unesp

Desenvolvimento de Aplicações para Internet

Aula 8

Celso Olivete Júnior

olivete@fct.unesp.br

PHP - Orientação a objetos
na aula passada

- Orientação a Objetos
  - Classes
  - Objetos
  - Construtores
  - Destrutores

PHP - Orientação a objetos
na aula passada

- Construtores ➔ exemplo: classe pai e classe filha

```
<?php
class BaseClass {
   public function __construct() {
       echo "In BaseClass constructor\n";
   }
}

class SubClass extends BaseClass {
   public function __construct() {
       parent::__construct();
       echo "In SubClass constructor\n";
   }
}

$obj = new BaseClass();
$obj = new SubClass();
?>
```

PHP - Orientação a objetos
na aula passada

- Destrutor: chamado assim que todas as referências a um objeto forem removidas ou quando o objeto for explicitamente destruído ou qualquer ordem na sequência de encerramento. Exemplo:

```
<?php
class destrutora
{
    private $nome;
    public function __construct()
    {
        $this->nome = "Exemplo";
        echo "Construtor...".$this->nome;
    }

    public function __destruct()
    {
        echo "<br>Destruindo...".$this->nome;
    }
}
$obj = new destrutora;
$obj = null;
?>
```

PHP - Orientação a objetos

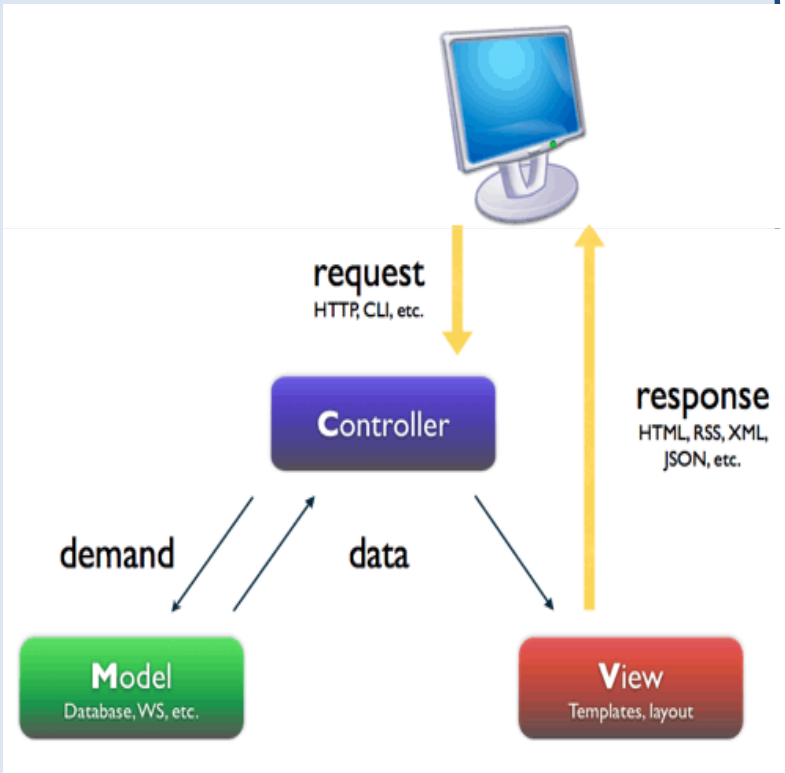
Aula de hoje

- Modelo MVC (Model-view-controller)
  - Um dos padrões de projeto mais populares quando se trabalha em PHP orientado a objetos.
  - A programação fica organizada em três camadas

