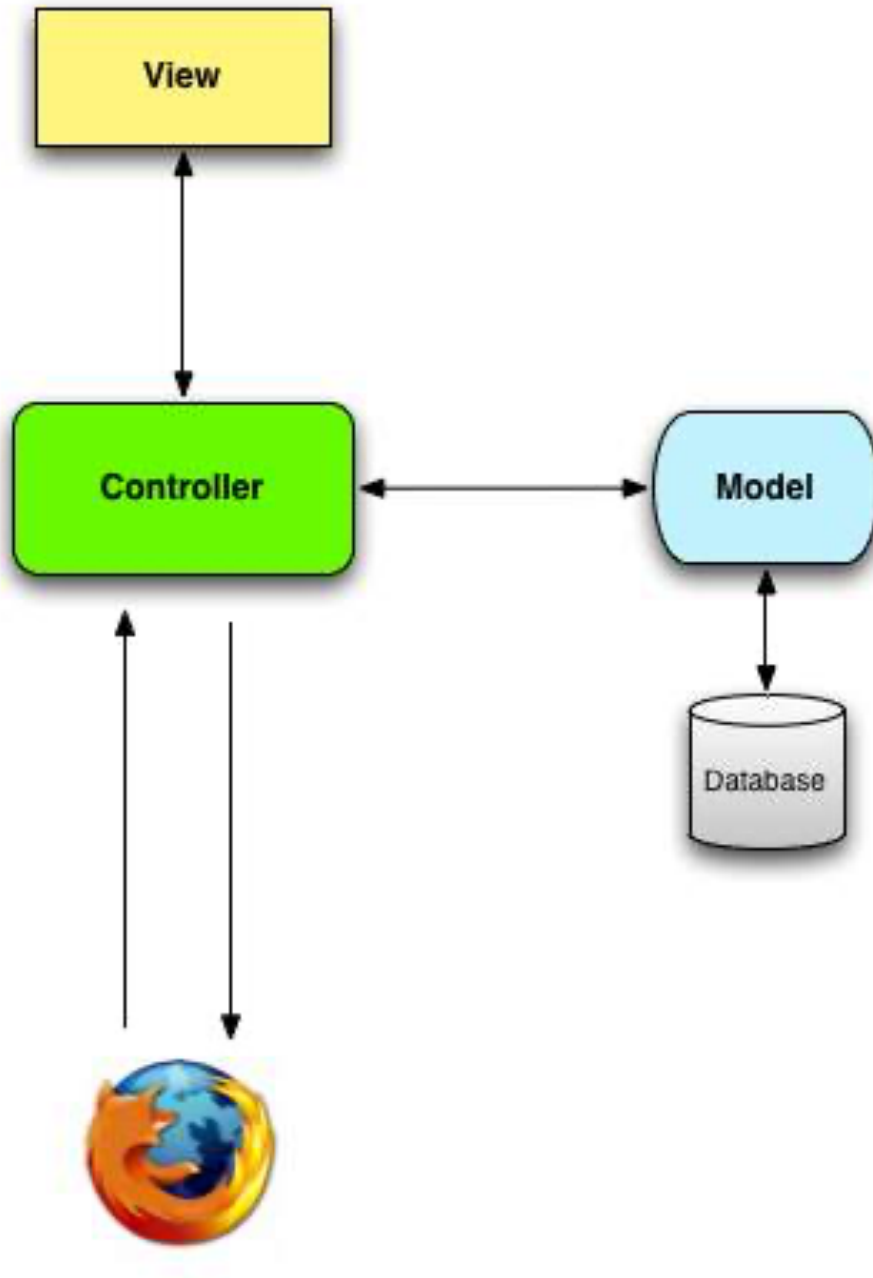


Arquitectura

Model-View-Controller



PHP com template engines

Interactions

1. A ***model*** stores data that is retrieved by the controller and displayed in the view.
2. A ***view*** generates new output to the user based on changes in the model.
3. A ***controller*** can send commands to the model to update the model's state. It can also send commands to change the view's presentation of the model.

HTML templates e PHP

Um documento HTML com código PHP embebido (.php) tem desvantagens:

- É difícil alterar a apresentação (estrutura e/ou o layout) do documento sem mexer no código PHP
- Solução: utilizar HTML templates

HTML templates e PHP (cont.)

