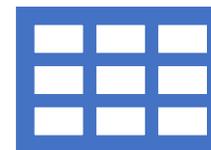


PROGRAMAÇÃO ORIENTADA A OBJETOS EM PHP E MVC

Prof. Me. Hélio Esperidião





Arquitetura de Software

- A arquitetura de software define a maneira como os componentes de software estão organizados e interagem entre si para atender aos requisitos funcionais e não funcionais do sistema.
- A arquitetura de software é responsável por fornecer uma visão geral do sistema, incluindo seus principais elementos, suas propriedades e as relações entre eles.



MVC

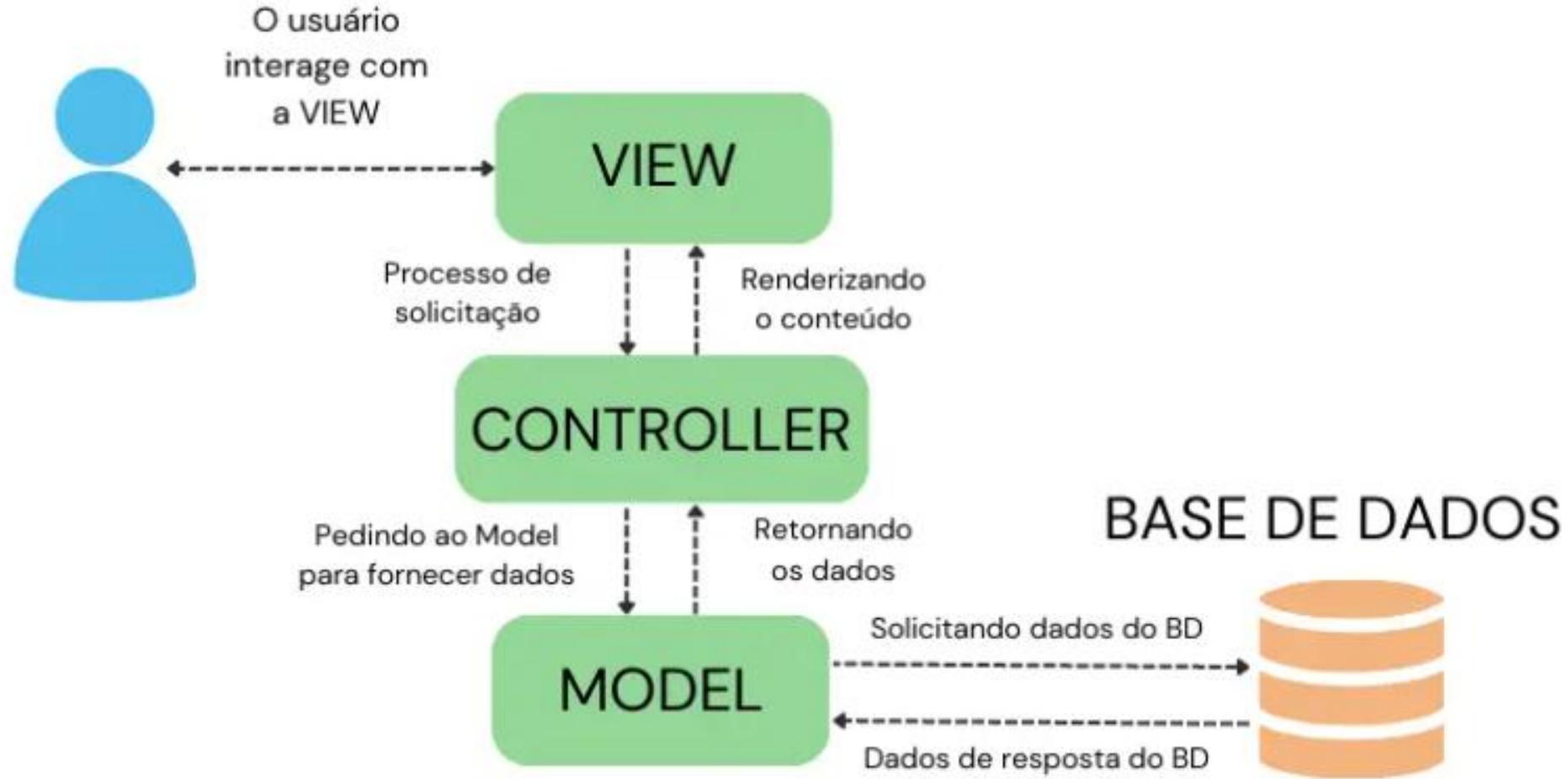
- MVC (Model-View-Controller) é um padrão arquitetural amplamente utilizado no desenvolvimento de software, especialmente em aplicações web e aplicativos para dispositivos móveis.



