

LIGA DE ENSINO DO RIO GRANDE DO NORTE
FARN - Faculdade Natalense para o Desenvolvimento do Rio
Grande do Norte

Igor Saraiva Brasil

Conceitos de Redes Sociais e Web 2.0 aplicados a um
sistema de clipping de notícias

Monografia apresentada ao Curso de Pós-Graduação
em Desenvolvimento de Sistemas Corporativos da FARN
- Faculdade Natalense para o Desenvolvimento do Rio
Grande do Norte como requisito obrigatório para obter
o título de especialista em Desenvolvimento de
Sistemas Corporativos.

Orientador: Ricardo Wangel, M.Sc.

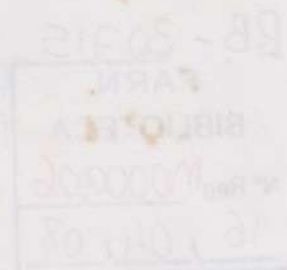
Natal
2007

IGOR SARAIVA BRASIL**Conceitos de Redes Sociais e Web 2.0 aplicados a um sistema de clipping de notícias**

Monografia apresentada ao Curso de Pós-Graduação de Desenvolvimento de Sistemas Corporativo da FARN - Faculdade Natalense para o Desenvolvimento do Rio Grande do Norte como requisito obrigatório para obter o título de especialista em Desenvolvimento de Sistemas Corporativos.

Orientador: Ricardo Wendell, M.sc

**Natal
2007**



IGOR SARAIVA BRASIL

Conceitos de Redes Sociais e Web 2.0 aplicados a um sistema de clipping de notícias

Catálogo na Publicação – Biblioteca da FARN
Setor de Processos Técnicos

Brasil, Igor Saraiva

Conceitos de Redes Sociais e Web 2.0 aplicados a um sistema de clipping de notícias / Igor Saraiva Brasil. – Natal: FARN, 2007. 63f.

Orientador: Ricardo Wendell (M.Sc.)

Monografia (Especialização em Desenvolvimento de Sistemas Corporativos) – Faculdade Natalense para o Desenvolvimento do Rio Grande do Norte.

1. Web 2.0 – Monografia. 2. Modelagem UML – Monografia. 3. Redes Sociais – Monografia. 4. Clipping de Notícias – Monografia. I. Título.

RN/FARN/BC

CDU 004.738.5

RB - 30715

FARN

BIBLIOTECA

Nº Reg. 1000906

16 / 04 / 08

004.738.5

B83c

2007

ex 01

IGOR SARAIVA BRASIL

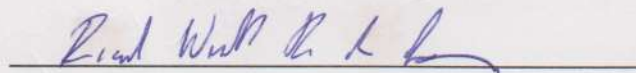
RESUMO

Atualmente a internet vive o momento de sua segunda geração, chamada Web 2.0, onde novos conceitos e novas formas de relacionamento estão surgindo. A consolidação desta "nova internet" e a procura por aplicações da Web 2.0 em si, faz com que as empresas busquem meios para aumentar a produtividade no desenvolvimento de seus sistemas, influenciando a utilização de novas ferramentas e metodologias. Neste contexto, a adoção do framework Ruby on Rails tem sido uma boa alternativa, que, combinada com uma metodologia de desenvolvimento ágil, vem demonstrando bons resultados e ganhos de produtividade. Este trabalho tem por objetivo mostrar o sistema de Clipping de Notícias desenvolvido numa plataforma ágil, apresentando as principais características que definem o sistema.

Conceitos de Redes Sociais e Web 2.0 aplicados a um sistema de clipping de notícias

Palavras-chave: Web 2.0, Redes Sociais, Ágil.

Monografia apresentada ao Curso de especialização em Desenvolvimento de Sistemas Corporativos da Faculdade Natalense para o Desenvolvimento do Rio Grande do Norte, aprovada em 01 de março de 2008, pelo banca composta pelos seguintes membros:



Ricardo Wendell, M.sc

RESUMO

Atualmente a internet vive o momento de sua segunda geração, chamada Web 2.0, onde novos conceitos e novas formas de relacionamentos estão surgindo. A consolidação desta "nova internet" e a procura por aplicações ditas Web 2.0 em alta, fez com que as empresas buscassem meios para aumentar a produtividade no desenvolvimento de seus sistemas, influenciando a utilização de novas ferramentas e metodologias. Neste contexto, a adoção do framework Ruby on Rails tem sido uma boa alternativa que, combinada com uma metodologia de desenvolvimento ágil, vem demonstrado bons resultados e ganhos de produtividade. Este trabalho tem por objetivo mostrar o sistema de Clipping de Notícias, desenvolvido numa plataforma ágil, apresentando as principais características que definem uma aplicação Web 2.0, além de evidenciar os conceitos de redes sociais através da manutenção de uma lista de amigos.

Palavras-chave: Web 2.0. Redes Sociais. Ágil.

ABSTRACT

Currently the internet lives the moment of its second generation, called Web 2.0, where new concepts and new forms of relationships are emerging. The consolidation of this "new internet" and the rising demand for such Web 2.0 applications has caused companies to seek ways to increase the productivity in the development of their systems, influencing the use of new tools and methodologies. In this context, the adoption of the Ruby on Rails framework has been seen a good alternative, combined with a agile development methodology, has shown good productivity gains. This paper aims to show a News Clipping System, developed in a agile platform, presenting the main features that define an application Web 2.0, in addition to demonstrating the concepts of social networks by maintaining a list of friends.

Keywords: Web 2.0. Social Networks. Agile.

Figura 5 – Active Record

Figura 6 – Diagrama de Caso de Uso

Figura 7 – Diagrama de Classes

Figura 8 – Estrutura de aplicação Ruby on Rails

Figura 9 – Classe User (usuário)

Figura 10 – Migração do modelo User (usuário)

Figura 11 – Arquivo de configuração do roteamento routes.rb

Figura 12 – Relacionamento Polimórfico

Figura 13 – Relacionamento has_many belongs_to para Notícia e Clipping

Figura 14 – Controlador de Categorias

Figura 15 – Classe Categoria

Figura 16 – Tela Editando relacionamento com o amigo

LISTA DE FIGURAS

LISTA DE QUADROS

- Figura 1 – Distribuição das redes sociais pelo mundo
- Figura 2 – Tag Clouds do Flickr
- Figura 3 – Requisição Ajax
- Figura 4 – Fluxo MVC
- Figura 5 – Active Record
- Figura 6 – Diagrama de Caso de Uso
- Figura 7 – Diagrama de Classes
- Figura 8 – Estrutura da aplicação Ruby on Rails
- Figura 9 – Classe User (usuário)
- Figura 10 – Migração do modelo User (usuário)
- Figura 11– Arquivo de configuração do roteamento routes.rb
- Figura 12 – Relacionamento Polimórfico
- Figura 13 – Relacionamento has_many belongs_to para Noticia e Clipping
- Figura 14 – Controlador da Categorias
- Figura 15 – Classe Categoria
- Figura 16 – Tela Editando relacionamento com o amigo

LISTA DE QUADROS

Quadro 1 - Modelo User / Tabela users

Quadro 2 - Modelo Tag

Quadro 3 - Modelo Tagging

Diagrama 1 - Tabela de Usuários

Diagrama 2 - Classificação XEN

LISTA DE LISTAGENS SIGLAS

Listagem 1 - Código das Classes `acts_as_taggable`

Listagem 2 - Tag Clouds do Flickr

Listagem 3 – Tabela de Usuários

Listagem 4 – Classificação XFN

DOM - Document Object Model (Modelo de Objetos do Documento)

DRY - Don't Repeat Yourself (não se repita)

HTML - HyperText Markup Language (Linguagem de Marcação de Hipertexto)

JVM - Java Virtual Machine (Máquina Virtual Java)

MVC - Model, View and Controller (Modelo, Visão e Controlador, Padrão de Projeto)

ORM - Mapeamento Objeto-Relacional (Object Relational Mapping)

RJS - Rails JavaScript

RoR - Ruby on Rails

RSS - Really Simple Syndication

W3C - World Wide Web Consortium

XHTML - eXtensible HyperText Markup Language

XFN - XHTML Friends Network

XSL - eXtensible Markup Language (Linguagem extensível de formatação)

LISTA DE ABREVIATURAS E SIGLAS

AJAX – Javascript e XML Assíncrono

CEO - Chief Executive Officer

CIO - Chief Information Officer

DAO- Data Access Object

DOM - Document Object Model (Modelo de Objetos do Documento)

DRY - Don't Repeat Yourself (não se repita)

HTML -HyperText Markup Language (Linguagem de Marcação de Hipertexto)

JVM – Java Virtual Machine (Maquina Virtual Java)

MVC –Model, View and Controller (Modelo, Visão e Controlador. Padrão de Projeto)

ORM – Mapeamento Objeto-Relacional (Object Relacional Mapping).

RJS - Rails JavaScript

RoR – Ruby on Rails

RSS – Really Simple Syndication

W3C - World Wide Web Consortium

XHTML - eXtensible Hypertext Markup Language

XFN – XHTML Friends Network

XML - EXtensible Markup Language (Linguagem extensível de formatação).

5.1.2 Banco de Dados.....	39
5.2 MODELAGEM DO SISTEMA.....	41
5.2.1 Diagrama de caso de uso.....	41
5.2.2 Diagrama de Classes.....	42
5.2.3 Organização de Pastas do Projeto Rails.....	42
5.3 IMPLEMENTAÇÃO DOS CASOS DE USO DO SISTEMA.....	43
5.3.1 Gerencia usuário.....	44
5.3.2 Modelo usuário (user).....	44
5.3.3 Gerenciar perfis do usuário.....	45
5.3.4 Gerenciar Notícias.....	48
5.3.4.1 Interação Visão / Controlador.....	48
5.3.4.2 Fazendo cadastramento da notícia usando tags.....	49
5.3.4.3 Criando feeds RSS.....	52
5.3.5 Fazer clipping.....	53
5.3.5.1 Ajax.....	54
1 INTRODUÇÃO.....	11
2 WEB 2.0.....	14
2.1 REDES SOCIAIS E INTERNET.....	16
2.2 AGREGAÇÃO DE CONTEÚDO DA INTERNET (WEB SYNDICATION).....	18
2.3 FOLKSONOMIA.....	20
2.4 AJAX.....	21
2.5 MICROFORMATS.....	24
3 RUBY ON RAILS.....	26
3.1 RUBY.....	27
3.2 RAILS.....	27
3.2.1 Modelo.....	28
3.2.1.1 Framework ActiveRecord.....	29
3.2.1.2 Migrações.....	31
3.2.2 Visão.....	31
3.2.3 Controlador.....	32
3.3 PRODUTIVIDADE.....	32
3.3.1 Convenção sobre configuração.....	32
3.3.2 Não se repita - DRY (don't repeat yourself).....	33
3.3.3 Desenvolvimento ágil.....	34
4 MODELAGEM UML.....	35
5 SISTEMA CLIPPING DE NOTÍCIAS.....	37
5.1 AMBIENTE DE DESENVOLVIMENTO.....	38
5.1.1 IDE.....	39

