

Relatório Final

MAC0416/5855 - Tópicos Especiais em Desenvolvimento para Web

Israel Danilo Lacerra #USP 5489057

Omar M. A. Ajoue #USP 5381885

30 de junho de 2009

1. Análise de recursos relacionados à Web 2.0

O site escolhido refere-se à revista TIME (www.time.com), uma conceituada revista de notícias e conteúdo nos Estados Unidos. Seu site é bastante completo e com conteúdo atualizado diariamente, tornando-o um portal informativo de notícias muito acessado, aliado à CNN, trazendo todas as notícias mais importantes e de última hora ao leitor a todo momento.

Em sua página inicial, são apresentadas notícias recentes e manchetes sobre os assuntos de maior repercussão além de notícias do tipo “última hora”. Apresenta também notícias “*must read*”, indicando ao leitor onde começar para se atualizar no cenário atual.

The screenshot shows the TIME.com homepage with a prominent headline about Michael Jackson. The layout includes a top navigation menu, a search bar, and several content blocks such as 'Must Reads', 'Must See', and 'Latest Headlines'. A 'ROLEX' advertisement is visible in the top right corner.

Como primeiro traço da Web 2.0 que pode ser visto, logo na página inicial, o site permite que o usuário obtenha Seeds RSS, receba newsletters e provê podcasts para os leitores.

Algo curioso que encontramos na home da TIME é uma matéria que diz como usar o Twitter e o Facebook para procurar emprego:

TIME
Business & Technology

Using Twitter & Facebook:
Find a Job

Apple Unveils the New iPhone »
New GM Ad: Restarting the Engines? »
Consumer Spending - Drag on Rebound? »

Em todo o site, é possível encontrar anúncios de sites de terceiros relacionados ao tema apresentado. As notícias são organizadas em estrutura hierárquica, classificadas por assunto. O que mais sentimos falta foi a ausência de tags. O site podia disponibilizar um simples tag cloudy onde o usuário poderia facilmente navegar pelos assuntos mais lidos.

Na página de visualização de notícias, encontramos mais um traço forte relacionado à Web 2.0: é possível interagir compartilhando a notícia com outros sistemas que fornecem uma API externa, como Buzz up!, Facebook, Twitter e outros.

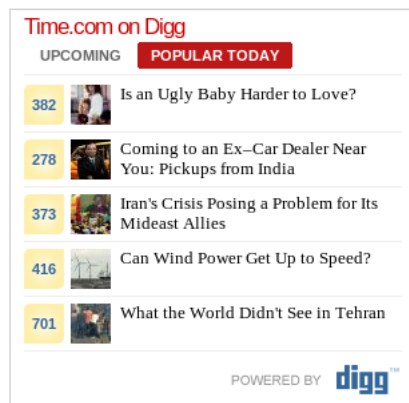
Logo abaixo do corpo de uma notícia há recomendação por usuário, onde são indicadas outras notícias que foram lidas por pessoas que leram a notícia atual, e também há recomendação de itens relacionados, onde são indicadas notícias que tenham tema relacionado com a notícia atual.

Para se manter "conectado" ao site, o usuário possui algumas opções como newsletter, rss e até usar o Twitter.

Ao lado da notícia temos também um pequeno ranking das notícias da TIME no Digg.

— still only about 4% of all public schools. Charters, which are funded with public dollars but are typically free of school-district and teacher-union restrictions, have typically been regarded as labs of innovation (though a recent Stanford University study makes the case that charter-school quality can range greatly, from great to not so great). Many charter principals have full control over the hiring and firing of teachers, full control of what curriculum they choose to teach on a daily and weekly basis, and full control over how long their school day should be — all the things that are typically dictated from a central office in a typical public school.

That freedom from bureaucracy is exactly why people go into the charter-school business in the first place, says Nelson Smith, president and CEO of the National Alliance for Public Charter Schools. And it's exactly what would need to stay in place if school districts are serious about having charter operators come in and take over parts of, or entire, failing schools. "It shouldn't be thought of as trying to change the ocean liner's course," says Smith, but rather, to carry out his



O ponto forte do site é que possui muitos recursos para integração com serviços de comunidade (Twitter, Facebook, etc) que são apresentados para os usuários em momentos interessantes e com bastante facilidade. O grande ponto fraco é realmente a ausência de tags. Não faria mal nenhum uma pequena tag *cloudy* na página inicial.

2. Sistema Desenvolvido

Especificações

Nosso sistema consiste basicamente em uma livraria virtual cujo nome é “*We book your face*”, proveniente da célebre frase “*Tea with me that I book your face*” ou “Chá comigo que eu livro a sua cara”.