Padrão MVC

- **Modelo**
 - Todas as **CLASSES** da aplicação ficarão armazenadas nesse diretório
- **Visualização**
 - Todos os arquivos de front-end ficarão armazenados nesse diretório.
- **Controle**
 - Arquivos que recebem dados do front-end, utilizam as **CLASSES** e enviam respostas para o front-end.

Arquitetura MVC



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.



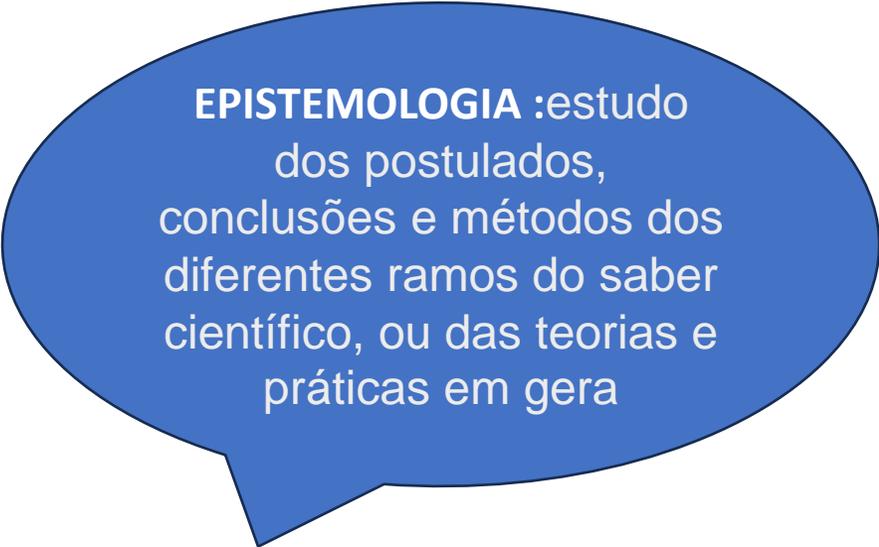
Paradigma

- Paradigma

- do latim tardio paradigma, do grego *παράδειγμα*, derivado de *παραδείκνυμι* «mostrar, apresentar, confrontar»)
- É um conceito das ciências e da **epistemologia** que define um exemplo típico ou modelo de algo. É a representação de um padrão a ser seguido.

- Paradigma

1. Um exemplo que serve como modelo; padrão.
2. É a representação de um padrão a ser seguido.



EPISTEMOLOGIA :estudo dos postulados, conclusões e métodos dos diferentes ramos do saber científico, ou das teorias e práticas em gera

Paradigmas de programação

- O paradigma de programação é uma forma ou abordagem que guia a maneira como os desenvolvedores organizam e estruturam o código.
- Define a metodologia utilizada para conceber a execução e a arquitetura do programa.

O que é uma classe?

- Em **Programação Orientada a Objetos (POO)**, uma **classe** é um **MODELO** ou **MOLDE** para criar objetos. Ela define um conjunto de **atributos** (dados) e **métodos** (comportamentos) que os objetos dessa classe terão.

Atributos

- São as **características** ou **propriedades** de um **objeto**.
- Representam o estado de um objeto e podem ser de diferentes tipos de dados, como **números**, **strings**, **booleanos**, entre outros.
- Se tivermos uma classe(Modelo) "**Pessoa**" com os atributos "**nome**", "**idade**" e "**sexo**",
- Podemos criar vários objetos dessa **classe "Pessoa"**, 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

Métodos

- São **funções** que pertencem a uma classe ou objeto e são usadas para realizar ações ou operações em objetos dessa classe.
- 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.

Instanciar

Instanciar significa criar um objeto a partir de uma classe.

Cada instância tem seus próprios dados (atributos) e pode executar os métodos definidos na classe.

Podemos criar várias instâncias de uma mesma classe, cada uma representando um elemento individual do modelo.

