

Programação Orientada a objetos em php

Prof. Me. Hélio Esperidião

O que é POO?

- A abordagem de desenvolvimento de software chamada Programação Orientada a Objetos (POO) se baseia na estruturação do software em Classes, que por meio de objetos interagem entre si para realizar uma determinada tarefa.





Abstração

- A capacidade de se concentrar nos aspectos fundamentais de um contexto, desconsiderando características secundárias ou menos importantes, é conhecida como abstração.
- Na modelagem orientada a objetos, uma classe representa uma abstração de uma entidade presente no domínio do sistema de software. Algumas classes comuns incluem Pedido, Produto e Cliente.

Atributos

- Atributos em Programação Orientada a Objetos (POO) são as características ou propriedades de um objeto.
- Eles representam o estado de um objeto e podem ser de diferentes tipos de dados, como números, strings, booleanos, objetos, entre outros.
- Por exemplo, se tivermos uma classe "Pessoa" com os atributos "nome", "idade" e "sexo", podemos criar vários objetos dessa classe, cada um com valores diferentes para esses atributos.
- O objeto "João", por exemplo, pode ter o atributo "nome" definido como "João", "idade" como "30" e "sexo" como "masculino", enquanto
- O objeto "Maria" pode ter o atributo "nome" definido como "Maria", "idade" como "25" e "sexo" como "feminino"

Métodos

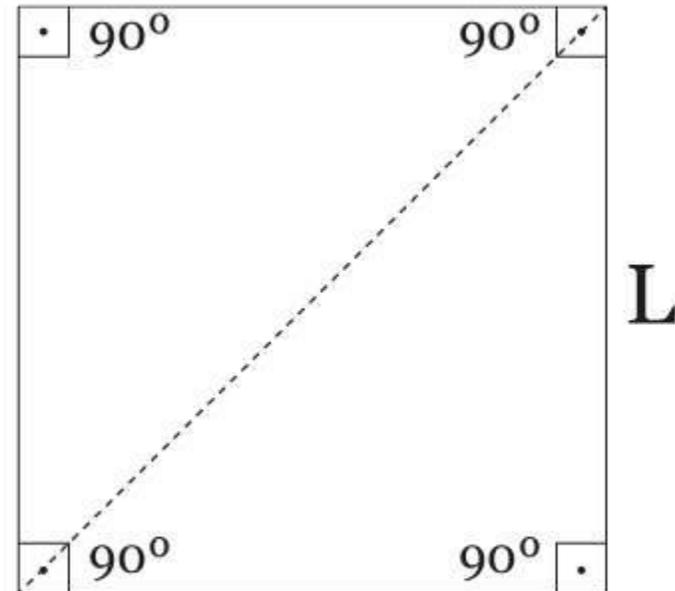
- Métodos em Programação Orientada a Objetos (POO) são funções que pertencem a uma classe e são usadas para realizar ações ou operações em objetos dessa classe.
- Eles representam o comportamento do objeto e permitem que o objeto interaja com o mundo exterior.
- Por exemplo, se tivermos uma classe "Pessoa" com os métodos "andar", "falar" e "trabalhar", podemos chamar esses métodos em objetos dessa classe para realizar essas ações. O método "andar" pode definir como o objeto se move, o método "falar" pode permitir que o objeto emita sons, e o método "trabalhar" pode definir como o objeto executa uma tarefa.

Abstraindo. ..

- Quais as propriedades/Atributos de um Quadrado?
- Quais os métodos ou ações poderiam ser realizadas por uma classe Retangulo?