No início do projeto estabelecemos como objetivo que o site cumprisse com os seguintes casos de uso:

- busca de livros;
- inserção de livros em um carrinho de compras;
- finalização do pedido;
- feedback de comentários quanto ao serviço de entrega;
- comentários a respeito dos livros.

Porém só foi possível implementar dois desses casos de uso: inserção de livros em um carrinho e a finalização do pedido. Além disso implementamos uma busca por tags (em uma *tag cloudy*) e uma tela para cadastro de livros, que só deve ser vista pelo usuário administrador.

Arquitetura

As classes do nosso projeto encontram-se divididas em 12 pacotes. Nosso projeto não é grande, mas procuramos fazer com que a estrutura das classes fosse bem razoável, de forma que continuasse intuitiva com o crescimento do projeto.

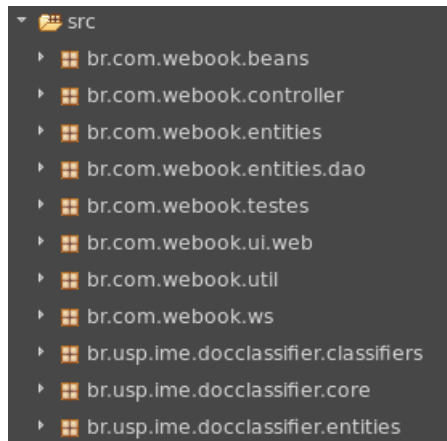


Figura 1: Estrutura das classes do projeto.

A família `br.com.webook.*` consiste nas classes efetivamente do nosso projeto. Já o conjunto `br.usp.ime.docclassifier.*` são herdadas do grupo de Filtragem de documentos, formado pelo Thiago Coraini e pelo Israel, que é um dos integrantes da nossa dupla. Além de algumas mudanças nas classes do classificador de documentos, adicionamos o pacote `br.usp.ime.docclassifier.entities` que possui as entidades onde guardaremos as informações sobre o classificador.

Faremos aqui alguns comentários sobre os pacotes `br.com.webook.ui.web` e `br.com.webook.util` e suas funcionalidades. Os demais pacotes terão suas funcionalidades explicadas mais a frente, quando falaremos das tecnologias utilizadas.

O pacote `br.com.webook.util` possui uma série de classes e utilitários que nos simplificam algumas tarefas. Os três principais são o `WeBookEm`, `WeBookServlet` e o `WeBookSession`. O `WebookEm` é utilizado para facilitar o uso do `entityManager` do JPA e é usado em qualquer lugar do projeto onde a funcionalidades da JPA. A classe `WeBookServlet` consiste na única servlet do nosso projeto, que faz parte do nosso framework que será explicado mais abaixo. As classes `WeBookSession` e `WeBookSessionAttributes` são responsáveis por fazer o relacionamento da camada de negócios com os atributos que guardamos na sessão.

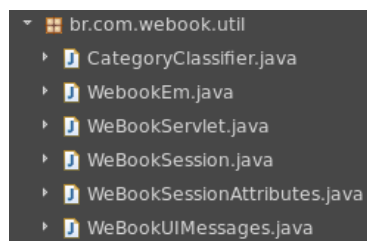


Figura 2: Pacote `br.com.webook.util`.

No início do projeto implementamos um pequeno framework para facilitar a comunicação entre as jsp e a camada de negócios. Implementamos uma única servlet que seria responsável por atender as requisições e receberia como parâmetro o nome da ação que seria efetuada. Assim, uma chamada de uma jsp à camada de negócios deveria ser feita, por exemplo, da seguinte maneira: `WeBookServlet?action=UsuarioSignupAction`.

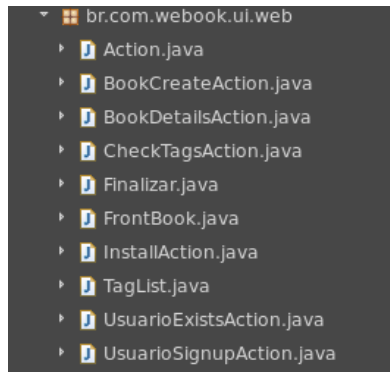


Figura 3: Pacote br.com.webook.ui.web, com as actions do framework.

A *action* passada como parâmetro deve ser um nome de uma classe que implementa a interface *Action* que possui um único método *execute* que recebe os parâmetros da chamada e efetua a ação. Esse simples framework se mostrou muito útil pra nós e foi usado até o final do projeto.

Tecnologias

Conforme as tecnologias foram apresentadas, incrementamos o sistema tentando usar sempre uma das novas tecnologias. A primeira das tecnologias que usamos foi o **AJAX**, que foi usado primeiramente junto com **JQuery** na tela de cadastro de usuários e depois em outras telas. Para fazer a requisição AJAX tanto nessa tela quanto em todas as outras que usamos AJAX optamos por usar o **JSON**, pois tem uma estrutura mais simples e atendia as nossas necessidades normalmente.

