

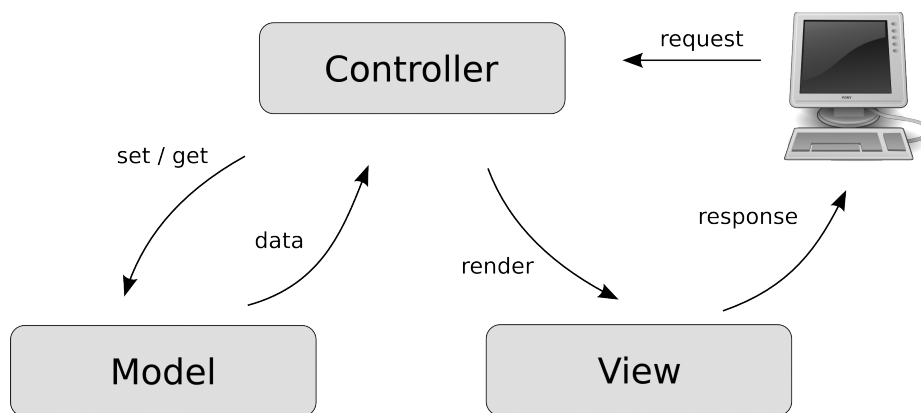
Módulo 6 – Apresentação e controle

Neste workbook do módulo 6, revisaremos os principais conceitos vistos nas videoaulas.

MVC

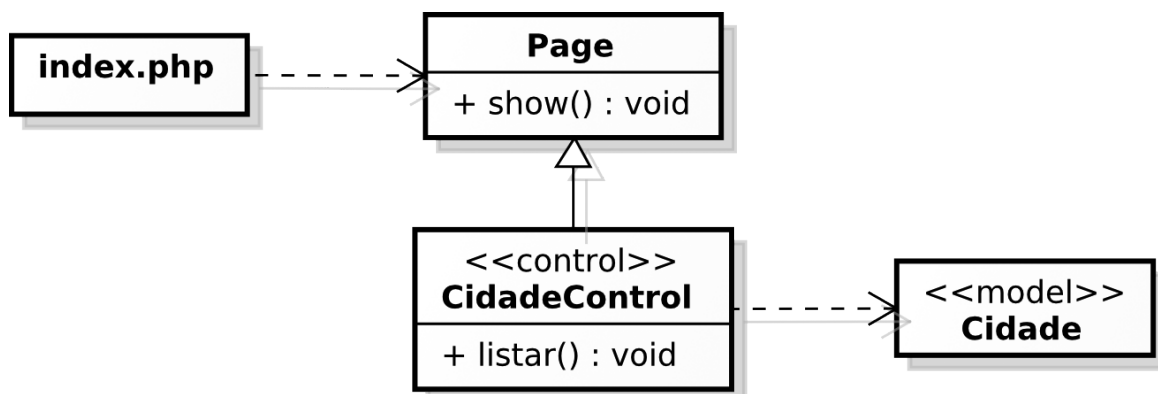
Um padrão de projetos no qual as classes são separadas em responsabilidades:

- Model: Classe que representa o modelo de dados.
- View: Classe que representa a visualização (fronteira de interface).
- Controller: Classe que controla os inputs, e delega as ações.



Front Controller

Um mecanismo frontal que recebe todas requisições do sistema. Pode concentrar ações relativas à sessão, timezone, segurança, internacionalização, dentre outros.



index.php

```
<?php
if ($_GET) {
    $class = $_GET['class'];
    if (class_exists($class)) {
        $pagina = new $class;
        $pagina->show();
    }
}
```

REST

Exemplo de servidor REST. Recebe a classe, o método e o request, e executa aquele método codificando o retorno no formato JSON.

rest.php

```
<?php
header('Content-Type: application/json; charset=utf-8');

...

class LivroRestServer {
    public static function run($request) {
        $class = isset($request['class']) ? $request['class'] : '';
        $method = isset($request['method']) ? $request['method'] : '';
        $response = NULL;

        try {
            if (class_exists($class)) {
                if (method_exists($class, $method)) {
                    $response = call_user_func(array($class, $method), $request);
                    return json_encode( array('status' => 'success', 'data' => $response));
                }
            }
        }
        catch (Exception $e) {
            return json_encode( array('status' => 'error', 'data' => $e->getMessage()));
        }
    }
}

print LivroRestServer::run($_REQUEST);
```

Componentes

Classes que representam componentes de interface. Bom índice de reaproveitamento, uso de herança, composição.

Lib/Livro/Widgets/Form/SimpleForm.php

```
<?php
namespace Livro\Widgets\Form;

class SimpleForm {
    private $name, $action, $fields, $title;

    public function __construct($name) {
        $this->name = $name;
        $this->fields = array();
        $this->title = '';
    }

    public function setTitle($title) {
        $this->title = $title;
    }
}
```

Template View

Separa a apresentação por meio do uso de gabaritos HTML. Ex: Biblioteca Twig.

App/Control/TwigSampleControl.php

```
<?php
use Livro\Control\Page;

class TwigSampleControl extends Page {
    public function __construct() {
        parent::__construct();

        $loader = new Twig_Loader_Filesystem('App/Resources');
        $twig = new Twig_Environment($loader);
        $template = $twig->loadTemplate('form.html');

        $replaces = array();
        $replaces['title'] = 'Título';
        $replaces['action'] = 'index.php?class=TwigSampleControl&method=onGravar';
        $replaces['nome'] = 'Maria';
        $replaces['endereco'] = 'Rua das flores';
        $replaces['telefone'] = '(51) 1234-5678';

        $content = $template->render($replaces);
        echo $content;
    }
}
```

Livro PHP Programando com Orientação a Objetos



www.adianti.com.br/phpoo