
14
15
16  ?>
```

(Autor – 2025)

Figura 3 – Classe Soma com uma Interface implementada



```
1  <?php
2
3  final class Soma implements IOperacao{
4
5
6      public function calcula():float
7      {
8
9          return $this->numero1 + $this->numero2;
10
11      }
12
13  }
14
15
16  ?>
```

(Autor – 2025)

Figura 4 – Classe Dividir com Interface implementada



```
1  <?php
2
3  final class Dividir implements IOperacao{
4
5
6
7      public function calcula():float
8      {
9
10         return $this->numero1 / $this->numero2;
11     }
12 }
13
14 }
15
16 ?>
```

(Autor – 2025)

Diferença entre classe Abstrata x Interface

A classe Abstrata é uma classe incompleta, representando um conceito bem geral do que queremos, como se fosse um molde para outras classes (uma Classe Mãe), podendo ter atributos e métodos dentro dela, além de que ela não possui instância direta, apenas indiretas por meio de suas filhas, esse conceito de classe está ligado diretamente a **Herança**.

A interface tem como objetivo criar uma padronização / regra de como serão os comportamentos/assinatura (métodos) utilizados pela classe, sendo um “contrato” com regras que devem ser obrigatoriamente seguidas, visando que todas as classes sigam a mesma estrutura.

Por que usar interfaces pode melhorar a manutenção e a extensibilidade do código?

Pois a interface nos permite ter um controle maior sobre as estruturas em nossas classes, já que ao implementar, elas devem seguir uma padronização imposta pela Interface no qual criamos, e além da manutenção ser direta e visível seu funcionamento.

REFERÊNCIA

EV. **POO – Programação Orientada a Objetos**. Disponível em:
<https://lms.ev.org.br/mpls/Custom/Cds/COURSES/3236-POO/curso/pdfs/ebook.pdf>.
Acesso em: 16 nov. 2025.

EDUCAÇÃO PERNAMBUCO. **Caderno INFO (Programação Web Orientada a Objetos 2018)**. Disponível em:
[https://sisacad.educacao.pe.gov.br/bibliotecavirtual/bibliotecavirtual/texto/Caderno_INFO\(Programacao_Web_Orientada_a_Objeto_2018\).pdf](https://sisacad.educacao.pe.gov.br/bibliotecavirtual/bibliotecavirtual/texto/Caderno_INFO(Programacao_Web_Orientada_a_Objeto_2018).pdf). Acesso em: 17 nov. 2025.

MEC / REDE ETEC. **Programação Orientada a Objetos**. Disponível em:
https://redeetec.mec.gov.br/images/stories/pdf/eixo_infor_comun/tec_inf/081112_programa_obj.pdf. Acesso em: 17 nov. 2025.