- Um template é um documento HTML com campos (“placeholders”) que são substituídos pelo valor de variáveis PHP
- O preenchimento dos campos é feito com recurso a uma template engine: (uma classe escrita em PHP)
- Vamos mostrar com dois exemplos duas template engines:
 1. PEAR Integrated Template (IT)
 2. Smarty

Exemplo

Pretende-se construir dinamicamente a seguinte tabela:



| nome | ano | duração | aCores | estúdio | realizador |
|---------------------------|------|---------|--------|-----------------|-------------------|
| Star Wars | 1977 | 124 | 0 | Fox | George Lucas |
| Empire Strikes Back | 1980 | 143 | 0 | Fox | Irvin Kershner |
| Return of the Jedi | 1983 | 165 | 0 | Fox | Richard Marquand |
| Mighty Ducks | 1991 | 104 | 0 | Disney | Stephen Herek |
| Wayne's World | 1992 | 95 | 0 | Paramount | Penelope Spheeris |
| Lion King | 1994 | 122 | 0 | Disney | Roger Allers |
| Pocahontas | 1995 | 115 | 0 | Disney | Mike Gabriel |
| Addams Family | 1991 | 102 | 0 | Paramount | Barry Sonnenfeld |
| King Kong | 1933 | 0 | 0 | Universal | Merian Cooper |
| King Kong | 1976 | 130 | 0 | Universal | John Guillermin |
| Gone With the Wind | 1939 | 181 | 0 | Paramount | Victor Fleming |
| Basic Instinct | 1986 | 100 | 0 | Paramount | Paul Verhoeven |
| Total Recall | 1990 | 110 | 0 | Fox | Paul Verhoeven |
| Indiana Jones | 1981 | 130 | 0 | Universal | Steven Spielberg |
| The Fugitive | 1993 | 120 | 0 | Universal | Andrew Davies |
| Bridges of Madison County | 1995 | 147 | 0 | Universal | Clint Eastwood |
| Batman Returns | 1992 | 122 | 0 | Warner Brothers | Tim Burton |
| Top Gun | 1986 | 145 | 0 | Fox | Tony Scott |
| Eyes Wide Shut | 1999 | 148 | 0 | Paramount | Stanley Kubrik |
| Moulin Rouge | 2001 | 124 | 0 | Fox | Baz Luhrmann |
| JFK | 1996 | 130 | 0 | Fox | Oliver Stone |
| Dances with Wolves | 1990 | 132 | 0 | Fox | Kevin Costner |
| Hackers | 1989 | 120 | 0 | Universal | George Lucas |

PHP com template engines

Código PHP sem template (filmes.php)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0  
Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-  
transitional.dtd">  
  
<html>  
<head>  
    <title>base de dados de filmes</title>  
    <meta http-equiv="Content-Type"  
content="text/html; charset=iso-8859-1" />  
</head>  
<body>
```

Código PHP sem template (cont)

```
<h2>Filmes</h2>
<table border=2>
<tr>
<th align=left>nome</th>
<th align=left>ano</th>
<th align=left>duração</th>
<th align=left>aCores</th>
<th align=left>estúdio</th>
<th align=left>realizador</th>
</tr>
```

Código PHP sem template (cont)

```
<?php
```

```
include 'inc/db.inc';
```

```
// ligação à base de dados
```

```
$db =
```

```
dbconnect ($hostname, $db_name, $db_user, $db_passwd);
```

```
// criar query numa string
```

```
$query = "SELECT * FROM filmes";
```

```
// executar a query
```

```
$result = mysql_query($query, $db);
```



```

$ntows = mysql_num_rows($result);

//encher a tabela
for($i=0; $i<$ntows; $i++) {
    $tupla = mysql_fetch_array($result,MYSQL_ASSOC);
    $nome = $tupla['nome'];
    $ano = $tupla['ano'];
    $duracao = $tupla['duracao'];
    $aCores = $tupla['acores'];
    $nomeEstudio = $tupla['nomeestudio'];
    $nomeRealizador = $tupla['nomerealizador'];
    printf("<tr>");
    printf("<td>%s</td>", $nome );
    printf("<td>%s</td>", $ano );
    printf("<td align=right>%s</td>", $duracao );
    printf("<td align=center>%s</td>", $aCores );
    printf("<td>%s</td>", $nomeEstudio );
    printf("<td>%s</td>", $nomeRealizador );
}

```

```
        printf("</tr>\n");
    } // end for

    // fechar a ligação à base de dados
    mysql_close($db);
?>
```

```
</table>
```

```
</body>
```

```
</html>
```

Código PHP sem template (cont.)

Conclusão:

- HTML e código PHP misturados
- Código PHP responsável pela criação de HTML

Código PHP com template

- Separação total entre o documento HTML e o código PHP
- O template é um documento HTML que define a estrutura e o layout do documento HTML final
- O código PHP (controlador) é apenas responsável pela atribuição de valores às variáveis (placeholders)

Filmes2_TemplateIT.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">

<html>
<head>
  <title>base de dados de filmes</title>
  <meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1" />
</head>

<body>
<h2>Filmes</h2>
<table border=2>
```

```

<tr>
  <th align=left>nome</th>
  <th align=left>ano</th>
  <th align=left>duração</th>
  <th align=left>aCores</th>
  <th align=left>estúdio</th>
  <th align=left>realizador</th>
</tr>
<!-- BEGIN FILMES -->
<tr>
  <td>{NOME}</td>
  <td>{ANO}</td>
  <td align=right>{DURACAO}</td>
  <td align=center>{ACORES}</td>
  <td>{NOMEESTUDIO}</td>
  <td>{NOMERREALIZADOR}</td>
</tr>
<!-- END FILMES -->
</table></body></html>