1 – Model

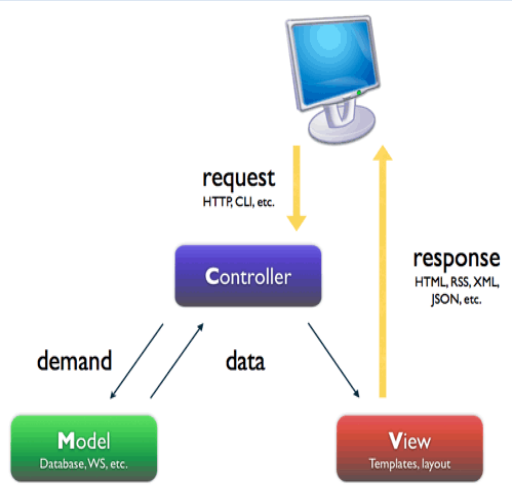
2 – View

3 – Controller



PHP - Orientação a objetos
Modelo MVC

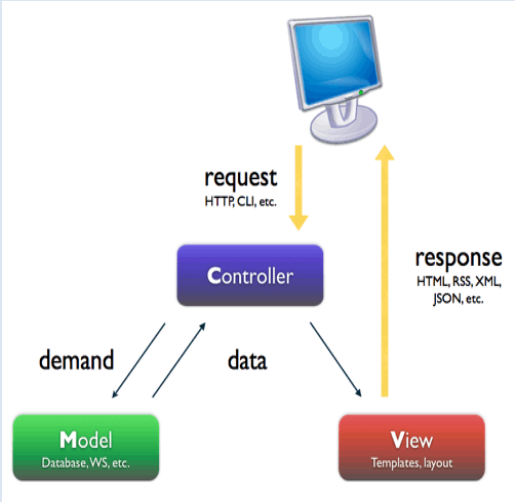
- ***Controller***
  - Onde serão processadas todas as requisições
  - O controle também acessa o ***Model*** afim de obter determinadas informações.
  - Toda lógica da aplicação (validações, atribuições, etc) é feita no ***Controller***.
  - É o gerenciador da aplicação



PHP - Orientação a objetos
Modelo MVC

☐ *Models*

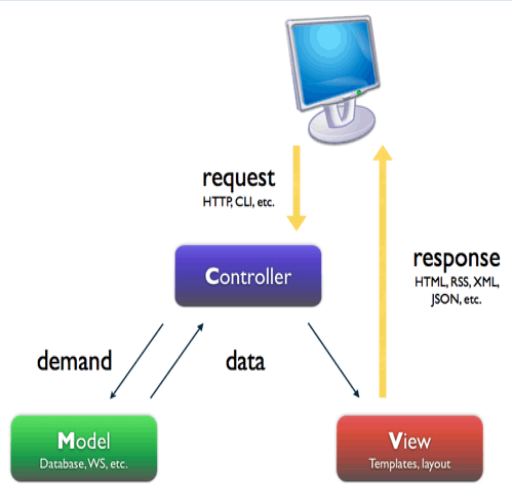
- É o modelo da aplicação, onde são definidos propriedades e atributos.
- Na arquitetura MVC não deve haver interação entre ***modelos*** e ***views***. Toda a lógica é manipulada pelos ***controllers***.



PHP - Orientação a objetos
Modelo MVC

❑ *View*

- ❑ É a camada de visualização da sua aplicação, onde ela apenas apresenta o que foi obtido através do ***controle***.
- ❑ É o que chega ao usuário final, a parte visual, de interface. A visão não deve ter nenhuma lógica de código, apenas a exibição dos dados.



PHP - Orientação a objetos
Modelo MVC

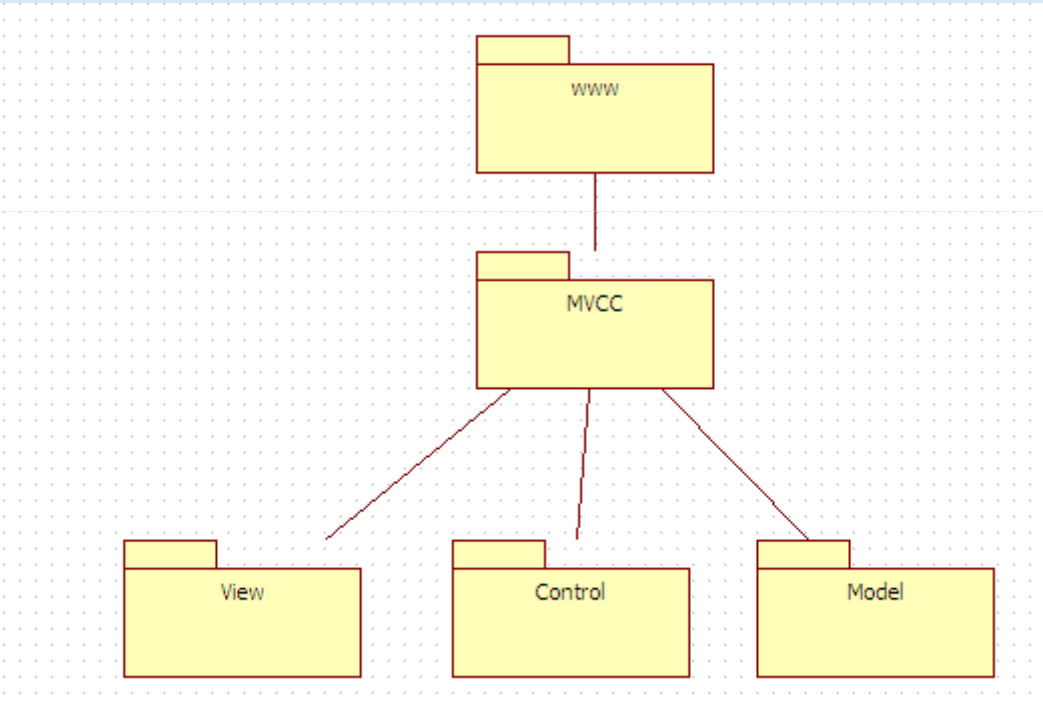
- Vantagens
  - Organização
  - Legibilidade
  - Maior controle de manutenção
  - Integração de programadores e webdesigners