Classe (Quadrado)

- Atributos (Características)
 - Lado
- Métodos
 - `area()`
 - `diagonal()`
 - `perimetro()`

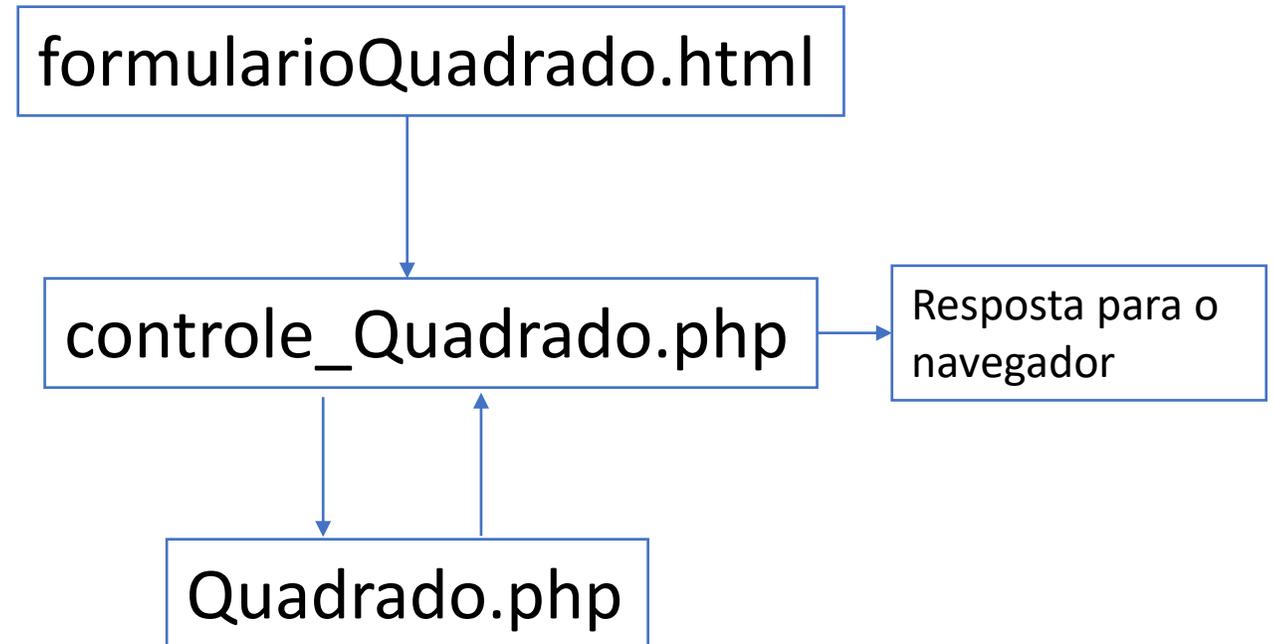


Arquivos / Arquitetura

- Formulário:
 - formularioQuadrado.html
- "PáginaDestino"
 - controle_Quadrado.php
- Classe:
 - Quadrado.php

Estrutura de hierarquia.

- Pasta
 - formularioQuadrado.html
 - controle_Quadrado.php
 - Quadrado.php



Classe: Quadrado.php

```
<?php
class Quadrado{
    private $lado;
    public function setLado($novoLado){
        $this->lado = $novoLado;
    }
    public function getLado(){
        return $this->lado;
    }
    public function area(){
        $a = $this->lado * $this->lado;
        return $a;
    }
    public function perimetro(){
        $p = 4* ($this->getLado());
        return $p;
    }
    public function diagonal(){
        //d=10√2 cm
        $d =$this->getLado()* sqrt(2);
        return $d;
    }
}
?>
```

Para cada atributo deve existir um get e um set

Métodos: Ações da classe

Usando uma classe:

Para utilizar uma classe é necessário fazer uma instancia da classe
Quando instanciamos uma classe geramos objetos da classe.

