

# Linguagem de Programação PHP

Estrutura de Linguagens

Uerj/2019-2

Integrantes: Dennis Ribeiro Paiva

Vinicius Sathler

# Origens e Influências

Inicialmente PHP se referia a "Personal Home Page Tools", um conjunto de scripts CGI escritos em linguagem C desenvolvidos por Rasmus Lerdorf em 1994 com o propósito de acompanhar visitas em seu currículo online. Em 1995, Rasmus liberou seu código fonte ao público.

Com o aumento da demanda por aplicações web simples e dinâmicas e com o apoio de usuários na correção de bugs o PHP se expandiu, estando presente em centenas de milhões de domínios atualmente.

A linguagem PHP foi desenvolvida propositalmente para ser parecida com a linguagem C, tornando fácil a adoção de desenvolvedores habituados com essa linguagem e outras similares, foi influenciada também pelas linguagens Perl e Java e possui sintaxe HTML embutida.

# Classificação

PHP é uma linguagem interpretada, imperativa, funcional, orientada a objeto, procedural e de tipagem fraca e dinâmica.

Apesar de possuir aplicações em outras áreas, a linguagem PHP é utilizada principalmente em desenvolvimento server-side web, sendo executado junto à um servidor web.

```
<?php
    $x = '3'; // x é uma string de um caractere
    $y = 2; // y é um inteiro
    $exemplo = $x/$y; //exemplo é um numero real 1.5
    $exemplo = $x*$y; //exemplo é um numero inteiro 6
    $exemplo = $x.$y; //exemplo é uma string "32"

?>
```

# Arrays

Arrays em PHP funcionam como um mapa que relaciona chaves e valores, onde as chaves podem ser tanto valores inteiros quanto strings, e os valores podem ser de qualquer tipo suportado em PHP, inclusive objetos e métodos

Internamente um array em PHP funciona como uma hashtable com encadeamento exterior.

```
$array = [27,  
1.8,  
"chave" => "valor",  
9 => [1,2,3,"quatro"]  
];
```

```
echo $array[1] //27  
echo $array['chave'] //valor  
echo $array[9][3] //quatro
```

# PHP - HTML

PHP possui html embutido, ou seja, é capaz de escrever páginas HTML completas, assim como pode receber informações da página HTML, como por exemplo, dados de um formulário.

```
<?php
$user = [ "name" => "Batman", "id" =>
"Bruce Wayne" ];
?>
<h1>Tela Inicial</h1>
<h3><?="bem vindo
".substr($user['id'], 0, 4)."...
quero dizer ". $user['name'] ?><h3>
```

## Tela Inicial

bem vindo Bruc... quero dizer Batman

# Reflexão

Reflexão é a capacidade de um programa de observar e modificar sua estrutura e comportamento.

Em PHP e outras linguagens orientadas a objeto que dão suporte a reflexão não só é possível inspecionar objetos e métodos em tempo de execução, como também é possível alterá-los e até mesmo burlar restrições de visibilidade do programador à campos e métodos do objeto em questão. Isso é especialmente útil quando o programador não tem acesso à implementação da classe.

# Reflexão - Acessando campos

```
class Pessoa{  
    private:  
        int RG;  
        string nome;  
  
    public:  
        Pessoa(int rg, string nome){  
            this->RG = rg;  
            this->nome = nome;  
        }  
  
        string getNome(){  
            return this->nome;  
        }  
  
        int getRG(){  
            return this->RG;  
        }  
}
```

Em c++, uma linguagem orientada a objeto sem suporte a reflexão, é necessário chamar um método de classe para ter acesso aos campos protegidos da classe.

# Reflexão - Acessando campos

```
class Pessoa{  
    private $RG;  
    private $nome;  
  
    function __construct($rg, $nome){  
        $this->RG = $rg;  
        $this->nome = $nome;  
    }  
}
```

```
$fieldNome = $reflectionClass->getProperty("nome");  
$fieldRG = $reflectionClass->getProperty("RG");  
//mudando a permissao de acesso aos campos escolhidos  
$fieldNome->setAccessible(true);  
$fieldRG->setAccessible(true);  
echo $fieldNome->getValue($usuario).'  
'.$fieldRG->getValue($usuario); //Bruce Wayne 123456789
```

```
//o objeto reflectionClass armazena informações sobre a classe  
Pessoa  
$reflectionClass = new ReflectionClass("Pessoa");  
//criando uma nova instancia de Pessoa a partir de reflectionClass  
$usuario = $reflectionClass->newInstance(123456789, "Bruce Wayne");
```



# Reflexão - Acessando métodos

```
...  
private function whoHeIs(){  
    $herois = ["Clark Kent" => "Super Man", "Bruce Wayne"  
=> "Batman", "Diana Prince" => "Mulher-Maravilha"];  
    if(isset($herois[$this->nome])){  
        return $herois[$this->nome];  
    }  
    else{  
        return "Pessoa comum";  
    }  
}  
...  
  
//method recebe um objeto reflectionMethod  
$method = $reflectionClass->getMethod("whoHeIs");  
  
//mudando acesso ao metodo  
$method->setAccessible(true);  
  
//chamando o metodo para o objeto antes instanciado  
echo $method->invoke($usuario); //Batman
```

# Traits

Em PHP traits são pedaços de código que definem propriedades e métodos que podem ser utilizados por diferentes classes.

Recomenda-se o uso de traits em PHP quando um conjunto de classes sem relação hierárquica possuem um conjunto comum de campos e métodos, já que o PHP não suporta herança múltipla.

# Traits

```
class Passaro extends Animal{  
    public void Voar(){  
        System.out.println(this.getClass().getSimpleName()+" levantou voo.");  
    }  
}
```

```
class Aviao extends Veiculo{  
    public void Voar(){  
        System.out.println(this.getClass().getSimpleName()+" levantou voo.");  
    }  
}
```

```
class Pipa extends Brinquedo{  
    public void Voar(){  
        System.out.println(this.getClass().getSimpleName()+" levantou voo.");  
    }  
}
```

# Traits

```
trait Voo{  
    public function Voar(){  
        echo get_class($this). " levantou voo<br>";  
    }  
}
```

```
class Passaro extends Animal{  
    use Voo;  
}
```

```
class Aviao extends Veiculo{  
    use Voo;  
}
```

```
class Pipa extends Brinquedo{  
    use Voo;  
}
```

# Referências

Nikic - Understanding PHP internal array implementation.

<https://nikic.github.io/2012/03/28/Understanding-PHPs-internal-array-implementation.html>

Acesso em 07 Ago. 2019.

PHP.net - Reflection Book. [https://www.php.net/manual/pt\\_BR/book.reflection.php](https://www.php.net/manual/pt_BR/book.reflection.php)

Acesso em: 03 Ago. 2019

Wikipedia - Reflection Programming.

[https://en.wikipedia.org/wiki/Reflection\\_\(computer\\_programming\)](https://en.wikipedia.org/wiki/Reflection_(computer_programming))

Acesso em: 03 Ago. 2019

Especializati Blog - Traits no PHP. <https://blog.especializati.com.br/traits-no-php/>

Acesso em 06 Ago. 2019