Após fazer a tela de cadastro de usuários, o próximo passo era a tela de login, onde já aproveitamos para usar **JSF**. Sendo assim essa tela se tornou uma das únicas onde nosso framework não é utilizado, pois usamos um *managed bean* do JSF. O uso de *managed beans* ficou restrito a essa tela, pois principalmente pelo fato dele precisar de uma configuração adicional no faces-config.xml para cada nova funcionalidade, optamos por continuar usando nosso framework.

Essa mesma tela de login foi refeita usando **Struts**, mas na implementação final foi mantida a tela com JSF. Não foi possível aprender muito profundamente sobre Struts, mas achamos interessante que ele se assemelhava muito ao nosso framework, mas tinha mais recursos. Mesmo assim acabamos seguindo com nosso framework pela simplicidade, e o Struts foi abandonado. A impressão que tivemos é que o Struts foi uma primeira tentativa de se criar um framework MVC, mas com idéias bem simples que talvez tenham melhorado muito no Struts 2, que infelizmente não estudamos.

Para facilitar a comunicação com o banco de dados usamos a implementação de **JPA** do **Hibernate**. Tudo que usamos era da especificação JPA, optamos por não usar nada específico do Hibernate. Mesmo que dificilmente trocariamos de implementação a idéia era não ficar preso ao Hibernate e, além disso, acreditamos que se tivéssemos problemas com coisas especificadas na JPA, seria mais fácil obtermos soluções. O JPA foi mantido no sistema, e todas as nossas tabelas foram mapeadas com o uso dele.

O framework **Spring** foi o que nos pareceu mais poderoso, mas também o mais complexo para se implantar num projeto já em andamento. Depois de alguns problemas com configuração e bibliotecas incompatíveis conseguimos enfim usar um pouco de sua funcionalidade MVC no nosso carrinho de compras. O processo de compra de um livro é direcionado a uma servlet mapeada ao controlador do Spring. Sua servlet foi configurada para direcionar o controle às classes que fazem parte do pacote `br.com.webook.controller`.

Utilizamos **JUnit** e **Selenium** para realizar alguns testes no sistema e entender melhor essas ferramentas, porém por falta de tempo não fizemos testes que cubrissem todas as nossas funcionalidades e classes. Os testes encontram-se no pacote `br.com.webook.testes`.

Nosso sistema possui também um webservice baseado em **SOAP** bem simples, que faz uma busca e devolve uma lista de livros. O serviço se encontra no pacote `br.com.webook.ws` e sua descrição pode ser vista em <http://pcgerosa.ime.usp.br:8280/webookyourface/webook.wsdl>.

Por fim implementamos uma tela para cadastro de livros usando **Ruby on Rails**. A tela ficou bem melhor do que a tela que temos para essa finalidade na versão final, mas infelizmente não conseguimos integrar ela com o nosso sistema, gerando um war ou coisa parecida e acabamos optando por deixar essa funcionalidade implementada do jeito que estava em java, pois o uso dessa tela era interessante quando implantávamos a aplicação no pcgerosa, por exemplo. Também não chegamos a cogitar a hipótese de migrar todo o projeto para Ruby, pois seria necessário mais tempo.

Na área de inteligência coletiva, pelo intuito do nosso site - vender livros -, é importante que nossos visitantes tenham facilidade para encontrar o que desejam, seja procurando algo específico, como um título, ou um tema. Pensando nesta idéia, concordamos que seria útil atribuir aos livros o conceito de etiquetas, ou "tags". Implementamos um *tag cloudy* na página inicial, onde é possível clicar em uma tag e ver os livros com aquela tag.

As tags são atribuídas na tela de cadastro de livros, que só pode ser vista pelo administrador (`admin@webook.com.br`). Ao escrever a descrição do livro, enviamos uma requisição ajax que sugere algumas tags para o usuário. O algoritmo usado foi uma variação do classificador bayesiano. A alteração foi bem simples. O algoritmo de Bayes pegava todas as categorias aos quais um texto poderia ser aceito, e indicava a com maior probabilidade. Alteramos o algoritmo para que ele retorne todas as categorias (tags no nosso caso) que tem uma probabilidade acima de um certo limite. O resultado foi razoável. Após cadastrar cerca de 30 livros, as sugestões começam a ser muito interessantes. Como nosso projeto tem ênfase mais nessa disciplina, e não terá milhares de livros cadastrados, achamos por bem fazer sugestões pra tags que tenham uma probabilidade até meio baixa. O resultado é que algumas vezes as sugestões não fazem sentido. Mas num projeto maior, poderíamos aumentar o limite mínimo das probabilidades e os resultados passariam a ser muito interessantes.