5.1.2 Banco de Dados.....	39
5.2 MODELAGEM DO SISTEMA.....	41
5.2.1 Diagrama de caso de uso	41
5.2.2 Diagrama de Classes.....	42
5.2.3 Organização de Pastas do Projeto Rails.....	42
5.3 IMPLEMENTAÇÃO DOS CASOS DE USO DO SISTEMA.....	43
5.3.1 Gerencia usuário.....	44
5.3.2 Modelo usuário (user)	44
5.3.3 Gerenciar perfis do usuário.....	48
5.3.4 Gerenciar Notícias.....	48
5.3.4.1 Interação Visão / Controlador.....	48
5.3.4.2 Fazendo cadastramento da notícia usando tags.....	49
5.3.4.3 Criando feeds RSS.....	52
5.3.5 Fazer clipping.....	53
5.3.5.1 Ajax	54
5.3.6 Gerenciar Categorias	54
5.3.6.1 Controle de acesso	54
5.3.6.2 Validações.....	55
5.3.7 Gerenciar Amigos.....	56
5.3.7.1 Relacionamentos entre os modelos	56
5.3.7.2 Microformato XFN	57
5.3.7.3 Estabelecendo e Editando relacionamento com amigo.....	58
6 CONCLUSÃO	60
REFERÊNCIAS.....	62

1 INTRODUÇÃO

Antes do surgimento da Internet, a formação de redes sociais era dificultada pela falta de meios de comunicação eficazes em manter estes indivíduos em contato constante.

Inicialmente para constituir uma rede social, é necessário um objetivo em comum que seja compartilhado por um coletivo sendo necessário que os participantes estabeleçam vínculos e interconectem ações, condição para que haja compromisso com o grupo e pela causa escolhida por todos.

As redes podem ser analisadas dentro dos mais variados enfoques e áreas. Os parágrafos a seguir resumem os principais tipos de rede sociais:

- **Rede Social Primária:** é formada por todas as relações que as pessoas estabelecem durante a vida cotidiana, que pode ser composta por familiares, vizinhos, amigos, colegas de trabalho, organizações etc. as redes de relacionamento começam na infância e contribuem para a formação das identidades.
- **Rede Social Secundária:** é formada por profissionais e funcionários de instituições públicas ou privadas, por organizações não-governamentais, organizações sociais etc., e fornecem atenção, orientação e informação.
- **Rede Social Intermediária:** é formada por pessoas que receberam capacitação especializada, tendo como função a prevenção e apoio. Podem vir do setor da saúde, igreja e até da própria comunidade.

As redes sociais secundárias e intermediárias são formadas pelo coletivo, instituições e pessoas que possuem interesses comuns. Elas podem ter um grande poder de mobilização e articulação para que seus objetivos sejam atingidos.

Com o desenvolvimento das ferramentas tecnológicas, principalmente aquelas promovidas pelo advento da Internet, e ao acesso fácil a tecnologia promovida pelos computadores, fez da tecnologia um dos meios que promove mais rápido as mudanças sociais, fazendo surgir na sociedade novas formas de relação, comunicação e organização das atividades humanas.

Hoje, as redes sociais são uma das principais aplicações do conceito de Web 2.0, termo criado em 2003 pela empresa estadunidense O'Reilly Media para designar uma nova internet que tem como uma de suas características o conteúdo colaborativo. A segunda geração de comunidades, redes sociais virtuais, wikis, blogs agregadores de notícias são exemplos de aplicações Web 2.0.

As redes sociais apoiadas por computadores utilizam-se de diferentes recursos, entre eles: e-mails, fóruns, listas de discussão, grupos de notícias, Chats, Softwares Sociais como Orkut, MySpace, facebook e etc.

Os sites, que foram criados como veículos para que as pessoas se mantivessem em contato com seus amigos, começaram a atender aos interesses dos profissionais, oferecendo oportunidades de contato para negócios e divulgação.

O aproveitamento da Web 2.0 para negócios pode vir de vários modos, tanto no que se refere a prospecção de novos produtos, na utilização de ferramentas para atender o público interno de uma empresa. Nesse contexto a empresa pode analisar grupos de consumidores, que reunidos em comunidades acabam criando soluções que são aproveitadas pelas empresas, que adaptam e criam produtos baseados nessas informações. Em volta de desses produtos as empresas criam canais de comunicação, com a intenção de estreitar o relacionamento e fidelizar os consumidores, fazendo com que eles conheçam as empresas e contribuam com mais informações.

Um dos pioneiros em explorar o conteúdo gerado pelo usuário foi a empresa Amazon.com (VOSEN, 2007), que construiu um império vendendo livros na rede, em seu sistema de sugestões de produtos baseado na navegação do usuário. Segundo a empresa, cerca de 35% de todas as suas vendas são geradas a partir dessas sugestões.

Outra forma das empresas explorarem estes conceitos é a utilização de ferramentas disponíveis atualmente na internet, pois os usuários enxergam possibilidades de melhorar a produtividade e, principalmente, a comunicação com os demais colegas de trabalho. Mas ao mesmo tempo em que essas ferramentas promovem agilidade e interação, podem provocar problemas de segurança e perda

de produtividade pelo mau uso desses recursos. Então, o grande desafio nesse caso é o de encontrar uma maneira segura de permitir o uso de ferramentas totalmente abertas e disponíveis na Web.

Exemplos como Second Life, Facebook e Orkut criam seus espaços virtuais a fim de chegar nesse público que usam redes sociais. Para atender a públicos restritos, o mercado começou a se movimentar para redes sociais específicas como, por exemplo, redes sociais para o mercado corporativo, para artistas e milionários.

A fim de explorar os conceitos da Web 2.0, é proposto um sistema de clipping batizado "ClippingWeb", que consiste em um agregador de notícias onde os usuários podem adicionar notícias e avaliá-las. As características de rede social do sistema estão presentes nas avaliações e comentários das notícias, e também no fato do usuário poder manter uma lista de amigos. A fonte de pesquisa e seleção das notícias geralmente é feita na mídia impressa.

Atualmente a demanda de aplicações Web 2.0 está em alta, empresas de desenvolvimento buscam produtividade para desenvolver aplicações, utilizando metodologias ágeis e ferramentas de apoio. O Rails framework que popularizou a linguagem Ruby consegue se adequar às metodologias de desenvolvimento ágeis, prometendo a criação de aplicações Web 2.0 de forma bastante produtiva, a programação nesse framework é chamada de Ruby on Rails.

O projeto que se segue está dividido em cinco seções, que incluem (1) conceitos da Web 2.0; (2) desenvolvimento Web ágil com Ruby utilizando seu framework mais famoso o Rails; (3) um resumo da linguagem utilizada para modelar o sistema, a UML (unified modeling language); (4) desenvolvimento da aplicação objeto de estudo desse trabalho, que é um agregador de notícias, com alguns conceitos de redes sociais; e (5) conclusões acerca do trabalho desenvolvido.

2 WEB 2.0

Desde a sua criação no início dos anos 1990, a internet tem revolucionado nossas vidas e o mundo mais do que muitas outras tecnologias desenvolvidas na história recente.

Ao longo dos últimos 10 a 15 anos a internet se tornou uma grande fonte de informações que acessamos através de motores de buscas e portais, se tornando um grande repositório de conteúdo multimídia com custo baixo e até mesmo grátis. O lado comercial também se desenvolveu sendo explorado como plataforma de negócios na quais pessoas e companhias fazem grandes partes de suas atividades.

A tecnologia permitiu que a internet evoluísse e se desenvolvesse, e nos últimos quinze anos é notória a mudança de forma de utilização da internet por parte dos usuários, baseado na análise da contribuição e participação ao longo desse tempo.

Nesse período houve grande investimento na internet investidores buscando grandes lucros começaram a investir em empresas que na grande maioria eram só idéias de universitários que não tinham nem ao menos um plano de negócio. Muitas empresas surgiram nessa época e poucas sobreviveram, o evento que fez a seleção foi a *bolha da internet* (SEGARAN, 2007).

A bolha da internet, fenômeno observado entre 1995 e 2001, representa o momento mais importante da internet nos últimos anos. Iniciada em 1995 quando o Netscape (VOSSEN, 2007) fez um IPO (sigla em inglês para oferta pública inicial de ações), a bolha gerou bilhões de dólares em investimentos na internet e colaborou para o desenvolvimento de novas tecnologias e sites.

Entretanto devido a péssimos planos de negócio e a incapacidade de gerar receita, resultou na falência de muitas empresas. As empresas que sobreviveram a esse estouro tinham em comum características que são hoje conceitos da nova internet, mesmo antes desse conceito ser definido: conteúdo colaborativo,

agrupamento de usuários através de comunidades e fidelização são exemplos de característica dessas empresas.

Menos de uma década após o estouro da bolha da internet, a rede mundial mais uma vez lidera uma transformação na vida das pessoas e promete mudar a relação entre as empresas, pessoas e a tecnologia da informação. No centro disso tudo, está a colaboração e o uso da chamada *inteligência coletiva*(SEGARAN, 2007)..

Inicialmente envolto em polêmicas, nas quais o termo Web 2.0 foi acusado de se tratar apenas de um "golpe publicitário" para levantar as empresas "ponto com" após o estouro da bolha, o termo, criado pelo pioneiro da Internet e CEO da O'Reilly Media, Tim O'Reilly(O'REILLY, 2005), hoje esta presente nos portfólios da maiores empresas do mercado de TI e na cabeça da grande maioria dos CIOs e CEOs.

O termo foi publicado pela primeira em outubro de 2004. Na época, O'Reilly, no artigo "*What's is Web 2.0*" (O'REILLY, 2005), afirmou a idéia central da Web 2.0:

Web 2.0 é a mudança para uma internet como plataforma, e um entendimento das regras para obter sucesso nesta nova plataforma. Entre outras, a regra mais importante é desenvolver aplicativos que aproveitem os efeitos de rede para se tornarem melhores quanto mais são usados pelas pessoas, aproveitando a inteligência coletiva(O'REILLY, 2005).

Não demoraram a surgir contestações às afirmações de O'Reilly. Tim Berners Lee, diretor da W3C, entidade que supervisiona o desenvolvimento da internet, levantou dúvidas quanto à Web 2.0 ser realmente uma novidade. Para Lee, o conceito se tratava apenas de um jargão que ninguém entende, sendo que a versão 1.0 já previa uma conexão de pessoas e criava espaços interativos. Além disso, na opinião do diretor, as soluções e os padrões utilizados pela Web 2.0 teriam sido produzidos para Web 1.0:

O'Reilly, pouco mais de um ano após lançar o conceito, publicou sua tréplica e determinou que a Web 2.0 : *"É uma revolução de negócios na indústria da computação provocada pela tendência de se usar a Internet como plataforma"*. O que ele queria mostrar era que a mudança no comportamento dos usuários seria o motor da transformação e não a tecnologia em si. A principal regra: produzir aplicações que se encaixem no conceito de rede, se tornam melhores na medida em que são usadas pelo maior número de pessoas.

