

A UTILIZAÇÃO DE FRAMEWORKS LIVRES NO DESENVOLVIMENTO DE APLICAÇÕES WEB

Alyne Oliveira
Caroline Kinoshita
Erico Vinicius
João Vitor Figueiredo
Pedro Lacerda
Pedro Pádua

1. INTRODUÇÃO:

O desenvolvimento de aplicações web pode ser bastante facilitado através da utilização dos chamados Frameworks. Existem diversos exemplos de Frameworks, sendo vários deles softwares livres, que podem ser usados em conjunto com linguagens de programação para a criação de aplicações web. Como exemplo, temos o Django, que utiliza a linguagem Python, e o Rails, que utiliza o Ruby. Esse trabalho tem como o objetivo mostrar como a utilização de Frameworks pode acelerar e auxiliar o processo de criação e manutenção de aplicações web.

2. DOS FATOS

• O que é uma aplicação web?

Inicialmente, as aplicações possuíam o código subdividido, parte era mantido em um servidor e a outra parte rodava localmente na estação do usuário. Caso surgisse alguma atualização para o código da estação do usuário. A idéia da aplicação web é acabar com essa separação, centrando tudo que é necessário em um servidor, de maneira que esse “conteúdo” possa ser acessado de forma mais ampla e simplificada. Podemos dizer que aplicações web são sistemas projetados para serem acessados através de uma rede, seja ela a internet ou intranet, de forma em que nada é instalado nos computadores dos clientes.

• O que é um framework?

O framework pode ser definido como uma união de códigos comuns a diferentes projetos, possibilitando e facilitando a implementação de uma funcionalidade comum. O código mais genérico garante uma melhor reutilização. Existem frameworks para a maioria das linguagens, podendo ser desenvolvidos de acordo com a necessidade de um programador.

Pode-se dizer que é uma ferramenta facilitadora no momento da implementação de funcionalidades mais genéricas, padronizando, acelerando e melhorando o desenvolvimento de um projeto.

• Vantagens em se utilizar um framework para o desenvolvimento de aplicações Web

Os principais objetivos do uso de Frameworks é agilizar o processo de desenvolvimento e aumentar a qualidade, tanto da aplicação como do código e artefatos gerados no desenvolvimento da aplicação. Neste sentido, o que determina tais objetivos são as características do Framework, que proporcionarão benefícios para o desenvolvimento de aplicações. Tais características são:

- **Modularidade:** permite o encapsulando de detalhes da execução.
- **Reusabilidade:** possibilita a reaplicação de componentes genéricos em aplicações distintas.
- **Extensibilidade:** fornece recursos que permitem a especialização das funcionalidades nativas do Framework.
- **Integração:** permite a integração com outras aplicações e sistemas.
- **Padronização de projetos:** projetos distintos podem ser tratados com um mesmo padrão de Framework. Estes padrões representam soluções genéricas que podem ser reutilizadas e especializadas para problemas particulares.
- **Confiabilidade:** o Framework torna-se cada vez mais confiável a medida que este é utilizado massivamente por diferentes equipes e projetos.
- **Segurança.** o projeto do Framework pode prover requisitos de segurança de dados, acessos e transações; encapsulando a tal requisito.
- **Compatibilidade:** internamente o framework pode tratar diferenças de como os entre navegadores.
- **Legibilidade:** códigos estruturados e padronizados proporcionam melhor interpretação deste.

- **Django**

Django é um framework de código aberto para o desenvolvimento escrito na linguagem Python, e que segue o padrão MVC (model-view-controller). Foi criado originalmente para gerenciar vários sites de notícias do "The World Company", portal jornalístico da cidade de Lawrence, no Kansas, Estados Unidos. Seu nome foi inspirado no guitarrista de jazz Django Reinhardt. O framework tornou-se um projeto open source, publicado sob a licença BSD em Julho de 2005. Em Junho de 2008 foi anunciado que a Django Software Foundation iria ser responsável pelo projeto.

O Django tem como objetivo principal facilitar a criação de aplicações web complexas, que fazem o uso de bancos de dados, de maneira extremamente ágil. Por seguir e utilizar o princípio DRY, "Don't Repeat Yourself", o desenvolvedor aproveita ao máximo o código já feito. O desenvolvimento de websites de alta performance é mais rápido e com menos código. Ele enfatiza a reusabilidade e a possibilidade de se "plugar" e "desplugar" seus componentes. A linguagem Python é utilizada por todo o sistema: nos arquivos de configuração, regras de negócio e modelos de dados.

