

Universidade de São Paulo
Instituto de Matemática e Estatística

MAC5855 – Tópicos de Desenvolvimento para Web

Relatório Final

Leilão Online

Professor: Marco Gerosa

Alexandre Albano – NUSP: 4894836
Filipe Ferraz Salgado – NUSP: 4894728

28/06/2009

1 Análise das funcionalidades Web 2.0 do site Amazon.com

1.1 Busca

Dada uma consulta, uma palavra ou frase, uma busca é o processo de retornar documentos que são relevantes à consulta feita. A lista de documentos é geralmente devolvida em alguma ordem de relevância. Uma busca é basicamente o processo de criação de um vetor de termos que representa a consulta e a devolução daqueles documentos cujos vetores de termos são mais similares ao vetor da consulta.

The screenshot shows the Amazon search interface. At the top, there is a search bar with the text "High Performance" and a "GO" button. To the right are links for "Cart" and "Your Lists". Below the search bar, the results are displayed under the heading "High Performance". The first result is the book "High Performance MySQL: Optimization, Backups, Replication, and More" by Baron Schwartz, Peter Zaitsev, Vadim Tkachenko, and Jeremy Zawodny. It is a paperback from June 26, 2008, priced at \$31.49. The second result is "Designing Organizations for High Performance" by David P. Hanna, a paperback from January 11, 1988, priced at \$60.00. The third result is "High Performance Mind" by Anna Wise, an audio CD from March 17, 1998, priced at \$65.59. The fourth result is "The Wisdom of Teams: Creating the High-Performance Organization" by Jon R. Katzenbach and Douglas K. Smith, available as a Kindle edition.

Figura 1: Exemplo de busca

1.2 Tags

Tagging é o processo de adicionar texto livre, palavras ou pequenas frase, aos itens. Essas palavras-chave ou tags podem ser anexadas a qualquer coisa em uma aplicação: usuários, fotos, artigos, produtos, blogs, vídeos, podcasts, ...

Tags Customers Associate with This Product

First tag: [collective intelligence](#) (Satnam Alag on April 7, 2008)
Last tag: [data mining](#)

Sort by:

Popularity

[collective intelligence](#) (6)
[lucene](#) (5)
[data mining](#) (4)
[information retrieval](#) (4)
[machine learning](#) (3)
[recommendation engine](#) (3)

[weka](#) (3)
[ci](#) (2)
[drools_cep_jboss_rules](#) (1)
[enterprise applications](#) (1)
[expertise finding](#) (1)
[intelligent search](#) (1)

[java](#) (1)
[java_architectutre](#) (1)
[jdm](#) (1)
[nutch](#) (1)
[tag_infrastructure](#) (1)