Para instanciar:

```
$nomeObjeto = new NomeClasse();
```

```
$q = new Quadrado();
```

```
$q->setLado(5)
```

```
echo $q->getLado();
```

Utilizando a classe

```
<!--formularioQuadrado.html -->
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
</head>
<body>
  <form action="controle_Quadrado.php" method="get">
    <input type="number" step="0.01" name="txtLado"><br>
    <input type="submit" value="Calcular">
  </form>
</body>
</html>
```

Quadrado é uma classe
\$q é um objeto da classe Quadrado...
Quadrado pode gerar vários objetos

//controle_Quadrado.php

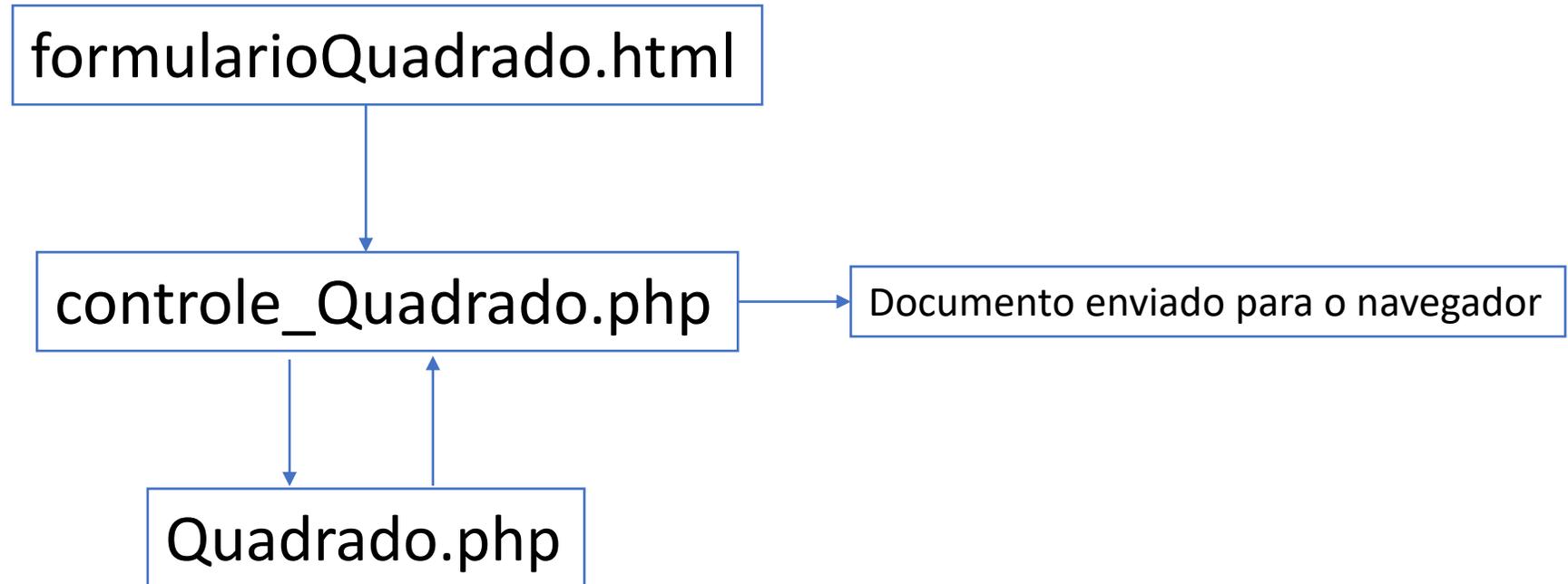
```
<?php
require_once "Quadrado.php"; //importa o arquivo da classe
$lado = $_GET['txtLado'];
$q = new Quadrado(); //instancia um novo quadrado
$q->setLado($lado); //Faça a chamada do método setLado

$respostaArea = $q->area(); //acessa o método área do objeto $q
$respostaPerimetro = $q->perimetro();
$respostaDiagonal = $q->diagonal();

printf("area =%.2f<br>",$respostaArea);
printf("perimetro = %.2f<br>",$respostaPerimetro);
printf("diagonal = %.2f<br>", $respostaDiagonal);

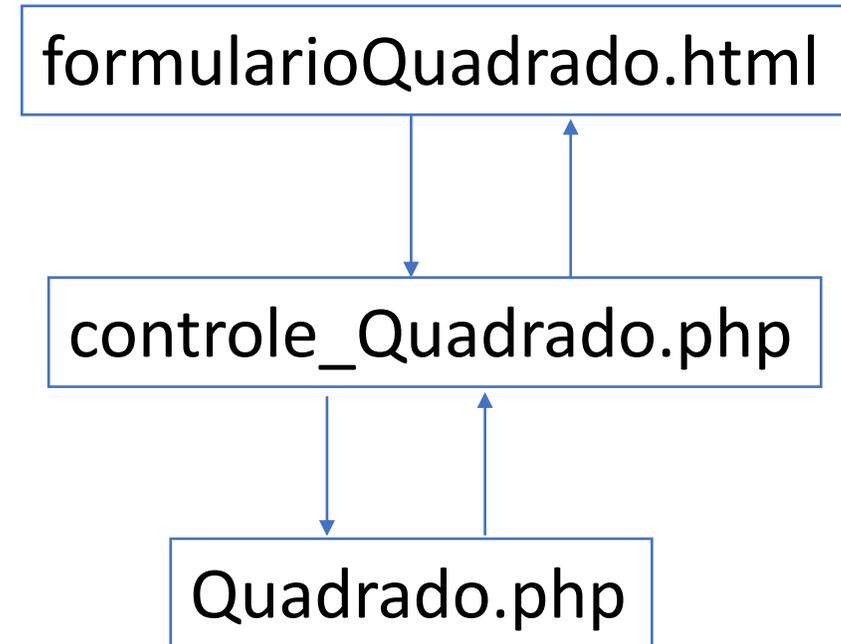
?>
```