Instância



Uma instância é um objeto criado a partir de uma classe. Quando você instancia uma classe, está criando um objeto específico baseado no modelo definido pela classe.



Analogia: Pense em uma classe como uma receita de bolo. A receita define os ingredientes (atributos) e o modo de preparo (métodos).



A **instância** é um bolo real feito a partir da receita.

Você pode criar vários bolos (instâncias) a partir da mesma receita (classe), mas cada bolo pode ter características diferentes (exemplo: sabor, tamanho).

Classe vs Objeto

CLASSE	OBJETO
É um modelo ou molde para criar objetos.	É uma instância de uma classe.
Define atributos (dados) e métodos (comportamentos).	Possui valores concretos para os atributos e pode executar os métodos.
Não ocupa espaço na memória até que um objeto seja criado.	Ocupa espaço na memória ao ser instanciado.
Representa uma ideia abstrata (exemplo: "Carro").	Representa um exemplo real daquela ideia (exemplo: "Um carro vermelho da marca X").

Regras para nomear classes

```
class Carro {}  
class ContaBancaria {}  
class UsuarioSistema {}
```

- A nomenclatura utilizada para definir o nome de classes segue o padrão de PascalCase (também chamado de UpperCamelCase).
 1. O nome da classe deve começar com letra maiúscula.
 2. Se houver mais de uma palavra, cada palavra deve iniciar com letra maiúscula.
 3. Não use caracteres especiais ou espaços (apenas letras e números, além de _ se necessário).
 4. O nome da classe deve ser descritivo, representando o que a classe faz.

Regras para nomear métodos

- A nomenclatura utilizada para definir nomes de **métodos** segue o padrão **camelCase** (também chamado de **lowerCamelCase**).
 1. O nome do método deve começar com letra minúscula.
 2. Se houver mais de uma palavra, a partir da segunda palavra, cada uma deve iniciar com letra maiúscula.
 3. O nome deve ser descritivo e representar a ação que o método executa.
 4. Não use caracteres especiais e abreviações.
 5. Métodos que retornam um valor booleano geralmente começam com "is" ou "has" (exemplo: `isAtivo()`, `hasPermissao()`).

Regras para nomear variáveis

- O nome da variável deve começar com uma letra minúscula.
- Se o nome tiver mais de uma palavra, a segunda palavra em diante deve começar com letra maiúscula.
- Não use caracteres especiais
- Não use palavras reservadas
- Use nomes descritivos que representem o propósito da variável.
- Evite nomes muito curtos ou muito longos
 - Exemplo ruim: a ou informacaoDoUsuarioSobreCadastroNoSistema
 - Exemplo bom: informacaoUsuario

Regras para o nomear arquivos de classes:

- Tipicamente um arquivo para cada classe.
- O nome do arquivo deve ser igual ao nome da classe.

Encapsulamento

- O encapsulamento é um dos pilares da Programação Orientada a Objetos (POO) e consiste em restringir o acesso direto aos atributos de uma classe, permitindo que eles sejam manipulados apenas através de métodos específicos (como get e set).
- O que é método?
- O que é atributo?
- O que é um método get?
- O que é um método set?

Importante

O que é método?	São “funções” de implementadas por uma classe.
O que é atributo?	São “variáveis” ou características que definem um objeto.
O que é um método get?	É uma “função” que retorna o conteúdo de um determinado atributo.
O que é um método set?	É uma “função” que insere o conteúdo de um determinado atributo.

Método set e get

- método set (ou setter) é usado para modificar o valor de um atributo privado.
- Vantagens do set e get
 - Permitem controle sobre os dados (ex.: validar entradas).
 - Impedem modificações indevidas dos atributos.
 - Melhoram a segurança e manutenção do código.
- Getters e Setters garantem o encapsulamento e segurança dos dados.
 - Evitam acesso direto aos atributos privados.
 - Permitem validações antes de alterar valores.
 - Boas práticas recomendam sempre encapsular atributos privados e expô-los por meio de métodos get e set!



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.

O processo de abstração.

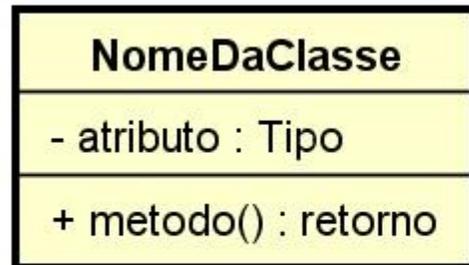
- O resultado do processo de abstração em programação orientada a objetos é uma classe.
- As classes podem ser consideradas **MODELOS** (moldes ou formas) que podem dar origem a diversos objetos.

