

DESENVOLVIMENTO PARA WEB II

Apresentação da disciplina



Professor: Alexandre Strapção Guedes Vianna – IFPE Igarassu
Alexandrevianna.net

Roteiro da Aula



1. Apresentar a Disciplina
 - a. Avaliações
2. Breve motivação sobre desenvolvimento para Web em Java

Cronograma 1º Unidade

01	Apresentação da Disciplina e Cronograma
02	Conceitos de Frameworks Web MVC
03	Conceitos de Frameworks Web MVC
04	Introdução ao CakePHP – Estrutura e Instalação
05	1º Aplicação com o CakePHP
06	1º Aplicação com o CakePHP
07	Scaffolding
08	Scaffolding
09	CRU Básico
10	ORM

Cronograma 1º Unidade

11	ORM
12	Segurança e Autenticação Usuários
13	Segurança e Autenticação Usuários
14	Segurança e Autenticação Usuários
15	Templates
16	Templates
17	Templates
18	REVISÃO
19	AVALIAÇÃO 1º UNIDADE
20	RESOLUÇÃO DA AVALIAÇÃO

Cronograma 2º Unidade

21	Rotas
22	Rotas
23	Segurança Avançado
24	Segurança Avançado
25	Helpers
26	Helpers
27	E-mails
28	E-mails
29	Migrations
30	Migrations

Cronograma 2º Unidade

31	ORM – Funcionalidades Avançadas
32	ORM – Funcionalidades Avançadas
33	ORM – Funcionalidades Avançadas
34	ORM – Funcionalidades Avançadas
35	REST
36	REST
37	REST
38	REVISÃO
39	AVALIAÇÃO 2º UNIDADE
40	ENTREGA DO PROJETO

Framework ?

Ferramentas...



Framework ?

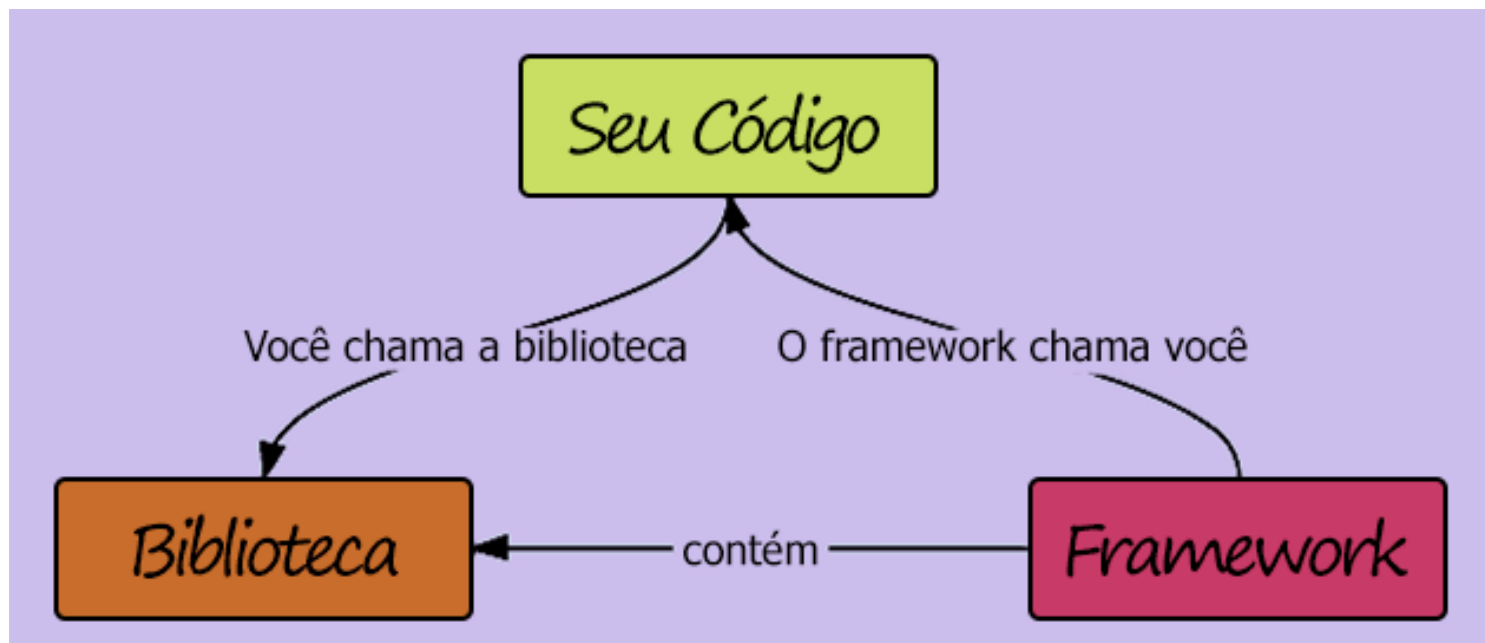
Um framework normalmente é um conjunto de bibliotecas para conseguir executar uma operação maior. É comum um framework encapsular os comportamentos da API em implementações mais complexas permitindo o seu uso de forma mais flexível, frequentemente através de extensões, configurações e inversões de controle. Como pode ser considerada uma camada em cima da API eventualmente pode simplificá-la em certo sentido. Ele costuma dar consistência a um conjunto de bibliotecas (pena que acontece o oposto em alguns casos).

