# Tópicos especiais de desenvolvimento web
# Trabalho Final

Cauê Haucke Porta Guerra
Cecilia Fernandes

27 de junho de 2009

# 1 Análise de recursos da Web2.0

O site escolhido foi a Amazon

## 1.1 Funcionalidades Web2.0

**Sistemas de recomendação**

Baseado em visualizações anteriores ou compras de um usuário, escolhe produtos que possam interessar a quem tem um histórico de gostos similares ou simplesmente produtos semelhantes àquele sendo visto.

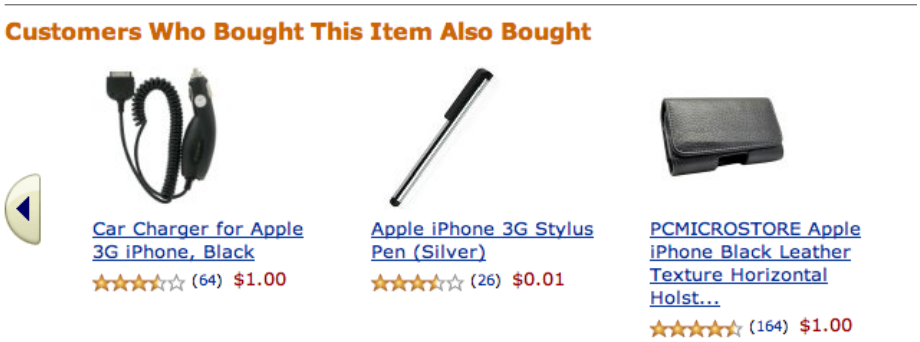


Figura 1: Recomendação baseada em compras

**Buscas, redes neurais e extração de informações de textos**

Digitando um texto ou palavras soltas no campo de busca do *website* da Amazon, a busca retorna resultados pertinentes. Além disso, conforme escrevemos uma palavra na busca, a página responde com uma lista das possíveis buscas que está sendo feita.

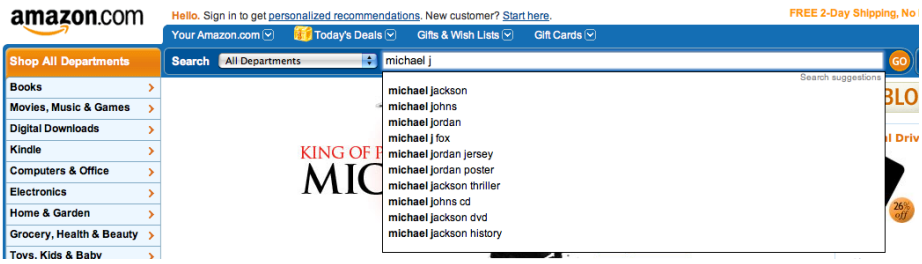


Figura 2: Busca textual

Certamente, também, um site do porte da Amazon, utiliza *Data Mining*, otimização, árvores de decisão, entre outros mecanismos da *Web2.0*.

# 2 Sistema desenvolvido

## 2.1 Especificação

O tema escolhido para o sistema foi um *site* de anúncio de vagas de empregos. Nele, empresas podem se logar e criar vagas de emprego, enquanto usuários, que não precisam se logar, são capazes de ver vagas disponíveis para sua especialidade. O site atual de cadastro de vagas de informática mais famoso do Brasil não conta com recursos web 2.0 e é terrível para procurar vagas interessantes. Foi exatamente isso que nos motivou a escolher esse tema.

Optamos por fazer a segunda e terceira fases do sistema de forma completamente diferente – mesmo a linguagem de programação variou. Assim, avaliamos abaixo, separadamente, as segunda e terceira fases.

Durante todo o desenvolvimento, fases 1, 2 e 3, pareamos na codificação dos sistemas. Participamos, assim, igualmente e ativamente nos produtos entregues.

## 2.2 Até a fase 2

A fase 2 refatorou o trabalho da fase 1 para que eles utilizassem as dez tecnologias que o professor julgou mais relevantes no mundo do Java atual. Escrevemos aqui sobre o que foi implementado durante toda a fase 2 e nota-se que algumas tecnologias de fato ficaram de fora. Algumas delas não foram implementadas, simplesmente, e outras não estão na versão final, mas podem ser encontradas em outras versões, nas entregas mais antigas no Paca.

### 2.2.1 Arquitetura

Já desde a fase 1, utilizamos arquitetura MVC na implementação do sistema. Por ser um sistema web padrão, essa arquitetura é simples e ideal para melhorar a separação entre modelo e visualização. Durante a fase 2, refatoramos o sistema para remover nosso *framework* caseiro feito em Servlets para colocar Struts e, posteriormente VRaptor. Ambas as bibliotecas ajudam a implementar e manter o MVC, isto é, a separação entre modelo, controlador e visualização.

### 2.2.2 Tecnologias escolhidas

- **Jquery + Ajax:** ao criar um usuário, uma chamada em Ajax para o servidor verifica se o nome de usuário já está sendo utilizado;
- **Struts:** um dos mais famosos e utilizados *frameworks* em Java para a web, o Struts poderia tomar o lugar da servlet principal da primeira fase;
- **Hibernate e JPA:** desde o princípio, utilizamos o Hibernate (a principal implementação da JPA) para ajudar nos CRUDs, na criação do banco e outros pormenores da persistência de dados;
- **JUnit e Selenium:** foram nossa apresentação. Ambos são bastante usados para, respectivamente, testes unitários e de aceitação/integração;

- **REST:** entregamos um sistema feito em VRaptor3 (framework de desenvolvimento web similar ao Struts), que possui suporte a REST
- **Ruby on Rails:** fizemos um sistema bem simples com apenas um modelo (vagas) e utilizando o scaffold.