Além desses citados acima existem muitos outros como o Myspace, Orkut, Facebook. Baseadas em padrões abertos e tendo como principal plataforma a internet essas aplicações tem em comum o fato de permitir bastante participação dos usuários, seja etiquetando e comentando fotos no FlickrR, escrevendo e editando os verbetes da Wikipédia, ou alimentando agregadores de notícias

2.1 REDES SOCIAIS E INTERNET

As Redes Sociais são uma das formas de representação dos relacionamentos afetivos ou profissionais dos seres humanos entre si ou entre seus agrupamentos de interesses mútuos. As redes sociais virtuais são as relações entre os indivíduos na comunicação mediada por computador (BROWN, 2000).

Atualmente existem vários ambientes Web dedicados à montagem de redes sociais. De uma forma ou de outra, todos os sistemas disponíveis compartilham, pelo menos, da capacidade de representar a conexão entre indivíduos. Os parágrafos a seguir resumem as características de destaque de alguns dos sistemas correntemente disponíveis na internet:

- **LinkedIn** - Ênfase na construção de redes baseadas em perfis profissionais e identificação de usuários intermediários capazes de facilitar o contato entre membros quaisquer do grupo de assinantes. Da mesma forma que o Spoke (THE OPEN, 2008), oferece a montagem de links baseada nos

contatos de e-mail já estabelecidos pelo usuário, através de upload da lista de endereços do Outlook.

- **Orkut** - Foco em relações de amizade, com ênfase na construção de comunidades virtuais, cuja evolução em termos de alguns parâmetros muito simples (tais como o número de membros e tópicos discutidos) pode ser acompanhada on-line.

Além desses citados acima existem muitos outros como o Myspace, Orkut, Facebook e Spaces. A principal característica desses sistemas dedicados à formação de redes sociais é a montagem de mapas de relacionamento entre indivíduos, através dos quais é possível visualizar as possibilidades de contato entre os membros de um determinado grupo. A figura 1 mostra a distribuição das redes sociais pelo mundo.

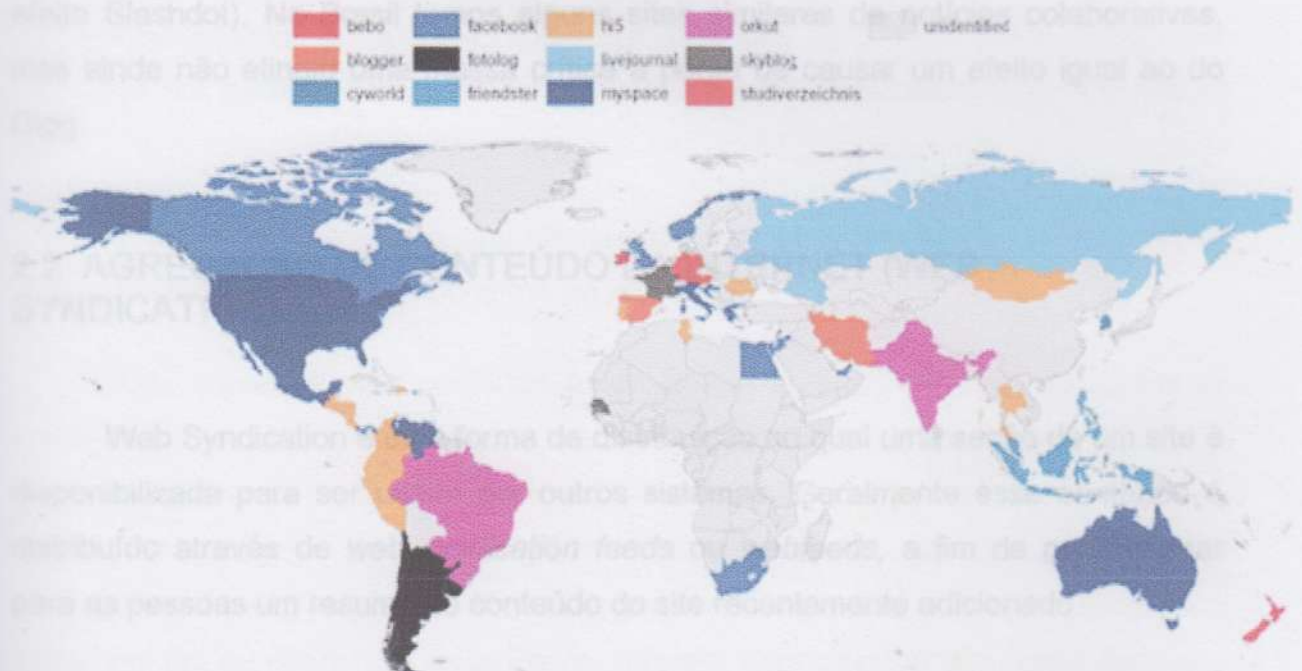


Figura 1 - Distribuição das redes sociais pelo mundo

Fonte: THE WORLD (2008)

Esses sistemas funcionam através da interação social, buscando conectar pessoas e proporcionar sua comunicação e, portanto, podem ser utilizados para fazer laços sociais. As pessoas levam em conta diversos fatores ao escolher conectar-se ou não a alguém.

Blogs e Fotologs também apresentam um campo interessante de estudo das redes sociais, na medida em que também possuem uma lista de amigos ou "blogs/fotologs" favoritos, bem como mecanismos de interação, tais como ferramenta de comentários, *trackbacks* e e-mails.

Para provar o poder das redes onde é explorado o conteúdo colaborativo, o melhor exemplo é o efeito Digg (VOSSSEN, 2007). Sites de notícias, blogs correm o risco de serem atacados, de repente, por uma massa de leitores ávidos por um de seus *posts*. Em geral isso acontece quando um *post* aparece na primeira página de serviços como o Digg ou Slashdot (VOSSSEN, 2007). Ambos os serviços publicam links interessantes e direcionam bastante tráfego, o suficiente para em alguns casos tirar um site do ar. O ataque amigo é conhecido como efeito Digg (antes chamado de efeito Slashdot). No Brasil temos alguns sites similares de notícias colaborativas, mas ainda não atingiu uma massa crítica a ponto de causar um efeito igual ao do Digg.

2.2 AGREGAÇÃO DE CONTEÚDO DA INTERNET (WEB SYNDICATION)

Web Syndication é uma forma de distribuição no qual uma seção de um site é disponibilizada para ser usado por outros sistemas. Geralmente esse conteúdo é distribuído através de *web syndication feeds* ou *webfeeds*, a fim de proporcionar para as pessoas um resumo do conteúdo do site recentemente adicionado.

Hoje o grande aumento de conteúdo na internet milhares de sites de notícias, vídeos, *blogs* e serviço de imagens começaram a disponibilizar *webfeeds* dos seus conteúdos. Os *webfeeds* são documentos baseados em XML que contém itens, que são sumários de notícias, *posts* de *blogs* ou qualquer outro conteúdo contendo *links* para versões maiores, tem como seus principais representantes o RSS e o ATOM.

Na prática, são usados para que um usuário de internet possa acompanhar os novos artigos e demais conteúdo de um site ou blog sem que precise acessar diretamente a página que possui o conteúdo. Sempre que uma atualização for publicada em determinado site, o assinante do feed poderá ler imediatamente.

O RSS é um subconjunto de "dialetos" XML que servem para agregar conteúdo ou "*Web syndication*" e é considerado uma das maneiras mais populares para publicação e compartilhamento de conteúdo, usado muito em sites de notícias e blogs.

A abreviatura do RSS é usada para se referir aos seguintes padrões:

- Rich Site Summary (RSS 0.91)
- RDF Site Summary (RSS 0.9 e 1.0)
- Really Simple Syndication (RSS 2.0)

O Atom é o nome dado a um estilo baseado em conteúdo XML e meta data, ou seja, é um protocolo ao nível da aplicação para publicar e editar Fontes Web que são periodicamente atualizadas, como por exemplo, Blogs. Os feeds devem ser formados em formato XML e são identificados como "*application/atom+xml media type*".

O funcionamento deles não difere das linguagens de marcação tradicionais, com cabeçalhos, títulos e corpo como no HTML 4.0.

Para consumir este conteúdo foram criados agregadores que concentram os feeds fazendo que o usuário reduza o tempo para se atualizar dos novos posts dos blogs preferidos, de notícias de sites especializados e de qualquer conteúdo oferecido através de *webfeeds*. Existe uma grande variedade de agregadores: os baseados em páginas da web tem como seus principais representantes o Google Reader e o Bloglines. Os agregadores baseados no software para desktop tem como destaque os *browsers* como o Internet Explorer 7, Firefox 2 e Opera. Diversas aplicações possuem leitores de feeds, pois se trata de um padrão aberto e muitas plataformas de desenvolvimento possuem APIs que facilitam a implementação de agregadores.

2.3 FOLKSONOMIA

A Internet vem introduzindo vários novos conceitos no mundo atual, a "folksonomia" é um dos que mais tem chamado atenção. O termo "folksonomia" contém o prefixo "folks" ("pessoas", em inglês) é a capacidade de várias pessoas indexarem conteúdo utilizando tags (palavras-chaves ou etiquetas). As indexações feitas obedecem aos critérios absolutamente pessoais e naturais de cada usuário, comunidade ou rede social que as criam e utilizam.

Muitos acreditam que a Web 2.0 encontra sua base na folksonomia. Apesar do nome um tanto complicado, as aplicações utilizando esta forma de indexação já são bem conhecidas, tendo em sites como del.icio.us, Flickr e YouTube seus maiores exemplos.

Estes sites, conhecidos por seus conteúdos colaborativos, se utilizam de tags definidas por seus próprios usuários para indexar as informações publicadas e são usadas como referência pelas comunidades para efetuarem suas próprias buscas.

Existem várias formas de apresentar essa categorização feita pelos usuários como uma simples lista das palavras-chaves ou partindo para representações visuais, como mapas de navegação ou diagramas, podemos assim facilitar a compreensão do conteúdo, transformando-o em estruturas visuais que combinam propriedades gráficas com a organização espacial dos dados.

A Nuvem de tags (*tag cloud*), com um exemplo apresentado na Figura 2, se popularizou devido ao fenômeno dos blogs. É uma representação visual das tags usadas no conteúdo do site, sendo elas tipicamente listadas em ordem alfabética, e frequentemente mostradas com o tamanho da fonte e cor baseada na sua popularidade. Usualmente as Tags são hyperlinks que levam a uma lista de conteúdo associada a tag.

berlin birthday black blackandwhite blue boston bw cali
 one camping canada canon car cat chicago china
 / clouds color concert day de dog england europe fall
 im florida flower flowers food football france friend
 i germany girl graffiti green halloween hawaii hiking holi
 house india ireland island italy japan july june kids la la
 ndon macro may me mexico mountain mountains museum r

Figura 2 - Tag Clouds do Flickr
 Fonte: FLICKR (2008).