```

Filmes2_TemplateIT.php

<?php

```
require_once "HTML/Template/IT.php";
```

```
include 'inc/db.inc';
```

```
// ligação à base de dados
```

```
$db = dbconnect($hostname, $db_name, $db_user, $db_passwd);
```

```
// criar query numa string
```

```
$query = "SELECT * FROM filmes";
```

```
// executar a query
```

```
$result = @ mysql_query($query, $db );
```

```
// Cria um novo objecto template
$template = new HTML_Template_IT('.');

// Carrega o template Filmes2_TemplateIT.tpl
$template->loadTemplatefile('Filmes2_TemplateIT.html',
true, true);

// mostra o resultado da query utilizando o template

$nrows = mysql_num_rows($result);
for($i=0; $i<$nrows; $i++) {
    $tuple = mysql_fetch_array($result,MYSQL_ASSOC);

    // trabalha com o bloco FILMES do template
    $template->setCurrentBlock("FILMES");

    $template->setVariable('NOME', $tuple['nome']);
    $template->setVariable('ANO', $tuple['ano']);
}
```



```
$template->setVariable('DURACAO', $tupla['duracao']);  
$template->setVariable('ACORES', $tupla['acores']);  
$template->setVariable('NOMEESTUDIO', $tupla['nomeestudio']);  
$template->setVariable('NOMEREALIZADOR', $tupla['nomerealizador']);  
  
// Faz o parse do bloco FILMES  
$template->parseCurrentBlock();  
  
} // end for  
  
// Mostra a tabela  
$template->show();  
  
// fecha a ligação à base de dados  
mysql_close($db);  
?>
```

Código PHP com Smarty Template

- Funcionamento idêntico ao Template PEAR IT
- Mais poderoso
- Ligeiramente mais complicada a instalação

Directórios necessários:

```
$ mkdir templates
```

```
$ mkdir templates_c
```

```
$ chmod 777 templates_c
```

(o directório `templates_c` tem que ter permissões de escrita!)

Smarty Template (IT)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">

<html>
<head>
    <title>base de dados de filmes</title>
    <meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1" />
</head>
<body>

<h2>Filmes</h2>
<table border=2>
<tr>
```

```
<th align=left>nome</th>
<th align=left>ano</th>
<th align=left>duração</th>
<th align=left>aCores</th>
<th align=left>estúdio</th>
<th align=left>realizador</th>
</tr>
```

```
{foreach item=filme from=$filmes}
<tr>
  <td>{$filme.nome}</td>
  <td>{$filme.ano}</td>
  <td align=right>{$filme.duracao}</td>
  <td align=center>{$filme.acores}</td>
  <td>{$filme.nomeestudio}</td>
  <td>{$filme.nomerealizador}</td>
</tr>
{/foreach}
</table></body></html>
```

Código PHP com Smarty Template

```
<?php
```

```
include 'db.inc';
```

```
// put full path to Smarty.class.php
```

```
require('/usr/share/php/smarty/libs/Smarty.class.php');
```

```
$smarty = new Smarty();
```

```
$smarty->template_dir = 'templates';
```

```
$smarty->compile_dir = 'templates_c';
```

```
// ligação à base de dados
```

```
$db = dbconnect($hostname, $db_name, $db_user, $db_passwd);
```

```
// criar query numa string
```

```
$query = "SELECT * FROM filmes";
```

```
// executar a query
```

```
$result = mysql_query($query,$db);

// vai buscar o resultado da query

$nrows = mysql_num_rows($result);
for($i=0; $i<$nrows; $i++)
    $films[$i] = mysql_fetch_array($result,MYSQL_ASSOC);

// faz a atribuição das variáveis do template smarty
$smarty->assign('filmes',$films);

// Mostra a tabela
$smarty->display('Filmes2_Smarty.tpl');

// fechar a ligação à base de dados
mysql_close($db);
?>
```

Para saber mais

- Template PEAR IT:

http://pear.php.net/package/HTML_Template_IT

- Template Smarty:

<http://www.smarty.net/>

- Exemplos:

<http://intranet.deei.fct.uaig.pt/DAW/templates/files.html>