Abstraindo. ..

- Quais as propriedades/Atributos de um Quadrado?

Diagrama de classes

- Pode representar uma classe por meio de símbolos.
- (-) significa privado, pode ser utilizado apenas dentro da classe
- (+) significa publico, pode ser utilizado em todas as outras classes.



EX:



Classe (Quadrado)

- Atributos (Características)
 - lado
- Métodos
 - calcularArea()
 - calcularDiagonal()
 - calcularPerimetro()

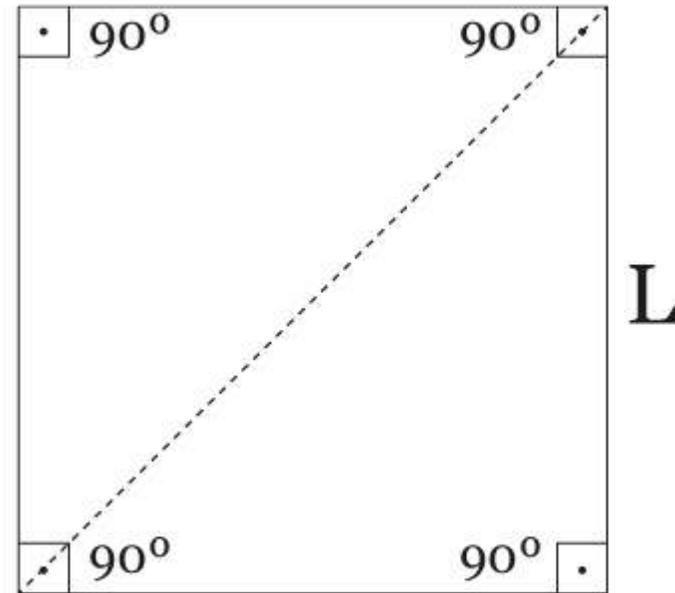
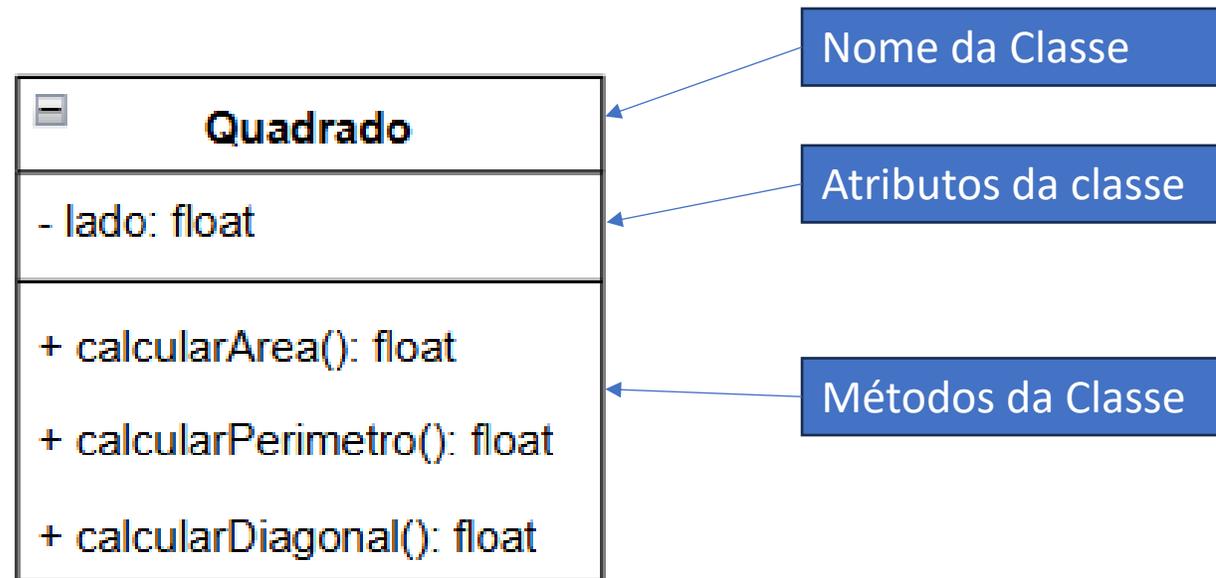