Arquitetura

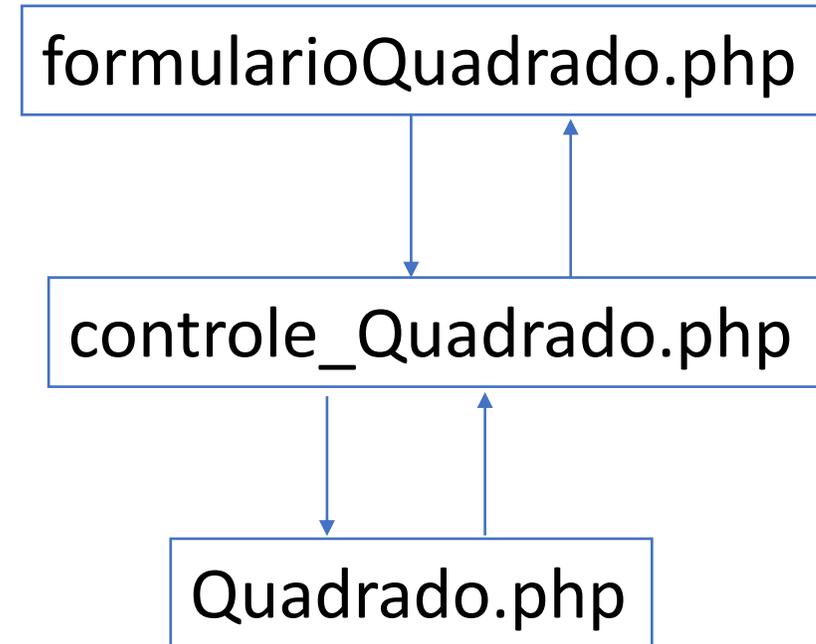


E se o controle_Quadrado.php enviase dados para formularioQuadrado.html?