PHP - Orientação a objetos
Modelo MVC

- Exemplo de aplicação com as manipulações (inserção, alteração, exclusão e listagem) ao BD

PHP - Orientação a objetos
Modelo MVC

- Estrutura de diretórios



PHP - Orientação a objetos
Modelo MVC

- Diagrama de classe da aplicação

Pessoa
-cod -nome -fone -rua
+setNome(nome) +setFone(fone) +setRua(rua) +retornaPessoas() +inserePessoa()

PHP - Orientação a objetos
Modelo MVC

- ***Model***
- Implementando o modelo computacional ➔ classe pessoa. Esse script (modelo_classe_pessoa.php) deverá ser salvo dentro da estrutura: www/mvc/model
- Dentro do diretório www/mvc defina um script responsável pela conexão ao MySql e por selecionar o BD

PHP - Orientação a objetos
Implementando a camada ***model***

- ***modelo_classe_pessoa:*** camada responsável por manipular dados (definir atributos e manipular o BD)

```
class pessoa
 {
    private $cod;
    private $nome;
    private $fone;
    private $rua;
    //setar atributos...
    public function setNome($nome)
    {
       $this->nome = $nome;
    } ...
   //retornar atributos...
    public function getNome()
    {
       return $this->nome;
    }...
```

PHP - Orientação a objetos
Implementando a camada ***model***

- ***modelo_classe_pessoa:*** continuação... Método responsável por retornar todas as pessoas do BD → retorna um array de objetos

Cria um obj conexão

```
class pessoa
 { ...
   public function retornaPessoas()
    {
     $conexao = new Conexao();
     $conexao->conectar();
     $query = "SELECT * FROM pessoa";
     $result = mysql_query($query);
     while ($dados = mysql_fetch_array($result))
     {
                $pessoa = new pessoa();
                 $pessoa->setNome($dados['nome']);
                 $pessoa->setFone($dados['fone']);
                 $pessoa->setRua($dados['rua']);
                 $ArrayPessoas[] = $pessoa;
         }
     return $ArrayPessoas;
     $conexao->desconectar(); //desconecta
    }
...
```

PHP - Orientação a objetos
Implementando a camada ***model***

- ***modelo_classe_pessoa:*** continuação... Método responsável por retornar todas as pessoas do BD → retorna um array de objetos

Cria um obj conexão

Cria e executa SQL

```
class pessoa
 { ...
   public function retornaPessoas()
    {
     $conexao = new Conexao();
     $conexao->conectar();
     $query = "SELECT * FROM pessoa";
     $result = mysql_query($query);
     while ($dados = mysql_fetch_array($result))
     {
                 $pessoa = new pessoa();
                  $pessoa->setNome($dados['nome']);
                  $pessoa->setFone($dados['fone']);
                  $pessoa->setRua($dados['rua']);
                  $ArrayPessoas[] = $pessoa;
         }
     return $ArrayPessoas;
     $conexao->desconectar(); //desconecta
    }
...
```