As camadas do MVC do Django consistem em:

- **Model:** Um mapeador objeto relacional que faz a ligação entre os modelos de dados, definidos como classes em Python, e o banco de dados relacional. Torna possível gerar tabelas no banco de dados e manipulá-las sem a necessidade de código SQL.
- **View:** Sistema de templates utilizado para projetar as interfaces que processam as requisições realizadas pelo usuário. Formulários podem ser gerados automaticamente podendo ser validados e utilizados para armazenar os dados no banco. Separa a parte de design e conteúdo do código das regras de negócio.
- **Controller:** Sistema faz o mapeamento das URL's da aplicação para as regras de negócio, que executam diversos tipos de ações, como consultar os dados no banco e mandar apresentar os dados na tela.

O Django fornece uma interface de administração, gerada de maneira automática e completamente pronta para a utilização, para criar, acessar, atualizar e remover os dados cadastrados no banco.

- **Ruby on Rails**

Ruby on Rails é um framework livre criado por David Heinemeier Hansson, um dos fundadores da 37Signals, para servir de base para as aplicações desenvolvidas em sua empresa que utilizavam a linguagem Ruby. Em 2003, ele disponibilizou o framework, assim como o seu código para a comunidade que hoje possui mais de 1.800 contribuidores. Este framework é considerado um meta-framework, por agrupar em si diversos outros frameworks disponíveis para Ruby, sendo eles: Active Record, responsável pela ligação entre a camada lógica do sistema e o banco de dados; Action Pack, responsável pela geração da interface com o usuário e as regras de negócio; Action Mailer, responsável pela entrega e

recebimento de e-mails; e Active Support, compreende diversas classes e bibliotecas utilizadas em aplicações desenvolvidas utilizando Ruby.

O objetivo do Rails é aumentar a produtividade dos desenvolvedores provendo uma série de recursos que agilizam a codificação. Como o Django, também utiliza o conceito DRY e prioriza as convenções ao invés de configuração manual exaustiva. Outra grande vantagem na utilização do Ruby on Rails é que a curva de aprendizado é menor, possibilitando os desenvolvedores com pouca experiência no framework desenvolver aplicações rapidamente com excelente qualidade.

3. CONCLUSÃO

A crescente popularidade e rotatividade de sites levaram a necessidade da criação de técnicas de desenvolvimento aprimoradas, que possam simplificar o processo de criação e manutenção desses portais.

Neste artigo foram apresentados dois frameworks: Django, que utiliza a linguagem Python; e Rails, que utiliza a Ruby. Eles são referência para o desenvolvimento com ênfase em aplicações web. Os frameworks propostos são completamente estruturados a partir de tecnologias livres de custos, tornando-se, portanto, economicamente viável ao mesmo tempo em que auxilia desenvolvedores que estão iniciando seus estudos no desenvolvimento de aplicações web.

Ambos frameworks possuem código aberto. Comparados aos frameworks privados, não deixam nada a desejar, o desenvolvimento e produtividade são equivalentes, senão superiores a algumas ferramentas que tem um alto custo para aquisição.

4. REFERÊNCIAS BIBLIOGRÁFICAS

http://pt.wikipedia.org/wiki/Ruby_on_Rails

<http://codeigniterbrasil.com/passos-iniciais/o-que-e-um-framework-definicao-e-beneficios-de-se-usar-frameworks/>

[http://en.wikipedia.org/wiki/Django_\(Web_framework\)](http://en.wikipedia.org/wiki/Django_(Web_framework))

http://en.wikipedia.org/wiki/Web_application

[http://pt.wikipedia.org/wiki/Django_\(framework_web\)](http://pt.wikipedia.org/wiki/Django_(framework_web))

<http://pt.wikipedia.org/wiki/Framework>

<http://rubyonrails.org/>

http://www.oficinadanet.com.br/artigo/1294/framework_o_que_e_e_para_que_serve

<http://www.rnp.br/newsgen/9710/n5-3.html>

<https://www.djangoproject.com/>