2.4 AJAX

O termo AJAX – Javascript e XML Assíncronos foi usado pela primeira vez por Jesse James Garrett em Fevereiro de 2005 (GARRET, 2007) para apresentar um conjunto de tecnologias que ele estava propondo a um cliente. Apesar do termo "Ajax" ter maior uso a partir de 2005, a história da tecnologia mostra que o início do Ajax remonta a uma década atrás com as iniciativas da Microsoft para o desenvolvimento do "Remote Scripting". A primeira aplicação que se beneficiou de conceito foi Outlook Web Access previsto na versão 2000 do Microsoft Exchange Server. Entretanto, foi o Gmail do Google a aplicação que divulgou e difundiu o uso do Ajax.

A ideia de se construir aplicações para internet mais atrativas não é nova, isto por que usuários acostumados a aplicações desktop se depararam com aplicações lentas na Web, baseada em requisições e respostas, onde qualquer ação significava o redesenho da página inteira numa velocidade de conexão que normalmente é lenta. Nos últimos anos houve várias tentativas de se obter um resultado parecido com uma aplicação desktop utilizando tecnologias como frames HTML (incluindo os

iFrames) e DHTML (capacidade de se modificar a estrutura de tags HTML de uma página programaticamente, por meio de código JavaScript embutido na própria página). Mas todas elas de maneira isolada, ou esbarravam em alguma limitação do próprio navegador, ou criavam suas próprias cotas de problemas, limitando seu uso.

O surgimento do objeto *XMLHttpRequest*, uma extensão inicialmente proprietária do JavaScript, permitiu usar os recursos do HTML, CSS (Folhas de Estilos em cascata) e JavaScript de forma a criar efeitos que melhoram a usabilidade dos sites. Este objeto era originalmente parte do conjunto de controles ActiveX fornecido pela Microsoft para manipulação de documentos XML, e posteriormente foi incorporado ao conjunto de objetos pré-definidos do interpretador de JavaScript dos demais navegadores.

A grande novidade introduzida pelo objeto *XMLHttpRequest* foi a capacidade de gerar requisições HTTP diretamente a partir do código Javascript e depois processar os documentos retornados por estas requisições sem a intervenção do visualizador de HTML embutido no navegador Web. O documento retornado seria preferencialmente interpretado como um documento XML, e seu conteúdo depois processado pelo código Javascript por meio da API DOM.

O uso do XML como formato de dados fornece ao código JavaScript independência da tecnologia empregada no servidor Web.

Os desenvolvedores do objeto ActiveX original da Microsoft estavam atentos ao fato de que uma requisição HTTP na internet poderia demorar vários segundos para ter sua resposta inteiramente recebida pelo navegador. Assim conceberam o *XMLHttpRequest* com suporte a requisições assíncronas. Em vez de o código Javascript ficar esperando pelo documento de retorno, o método responsável pelo envio da requisição retorna imediatamente. Isto permite ao JavaScript atender a outros eventos gerados pela interação do usuário com a página.

Uma requisição *XMLHttpRequest* pode ser ilustrada de acordo com a figura 3:

Alguns frameworks ajudam a construir aplicações com AJAX, tais como Prototype, Script.aculo.us, Ext.js, Ajax.Net.

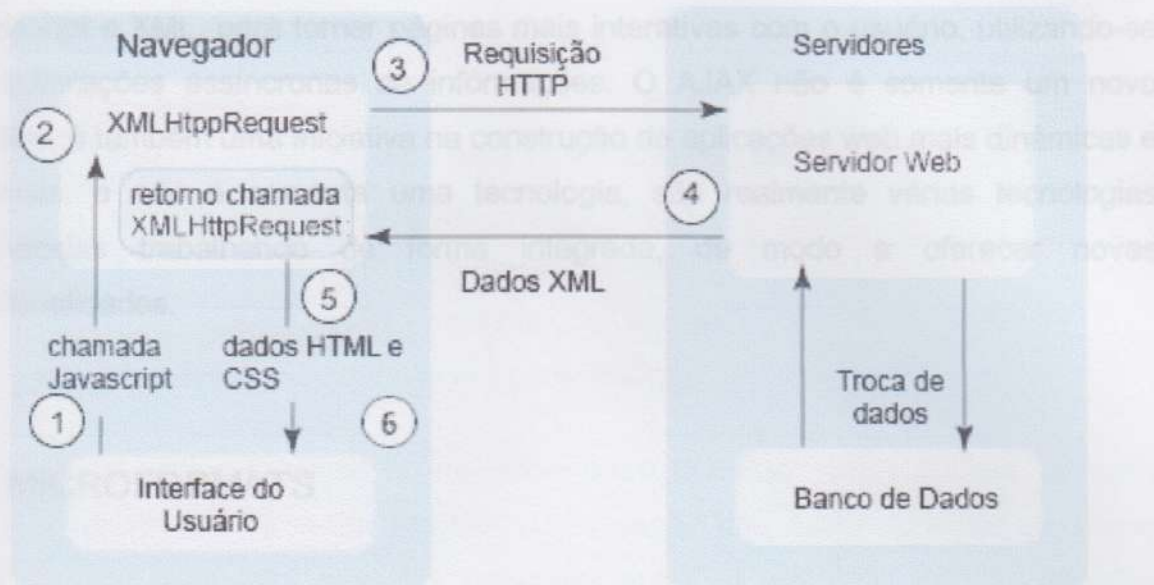


Figura 3 - Requisição AJAX
 Fonte: Dados da pesquisa, 2008

1. É carregada uma página HTML contendo o estado inicial da interface com usuário da aplicação;
2. Uma função Javascript nesta pagina responde a determinado evento disparado pelos tag HTML da própria pagina;
3. A função Javascript envia uma requisição HTTP por meio do XMLHttpRequest, e configura outra função Javascript para ser executada ao chegar a resposta da requisição. O navegador segue normalmente o processamento de outros eventos provocados pelo usuário;
4. A resposta à requisição HTTP é recebida e então o objeto XMLHttpRequest dispara o evento correspondente, provocando a execução da função javascript configurada anteriormente.
5. A função javascript de tratamento da resposta utiliza os dados contidos na resposta para atualizar o estado e o conteúdo dos tags HTML na página;
6. O navegador volta a aguardar por eventos provocados pelos usuários.

Alguns frameworks ajudam a construir aplicações com AJAX , tais como: Prototype, Script.aculo.us, ExtJs , Ajax.Net.

O Ajax é o uso sistemático de tecnologias providas por navegadores, como Javascript e XML, para tornar páginas mais interativas com o usuário, utilizando-se de solicitações assíncronas de informações. O AJAX não é somente um novo modelo, é também uma iniciativa na construção de aplicações web mais dinâmicas e criativas, e não é somente uma tecnologia, são realmente várias tecnologias conhecidas trabalhando de forma integrada, de modo a oferecer novas funcionalidades.

2.5 MICROFORMATS

Microformats é uma série de especificações que tem como foco principal relacionar a informação ou os dados com os humanos em primeiro lugar e em segundo com as máquinas. É uma nova maneira de se pensar sobre dados. Esta série de especificações constitui um "dicionário" de conteúdo semântico para (X)HTML que tem como base os Web standards e é escrito para descrever a informação de forma mais simples possível.

A função destas especificações é enriquecer a informação inserida em páginas Web com meta informação. "Meta" é uma palavra de origem grega que significa "além de" (beyond) e é usada geralmente como prefixo em palavras que indicam conceitos que explicam ou falam de outros conceitos. Esta é a função das metas tags: fornecer informações que estão além daquilo que é visto em um primeiro momento, ou seja, o próprio conteúdo. Mas microformats se refere a descrever trechos de conteúdos específicos de um site, como datas de eventos, informações de contato, descrição de links etc, coisas que estão além do escopo das meta tags. Ou seja, tratando "Pequenos formatos" (micro + format) de dados (informação) válidos no código do seu XHTML é possível enriquecer a maneira com que nós lidamos com a informação e a maneira com que as máquinas armazenam, indexam e organizam toda essa meta-informação. A função dessas especificações é fornecer o máximo de meta-informação sobre o conteúdo que você insere, ou seja, descrevendo os seus próprios dados.

A lista abaixo mostra as principais especificações do micro formato:

- hCard - para informação de contato
- hCalendar - para eventos
- hReview - para revisões
- XFN - para relações sociais
- rel-license - para licenças

Como várias pessoas online começaram a formar redes sociais, serviços como a Technorati tentaram mostrar como esses nós estão conectados. Esses serviços são úteis para descobrir conexões entre os nós, mas não as relações humanas das pessoas responsáveis pelos nós. XFN resolve esse problema definindo alguns valores que descrevem relações humanas, colocados no atributo *rel* de um hiperlink em HTML/XHTML. Usando XFN os autores podem mostrar que o destino de um hiperlink pertence à um amigo, colega de trabalho ou ao próprio autor.

Estas informações podem ser usadas por alguma aplicação para mapear relacionamentos de usuários de uma rede social, podendo também relacionar com outras redes. A marcação é muito simples de ser entendida por humanos e aplicações, é muito simples desenvolver aplicações utilizando estas informações.

Com o atributo *rel* você pode especificar um número de relacionamentos separados por espaços.

Devido o atributo *rel* ser parte de uma tag normal, é muito simples de implementar e transparente ao usuário. Entretanto, pode ser usado também para desenhar utilizando css, javascript caso deseje. Na implementação veremos maiores detalhes sobre esta especificação.

3 RUBY ON RAILS

Durante a era “pós-bolha”, os orçamentos de TI caíram mais do que a demanda por seus serviços, e isso levou os gerentes a buscar soluções com melhor custo/benefício e menores prazos. Devido à busca por produtividade as empresas se sentiram motivadas a utilizar metodologias ágeis.

Os princípios das metodologias ágeis surgiram com o Manifesto Ágil consenso dos participantes de um encontro ocorrido em 2001, onde reuniu um grupo de 17 pensadores do desenvolvimento de softwares, o termo Desenvolvimento Ágil passou a descrever abordagens de desenvolvimento que seguissem estes princípios, que são apresentados a seguir:

- Pessoas e iteração são mais importantes que processos e ferramentas
- Software funcionando é mais importante que uma documentação extensa.
- O relacionamento com o cliente é mais importante que a negociação do contrato.
- Responder às mudanças é mais importante que seguir o planejamento

O que é para ser entendido é que, mesmo havendo valor nos itens à direita, valorizamos mais os itens à esquerda.

Os métodos ágeis se adaptam muito bem à pequenas empresas baseadas na Internet. O desenvolvimento ágil de software permite que elas consigam maior produtividade e realizem projetos significativos com equipes pequenas e orçamentos baixos.