- **formularioQuadrado.html** deve ser um **arquivo.php** para ser capaz de manipular dados vindos de **controle_Quadrado.php**



Nova arquitetura

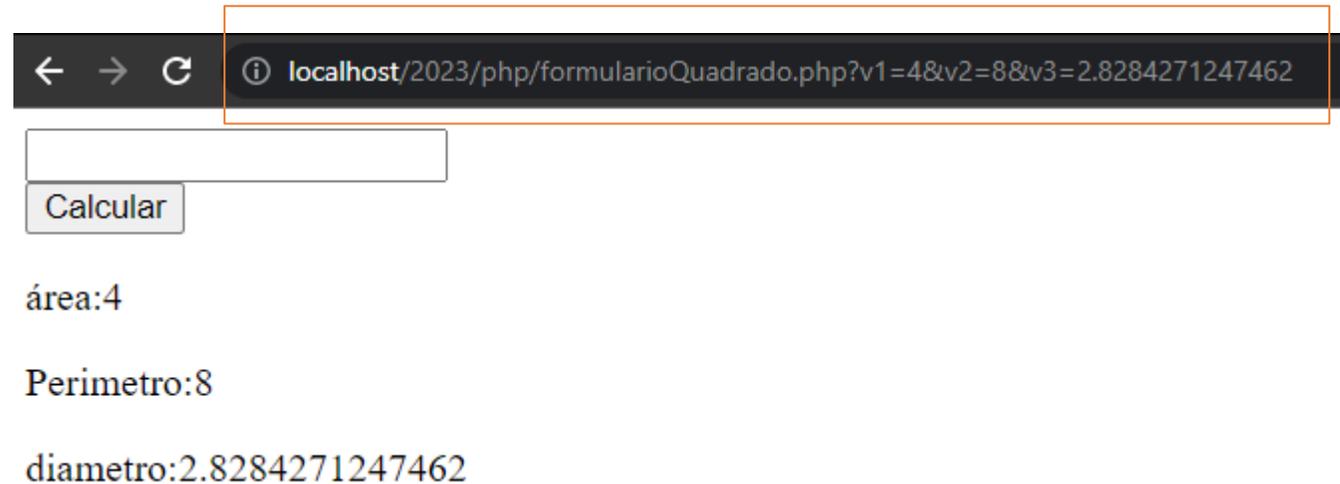


formularioQuadrado.php

```
<?php
$a=0; $p=0; $d=0;
if(isset($_GET['v1'])){
    $a = $_GET['v1'];
    $p = $_GET['v2'];
    $d = $_GET['v3'];
}
?>
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
</head>
<body>
    <form action="controle_Quadrado.php" method="get">
        <input type="number" step="0.01" name="txtLado"><br>
        <input type="submit" value="Calcular">
    </form>
    <div>
        <p>área:<?php echo $a;?></p>
        <p>Perimetro:<?php echo $p;?></p>
        <p>diametro:<?php echo $d;?></p>
    </div>
</body>
</html>
```

controle_Quadrado.php

```
<?php
    require_once "Quadrado.php";
    $lado = $_GET['txtLado'];
    $q = new Quadrado();
    $q->setLado($lado);
    $a = $q->area();
    $b = $q->perimetro();
    $c = $q->diagonal();
    $variaveis="v1=$a&v2=$b&v3=$c";
    header("location: formularioQuadrado.php?$variaveis");
?>
```

