



**INSTITUTO  
FEDERAL  
Pernambuco**

---

**Campus Garanhuns**

**DICIPLINA:  
TÓPICOS AVANÇADOS  
EM PROGRAMAÇÃO  
ALUNO:  
DANIEL DA SILVA CALADO**



**PARADIGMA FUNCIONAL**

# INTRODUÇÃO

- Embora o PHP seja uma linguagem de programação orientada a objetos, é possível usar o paradigma funcional em PHP para escrever código mais conciso, modular e fácil de entender.

## FUNÇÕES DE ALTA ORDEM

- Funções de alta ordem, são funções que recebem outras funções como argumentos ou retornam funções como resultado.

### array\_map()

```
1 <?php
2 function multiplicar($num) {
3     return $num * $num;
4 }
5 $numeros = [5, 13, 77, 654];
6 $resultado =
7     array_map("multiplicar", $numeros);
8 // Imprime o array
9 print_r($resultado)
10 ?>
```

### Saida:

```
Array
(
    [0] => 25
    [1] => 169
    [2] => 5929
    [3] => 427716
)
```

## COMPOSIÇÕES DE FUNÇÕES

- É uma técnica, em que você combina funções simples para criar funções mais complexas.

```
<?php
function adicionar($valor1, $valor2) {
    return $valor1 + $valor2;
}
function dobrar($valor) {
    return $valor * 2;
}
$adicionar_e_dobrar = function ($valor1, $valor2) {
    return dobrar(adicionar($valor1, $valor2));
};
echo $adicionar_e_dobrar(6, 40); // retorno: 92
?>
```

## IMUTABILIDADE

- A imutabilidade refere-se à característica de que uma vez que um valor é criado, ele não pode ser alterado.
- Não tem como definirmos um objeto essencialmente 100% imutável no PHP

```
25  <?php
26  class Pessoa {
27      protected $nome;
28      protected $idade;
29      public function getNome() {return $this->nome;}
30      public function getIdade() {return $this->idade;}
31  }
32  $pessoa = new Pessoa("João", 25);
33  $pessoa->nome = "Maria";
34  echo $pessoa->getNome();
35  ?>
```

## FUNÇÃO PURA

- Uma função pura é uma função que não possui efeitos colaterais e sempre retorna o mesmo resultado quando é chamada com os mesmos argumentos.

```
37  <?php
38  function calcular_area_circulo($raio) {
39      return pi() * pow($raio, 2);
40  }
41  function ordenar_numeros_pura($numeros) {
42      $copia = $numeros;
43      sort($copia);
44      return $copia;
45  }
46  ?>
```

## RECURSÃO

- A recursão é uma técnica utilizada em programação para que uma função chame a si mesma repetidamente até atingir uma condição de parada.

```
48  <?php
49  function somaArray(array $arr): int {
50      if (count($arr) == 0) {
51          return 0;
52      } else {
53          return $arr[0] + somaArray(array_slice($arr, 1));
54      }
55  }
56  ?>
```

## CURRYING

- Currying é uma técnica de programação funcional em que uma função com vários argumentos é transformada em uma série de funções, cada uma com um argumento.

```
58  <?php
59  function adicionar($a, $b) {
60      return $a + $b;
61  }
62
63  $addFive = function($c) use ($adicionar) {
64      return add(5, $c);
65  };
66
67  echo $addFive(10);
68  ?>
```