PHP - Orientação a objetos
Implementando a camada ***model***

- ***modelo_classe_pessoa:*** continuação... Método responsável por retornar todas as pessoas do BD → retorna um array de objetos

```
class pessoa
 { ...
   public function retornaPessoas()
    {
     $conexao = new Conexao();
     $conexao->conectar();
     $query = "SELECT * FROM pessoa";
     $result = mysql_query($query);
     while ($dados = mysql_fetch_array($result))
     {
                  $pessoa = new pessoa();
                   $pessoa->setNome($dados['nome']);
                   $pessoa->setFone($dados['fone']);
                   $pessoa->setRua($dados['rua']);
                   $ArrayPessoas[] = $pessoa;
          }
     return $ArrayPessoas;
     $conexao->desconectar(); //desconecta
    }
...
```

Cria um obj conexão

Cria e executa SQL

Define um objeto pessoa e seta atributos

PHP - Orientação a objetos
Implementando a camada ***model***

- ***modelo_classe_pessoa:*** continuação... Método responsável por retornar todas as pessoas do BD → retorna um array de objetos

```
class pessoa
 { ...
   public function retornaPessoas()
    {
     $conexao = new Conexao();
     $conexao->conectar();
     $query = "SELECT * FROM pessoa";
     $result = mysql_query($query);
     while ($dados = mysql_fetch_array($result))
     {
                  $pessoa = new pessoa();
                   $pessoa->setNome($dados['nome']);
                   $pessoa->setFone($dados['fone']);
                   $pessoa->setRua($dados['rua']);
                   $ArrayPessoas[] = $pessoa;
          }
     return $ArrayPessoas;
     $conexao->desconectar(); //desconecta
    }
```

Cria um obj conexão

Cria e executa SQL

Define um objeto pessoa e seta atributos

Retorna um array de objetos do tipo pessoa

PHP - Orientação a objetos

Implementando a camada ***model***

- ***modelo_classe_pessoa:*** continuação... Método responsável por inserir pessoas do BD

Cria um obj conexão

```
class pessoa
 { ...
  public function inserePessoa($pessoa)
    {
     $conexao = new Conexao();
     $conexao->conectar();
     $nome = $pessoa->getNome();
     $fone = $pessoa->getFone();
     $rua = $pessoa->getRua();

     $sql = "insert into pessoa(nome,fone,rua)
           values('$nome','$fone','$rua')";

     mysql_query($sql) or die ('Erro ao Inserir');
     $conexao->desconectar();
     echo '<h1>Pessoa inserida com sucesso</h1>';
    }
}
```

PHP - Orientação a objetos

Implementando a camada ***model***

- ***modelo_classe_pessoa:*** continuação... Método responsável por inserir pessoas do BD

Cria um obj conexão

Recupera atributos

```
class pessoa
 { ...
  public function inserePessoa($pessoa)
    {
     $conexao = new Conexao();
     $conexao->conectar();
     $nome = $pessoa->getNome();
     $fone = $pessoa->getFone();
     $rua = $pessoa->getRua();

     $sql = "insert into pessoa(nome,fone,rua)
           values('$nome','$fone','$rua')";

     mysql_query($sql) or die ('Erro ao Inserir');
     $conexao->desconectar();
     echo '<h1>Pessoa inserida com sucesso</h1>';
    }
}
```

PHP - Orientação a objetos
Implementando a camada ***model***

- ***modelo_classe_pessoa:*** continuação... Método responsável por inserir pessoas do BD

```
class pessoa
 { ...
  public function inserePessoa($pessoa)
    {
     $conexao = new Conexao();
     $conexao->conectar();
     $nome = $pessoa->getNome();
     $fone = $pessoa->getFone();
     $rua = $pessoa->getRua();

     $sql = "insert into pessoa(nome,fone,rua)
           values('$nome','$fone','$rua')";

     mysql_query($sql) or die ('Erro ao Inserir');
     $conexao->desconectar();

    }
}
```

Cria um obj conexão

Recupera atributos

Cria e executa instrução SQL

PHP - Orientação a objetos

Implementando a camada ***control***

- ***controle_classe_cliente***: camada responsável pelo controle da aplicação.
  - Salve o arquivo em www/mvc/control