Em busca desta maior produtividade surgiram diversos frameworks, nesse mercado o framework Rails, baseado na linguagem Ruby se destacou, por utilizar linguagens dinâmicas. A seguir serão analisadas as características do desenvolvimento Ruby on Rails.

A implementação do framework segue o padrão arquitetural MVC (Modelo (Model) – Visão (View) – Controlador (Controller) (MVC)), que é um padrão de

3.1 RUBY

Ruby é uma linguagem de programação interpretada, com tipagem dinâmica e forte, orientada a objetos com vastas semelhanças com Perl, SmallTalk e Python. Projetada tanto para a programação em grande escala quanto para codificação rápida, tem um suporte a orientação à objetos desenhado cuidadosamente para ser completo e aberto à melhorias. A linguagem foi criada pelo japonês Yukihiro Matsumoto, que aproveitou as melhores idéias das outras linguagens da época. Ruby é bastante usada no mundo todo e possui uma vasta biblioteca de módulos chamada "Arquivo de Aplicação de Ruby" (RAA). Existe, ainda, uma ferramenta bastante útil para instalação de bibliotecas, chamada Ruby Gems.

Ultimamente, devido à grande exposição de um framework Web feito em Ruby, o Rails fez com que a linguagem se tornasse alvo de grandes elogios e grandes críticas, justamente pela sua praticidade. A praticidade, inclusive, é um dos conceitos básicos desta linguagem: "Você deve fazer códigos que resolvam seus problemas, e não os problemas da linguagem".

Os conceitos que fizeram o Ruby ter sucesso já estão nos planos das próximas versões das plataformas Java e .Net.

3.2 RAILS

O Ruby on Rails como é chamada a programação feita sobre o mais popular framework, originou-se como um aplicativo chamado Basecamp(THOMAS, 2007), uma solução criada pelo dinamarquês David Heinemeier Hansson.

A implementação do framework segue o padrão arquitetural MVC (Modelo (Model) – Visão (View) – Controlador (Controller) (MVC)), que é um padrão de

projeto, que separa o pedido do modelo de dados (modelo), interface do usuário (visão), lógica e controle (controlador) em três componentes distintos.

A figura 4 ilustra quando o navegador solicita uma página Web a uma aplicação MVC, a página fala exclusivamente ao controlador. O controlador recolhe os dados necessários a partir de um ou mais modelos e retorna a resposta à sua solicitação através de uma visão. Esta separação de componentes significa que qualquer mudança que é feita para um componente tem um efeito mínimo sobre os outros dois.

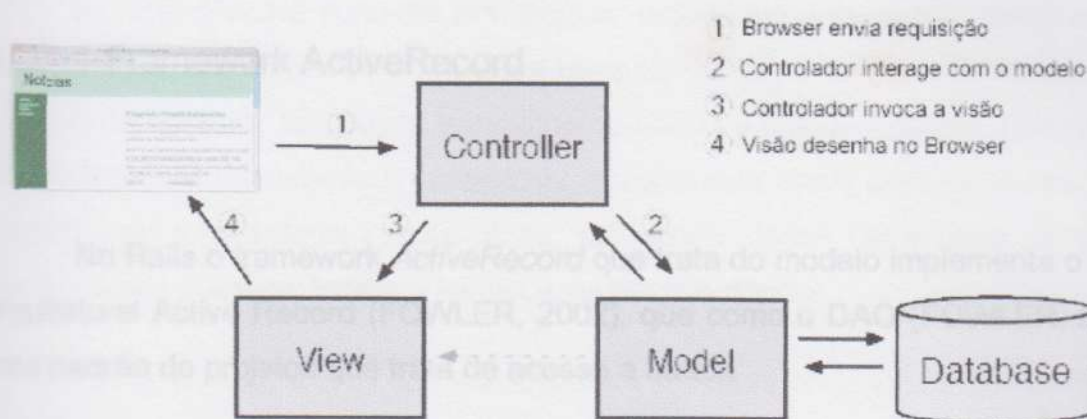


Figura 4 – Fluxo MVC

Fonte:

1. Browser envia uma requisição
2. Controlador interage com o modelo
3. Controlador invoca a visão
4. Visão desenha no browser

3.2.1 Modelo

O Modelo diz respeito ao gerenciamento da informação e ao comportamento da aplicação. O Modelo, numa concepção simplificada comum a alguns desenvolvedores, seria uma mera representação do conteúdo do banco de dados.

Na verdade, no contexto de um sistema de informação, ele é constituído pelas entidades de domínio e pelas regras de negócio intrínsecas a essas entidades, atuando como uma camada de validação e abstração entre a interface com o usuário e os dados de negócio. O servidor de banco de dados, portanto, constitui apenas a camada de persistência do Modelo.

Características:

- Mantém o relacionamento entre objetos e o banco de dados
- É responsável por: validações, associações, transações, etc.

3.2.1.1 Framework ActiveRecord

No Rails o framework *ActiveRecord* que trata do modelo implementa o padrão arquitetural Active Record (FOWLER, 2002), que como o DAO (FOWLER, 2002) é uma padrão de projetos que trata de acesso a dados.

No Active Record, cada classe é responsável por sua persistência e qualquer lógica domínio que atua sobre os dados. O Active Record usa a abordagem mais óbvia, colocando a lógica de acesso a dados no domínio do objeto.

A essência do Active Record é o modelo de domínio, em que sua estrutura de classes se iguala a estrutura do banco de dados. Cada Active Record é responsável por recuperar e salvar para o banco de dados e também tratar qualquer lógica domínio que age sobre os dados.

A figura 5 mostra um objeto que envolve uma estrutura de dados de um recurso externo, como uma linha de uma tabela de banco de dados, e adiciona lógica de domínio a esse objeto.

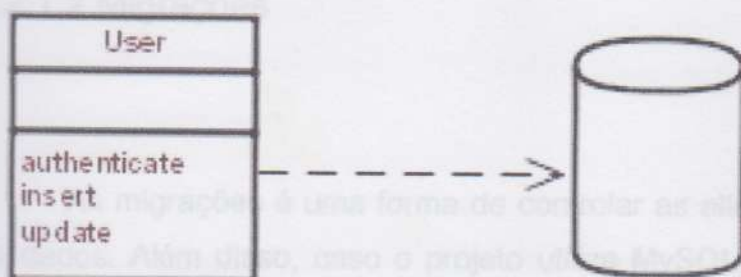


Figura 5 - Active Record

Fonte: Dados de pesquisa, 2008

O ActiveRecord consiste em mapear ordenadamente dados relacionais em objetos Ruby, incorporando o acesso a base de dados, e acrescentando aos dados a lógica do domínio. Um objeto transporta os dados e comportamento. Grande parte destes dados é persistente e precisa ser armazenada numa base de dados. Active Record usa a abordagem mais óbvia, colocando dados de acesso no domínio objeto. Desta maneira, facilita a leitura e escrita dos dados.

Active Record é uma boa escolha para sistemas onde lógica do domínio não é complexa, como criar, ler, atualizar e excluir. Validações baseadas em um único registro funcionam bem nesta estrutura, portanto ideal para um framework que propõem agilizar o desenvolvimento de aplicações web, que na maioria das vezes não possui grande complexidade.

Já o DAO, outro padrão que trata de acesso a objetos, cria uma camada na aplicação separando as rotinas de acesso a banco de dados, geração de SQL, das rotinas de negócio de tal modo que os objetos de domínio não saibam como estão sendo gravados. No DAO a responsabilidade para fazer a comunicação fica com as classes DAOs, no caso do ActiveRecord o próprio objeto de domínio trata de forma transparente o acesso aos dados.

Outra vantagem do Active Record e outro ORM é que no primeiro não há necessidade de configurar mapeamento, o que não acontece na grande maioria dos frameworks ORM como Hibernate que usa XML ou *annotations* para configurar os mapeamentos. Para mapear o Active Record usar convenção sobre configuração.

3.2.1.2 Migrações

As migrações é uma forma de controlar as alterações do esquema de banco de dados. Além disso, caso o projeto utilize MySQL, por exemplo, e depois mude para Postgres, a geração das tabelas permanece intacta, pois o Rails abstrai os comandos e os tipos de dados específicos de cada SGBD. O primeiro passo de nosso sistema é a criação dos modelos, o rails para cada modelo ele cria automaticamente uma classe de migração.

Características:

As classes contem dois métodos:

- self.up é chamado quando a migração é aplicada.
- self.down é chamado quando queremos desfazer a migração.

As migrações são uma boa maneira de poupar tempo na criação das tabelas, visto que seus comandos podem ser executados em qualquer banco de dados suportado pelo Rails. Os metodo "UP" e "DOWN", são inversos, logo tudo que é feito no comando "UP", o comando "DOWN" tratará de desfazer.

3.2.2 Visão

A Visão é responsável por apresentar as entidades de domínio ao usuário, constituindo a parte visível do sistema. Acessa os dados do Modelo e especifica como os dados do Modelo são mostrados ao usuário, como, por exemplo, um texto dentro de um *TextBox*.

Características:

- Templates HTML com scripts Ruby in-line (como JSP, ASP, PHP, etc.)
- Suporte a AJAX de fácil utilização

3.2.3 Controlador

Controlador é o elemento que garante o total desacoplamento entre essas duas partes principais da aplicação. O Controlador interpreta as ações do usuário provenientes da Visão e comanda a execução das regras de negócio contidas no Modelo, além disso, comanda a Visão para que ela apresente adequadamente a informação ao usuário.

Características:

- Ações são mapeadas por reflexão a métodos no controller
- Suporte para: filtros, layouts, caching, sessões de usuário, etc.
- URLs são tipicamente na forma "/controller/action"

3.3 PRODUTIVIDADE

Buscando produtividade para o desenvolvedor o Rails segue três conceitos que visam aumentar a produtividade do desenvolvedor:

- Convenção sobre configuração
- Não se Repita - *DRY (don't repeat yourself)*
- Desenvolvimento Ágil

3.3.1 Convenção sobre configuração

Na maioria dos casos, usamos convenções no dia-a-dia da programação, em geral para facilitar o entendimento e manutenção por parte de outros

desenvolvedores. Sabendo disso, e sabendo que o tempo gasto em configurações é extremamente alto, decidiu-se adotar esse conceito.

Heinemeier Hansson propôs criar o Rails de maneira que não precisasse de excessiva configuração, desde que algumas normas convenções sejam seguidas. O resultado é que pequenos arquivos de configuração são necessário. De fato, se não existe necessidade de mudar alguns desses padrões, o Rails só precisará de um único arquivo de configuração para rodar uma aplicação. O arquivo é usado para estabelecer conexão com o banco de dados: que fornece o mínimo necessário como tipo de servidor, nome de servidor, nome de usuário e senha.

Outras convenções que são prescritas pelo Rails incluir os nomes das tabelas de bancos de dados, bem como o processo pelo qual os seus controladores encontra os modelos correspondentes e visões.

