

UNIVERSIDADE FEDERAL DE PERNAMBUCO  
CENTRO DE INFORMÁTICA  
MESTRADO PROFISSIONAL EM CIÊNCIA DA COMPUTAÇÃO



ATIVIDADE AVALIATIVA:  
RELATÓRIO - DIFERENÇAS OBSERVADAS ENTRE OS FRAMEWORKS RAILS E GRAILS  
DURANTE A IMPLEMENTAÇÃO DE PROJETOS DE PRÁTICA

**Disciplina: Tecnologia de Programação para a Internet**

Alunos:  
ALAN DE OLIVEIRA SILVA  
BRUCE FÁBIAN REIS ALBUQUERQUE  
ERLAN FONSECA DE SOUZA  
MARCOS ADRIEL SAMPAIO ROST

**Área de concentração:** Sistemas de Informação

Recife-PE  
2015

## 1. Introdução:

O presente relatório apresenta algumas observações feitas durante a implementação de projetos de prática apresentados como parte da avaliação para as disciplinas de Engenharia de Software e de Tecnologias de Programação para a Internet, utilizando os frameworks de desenvolvimento *Ruby on Rails* e *Grails* respectivamente. Esses frameworks agilizam o processo de desenvolvimento de aplicações para a internet, tornando-o menos complicado e possibilitando uma melhor manutenibilidade das aplicações criadas.

O *Ruby on Rails* é um framework *open-source* escrito na linguagem de programação *Ruby*, criado por David Heinemeier Hansson, a partir de um produto de sua empresa, chamado “Basecamp”. Desse modo, esse framework oferece recursos que visam tornar a experiência do desenvolvedor mais agradável, baseando-se nos seguintes princípios: convenção à configuração; não se repita; e, automação de tarefas repetidas.

Por sua vez, o *Grails* é um framework que propõe a aplicação dos mesmos princípios citados acima ao desenvolvimento de aplicações para a internet na plataforma Java. Para isso, utiliza-se da linguagem de programação de *Groovy* e, por estar sua profunda integração com a plataforma Java, pode integrar-se a bibliotecas e códigos Java. Desse modo, esse framework apresenta a proposta de aumento da produtividade, ao preservar o programador de detalhes de configuração.

## 2. Diferenças observadas:

Os frameworks *Ruby on Rails* e *Grails* apresentam muitas similaridades, as quais atribuímos à proximidade entre as motivações que levaram à criação dessas duas ferramentas. De qualquer, as experiências de implementação em ambos os ambientes nos permitiu observar algumas diferenças que apresentamos no quadro a seguir:

<i>Ruby on Rails</i>	<i>Grails</i>
<ul style="list-style-type: none"><li>• Linguagem de desenvolvimento é o <i>Ruby</i>.</li><li>• Persistência de dados por meio do <i>ActiveRecord</i>.</li><li>• Permite apenas tipagem dinâmica de variáveis.</li><li>• Utiliza-se do framework de testes</li></ul>	<ul style="list-style-type: none"><li>• Linguagem de desenvolvimento é o <i>Groovy</i> e/ou o Java explícito.</li><li>• Persistência de dados por meio do <i>Hibernate</i>.</li><li>• Permite tanto a tipagem estática de variáveis, quanto a dinâmica.</li><li>• Utiliza-se do framework de testes</li></ul>

Rspec. <ul style="list-style-type: none"> <li>• Apresentou maior leveza e menor consumo de recursos do hardware pela aplicação criada.</li> <li>• Apresentou uma estrutura de pastas intuitiva, facilitando a configuração do ambiente.</li> <li>• Documentação extensa e atualizada, mantida pela comunidade.</li> </ul>	JUnit. <ul style="list-style-type: none"> <li>• Apresentou um alto consumo de recursos de hardware para execução da aplicação criada.</li> <li>• Apresentou uma estrutura de pastas mais complexa, o que dificultou a configuração do ambiente.</li> <li>• Pode ser caracterizado como um “framework de frameworks” Java, por integrar Spring, Hibernate, Gradle, etc.</li> <li>• Ampla documentação, porém mostrou-se mais limitada.</li> </ul>
---	--

### 3. Conclusões:

A realização das atividades de prática propostas pelas disciplinas de Engenharia de Software e de Tecnologias de Programação para a Internet nos possibilitou estabelecermos nosso primeiro contato com esses dois importantes frameworks de desenvolvimento para a internet, os quais representam grandes comunidades de desenvolvedores no mundo. Em ambos frameworks, notamos a preocupação em oferecer um ambiente de programação que possibilite agilidade e produtividade, e que seja seguro e estável. Ao mesmo tempo, nos vimos diante de uma considerável curva de aprendizagem se quisermos adotar qualquer um desses ambientes para a implementação de soluções em nossos locais de trabalho hoje.

### Referências:

DAVIS, Scott; RUDOLPH, Jason. *Getting started with Grails*. 2 ed. C4Media, InfoQ.com, 2010. Disponível em <<http://www.infoq.com/minibooks/grails-getting-started>>, acessado em 21/08/2015.

BECKWITH, Burt. *Programming Grails*. Gravenstein Highway North, Sebastopol: O'Reilly Media, 2013.

FLANAGAN, David; MATSUMOTO, Yukihiro. *The Ruby programming language*. Gravenstein Highway North, Sebastopol: O'Reilly Media, 2008.

FUENTES, Vinícius. *Ruby on Rails*. São Paulo: Casa do Código, 2012.