PHP - Orientação a objetos
Implementando a camada ***control***

- ***controle_classe_cliente:*** acessando a camada model

método que faz o controle do método retornaPessoas() da camada modelo

```
<?php
 require('../model/modelo_classe_pessoa.php');

 class Controle {

  private $modelo_pessoa;

  public function Controle_Listagem()
  {
   $this->modelo_pessoa = new pessoa();
   return $this->modelo_pessoa->retornaPessoas();
  }
 ...
```

PHP - Orientação a objetos
Implementando a camada ***control***

- ***controle_classe_cliente:*** acessando a camada model

```
<?php
 require('../model/modelo_classe_pessoa.php');

 class Controle {

  private $modelo_pessoa;

  public function Controle_Listagem()
  {
   $this->modelo_pessoa = new pessoa();
   return $this->modelo_pessoa->retornaPessoas();
  }
  ...
```

método que faz o controle do método retornaPessoas() da camada modelo

Define um objeto para acessar a camada modelo

PHP - Orientação a objetos
Implementando a camada ***control***

- ***controle_classe_cliente:*** acessando a camada model

```
<?php
 require('../model/modelo_classe_pessoa.php');

 class Controle {

  private $modelo_pessoa;

  public function Controle_Listagem()
  {
   $this->modelo_pessoa = new pessoa();
   return $this->modelo_pessoa->retornaPessoas();
  }
   ...
```

método que faz o controle do método retornaPessoas() da camada modelo

Define um objeto para acessar a camada modelo

Retorna os objetos obtidos a partir da camada modelo - método retornaPessoas()

PHP - Orientação a objetos
Implementando a camada ***control***

- ***controle_classe_cliente:*** acessando a camada model

método que recebe os valores vindos da ***View*** e faz o controle do método inserePessoa da camada modelo

```
<?php
 require('../model/modelo_classe_pessoa.php');

 class Controle {
   ...
   public function Controle_Insere($nome,$fone,$rua)
   {
     $this->modelo_pessoa = new pessoa();

     $this->modelo_pessoa->setNome($nome);
     $this->modelo_pessoa->setFone($fone);
     $this->modelo_pessoa->setRua($rua);
     $this->modelo_pessoa->inserePessoa($this->modelo_pessoa);

   }
?>
```

PHP - Orientação a objetos
Implementando a camada ***control***

- ***controle_classe_cliente:*** acessando a camada model

```
<?php
 require('../model/modelo_classe_pessoa.php');

 class Controle {
   ...
   public function Controle_Insere($nome,$fone,$rua)
   {
     $this->modelo_pessoa = new pessoa();

     $this->modelo_pessoa->setNome($nome);
     $this->modelo_pessoa->setFone($fone);
     $this->modelo_pessoa->setRua($rua);
     $this->modelo_pessoa->inserePessoa($this->modelo_pessoa);

   }
?>
```

método que recebe os valores vindos da ***View*** e faz o controle do método inserePessoa da camada modelo

Define um objeto para acessar a camada modelo

PHP - Orientação a objetos
Implementando a camada ***control***

- ***controle_classe_cliente:*** acessando a camada model

```
<?php
 require('../model/modelo_classe_pessoa.php');

 class Controle {
   ...
   public function Controle_Insere($nome,$fone,$rua)
   {
     $this->modelo_pessoa = new pessoa();

     $this->modelo_pessoa->setNome($nome);
     $this->modelo_pessoa->setFone($fone);
     $this->modelo_pessoa->setRua($rua);
     $this->modelo_pessoa->inserePessoa($this->modelo_pessoa);

   }
?>
```

método que recebe os valores vindos da ***View*** e faz o controle do método inserePessoa da camada modelo

Define um objeto para acessar a camada modelo

Seta os valores vindos da ***View*** e acessa o método inserePessoa da camada modelo passando um objeto pessoa por parâmetro

PHP - Orientação a objetos

Implementando a camada ***view***

- ***View_classe_pessoa:*** apresenta o que foi obtido através do controle.
- Salve o arquivo em www/mvc/view

PHP - Orientação a objetos
Implementando a camada ***view***

- Inserindo os dados...