3.3.2 Não se repita - DRY (don't repeat yourself)

O Rails apóia-se no princípio da DRY (*don't repeat yourself*, não se repita) que diz que cada parte do conhecimento de uma aplicação deve ter uma única e inequívoca representação. E esse conhecimento não se resume apenas ao código, se estende também para esquemas de banco de dados, planos de testes e documentação. Ao eliminar a duplicidade e reaproveitar o código o desenvolvedor estará melhorando a qualidade do software, por promover flexibilidade e facilidade de manutenção.

Por exemplo, quando se planeja mudar um comportamento de uma aplicação que tem como o princípio o DRY, você não precisa mudar o código da aplicação. Alterando-se apenas o banco de dados, tudo o que se baseia nessas informações é atualizado automaticamente.

3.3.3 Desenvolvimento ágil

A adoção de métodos e práticas ágeis, como SCRUM, desenvolvimento orientado a testes (TDD - Test-driven Development) e XP é crescente. O método de desenvolvimento ágil utiliza uma abordagem adaptativa, onde pequenas equipes (normalmente constituídos por menos de dez desenvolvedores) interagem em pequenas unidades do projeto. Antes de iniciar uma iteração, a equipe reavalia as prioridades para a aplicação que está sendo construída (estas prioridades podem ser feitas durante a iteração anterior, para que caso tenha necessidade ter tempo para ajustá-las). Desenvolvedores ágeis projetam seus sistemas de maneira *Top-Down* (cima para baixo), que começa com a concepção (que pode ser tão simples como um esboço de uma interface, em uma folha de papel).

Quando uma aplicação é construída usando métodos ágeis, é menos provável que ocorra perda de controle durante o ciclo de desenvolvimento, devido aos contínuos esforços da equipe de ajustar as prioridades. Ao passar menos tempo criando especificações funcionais, desenvolvedores utilizando metodologias ágeis podem realmente saltar, e iniciar um projeto no seu desenvolvimento. Tem como características de destaque o fato de não ter fases de compilação, geração de código, deploy e re-inicialização.

Apesar de processos não ser foco do trabalho, é interessante conhecer para melhor aproveitar as características de produtividade do Ruby on Rails no processo de desenvolvimento de um sistema. Existem vários processos ágeis dentre eles o SCRUM e XP, que é direcionada para planejamento e acompanhamento de projetos

- Análise de design preliminar (diagrama de classes, diagrama de sequência e de estados).

- Modelagem Física

- Design detalhado (diagrama de classes, modelagem do banco de dados, componentes, implantação)

4 MODELAGEM UML

O Unified Modeling Language (UML) é uma linguagem padrão para documentar projetos de software. O UML pode ser utilizado para visualizar, especificar, construir, documentar os elementos de um sistema baseado em software. O UML foi concebido para modelar sistemas do mais variados tipos: sistemas de tempo real, sistemas distribuídos baseado na Web, sistemas de informação e outros.

O UML não é apenas uma linguagem visual; os modelos podem ser utilizados para gerar código escrito em diferentes linguagens, como Java e C#, e tabelas de banco de dados relacionais. Ferramentas CASE como o Rational Rose e JUDE, provêm este recurso.

Tipicamente, a modelagem de um sistema é desenvolvida em três etapas sequenciais:

- Modelagem Conceitual
 - Modelagem dos casos de uso do negócio (casos de uso do negócio e diagramas de atividade).
 - Modelagem das entidades de negócio (classes e objetos do negócio, diagramas de seqüência).
- Modelagem Lógica
 - Definição dos requisitos (modelagem de casos de uso do sistema e descrição de casos de uso).
 - Análise de *design* preliminar (diagrama de classes, diagrama de seqüência e de estados).
- Modelagem Física
 - *Design* detalhado (diagrama de classes, modelagem do banco de dados, componentes, implantação)

Os diagramas abaixo apresentados são os que estão sendo usados nesse projeto:

- **Diagrama de Classes**

Os Diagramas de Classes são utilizados para representar a estrutura estática do sistema, composta pelas classes de negócio, classes de interface com usuário e com outros sistemas, e as classes de controle responsáveis pelo controle de transações.

- **Diagrama de Casos de Uso**

O Diagrama de Casos de Uso é o diagrama mais geral e informal da UML, sendo utilizado normalmente nas fase de Levantamento e Análise de Requisitos do Sistema. Apresenta uma linguagem simples e de fácil compreensão para os usuário possam ter uma idéia geral de como o sistema irá se comportar.

Os Casos de usos são usados para capturar os requisitos funcionais do sistema. Geralmente os casos de uso são agrupado em pacotes, contendo casos de uso de alta coesão e baixo acoplamento.

- **Diagrama de Sequência**

O Diagrama de Sequência preocupa-se com a ordem temporal em que as mensagens são trocada entre os objetos envolvidos em um determinado processo.

Nos parágrafos seguintes contém uma breve descrição dos principais módulos do sistema: Gerenciamento de Usuário, Gerenciamento de Notícias e Gerenciamento de Lista de o Amigos.

No módulo de gerenciamento de usuário conta com cadastro, atualização, navegação e perfil. O usuário "Editor" apenas poderá atualizar o seu cadastro, enquanto que o administrador terá acesso a edição de cadastros, associação e remoção de perfil.

5 SISTEMA CLIPPING DE NOTÍCIAS

Hoje uns dos principais métodos que as empresa utilizam para calcular o valor de sua imagem veiculado nos meios de comunicações é o clipping. O clipping é um serviço de pesquisa e seleção das principais notícias publicadas na mídia impressa relacionadas a um determinado assunto. O principal benefício do Clipping para as empresa é o de imprimir agilidade no processo de obtenção de notícias e informações, permitindo que diversos setores de uma empresa como, por exemplo, assessoria de imprensa e comunicação, marketing e vendas, tomem decisões estratégicas com base em informações mais consistentes e atualizadas. Usualmente processo do clipping é feito através da medição de recortes de jornais e da avaliação do impacto da noticia na imagem da empresa.

O *clippingweb* é um sistema agregador de noticias, onde o usuário irá postar suas noticias e avaliar, e em períodos que desejar o usuário pode verificar um histórico em forma de gráfico ou relatório. A ideia é que o usuário mantenha um histórico de noticia assim podendo analisar e quantificar o peso que as mesmas teve num decorrer de um período.

Os usuários também poderão avaliar, comentar, agrupar as noticias e também manter uma lista de amigos. O posicionamento da noticia será feito por data e/ou por popularidade

Nos parágrafos seguintes contem uma breve descrição dos principais módulos do sistema: Gerenciamento de Usuário, Gerenciamento de Noticias e Gerenciamento da Lista de o Amigos.

No Modulo de gerenciamento de usuário conta com cadastro, atualização, inativação e perfis. O usuário "Editor" apenas poderá atualizar o seu cadastro, enquanto que o administrador terá acesso a edição de cadastros, associação e criação de perfis.

O Modulo de gerenciamento de noticias conta com cadastro, atualização, exclusão de noticias, clipping, comentários. O cadastramento de noticias será feito por usuário logado com o perfil "Editor" podendo cadastrar, atualizar e excluir noticias criada por ele. O clipping que é uma associação da noticia com um usuário conterà apenas um campo de avaliação da noticia como positiva, neutra ou negativa. O sistema contara com gráficos e relatórios para verificar histórico das noticias. As noticias estarão associada sempre a um ou mais categorias, esse modulo também contara com o sistema de tags para a noticia.

Por fim o modulo de gerenciamento de amigos onde poderemos adicionar, remover usuários da nossa lista de amigos, visualizar a ultima atividade dos usuários.

Para implementação dos modelos, se seguirmos as convenções, não precisaremos adicionar nada. Os modelos sempre andam juntos com as migrações.

Por padrão o Rails vai pluralizar os nomes dos modelos na hora de criar as tabelas correspondentes, este padrão pode ser desabilitado adicionando a seguinte linha no arquivo *environment.rb*

- `ActiveRecord::Base.pluralize_table_names = false`

No clipping esta opção é verdadeira as tabelas estão sendo criadas pela pluralização do nome dos modelos. Para aproveitar algumas convenções do Rails algumas tabelas foram criadas toda em inglês, outras apenas algumas colunas ficaram em inglês.

5.1 AMBIENTE DE DESENVOLVIMENTO

Neste capitulo será apresentado o ambiente de desenvolvimento usado para desenvolver o projeto clippingweb Ruby on Rails.

5.1.1 IDE

Para desenvolvimento em Ruby / Ruby on Rails será usada a IDE NetBeans 6, um projeto de código aberto de sucesso, com uma grande base de usuários, uma comunidade crescente e perto de 100 parceiros mundiais. A Sun Microsystems fundou o projeto NetBeans em junho de 2000 e continua sendo seu principal patrocinador.

Hoje, existem dois produtos: a IDE NetBeans (NetBeans IDE) e a Plataforma NetBeans (NetBeans Platform).

A IDE NetBeans é um ambiente de desenvolvimento - uma ferramenta para programadores, que lhe permite escrever, compilar, debugar e instalar programas. A IDE é completamente escrita em Java, mas pode suportar qualquer linguagem de programação. Existem também um grande número de módulos para estender a IDE NetBeans. A IDE NetBeans é um produto livre, sem restrições de como ela pode ser usada. O Banco de Dados MySQL é o que melhor se integra ao Ruby on Rails.

5.1.2 Banco de Dados

O MySQL é um SGBD relacional que utiliza a linguagem padrão SQL (structured Query Language) e é largamente utilizado em aplicações para a internet. É o mais popular entre os banco de dados com código fonte aberto. Há milhões de instalações do MySQL em todo mundo, inclusive sites com alto volume de dados e tráfego, como Associated Press, Google, NASA, entre outros.

O MySQL é um alternativa atrativa porque, mesmo possuindo uma tecnologia complexa de banco de dados, seu custo é bastante baixo. Tem como destaque suas características de velocidade, escalabilidade e confiabilidade, o que vem fazendo

com que ele seja adotado por departamentos de TI (Tecnologia da Informação), desenvolvedores Web e pacotes de softwares.

A seguir são listadas algumas vantagens do MySQL:

- Número ilimitado de utilização por usuários simultâneos;
- Capacidade de manipulação de tabelas com grande quantidade de registros;
- Alta velocidade na execução de comandos;
- Fácil e eficiente controle de privilégios de usuários.

Para desenvolvimento de sites dinâmicos, o MySQL forma uma excelente dupla com linguagens de script como PHP e Ruby on Rails, tanto para pequenos websites como para grandes portais.

A seguir veremos Tipos de Tabelas do MySQL, pois como Rails produz aplicações orientadas a acesso a dados é necessário saber escolher o tipo de tabela, afim de aumentar o desempenho da aplicação.