Diagrama da classe Quadrado



Classe: Quadrado.php

```
<?php
```

```
class Quadrado{
```

```
    private $lado; //atributo da classe
```

```
    public function calcularArea() {  
        $a = $this->lado * $this->lado;  
        return $a;  
    }
```

```
    public function calcularPerimetro() {  
        $p = 4 * ($this->getLado());  
        return $p;  
    }
```

```
    public function calcularDiagonal() {  
        //d=l*√2 cm  
        $d = $this->getLado() * sqrt(2);  
        return $d;  
    }
```

```
    public function setLado($novoLado) {  
        $this->lado = $novoLado;  
    }
```

```
    public function getLado() {  
        return $this->lado;  
    }
```

```
}
```

Para cada atributo deve existir um get e um set

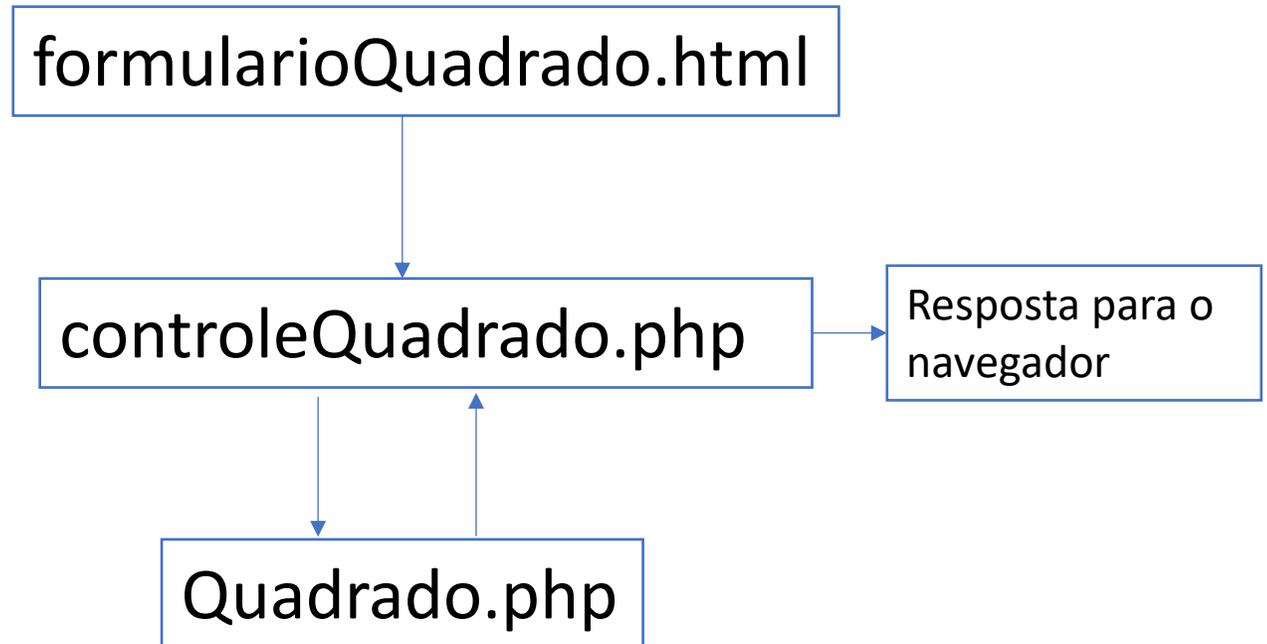
Caso uma classe possua 10 atributos,
somando os métodos
Get e set obtemos um total de?

Arquivos / Arquitetura

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

Estrutura de hierarquia.

