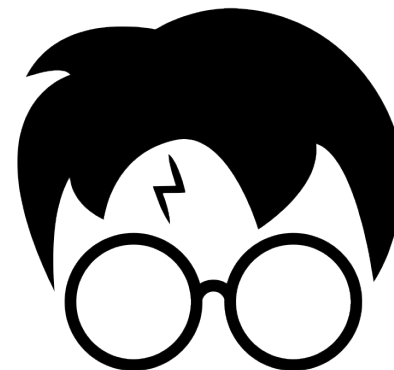


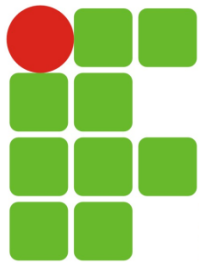


INSTITUTO FEDERAL
Farroupilha
Campus Avançado Uruguaiana



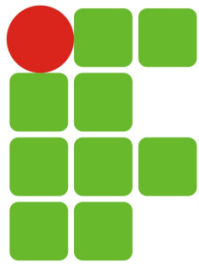
[POO] Métodos Mágicos

Prof. Leandro Martins Dallanora
leandro.dallanora@iffarroupilha.edu.br



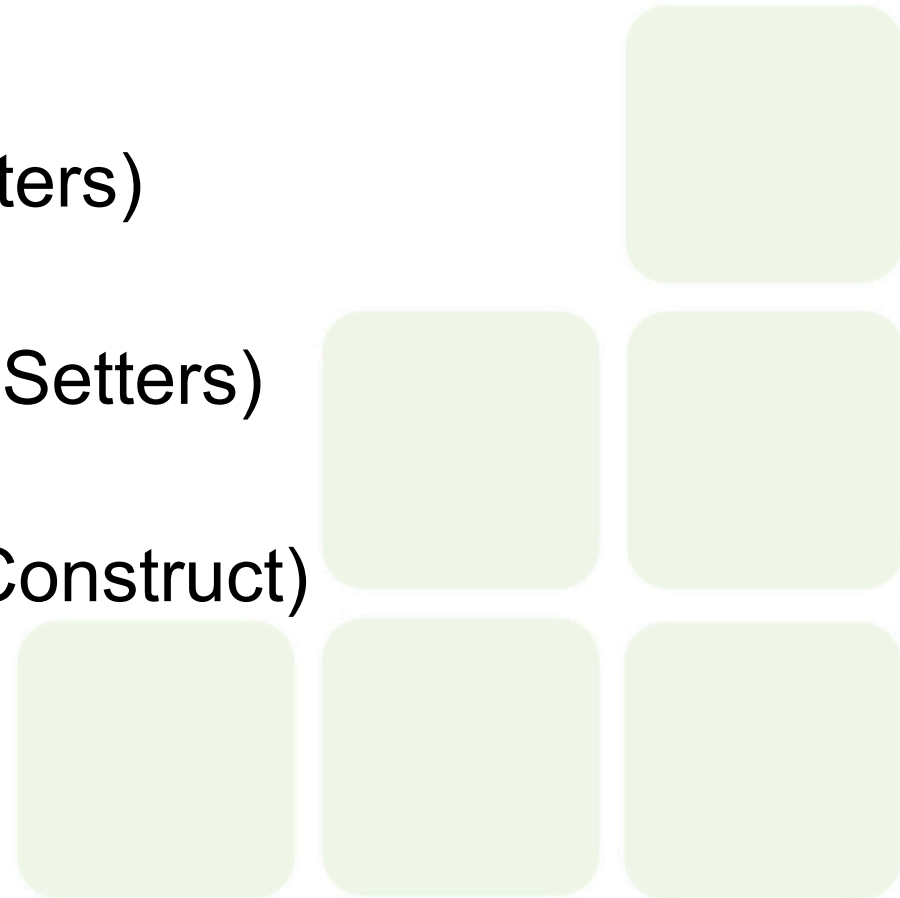
Introdução

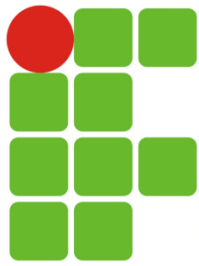
- Para facilitar o uso dos objetos, o PHP provê uma série de, **métodos mágicos**, ou **métodos especiais**, chamados quando certas ações comuns ocorrem com objetos.
- Isso permite executar várias tarefas úteis com certa facilidade.
- Como por exemplo, o controle da nossa classe, permitindo acesso e manipulação dos nossos atributos.
- Ou ainda realizar pré-configurações no momento de instanciar a classe.