O tipo MyISAM, são as tabelas do tipo padrão do MySQL. São as mais rápidas porque não possuem certos recursos que os outros tipos de tabelas têm, como por exemplo, as tabelas MyISAM não suportam chaves estrangeiras nem suportam transações. A maior vantagem destas tabelas é a velocidade de pesquisa e de atualizações. O MyISAM é apropriado para tabelas cujos dados não mudam com frequência. Um exemplo disso é uma tabela de cidades e estados. Como geralmente este tipo de tabela é usado apenas como consultas, não há a necessidade do uso de InnoDB.

InnoDB são tabelas que permitem transações e mais alguns recursos que as MyISAM não suportam. InnoDB funciona mais rápido que MyISAM quando há modificações constantes nos dados, uma vez que este tipo de armazenagem usa a proteção por registros (row locking) e não a proteção por tabelas (table locking) como o faz o MyISAM. Contudo, em várias situações, InnoDB é mais lento que MyISAM, devido ao fato de que InnoDB funciona com transações. A armazenagem

padrão do MySQL é MyISAM. Outro ponto interessante são as restrições de chaves estrangeiras (foreign key constraints) e transações. Estes recursos estão disponíveis apenas no InnoDB.

5.2 MODELAGEM DO SISTEMA

Para melhor visualizar a domínio da aplicação, nos capítulos seguintes veremos o diagrama de caso de uso e diagrama de classes.

5.2.1 Diagrama de caso de uso

O diagrama de caso de uso do sistema ClippingWeb é ilustrado na figura 6.

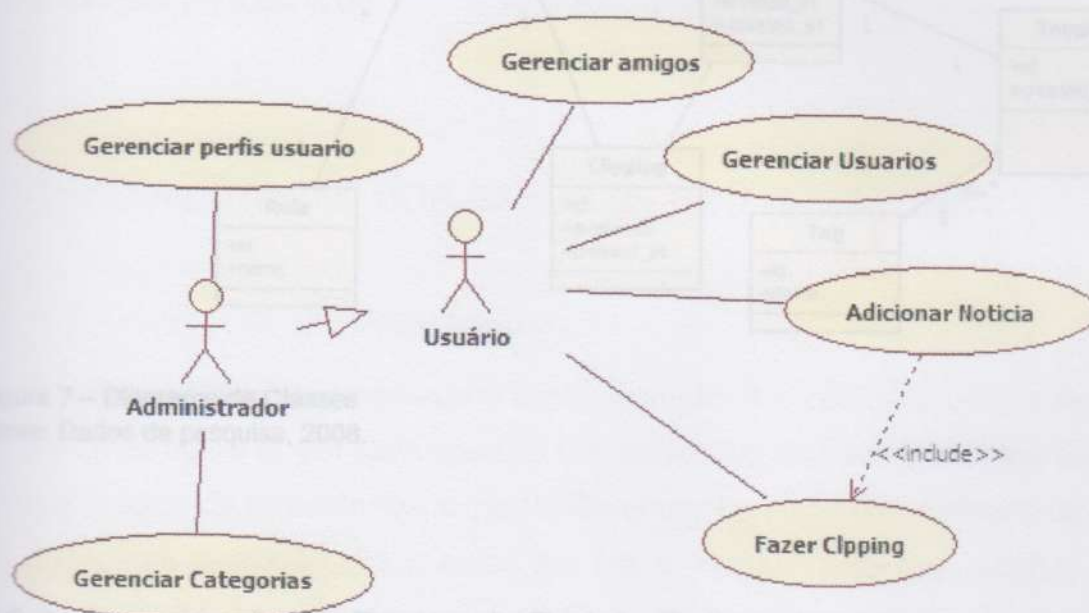


Figura 6 – Diagrama de Caso de Uso
Fonte: Dados da pesquisa, 2008.

5.2.2 Diagrama de Classes

O diagrama de classes do sistema de Clipping, na figura 7, onde é evidenciado suas principais entidades e seus relacionamentos.

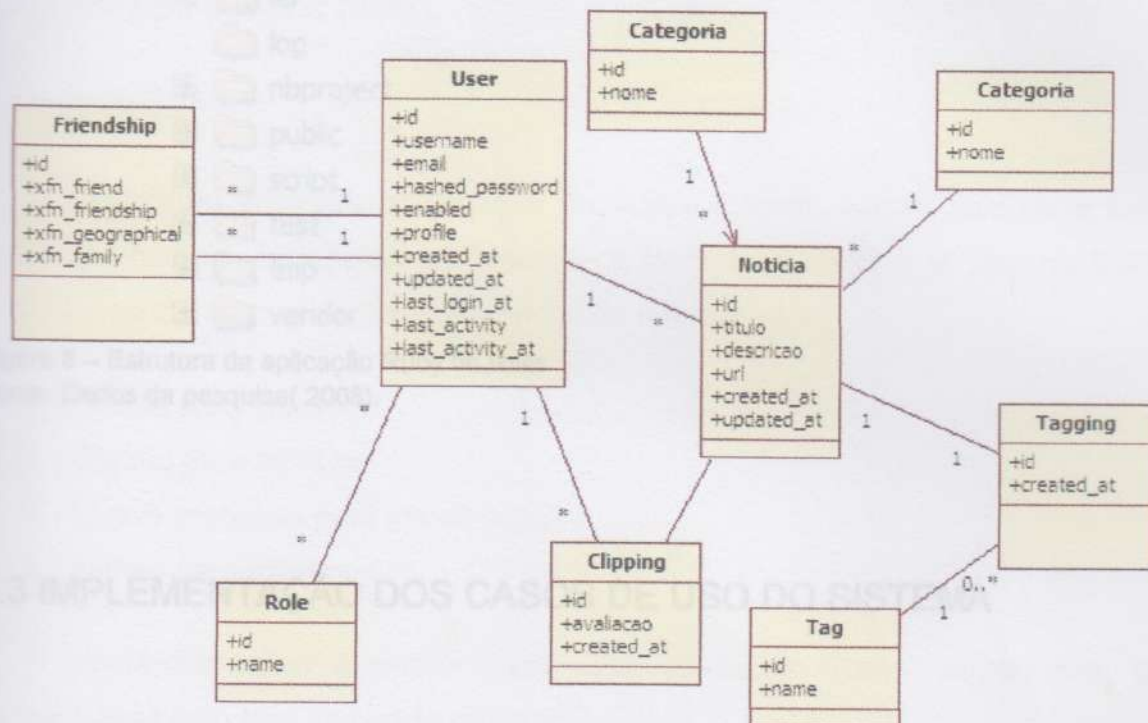


Figura 7 – Diagrama de Classes

Fonte: Dados da pesquisa, 2008.

5.2.3 Organização de Pastas do Projeto Rails

Para criar o projeto Ruby on Rails, depois que esta instalado e configurado o comando rails "nome da aplicação" irá gerar uma estrutura de pastas mostrada na figura 8.

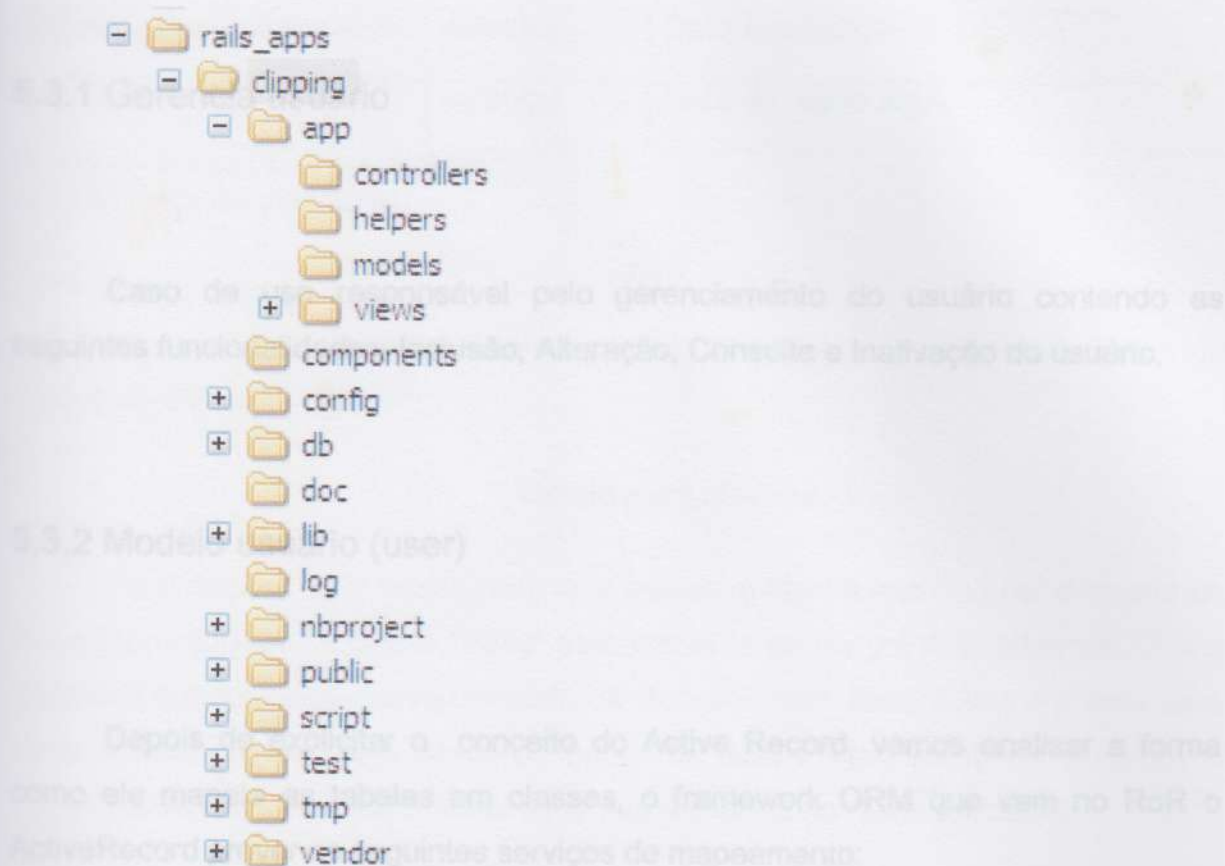


Figura 8 – Estrutura da aplicação Ruby on Rails
 Fonte: Dados da pesquisa(2008).

5.3 IMPLEMENTAÇÃO DOS CASOS DE USO DO SISTEMA

Usaremos como exemplo o esquema de tabela "users" do sistema, que formará pela estrutura mostrada no quadro 1

Nesta seção é apresentada a implementação dos casos de uso mostrado no diagrama da figura 6. Em cada caso de uso será dado foco em diferentes assuntos, e estar dividido da seguinte forma: no (1) Gerenciar usuário será explicado um pouco do framework ActiveRecord e como ele faz o mapeamento das tabelas; no (2) Gerenciar Perfis, os relacionamentos do Rails; (3) Gerenciar notícias, destaque para a criação de tags e feeds RSS; (4) Clipping com o uso do Ajax ; (5) Gerenciar categoria será mostrado o controle de acesso e validações; e (6) Gerenciar Amigos será mostrado a implementação do conceito de redes sociais através de uma lista de amigos.

created_at	nome	Data de Criação

5.3.1 Gerencia usuário

Caso de uso responsável pelo gerenciamento do usuário contendo as seguintes funcionalidades: Inclusão, Alteração, Consulta e Inativação do usuário.