PHP - Orientação a objetos
Implementando a camada ***view***

- ***View_classe_pessoa:*** acessa a camada controle

```
<?php
     require('../Control/controle_classe_pessoa.php');
     if(!empty($_POST['b_Inserir'])) //se optou por inserir
     {
     $controle_inserir_Pessoa = new Controle();
     $nome = $_POST['nome']; $fone = $_POST['fone']; $rua = $_POST['rua'];
     $controle_inserir_Pessoa->Controle_Insere($nome,$fone,$rua);
     echo '<h1>Pessoa inserida com sucesso</h1>';
     }
...
?>
```

inclui a classe ***Control*** no ***view***

PHP - Orientação a objetos
Implementando a camada ***view***

- ***View_classe_pessoa:*** acessa a camada controle

```
<?php
     require('../Control/controle_classe_pessoa.php');
     if(!empty($_POST['b_Inserir'])) //se optou por inserir
     {
     $controle_inserir_Pessoa = new Controle();
     $nome = $_POST['nome']; $fone = $_POST['fone']; $rua = $_POST['rua'];
     $controle_inserir_Pessoa->Controle_Insere($nome,$fone,$rua);
     echo '<h1>Pessoa inserida com sucesso</h1>';
...  }
?>
```

inclui a classe ***Control*** no ***view***

Cria um objeto para acessar a camada de **controle**

PHP - Orientação a objetos
Implementando a camada ***view***

- ***View_classe_pessoa:*** acessa a camada controle

```
<?php
     require('../Control/controle_classe_pessoa.php');
     if(!empty($_POST['b_Inserir'])) //se optou por inserir
     {
     $controle_inserir_Pessoa = new Controle();
     $nome = $_POST['nome']; $fone = $_POST['fone']; $rua = $_POST['rua'];
     $controle_inserir_Pessoa->Controle_Insere($nome,$fone,$rua);
     echo '<h1>Pessoa inserida com sucesso</h1>';
...  }
?>
```

inclui a classe ***Control*** no ***view***

Cria um objeto para acessar a camada de **controle**

Chama o método para inserir uma pessoa, a partir da camada **controle**

PHP - Orientação a objetos

Implementando a camada ***view***

- Exibindo os dados...

PHP - Orientação a objetos
Implementando a camada ***view***

- ***View_classe_pessoa:*** acessa a camada controle

```
<?php
     ...
 if(!empty($_POST['b_Exibir']))
 {
     require('../Control/controle_classe_pessoa.php');
     $controle_vizualizar_pessoas = new Controle();
     $Pessoas_Retornadas = $controle_vizualizar_pessoas->Controle_Listagem();

     foreach( $Pessoas_Retornadas as $p ){
      echo '<br>';
      echo "<br>".$p->getNome();
      echo "<br>".$p->getFone();
      echo "<br>".$p->getRua();
     }
  }
?>
```

Cria um objeto para acessar a camada de **controle**

PHP - Orientação a objetos
Implementando a camada ***view***

- ***View_classe_pessoa:*** acessa a camada controle

```
<?php
     ...
 if(!empty($_POST['b_Exibir']))
 {
     require('../Control/controle_classe_pessoa.php');
     $controle_vizualizar_pessoas = new Controle();
     $Pessoas_Retornadas = $controle_vizualizar_pessoas->Controle_Listagem();

     foreach( $Pessoas_Retornadas as $p ){
      echo '<br>';
      echo "<br>".$p->getNome();
      echo "<br>".$p->getFone();
      echo "<br>".$p->getRua();
     }
  }
?>
```

Cria um objeto para acessar a camada de **controle**

Obtêm o array de pessoas a partir da camada **controle**

PHP - Orientação a objetos
Implementando a camada ***view***

- ***View_classe_pessoa:*** acessa a camada controle

```
<?php
     ...
 if(!empty($_POST['b_Exibir']))
 {
     require('../Control/controle_classe_pessoa.php');
     $controle_vizualizar_pessoas = new Controle();
     $Pessoas_Retornadas = $controle_vizualizar_pessoas->Controle_Listagem();

     foreach( $Pessoas_Retornadas as $p ){
      echo '<br>';
      echo "<br>".$p->getNome();
      echo "<br>".$p->getFone();
      echo "<br>".$p->getRua();
     }
  }
?>
```

Cria um objeto para acessar a camada de **controle**

Obtêm o array de pessoas a partir da camada **controle**

Percorre o array e imprime os dados da pessoa

PHP - Orientação a objetos
Modelo MVC

- Exercício:
  - Fazer as manipulações: alterar, excluir e localizar um determinado registro