3. Conclusões sobre o curso

Houve certo aprendizado na disciplina. Entendemos um pouco de como funciona a WEB, porém, pela grande quantidade de temas abordados e à urgência para entrega dos projetos, ficou difícil acompanhar o curso e ter um nível maior de aprendizado.

Alguns tópicos foram um pouco redundantes, como por exemplo, Struts. Afinal, apresentamos também Spring, que é uma ferramenta mais completa e também cobre as funcionalidades do Struts. Os trabalhos relacionados a web services poderiam ser unificados também.

Como pontos positivos, exploramos diversos aspectos da linguagem, várias ferramentas que nos proporcionaram uma ampla visão da capacidade da WEB e nos deparamos com o conceito de Inteligência Coletiva.

Deixamos um pouco de lado o conceito de que computadores são máquinas e tornamos os sites, que são, de certa forma, apenas um amontoado de código, em um corpo com vida. Vida esta dada pelas tecnologias implementadas, algoritmos inteligentes de escolha, busca e contextualização, dando sentido para criar um ambiente propício para a criação e descoberta de conteúdo.

A WEB tem crescido muito nos últimos anos e sua aplicação está se tornando cada vez mais diversa. A quebra do paradigma estático é uma mudança necessária para um ambiente que foi criado por humanos e principalmente: está sendo utilizado por humanos.

Abaixo descreveremos nossas conclusões individuais sobre o curso.

Omar Ajoue

Apesar de já trabalhar com desenvolvimento de web sites, o conceito utilizado é diferente e a linguagem também.

Acredito que seria interessante também fornecer aos alunos do curso um pequeno modelo pronto de um site junto das instruções de como utilizá-lo, pois o início do projeto é sem dúvida o mais complicado: entender como que tudo funciona e de onde vem cada componente. Ver um projeto inicial já funcionando sem dúvida facilita o entendimento.

Por experiência do grupo, grande parte da dificuldade que tivemos foi de entender o processo inicial de cada ferramenta ou framework, afinal, era necessário aprender a partir de uma base bastante simples, que foram as apresentações dos trabalhos. Mesmo com exemplos, seria interessante ter um projeto em funcionamento para nos basearmos. Creio que nos próximos anos, tendo os projetos de anos anteriores em mãos, sem dúvida o curso ficará cada vez melhor, tanto por mais disponibilidade de projetos como pela própria experiência adquirida.

O aprendizado conceitual obtido foi de grande valor, mas devido à inexperiência com programação em java e com o funcionamento de frameworks, tive grande dificuldade em realizar algumas etapas do projeto.

O maior aproveitamento obtido foi no conteúdo teórico, pois o conteúdo prático serviu mais como fonte de atualização em quais ferramentas seriam úteis para realizar cada tarefa, e nem tanto o seu funcionamento.

Devido a este fato, minha contribuição na programação ficou defasada; apesar de ter implementado a interface MVC do spring e auxílio em outras ferramentas, o maior aproveitamento foi no aspecto de compreensão teórica do conteúdo apresentado. Como o Israel possuía mais

conhecimento sobre a linguagem Java pôde me auxiliar em diversas dúvidas. Devido a este fato, me dispus a ter uma maior participação na realização das apresentações. É importante ressaltar que nada foi feito às escuras e ambos os integrantes do grupo sabiam de tudo que acontecia no sistema, mas algumas partes da programação foram inteiramente feitas pelo Israel e apenas compreendidas por mim. Em suma, pode-se dizer que a programação foi dividida em um aspecto 60%-40% entre os integrantes.

Israel Lacerra

Acredito que o grande problema da disciplina realmente foi a quantidade exageradamente desnecessária de conteúdo. Acredito que seria bem melhor vermos uma ferramenta para cada contexto, e entendermos realmente o seu funcionamento. Do jeito que as coisas foram feitas, vários fundamentos interessantes deixaram de ser explorados. O excesso de conteúdo trouxe também uma dificuldade maior do que o esperado em uma matéria sem créditos trabalho. Alguns de nós tiveram que se decidir em fazer essa ou outra matéria alterando seu planejamento. Acho que pra quem não tinha algum conhecimento prévio das tecnologias vistas na disciplina, ficou praticamente impossível fazer os trabalhos e entender realmente o que estava se passando por baixo das implementações.

Fora isso, a intenção do curso com certeza foi boa. Não tínhamos nenhum curso com essa abordagem, principalmente no que diz respeito à Inteligência Colêtiva, que por sinal, devido a seu caráter mais teórico, foi a a parte onde mais podemos aprender, livres das especificidades das tecnologias da primeira parte do curso.