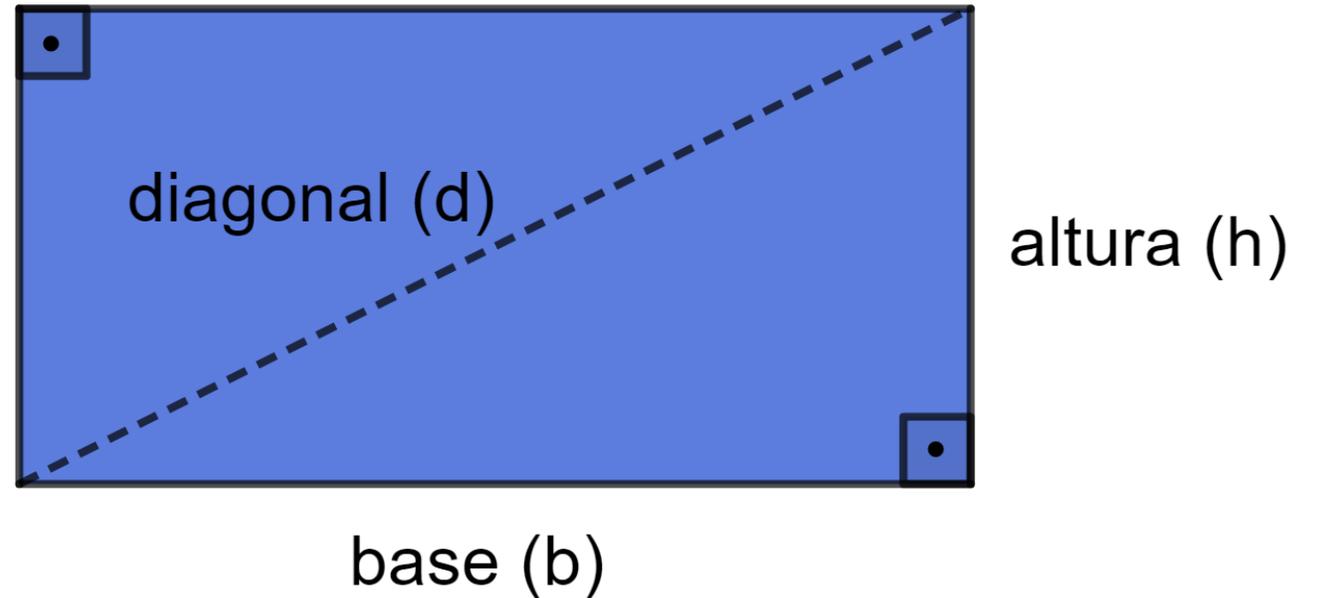


Abstraindo. ..

- Quais as propriedades/Atributos de um Retângulo?
- Quais os métodos ou ações poderiam ser realizadas por uma classe retângulo?

Classe (Retangulo)

- Atributos (Características)
 - base
 - altura
- Métodos
 - CalcularArea()
 - CalcularDiagonal()
 - CalcularPerimetro()



Retangulo.php

```
?php
class Retangulo{
    private $base;
    private $altura;
    public function setBase($novaBase){
        $this->base=$novaBase;
    }
    public function getBase(){
        return $this->base;
    }
    public function setAltura($novaAltura){
        $this->altura = $novaAltura;
    }
    public function getAltura(){
        return $this->altura;
    }
    public function calcularArea(){
        return ($this->altura*$this->base);
    }
    public function calcularDiagonal(){
        return sqrt(pow($this->altura,2) + pow($this->base,2));
    }
    public function calcularPerimetro(){
        return ($this->altura*2 +$this->base*2) ;
    }
}
?>
```

Para usarmos uma classe devemos fazer a instancia da mesma
tipicamente sua instância é realizada em outro arquivo.

retanguloCalcularArea.php

```
<?php
require_once "Retangulo.php";
$q1 = new Retangulo();

$q1->setBase(10);
$q1->setAltura(5);

$resultado = $q1->calcularArea();

echo $resultado;

?>
```

Recuperar dados de um formulário e usar uma classe

calcularArea.html

```
<!DOCTYPE html>
<html>
  <body>
    <form action="retanguloCalcularArea.php" method="get">
      <input type="text" name="b" placeholder="base"><br>
      <input type="text" name="a" placeholder="altura"><br>
      <input type="submit" value="Calcular">
    </form>
  </body>
</html>
```

retanguloCalcularArea.php

```
<?php
require_once "Retangulo.php";
if(isset($_GET['b']) && isset($_GET['a'])) {
  if(is_numeric($_GET['b']) && is_numeric($_GET['a'])) {
    $q1 = new Retangulo();
    $q1->setBase($_GET['b']);
    $q1->setAltura($_GET['a']);
    $resultado = $q1->calcularArea();
    echo $resultado;
  } else {
    echo "parâmetros em formato incorreto";
  }
} else {
  echo "parâmetros não recebidos";
}
?>
```