Figura 2: Exemplo de tags criadas por usuários

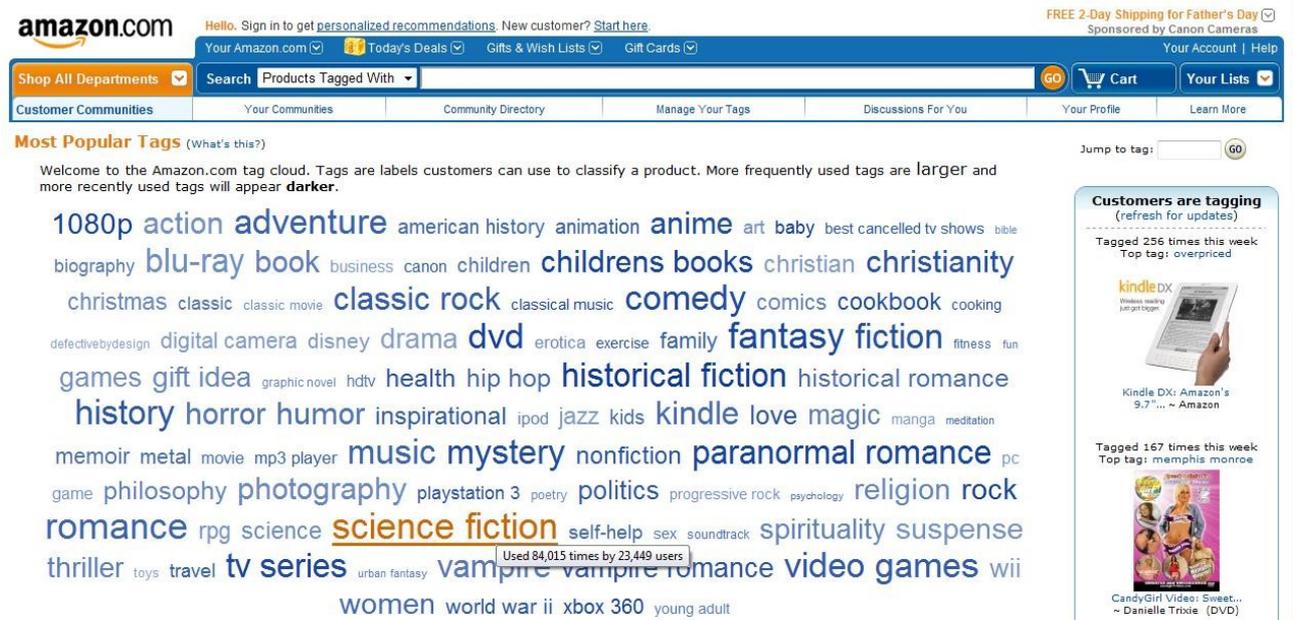


Figura 3: Exemplo de tag cloud

1.3 Recomendação

Mecanismos de recomendação mostram aos usuários itens que possam interessá-los. Eles levam em conta o contexto no qual os itens estão sendo mostrados e dados transacionais passados (com alta frequência, quem comprou este item também comprou estes) .

Para isso, são utilizados algoritmos de descoberta de regra de associação (Association Rule Discovery) sobre a base de dados que contém as transações (compras) dos usuários.



Figura 4: Exemplo de recomendação

What Do Customers Ultimately Buy After Viewing This Item?

- 80%** buy the item featured on this page:
Programming Collective Intelligence: Building Smart Web 2.0 Applications ★★★★★ (47)
\$26.39
- 6%** buy
RESTful Web Services ★★★★★ (37)
\$26.39
- 5%** buy
Building Scalable Web Sites: Building, scaling, and optimizing the next generation of web applications ★★★★★ (29)
\$26.39
- 4%** buy
JavaScript: The Good Parts ★★★★★ (46)
\$19.79

[Explore similar items](#)

Figura 5: Exemplo de recomendação

1.4 Reputação

O amazon marketplace (www.amazon.com/marketplace) é um ambiente virtual onde livrarias podem “terceirizar” seu serviço de e-commerce para a Amazon.com em troca de comissão sobre vendas. Desta maneira, pequenas livrarias ao redor do mundo podem vender seus livros (em particular, livros não disponíveis no amazon.com) utilizando toda a infraestrutura Amazon.com, inclusive de cobrança. A amazon.com então repassa a receita para a livraria e pede para o usuário um “feedback” em forma de avaliação, de 0 a 5 estrelas, desta loja virtual que o usuário efetuou uma compra.

Desta maneira, um sistema de reputação sobre as lojas virtuais é estabelecido, evitando lojas “fraudulentas” que não entregam o que foi pedido.

Na figura 6, um livro sendo oferecido por 2 “lojas virtuais” e pela própria Amazon.com:

The screenshot shows the Amazon Marketplace interface for the book "Galois Connections and Applications (Mathematics and Its Applications) (Hardcover)" by K. Denecke (Editor), et al. The product page includes a "Price at a Glance" box showing a list price of \$129.00, a new price of \$69.43, and a used price of \$71.43. Below the product information, there are three listings from different sellers:

Price + Shipping	Condition	Seller Information	Ready to buy?
\$69.43 + \$3.99 shipping LOW ITEM PRICE	New	Seller: PLANET_BOOKS Rating: ★★★★★ 91% positive over the past 12 months (583 ratings.) 1345 lifetime ratings. Shipping: In Stock. Ships from TX, United States. See Shipping Rates . See return policy . Comments: Brand New Book. Excellent Customer Service! Easy Returns. International shipping available upon request.	Add to Cart or Sign in to turn on 1-Click ordering.
\$109.25 + \$3.99 shipping	New	Seller: TABLETOPART Rating: ★★★★★ 84% positive over the past 12 months (1414 ratings.) 2719 lifetime ratings. Shipping: In Stock. Ships from NY, United States. Expedited shipping available. See Shipping Rates . See return policy . Comments: Brand New Book. Immediate Dispatch	Add to Cart or Sign in to turn on 1-Click ordering.
\$129.00 (this item ships for FREE with Super Saver Shipping) Details	New	amazon.com. <small>Ships from TX, United States. Expedited shipping available. See Shipping Rates. See return policy.</small>	Add to Cart or Sign in to turn on 1-Click ordering.