- htdocs
 - formularioQuadrado.html
 - controle_Quadrado.php
 - Quadrado.php



Utilizando a classe

formularioQuadrado.html

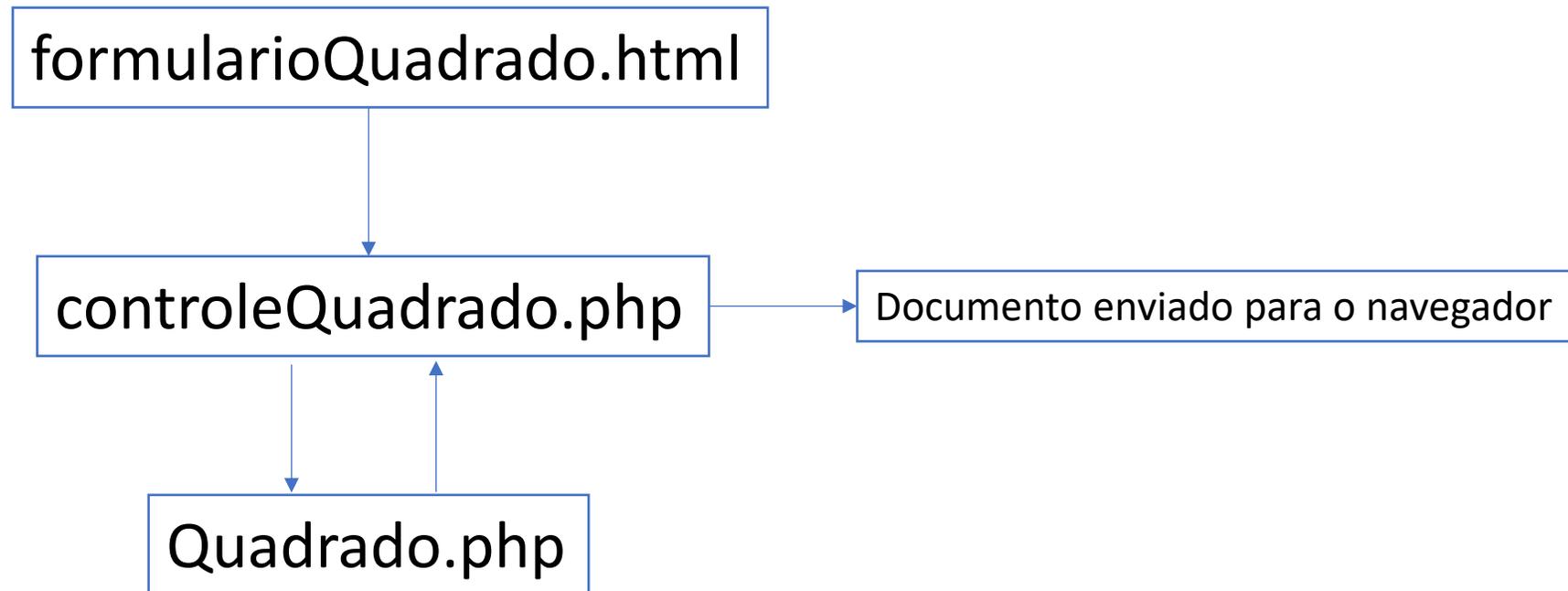
```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
</head>
<body>
  <form action="controleQuadrado.php" method="get">
    <input type="text" name="ladoQuadrado"><br>
    <input type="submit" value="Calcular">
  </form>
</body>
</html>
```

controleQuadrado.php

```
<?php
require_once "Quadrado.php"; //importa o arquivo da classe
if (isset($_GET['ladoQuadrado'])) {
    if (is_numeric($_GET['ladoQuadrado'])) {
        $lado = $_GET['ladoQuadrado'];
        $objetoQuadrado = new Quadrado(); //instancia um novo quadrado
        $objetoQuadrado->setLado($lado); //Faça a chamada do método setLado
        $respostaArea = $objetoQuadrado->calcularArea(); //acessa o método do objeto
        $respostaPerimetro = $objetoQuadrado->calcularPerimetro();
        $respostaDiagonal = $objetoQuadrado->calcularDiagonal();
        printf("area =%.2f<br>", $respostaArea);
        printf("perimetro = %.2f<br>", $respostaPerimetro);
        printf("diagonal = %.2f<br>", $respostaDiagonal);
    } else {
        echo "O lado fornecido não é um número";
    }
} else {
    echo "Não foi fornecido um lado para calcular";
}
• ?>
```