Introdução

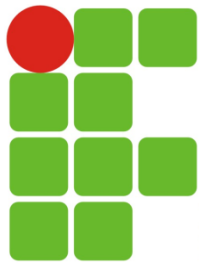
- Vamos aprender como funcionam os seguintes métodos mágicos:
- Métodos **Acessores** (Getters)
- Métodos **Modificadores** (Setters)
- Métodos **Construtores** (Construct)





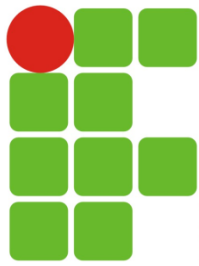
Métodos Acessores (Getters)

- O método **get**, “pegar”, serve para pegarmos informações de variáveis (atributos) da classe que são definidas como '*private*', porém esses métodos serão definidos como '*public*'.
- Mas por que criar métodos para acessar variáveis, se podemos acessar elas diretamente?
- Simples: **questão de segurança.**



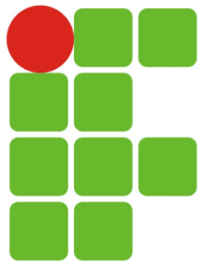
Métodos Acessores (Getters)

- As variáveis 'private' só podem ser acessadas de dentro da Classe.
- É como se elas fossem invisíveis foram do escopo da classe/objeto.
- Assim, evitamos que outros métodos, classes ou hackers tenham acesso aos dados de determinada classe, que muitas vezes podem ser dados privados, como é caso de aplicações para empresas e bancos.



Métodos Acessores (Getters)

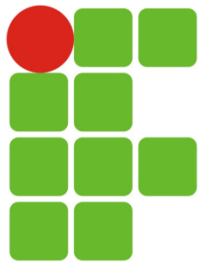
- Como tratar, então, essas variáveis?
- Aí que entra a questão dos métodos.
- Vamos permitir o acesso a essas variáveis, claro (senão não haveria sentido em usar esses atributos).
- Porém, vamos ter um total controle sobre essas variáveis através dos métodos.
- Usamos **get** para obter informações. Esse tipo de método sempre retorna um valor.



Métodos Acessores (Getters)

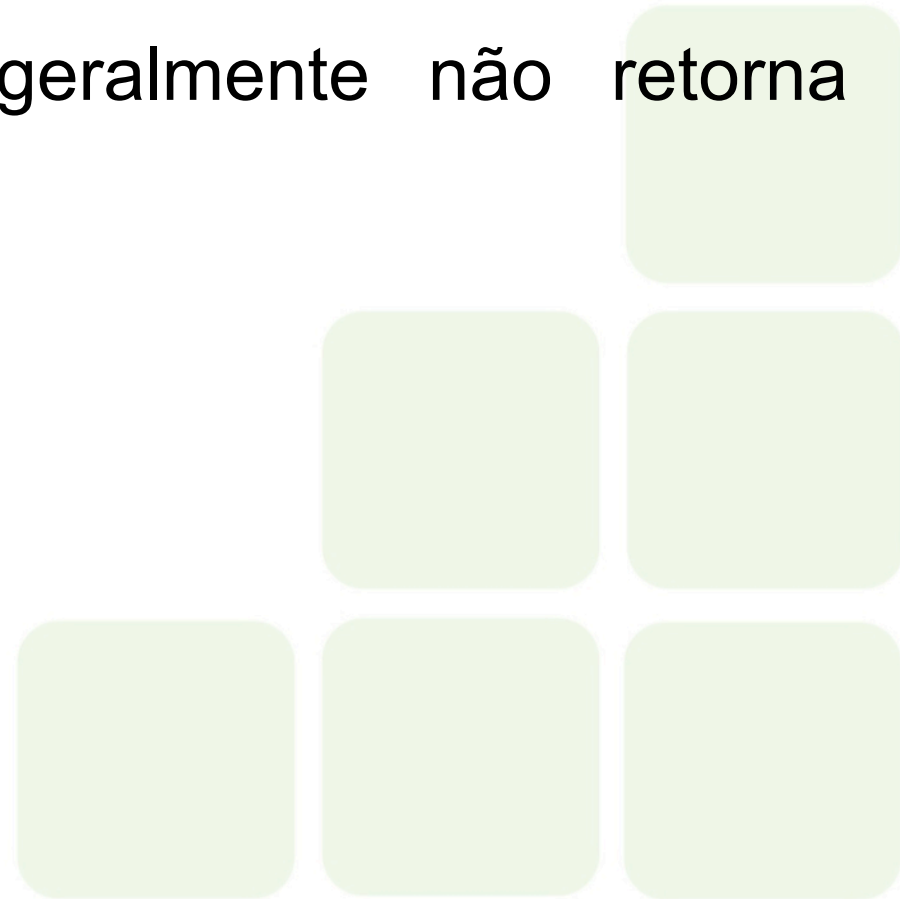
- Usando o exemplo da classe Caneta, visto anteriormente, temos o seguinte código:

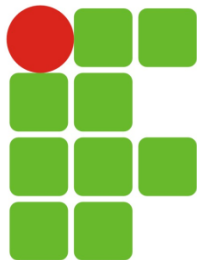
```
class Caneta {  
  
    public $modelo;  
    private $ponta;  
  
    public function getModelo(){  
        return $this->modelo;  
    }  
  
    public function getPonta(){  
        return $this->ponta;  
    }  
  
}
```



Métodos Modificadores (Setters)

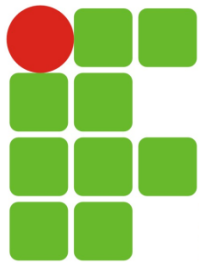
- Já o método **set**, “**editar**”, serve para que possamos colocar um valor em nossos atributos.
- Esse tipo de método geralmente não retorna valores.
- Vamos ver um exemplo...





Métodos Modificadores (Setters)

```
class Caneta {  
  
    public $modelo;  
    private $ponta;  
  
    public function setModelo($m){  
        return $this->modelo = $m;  
    }  
  
    public function setPonta($p){  
        return $this->ponta = $p;  
    }  
  
    // Métodos getters...  
  
}
```



Métodos Modificadores (Setters)

- Com os Getters e Setters definidos, podemos testar:

```
<?php
    include("Caneta.php");

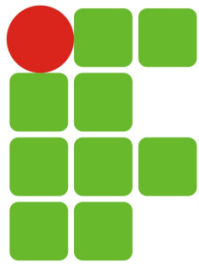
    $c1 = new Caneta;

    $c1->setModelo("BIC");
    $c1->setPonta(0.5);

    echo "Eu tenho uma caneta modelo ".$c1->getModelo();
    echo " de ponta ".$c1->getPonta();
```

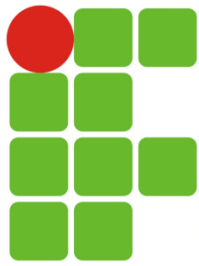
- E ver o resultado:

Eu tenho uma caneta modelo BIC de ponta 0.5



Métodos Construtores (Construct)

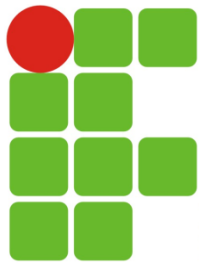
- O método **construct**, “**construtor**”, serve para que a gente faça pré-configurações no momento da instanciação da classe.
- Para criar-mos um método construtor, este deverá ter como nome a palavra **__construct()**, com dois underlines antes do nome.
- Vejamos um exemplo...



Métodos Construtores (Construct)

A classe Caneta com o construtor definido...

```
class Caneta {  
  
    private $modelo;  
    private $cor;  
    private $ponta;  
    private $tampada;  
  
    public function __construct($m, $c, $p){  
        $this->modelo = $m;  
        $this->cor = $c;  
        $this->ponta = $p;  
        $this->tampar();  
    }  
  
    public function tampar(){  
        $this->tampada = true;  
    }  
}
```



Métodos Construtores (Construct)

- Sua instanciação...

```
<?php
    include("Caneta.php");

    $c1 = new Caneta("BIC", "Azul", 0.5);
    print_r($c1);
```

- E o resultado:

```
Caneta Object
(
    [modelo:Caneta:private] => BIC
    [cor:Caneta:private] => Azul
    [ponta:Caneta:private] => 0.5
    [tampada:Caneta:private] => 1
)
```