Figura 6: Exemplo do Amazon Marketshare com as avaliações

Na figura 7, detalhe da reputação da loja virtual “planet_books”:



Figura 7: Exemplo de reputação da loja planet_books

2 Sistema Desenvolvido – Leilão Online

2.1 Especificação

R1	Usuário deve ser capaz de se cadastrar no sistema.
R2	Usuário deve ser capaz de efetuar login, se possuir cadastro.
R3	Usuário deve ser capaz de criar leilões com lotes após efetuar login.
R4	Usuário deve ser capaz de visualizar os leilões que criou.
R5	Usuário deve ser capaz de buscar leilões.
R6	Usuário deve ser capaz de visualizar um leilão e seus lotes.
R7	Usuário deve ser capaz de dar lances a um lote, se estiver logado.
R8	Usuário deve ser capaz fazer perguntas a respeito de um lote e o leiloeiro de responder.

2.2 Análise

2.2.1 Diagrama de Classes

A distribuição dos pacotes ficou da seguinte forma:

Pacote	Descrição
<i>src.actions</i>	Classes que agem sobre as entidades de negócio.
<i>src.actions.helper</i>	Classes que ajudam as classes do actions.
<i>src.actions.exception</i>	Classe usada para exceção.
<i>src.actions.listener</i>	Classe que atua na criação e destruição do contexto.
<i>src.actions.redirecionador</i>	Classe que atua no redirecionamento para autenticação dos usuários.
<i>src.aop.advice</i>	Classe utilizada como advice para logar ações dos usuários.
<i>src.dao</i>	Classes que realizam e gerenciam o acesso aos objetos de

	dados.
<i>src.dao.jdbc</i>	Idem, em particular para conexões JDBC utilizando o JPA e sua implementação Hibernate.
<i>src.db</i>	Classe que atua na inicialização do sistema para povoar a base de dados.
<i>src.model</i>	Classes que modelam as entidades de negócio.
<i>src.otimização</i>	Classes utilizadas para otimizar a escolha das tags
<i>src.search</i>	Classes responsáveis pela indexação e busca no Lucene
<i>src.struts</i>	Classes que precisam de autenticação. Interceptor's de admin e de usuarios para garantir tais autenticações.
<i>src.webservices.client</i>	Classes que consultam serviço disponibilizado sobre os dados estatísticos dos anunciantes.
<i>src.webservices.server</i>	Serviço disponibilizado para consultar dados estatísticos dos anunciantes
<i>cfg.spring</i>	Arquivos de configuração do Spring
<i>WebContent/selenium-test</i>	Testes de aceitação/integração com Selenium.

No diagrama 1 colocamos apenas as classes do pacote *src.model*:

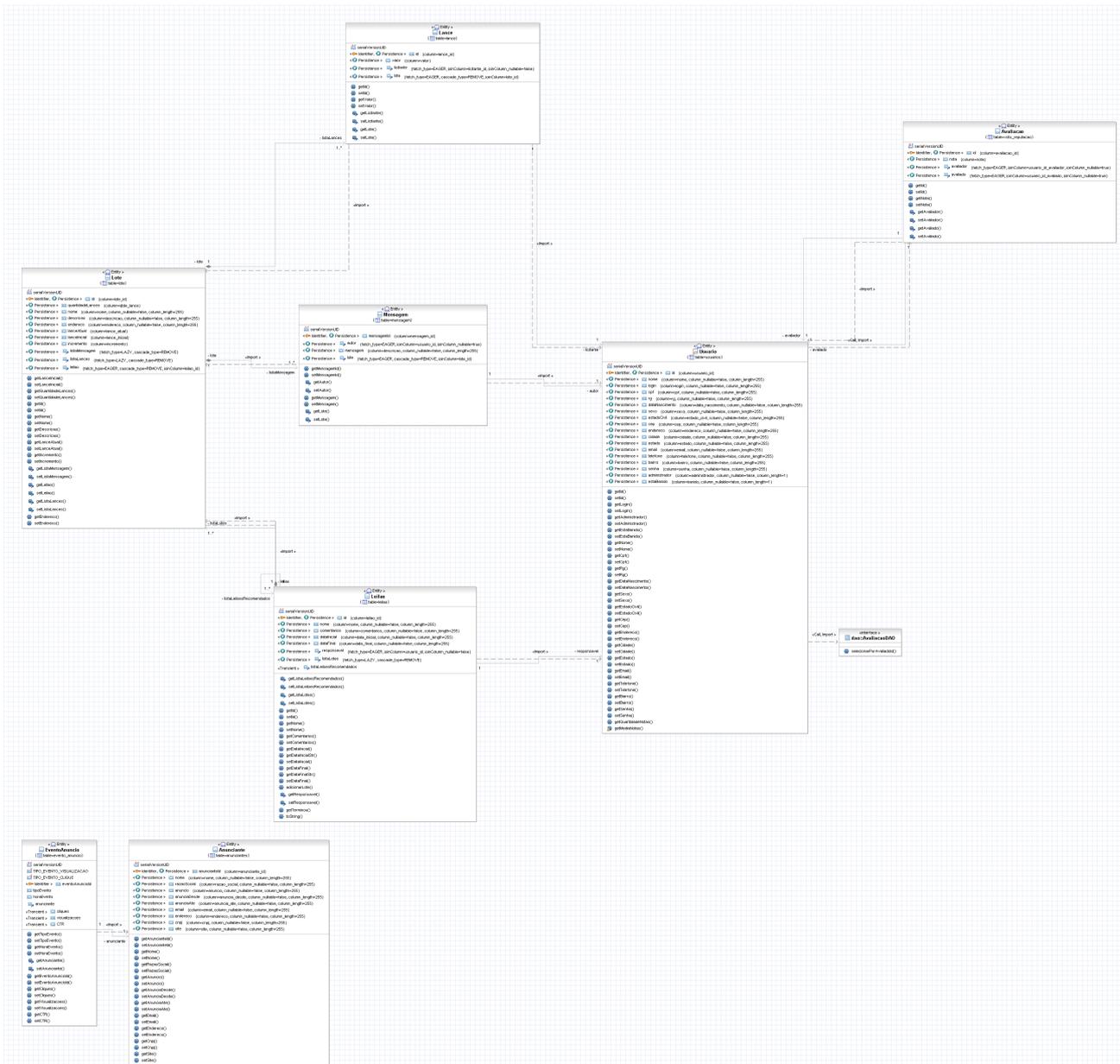


Diagrama 1: Diagrama de classes do pacote src.model

2.2.2 Diagrama de Casos de Uso

Caso	Descrição
caso 1	Um usuário já cadastrado efetua o login
caso 2	Um usuário efetua seu cadastro
caso 3	Um usuário cria um leilão (é necessário já ter efetuado login)
caso 4	Um usuário cria um lote (é necessário já ter criado um leilão para conter este lote)
caso 5	Um usuário lista seus leilões (é necessário já ter efetuado login)
caso 6	Um usuário lista os lotes de um leilão
caso 7	Um usuário busca um leilão (através de uma palavra-chave)
caso 8	Um usuário efetua um lance em um lote (é necessário já ter efetuado login e listado os lotes do leilão)
caso 9	Um usuário lista os lances que ele realizou em quaisquer lotes (é necessário já ter

	efetuado login)
caso 10	Um usuário avalia outro usuário (é necessário já ter efetuado login)
caso 11	Um usuário cria uma mensagem em um lote, comumente a fim de sanar alguma dúvida referente a este lote. (é necessário já ter efetuado login)

O diagrama 2 representa os casos de uso:

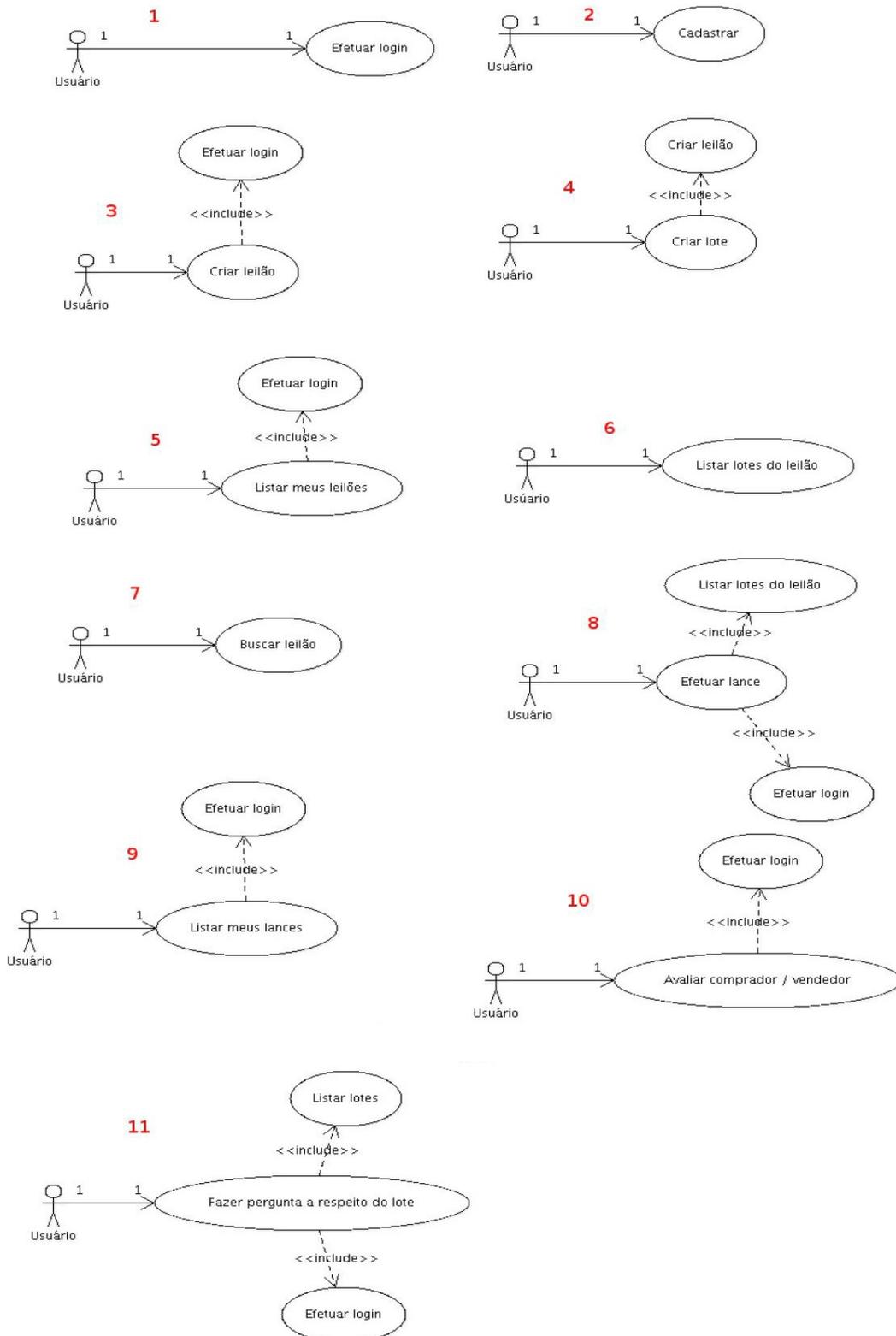


Diagrama 2: Diagrama de casos de uso

2.3 Justificativa das tecnologias escolhidas

2.3.1 AJAX, JSON

Optamos pela utilização de AJAX e JSON em alguns lugares da interface, pois através dessas tecnologias o sistema ganha em dinamicidade. Isso é possível graças ao AJAX que faz com que não seja necessário fazer uma nova requisição ao servidor a cada clique, o que deixaria o cliente esperando. Já o JSON é um formato para serialização de dados mais simples que o XML. Tudo isso junto deixa a navegação mais agradável.

2.3.2 jQuery

Hoje em dia utilizamos muito JavaScript (JS) para aproveitarmos os recursos do próprio navegador do cliente. Entretanto, JS em si é um tanto trabalhoso. Então, optamos por usar o jQuery, que é uma biblioteca JS, para facilitar todo o trabalho que seria feito através de JS puro: manipulação de elementos DOM, efeitos, estilos, ...

2.3.3 Struts

Optamos pelo Struts devido aos seus Interceptor's. Assim, estamos utilizando-o apenas para controlar as partes do sistemas que necessitavam de autenticação.

2.3.4 Hibernate / JPA

Optamos pelo JPA e pela sua implementação Hibernate pois essas tecnologias facilitam muito o mapeamento de dados modelados em sistemas OO para dados modelados em banco de dados e vice-versa.

2.3.5 Spring

Escolhemos o Spring por ele permitir injeção de dependência de maneira simples. No nosso caso, optamos por injetar os dao's, o que já facilitou bastante. Além disso, usamos o Spring AOP para logar toda ação do banco de dados. Isso é útil para uma eventual auditoria ou para coletar informações dos usuários.

2.3.6 Selenium

O Selenium foi escolhido pois foi a ferramenta mais adequada para realizar os testes de aceitação e integração. Adequada no sentido de ser a mais conhecida pelos integrantes do grupo e também uma das mais populares.