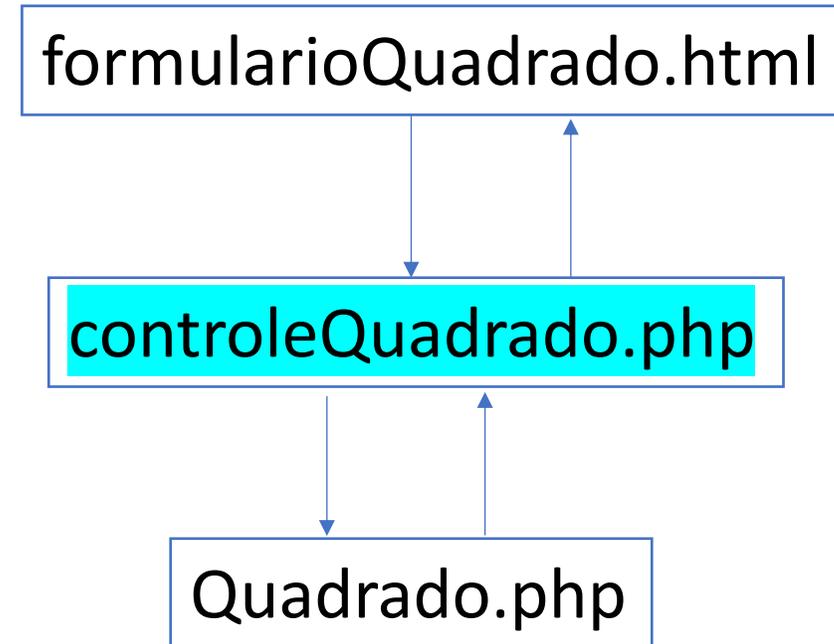
Arquitetura

é possível mostrar os dados no arquivo:
formularioQuadrado.html?



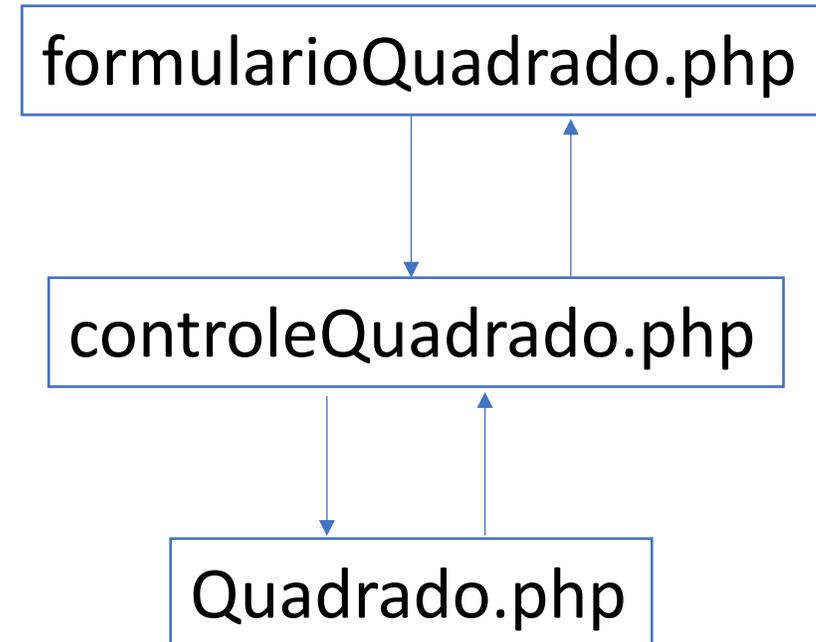
E se o `controleQuadrado.php` enviase dados para `formularioQuadrado.html`?

- `formularioQuadrado.html` deve ser um `arquivo.php` para ser capaz de manipular dados vindos de `controleQuadrado.php`



Nova arquitetura

- Se o arquivo de formulário é um php
 - É possível receber dados e programar no arquivo de formulário



header("Location: URL")

- Função header("Location: URL") no PHP é usada para redirecionar o navegador para outra página ou URL. Ela envia um cabeçalho HTTP informando ao navegador que deve carregar um novo endereço.
 - Exit() é utilizado para sair da execução do script e nada mais ser executado.

