

# Desenvolvimento PHP com MVC

Noção sobre desenvolvimento Web

O que são Padrões de Projeto

Padrões e o Universo de TI

Conhecendo os Padrões de Projeto

O MVC

As Camadas do MVC e o PHP

Mergulhando no MVC

# Objetivos do Curso

Apresentar uma visão geral sobre Design Pattern;

Dar uma visão sólida sobre MVC;

Entender como o MVC se comporta sozinho;

Explicar como o MVC se integra a alguns outros Patterns;

Entender como podemos aplicar MVC em uma aplicação PHP;

Servir de ponto de apoio para seu desenvolvimento

# Não são Objetivos

Dar capacidade de maestria em MVC;

Ser um curso único de MVC ou de programação em PHP;

Ser consultoria para sua empresa/produtos;

Substituir publicações ou estudo;

Apresentar outros Design Patterns;

Ser um treinamento de grande profundidade.

# Padrões de Projeto(DP)

Proposto para a área de Construção Civil na década de 70 por Christopher Alexandre;

Padrões de soluções para problemas comuns;

1987 primeiros padrões de projeto para computação;

1995 Lançamento do Livro Design Patterns;

Popularização em C++;

Independente de OOP, mas recomendável usar o Paradigma.

# Características de Um Padrão

Generalidade;

Equilíbrio;

Abstração;

Abertura;

Combinatoriedade.

# Vantagens de Usar Patterns

Redução do tempo de codificação

Reaproveitamento de Código;

Variedade;

Possibilidade de Generalização de sistema;

Menor tempo de manutenção;

Melhor interação na equipe de desenvolvimento;

Padronização da codificação.

# Frameworks que usam MVC

CakePHP;

Symfony;

Code Igniter;

Zend Framework;

Biscuit;

Prado;

Outros.

# Características do MVC

- Separação das Camadas de Desenvolvimento;
- Separação das Lógicas(Apresentação e Aplicação);
- Melhor Reaproveitamento do Código;
- Maior Facilidade de Documentação e Manutenção;
- Permite o desenvolvimento de multiplas aplicações;
- Separa o design da Programação;

# O MVC (Modelo Visualização Controle)

O MVC é um padrão de projeto baseado no problema de reescrever o código para cada página, ou tela, que fôssemos produzir, sendo que com a mistura entre rotinas de DML (Linguagem de Manipulação de Dados), montagem de interfaces e controle dos comportamentos. Tudo isso junto muitas vezes gerava um código ilegível o que dificultava a vida dos desenvolvedores.

Assim foi proposto o MVC como forma de resolver esse conjunto de problemas extremamente recorrente no universo do desenvolvimento de sistemas.

# A Camada de Modelo

Manipulação de arquivos texto

Sistemas de arquivos

XML

DML

Validações de dados \*

Acesso somente por um controle

\*Existe uma polêmica sobre aonde devemos manter as validações, se no controle ou no modelo, eu pessoalmente prefiro o modelo

# A Camada de Controle

Recebe as interações da visão;

Dispara o modelo;

Recebe a resposta do modelo;

Devolve a resposta que o modelo deve apresentar;

Ajax é acionado aqui e descarrega em uma view;

Pode, e recomenda-se, integrar-se com um  
FontControll

# A Camada de Visão

Pode um sistema de templates, como o Smarty;

Carrega os frontEnds gráficos, CSS, Javascript, XML para XSLT etc.;

Envia as interações para o controle;

Recebe a resposta do controle e trata de exibir de uma maneira humanamente legível ;