2.3.7 Webservice / SOA

Decidimos implementar um serviço que mostra as estatísticas de cada anunciante utilizando SOAP por ser uma tecnologia já estabelecida, com boa documentação e que utiliza um formato bem-definido (XML). Talvez, se REST continuar crescendo, poderíamos disponibilizar nos dois protocolos.

2.3.8 Mashup

Fizemos um mashup com o <http://tinygeocoder.com/> e <http://code.google.com/intl/pt-BR/apis/maps/index.html> para obter a localização dos lotes dos leilões. Optamos por fazer um mashup porque ele só tem a acrescentar à aplicação, uma vez que você pode utilizar recursos prontos na Web. Basta que tenhamos consciência que esses recursos podem ficar indisponíveis às vezes

2.4 Funcionalidades referentes à Web 2.0

2.4.1 Analisador Textual, Tags e Otimização

As tags são exibidas no resultado de cada pesquisa feita pelo usuário, baseadas nos leilões

devolvidos. Ao clicar numa tag, é como se o usuário realizasse uma nova busca com aquela palavra e, assim, a lista de tags se atualiza, permitindo uma navegação pelas tags.

Para construir a lista de tags utilizamos o analisador textual para criar um vetor de termos de cada leilão devolvido pela busca. Nessa criação levamos em conta o texto dos outros leilões (calculamos através de $tf \cdot idf$). Utilizando o otimizador com o algoritmo randômico, pegamos 10 termos (tags) da lista de termos de todos os leilões que havia sido calculada.

2.4.2 Analisador Textual e Recomendação

A recomendação é feita para sugerir leilões ao usuário assim que ele visualiza um leilão. Então, o usuário pode clicar em um leilão recomendado e, assim, ele irá visualizar esse novo leilão e a lista de recomendação será atualizada.

Utilizamos o analisador textual para calcular o vetor de termos de um leilão que será visualizado. Em seguida, calculamos a similaridade através do `dotProduct()` desse vetor com os vetores de termos dos outros leilões (os vetores dos leilões são pré-processados na subida do contexto da aplicação). Pegamos os 5 mais similares, descartando o primeiro que será o próprio leilão que será exibido.

2.5 Participação dos integrantes do grupo

Ambos tiveram participação em todas as partes do projeto. Apesar de programarmos separados, sempre nos falávamos, por e-mail ou MSN. Tomamos as decisões principais juntos e quando víamos alguma coisa que não concordávamos questionamos.

3 Conclusão sobre o aproveitamento da disciplina

Eu, Filipe, gostei do curso, talvez tenha sido o curso mais prático que fiz no IME. Era justamente o que muitos alunos reclamavam que faltava por aqui (é lógico que as matérias teóricas ainda são importantes).

A matéria abordou várias tecnologias que são utilizadas hoje em dia das quais é bom, no mínimo, se ter uma noção. Entretanto, como o próprio professor disse no começo do semestre, a matéria aborda muito tópicos e acho que acabou ficando muito puxado. Digo isso, porque praticamente larguei meu projeto de mestrado para ter um bom aproveitamento da matéria e mesmo assim não foi possível aprofundar nas tecnologias nem nas funcionalidades referentes à Web 2.0. Quem já conhecia ou utilizava as tecnologias no dia-a-dia até que conseguiu levar melhor o curso. Enfim, acho que seria melhor dividir o curso. Talvez um curso referente às tecnologias e outro referente às funcionalidades Web 2.0 com pré-requisito o primeiro curso.

Eu, Alexandre, também gostei do curso, principalmente pelo seu aspecto prático, mesmo quando estavam envolvidos tecnologias e algoritmos não triviais (como os algoritmos de inteligência coletiva: filtragem de documentos, recomendação, etc). O curso é bastante puxado e realmente merece o nome de “Tópicos Especiais”, não se trata de um curso de “Introdução ao Desenvolvimento Web”.

Algo que ficou devendo foram os computadores do CEC, que se revelaram de pouco auxílio ao desenvolvimento do projeto (em particular, a memória, processador e quota insuficientes, além disso a versão dos IDEs e plugins instaladas não foram apropriadas para a maioria dos grupos). Como reflexo, foram utilizados amplamente os “laptops” de membros da dupla ou computadores pessoais que alunos tinham em casa.

É minha opinião que os alunos e professores devem empenhar esforços para exigir uma infraestrutura de laboratório mais adequada, tendo em vista o direito de acesso, por parte dos alunos, aos recursos necessários para se cursar uma disciplina.