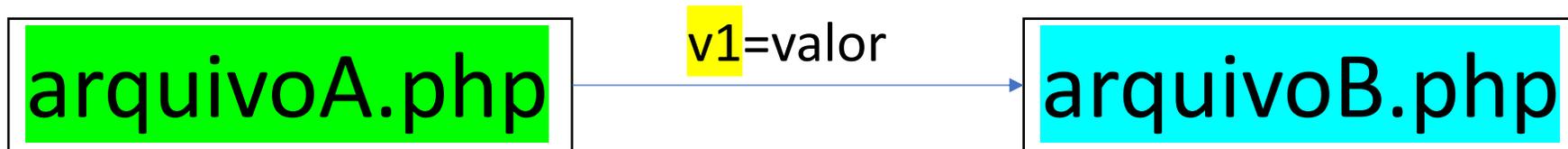
```
header("Location: https://exemplo.com");  
exit();
```

```
header("Location: arquivo.qualquer");  
exit();
```

```
header("Location: arquivo.php?parametro1=valor&parametro2=valor");  
exit();
```

Esquema de funcionamento

- Se houver no **arquivoA.php** `header("Location: URL")` para o **arquivoB.php**
 - `header("Location: arquivoB.php?v1=valor")`
 - O navegador irá redirecionar para o arquivo B e enviará a variável get v1

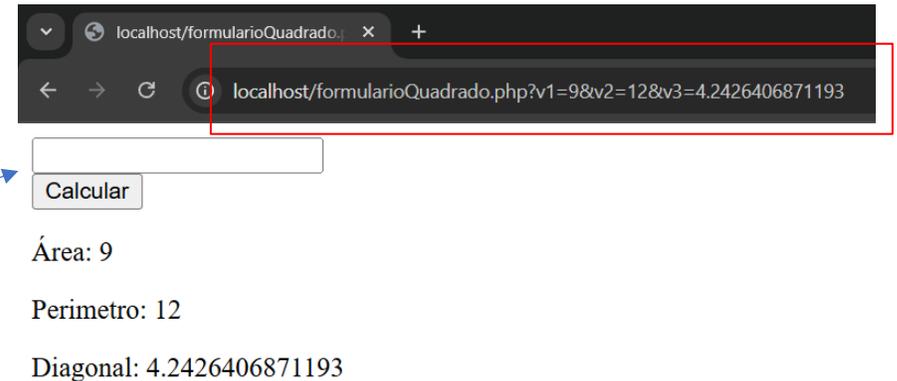


Funcionamento

- O arquivo de controle deve receber os dados vindos do formulário
- Os dados são validados
 - Caso sejam invalidados é enviada a variável get “**erro**” para o formulário
- É calculada a área, o perímetro e a Diagonal.
- É feito redirecionamento para o arquivo de formulário.
- No redirecionamento são enviadas as variáveis get:
 - **v1**, **v2** e **v3** que correspondem a área, perímetro e diagonal.

controleQuadrado.php

```
<?php
require_once "Quadrado.php"; //importa o arquivo da classe
if (isset($_GET['ladoQuadrado'])) {
    if (is_numeric($_GET['ladoQuadrado'])) {
        $lado = $_GET['ladoQuadrado'];
        $q = new Quadrado();
        $q->setLado($lado);
        $a = $q->calcularArea();
        $b = $q->calcularPerimetro();
        $c = $q->calcularDiagonal();
        $variaveis = "?v1=$a&v2=$b&v3=$c";
        header("location: formularioQuadrado.php$variaveis");
        exit();
    } else {
        $variaveis = "?erro=0 lado fornecido não é um número";
        header("location: formularioQuadrado.php$variaveis");
        exit();
    }
} else {
    $variaveis = "?erro=Não foi fornecido um lado para calcular";
    header("location: formularioQuadrado.php$variaveis");
    exit();
}
?>
```



```
<!DOCTYPE html>
<head>
  <meta charset="UTF-8">
</head>
<body>
  <form action="controleQuadrado.php" method="get">
    <input type="text" name="ladoQuadrado"><br>
    <input type="submit" value="Calcular">
  </form>
  <div>
    <?php
      $a = 0;
      $p = 0;
      $d = 0;
      if (isset($_GET['v1'])) {
        $a = $_GET['v1'];
        $p = $_GET['v2'];
        $d = $_GET['v3'];
        echo "<p>Área: $a</p>";
        echo "<p>Perimetro: $p</p>";
        echo "<p>Diagonal: $d</p>";
      } elseif (isset($_GET['erro'])) {
        $dadosErro = $_GET['erro'];
        echo "<p class='erro'>$dadosErro</p>";
      }
    ?>
  </div>
</body>
</html>
```

Quando o bloco 1 é executado?

Quando o bloco 2 é executado?

*Quando eu abrir o arquivo
formularioQuadrado.php
Entrará em algum if?*

1

2

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()