Framework ?

O framework costuma "tomar conta" da sua aplicação deixando "portas" para você acessar o que é importante para o seu objetivo. É comum ter uma grande interdependência entre seus componentes. Por estas duas razões, o framework pode se tornar um fardo quando ele não é bem feito, não é adequado ao que o programador precisa ou quando o programador não sabe bem como usá-lo.

Framework ?

Um *framework* chama você. É um estilo de vida. Seu projeto se compromete com ele.



Framework ?

Frameworks podem ser entendidos como plataformas de desenvolvimento. Eles possuem lacunas que devem ser preenchidas pelo programador para funcionar conforme sua necessidade. Pode ser visto como um esqueleto de uma aplicação. Frequentemente são confundidos, de forma certa ou errada, como um conjunto de classes.

Framework ?

Algumas vezes o termo é usado para determinar um conjunto de arquivos de bibliotecas de códigos binários, ou fontes ou até outros recursos para a aplicação que são relacionados.

Exemplos são os sistemas de GUI, de operações web, alguns ORMs, um conjunto de bibliotecas padrão de uma linguagem de programação, implementação de um padrão MVC, etc.

Algumas vezes uma biblioteca é ou se torna tão complexa que acaba virando um framework. Se isto é bom ou ruim deixo para você decidir.

Framework ?

Porque usar um framework de PHP?

Os desenvolvedores utilizam frameworks por vários motivos, e o maior deles é para agilizar o processo de desenvolvimento. A re-utilização de código em vários projetos vai economizar muito tempo e trabalho... Isso é garantido pois o framework já traz uma série de módulos pré-configurados (e funcionando) para fazer as mais variadas e comuns tarefas como envio de e-mails, conexão com o banco de dados, sanitização (limpeza) de dados e proteção contra ataques.

Estabilidade é outra grande vantagem dos frameworks.

Framework ?

E no contexto de Web ?

Acesso fácil aos dados, padronização, qualidade, manutenção, autenticação de usuários, rotas, segurança, performance...

Framework ?

Vantagens:

- Padronização: uma das grandes vantagens de um framework é a padronização do desenvolvimento. Por termos um conjunto já definido de classes e/ou funções, somos "forçados" a trabalhar conforme a ferramenta de escolha.
- Velocidade de desenvolvimento: por fazermos uso de módulos genéricos economizamos tempo por não ter que ficar "reinventando a roda" em cada projeto.
- Qualidade: os principais frameworks do mercado são muito bem testados em versões alfa, beta e release candidate (RC) além de serem mantidos por comunidades e/ou empresas experientes.

Framework ?

Vantagens:

- Manutenção: por seguirmos um padrão e termos código de alta qualidade, ganhamos na facilidade de manutenção, uma vez que sabemos onde encontrar o que precisamos.
- Comunidade: excelente meio de aprendizado onde podemos obter ajuda e ganhar experiência não só com a ferramenta e a linguagem mas com a cultura e forma de pensar.
- Segurança: sem dúvidas é um dos quesitos que preocupa muita gente (e com razão) e que é um fator muito considerado para lançamentos de patches e atualização nos frameworks mais conhecidos.

Framework ?

CakePHP



O CakePHP é uma grande opção para iniciantes e desenvolvedores avançados. Ele foi criado usando as bases e modelos do Ruby on Rails e é pesadamente focado no desenvolvimento ágil e rápido. Recentemente ele tem se tornado muito famoso por sua simplicidade e facilidade de uso.

Leitura recomendada

Frameworks no PHP: O que, quando, porque e qual? :

<http://blog.thiagobelem.net/frameworks-no-php-o-que-quando-porque-e-qual>

Porque Usar um Framework?

<http://glaucocustodio.com/2012/07/31/porque-usar-um-framework/>

pt.stackoverflow.com

<http://book.cakephp.org/3.0/pt/index.html>