## 2.3 O sistema desenvolvido na fase 3

### 2.3.1 Funcionalidades

As funcionalidades entregues pelo grupo foram:

- tags - tagcloud com as vagas cadastradas;
- busca textual - fazendo pesquisas por vagas. Utilizamos todo o processo de buscas textuais, como tokenização, normalização, steemming, lista de stopwords;
- REST - nosso sistema é RESTful, o que significa basicamente que pra uma mesma url temos ações diferentes que podem ser executadas de acordo com o método HTTP que estiver sendo utilizado para a requisição;
- autenticação e autorização - usado para o cadastramento de novas vagas e deve ser feito apenas por empresas interessadas em anunciar uma nova vaga;

### 2.3.2 Arquitetura

A arquitetura utilizada é a mesma utilizada em todo e qualquer projeto em Ruby on Rails. É utilizado o padrão MVC, com uma separação clara de responsabilidades, como demonstrado na screenshot abaixo.

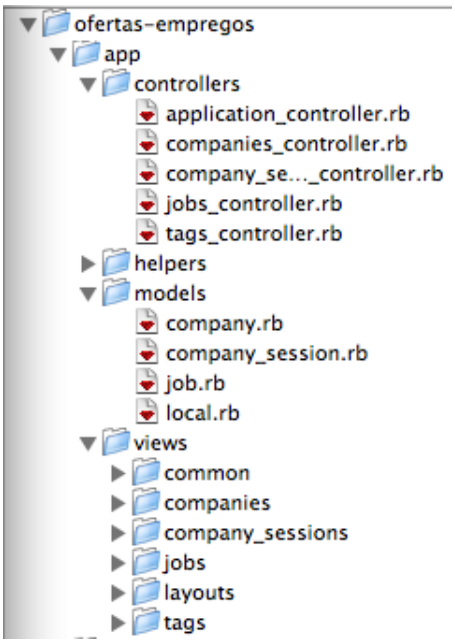


Figura 3: MVC de projetos em Ruby on Rails

Esse foi um dos motivos pela escolha de Ruby on Rails como linguagem que seria utilizada para a realização do projeto final: muita convenção e pouca configuração, resultando em economia de tempo.

### 2.3.3 Tecnologias escolhidas

Escolhemos utilizar Ruby on Rails, pois temos bastante familiaridade com essa linguagem/framework e além disso muitas das APIs disponibilizadas pelos outros grupos, estavam muito acopladas com os projetos deles e, portanto, daria um trabalho bem maior para a reutilização daqueles códigos. Além disso, utilizamos algumas *gems* (espécie de plugins e Jars para o mundo Ruby), que acrescentava suporte de maneira bem facilitada às funcionalidades solicitadas na fase 3. As *gems* utilizadas foram:

- act_as_taggable_on_steroids – para uso de tags / tagcloud;
- act_as_solr – busca textual;
- auth_logic – autenticação e autorização;

Além disso, ainda utilizamos javascript / AJAX, através da biblioteca Prototype, para verificar por tags que já foram escolhidas e sugerir tags para nossos usuários ao cadastrarem uma nova vaga. Prototype é um concorrente do JQuery, apresentado na fase 2.

# 3 Aproveitamento da disciplina

## 3.1 Pontos positivos

Um ponto positivo foi que tivemos uma matéria de Sistemas de cunho prático. Além disso, consideramos esta uma tentativa muito válida em aproximar os alunos às tecnologias que serão encontradas no mercado de trabalho.

## 3.2 Pontos negativos

Como ponto negativo, acreditamos que a carga de trabalho extra-classe foi muito grande e não de acordo com o esperado dados os 2 créditos trabalho da ementa da disciplina. Sentimos também falta do conhecimento prévio do professor sobre os temas abordados. As aulas de apresentação de Java para a web no início do semestre passaram muito rápido pela parte mais relevante para aqueles que não conheciam a linguagem ou a parte *web* dela.

Cremos, também, que foram ditas informações erradas em diversas palestras, o que poderia ter sido evitado (ou corrigido imediatamente) pelo professor. Em particular, um erro que permenece até o momento é que JSF é uma API do Java não de visualização, mas um *framework web* baseado em componentes. Não se trata da camada de visualização, mas da de controle. Não nos sentimos no direito de intervir nas apresentações de colegas já bastante nervosos apenas por estarem em destaque.

No mais, implementar um mesmo sistema com N tecnologias apenas porque elas estão em evidência é um dos erros que mais se vê em empresas que dizem usar tecnologia de ponta. Vê-se muitos projetos perdendo o foco por querer agregar "apenas mais essa" tecnologia do momento. Talvez fosse interessante deixar claro que não se deve usar tecnologias sem motivo real para elas – ao invés de estimular que os grupos o fizessem. Não há real razão para utilizar Struts junto com JSF, por exemplo, e vimos grupos que não o fizeram serem questionados a esse respeito.

## 3.3 Tópicos abordados

A seleção dos tópicos foi surpreendentemente representativa. Apenas, acreditamos que poderia-se eliminar o JSF (ou o Struts), o AOP e também MashUps em pról de, por exemplo, uma aula a mais de Ruby on Rails e ferramentas de automação de *build* como Ant, Ivy ou Maven.

## 3.4 Dedicação à matéria

O grupo não se dedicou muito a matéria por se tratar de um tema repetido para nós. Ambos trabalhamos com todas essas tecnologias diariamente, logo o conteúdo não representava nenhuma grande novidade, que fosse capaz de motivar e aumentar o interesse. O principal motivo nosso ao escolher a matéria foi ver o foco que seria dado numa matéria de prática sobre sistemas no IME.