



Prof. Taciano Balardin  
[taciano@ulbra.edu.br](mailto:taciano@ulbra.edu.br)

**E-MAIL:**

**[taciano@ulbra.edu.br](mailto:taciano@ulbra.edu.br)**



**SITE DA DISCIPLINA:**

**<http://www.taciano.pro.br/>**

**SENHA:**

**[@lpweb](#)**



PHP Orientado a Objeto

# LINGUAGEM DE PROGRAMAÇÃO WEB

## AULA 18

# Classes

- Classes são conjuntos de variáveis e funções que descrevem um objeto com suas características e ações.
- Classe apartamento (propriedades)
  - numero\_quartos
  - numero\_banheiros
  - preco\_aluguel
  - preco\_venda
  - tamanho
  - localizacao

# Classes

```
class nome_classe
{
    <variáveis>
    <procedimentos>
}
```

- *var* é usado para criar variáveis dentro de uma classe. Também pode ser usado a variável interna do PHP, *\$this* que serve para referenciar o próprio objeto;
- Para criar um novo objeto, deve ser atribuído a uma variável que representará o objeto a instrução *new* seguida do nome da classe;
- Já para acessar valores de uma array dentro de uma classe, devemos usar *->* seguido do nome do array e chave associada.

```
class Loja
```

```
{  
    var $itens;  
    function adiciona($codigo,$quantidade){  
        if(isset($this->itens[$codigo]))  
            $this->itens[$codigo] += $quantidade;  
        else  
            $this->itens[$codigo] = $quantidade;  
    }  
    function remove($codigo,$quantidade){  
        if($this->itens[$codigo] > $quantidade){  
            $this->itens[$codigo] -= $quantidade;  
        }  
        else{  
            return false;  
        }  
    }  
}
```

# Classe Loja

- Possui como variável o array itens que foi declarada usando a instrução var;
- Também possui duas funções para acrescentar ou remover itens do estoque;
- O objeto é criado quando fazemos a atribuição de new Loja à \$estoque;
- Depois de criado o objeto, \$estoque já acessa as variáveis e funções disponíveis na classe.

# Classe Loja

- Note que o símbolo `->` sempre deve ser usado quando trabalhando com objetos para acessar funções e variáveis.

```
$estoque->adiciona ("bermuda", 2) ;
```

```
$estoque->adiciona ("camiseta", 3) ;
```

```
echo "A loja possui {$estoque->itens ['bermuda']}  
bermudas." ;
```

```
echo "A loja possui {$estoque->itens ['camisetas']}  
camisetas" ;
```



# Private e Protected

- Utilizadas para criar métodos e variáveis privadas e protegidas dentro de uma classe (PHP 5);
- Uma variável ou método privados só podem ser acessados de dentro da classe que foram declarados;
- Uma variável ou método protegidos poderão ser acessados também pelas subclasses da classe onde foram declarados.

```
class Classe1 {  
    private $var1 = "Olá, var1!  <br/>";  
    protected $var2 = "Olá, var2!  <br/>";  
    protected $var3 = "Olá, var3!  <br/>";  
    function bomDia() {  
        print "Classe1 {$this->var1}<br/>";  
        print "Classe1 {$this->var2}<br/>";  
        print "Classe1 {$this->var3}<br/>";  
    }  
}  
  
class Classe2 extends Classe1 {  
    function bomDia() {  
        Classe1::bomDia();  
        print "Classe2 {$this->var1}<br/>";  
        print "Classe2 {$this->var2}<br/>";  
        print "Classe2 {$this->var3}<br/>";  
    }  
}
```