5.3.2 Modelo usuário (user)

Depois de explicitar o conceito do Active Record, vamos analisar a forma como ele mapeia as tabelas em classes, o framework ORM que vem no RoR o ActiveRecord prover os seguintes serviços de mapeamento:

1. Tabela para classes
2. Coluna para atributos
3. Chave primarias para ids de objetos
4. Linhas para objetos

Usaremos como exemplo o esquema da tabela "users" do sistema, que formada pela estrutura mostrada no quadro 1.

Nome	Tipo	Descrição
id	integer	Id do usuário
username	string	Nome do usuário
email	string	E-mail
hashed_password	string	Senha (criptografia hash)
enabled	boolean	Habilitado ou Desabilitado
profile	text	Texto livre
created_at	datetime	Data da Criação

updated_at	datetime	Data Atualização
last_login_at	datetime	Data do ultimo login

Quadro 1 – Modelo User / Tabela Users

Fonte: Dados da pesquisa(2008)..

Nos parágrafos seguintes será explicado os serviços de mapeamento provido pelo ActiveRecord.

Tabela para classes

Para mapear uma tabela para uma classe, a classe tem que ser derivada de ActiveRecord::Base. A classe "Base" está presente no pacote ActiveRecord. O que acontece quando uma classe derivada de ActiveRecord::Base é que a classe atua como intermediário com a tabela de banco de dados. Para identificar a tabela, o ActiveRecord assume que o nome da tabela é o plural do nome da classe.

Esse comportamento pode ser desativado inserindo na configuração do ambiente "ActiveRecord::Base.pluralize_table_names = "false" ou atribuindo a diretiva "set_table_name".

A figura 9 mostra a User, comandos de validação são colocados no modelo.

```
class User < ActiveRecord::Base

  attr_protected :hashed_password, :enabled
  attr_accessor :password

  validates_presence_of :username
  validates_presence_of :email
  validates_presence_of :password, :if => :password_required?
  validates_presence_of :password_confirmation, :if => :password_req

  validates_confirmation_of :password, :if => :password_required?

  ...
end
```

Figura 9 – Classe user (usuário)

Fonte: Dados da pesquisa(2008)..

A classe *User* possui uma migração que criará a tabela, figura 10 mostra a migração para tabela da Classe *User*.

```
class CreateUsers < ActiveRecord::Migration
  def self.up
    create_table :users do |t|
      t.column :username, :string, :limit=>64, :null=>false, :primary_key => true
      t.column :email, :string, :limit=>128, :null=>false
      t.column :hashed_password, :string, :limit=>64, :null=>false
      t.column :enabled, :boolean, :default => true, :null=>false
      t.column :profile, :text
      t.column :created_at, :datetime
      t.column :updated_at, :datetime
      t.column :last_login_at, :datetime
    end
  end

  def self.down
    drop_table :users
  end
end
```

Figura 9 – Migração do modelo User (usuário)
Fonte: Dados da pesquisa(2008)..

Como Rails abstrai os comandos e os tipos de dados específicos de cada SGBD, fica fácil mudar de banco de dados a qualquer momento. A seguir vamos ver como o Rails usando o Active Record mapeia as Tabelas:

Colunas para atributos

Uma vez que a tabela tenha sido mapeada para uma classe, não precisa de nenhuma configuração explícita para mapear as colunas. Isto por que o Active Record determina os atributos a partir de uma tabela de banco de dados, em tempo de execução. Tipicamente, Active Record reflete o esquema para configurar uma classe que mapeia uma tabela.

As colunas campos `created_at` e `updated_at` da tabela `users`, são atualizadas automaticamente pelo ruby on rails com a data de criação e atualização do objeto.

5.3.4 Gerenciar Notícias

Chaves primarias para ids

O Active Record por padrão assume o campo da tabela com o nome "id" o identificador do objeto, podendo ser alterado mudando a diretiva "set_primary_key".

No sistema de clipping todas as tabelas seguem o padrão usando como identificador da tabela o campo id.

Linhas para objetos

Quando alguma operação retorna informações de uma tabela, os valores da colunas das linhas retornadas se tornam valores dos atributos do objetos.

5.3.3 Gerenciar perfis do usuário

O sistema admitirá usuários privilegiados (administradores), capazes de visualizar a totalidade das ações implementadas pelos usuários. Este ambiente de administração oferecerá, portanto, tabelas e gráficos que permitam, entre outras, as seguintes funcionalidades:

1. Inativação do Usuário
2. Acesso a tabelas básicas do sistema.
3. Consulta histórico dos usuários.
4. Alterar perfis do usuário

O perfil padrão do sistema é o "Editor" permitindo assim ao usuário adicionar e editar notícias, fazer clipping e fazer consultas relacionada apenas a seu usuário.

5.3.4 Gerenciar Notícias

Caso de uso responsável pelo gerenciamento de notícias. Os usuários logados com perfil "Editor" poderão criar, atualizar as notícias e para apaga-las será necessário o perfil "Administrador". As notícias serão categorizada através de categorias pré definidas pelo administrador e indexadas através de tags definidas pelos usuários.

5.3.4.1 Interação Visão / Controlador

Para identificar qual controlador será responsável por atender uma determinada solicitação da aplicação, o Rails utiliza um mecanismo de roteamento de requisições automático que não necessita de configurações complexas, mas que pode também ser customizado de acordo com as necessidades de cada aplicação. A figura mostra detalhes do arquivo route.rb que contém a configuração do roteamento.

Nome	Tipo	Descrição
<pre>map.resources :noticias, :collection => {:admin => :get} map.resources :categorias, :collection => {:admin => :get} do categorias.resources :noticias, :name_prefix => 'categoria_' end</pre>		
<pre>map.resources :tags, :collection => {:admin => :get} do tags tags.resources :noticias, :name_prefix => 'tag_' end</pre>		
<pre>map.show_user '/user/:username', :controller => 'users', :action => 'show_by_username'</pre>		Identificador de tabela
<pre># Install the default route as the lowest priority. map.connect ':controller/:action/:id.:format' map.connect ':controller/:action/:id'</pre>		
		Tipo do Objeto

Figura 11 - Arquivo de configuração do roteamento routes.rb

Fonte: Dados da pesquisa(2008).

A regra principal de roteamento no Rails forma URLs segundo o padrão `/controlador/ação/id`, onde controlador é a classe responsável por atender a requisição especificada por aquela URL, ação é um método dentro da classe e id é um parâmetro opcional passado para identificar um objeto qualquer sobre o qual a ação será efetuada.

5.3.4.2 Fazendo cadastramento da noticia usando tags

Para cadastrar a noticia usando a categorização feita pelos usuários através de tags foi utilizado o componente "acts_as_taggable". O Plugin tem basicamente dois modelos, o Modelo Tag mostrado no quadro 2 contem apenas o identificador e o nome da tag, enquanto que o tagging mostrador na quadro 3 possui o identificador, identificador da tag, identificador do objeto e tipo do objeto.

O campo `taggable_id` e `taggable_type` define o relacionamento entre as classes.

Nome	Tipo	Descrição
id	integer	Identificador da tabela
name	string	Nome da tag

Quadro 2 – Modelo Tag

Fonte: Dados da pesquisa(2008)..

Nome	Tipo	Descrição
id	integer	Identificador da tabela
tag_id	integer	Id da tag
taggable_id	integer	Id do objeto
taggable_type	string	Tipo do Objeto

Quadro 3 – Modelo Tag

Fonte: Dados da pesquisa(2008)..

O `acts_as_taggable` estar preparado para marcar com tags qualquer tipo de objetos, sejam eles notícias, imagens, vídeos. Isso por que o modelo tagging se relaciona de forma polimórfica com vários tipos de objetos.

No Rails, uma associação polimórfica(THOMAS , 2006) é uma associação que liga objetos de diferentes tipos. O pressuposto é que esses objetos todos partilhem algumas características em comuns, mas que eles têm diferentes representações.

Na associação polimórfica, o modelo é associado a objetos de mais de um modelo de classe, como mostra a figura 13. A fim de salvar este relacionamento com exatidão, os nome das classes e ids dos objetos precisam ser gravados.



Figura 13 - Relacionamento Polimorfo

Fonte: Dados da pesquisa(2008)..

O código da listagem 1 mostra as duas classes de modelo do componente `'acts_as_taggable'`:

```
Class Tag < ActiveRecord::Base
```

```
  has_many :tagging
```

```
  ....
```

```
end
```

```
class Tagging < ActiveRecord::Base
```

```
belongs_to :tag
belongs_to :taggable, :polymorphic => true
```

```
...
```

```
end
```

Listagem 1 – Código das Classes `acts_as_taggable`

Fonte: Dados da pesquisa(2008)..

5.3.5 Para colocar a funcionalidade de tags em Notícia é necessário apenas acrescentar a diretiva **`acts_as_taggable`** no modelo, na view `new.rhtml` foi adicionado o código na listagem 2.

```
<% form_for :noticia do |f| %>
  ☐
  <p> Tags:<br /> <%= text_field_tag 'tags', params[:tags] %>
</p>
<p><%= submit_tag %></p>
<% end %>
```

Listagem 2 - Formulário `new.html`

Fonte: Dados da pesquisa(2008)..

Para o controlador de notícias basta apenas adicionar `@noticia.tag_with params[:tags] if params[:tags]`, que o mesmo se encarrega de salvar a tags e os relacionamentos, tratando automaticamente como tags as palavras separadas por espaço.

5.3.4.3 Criando feeds RSS

Após inserir a noticia estaremos apto a publicá-la, uma forma de fazer isso é criando feeds RSS. O Método `index` do controlador de noticias também responde a requisições para RSS e ATOM, entretanto pelo fato do XML ainda não estar definido

no rails, é necessário primeiramente especifica-lo em um arquivo .xml que define a estrutura do XML.

Para facilitar a geração de documentos XML o Rails emprega uma biblioteca chamada BUILDER. Esta biblioteca oferece uma maneira simples de criar arquivos XML válidos.

5.3.5 Fazer clipping

5.3.5.1 Ajax

Caso de uso responsável pelo clipping da noticia, que consiste na avaliação da noticia pelos usuários logados, que no ato da avaliação da mesma seleciona as opções: positiva, neutra ou negativa. Para cada avaliação não importa qual, o sistema contabilizará como voto de popularidade.

Como essa operação deve ser rápida o Ajax promoverá o comportamento ideal, por gerar uma requisição assíncrona e alterando apenas o totalizador da votação sem redesenho total da pagina. No cadastramento da noticia o editor já expressará sua opinião em relação a avaliação da mesma.

O relacionamento "has_many", mostrado na figura 13, define que uma Noticia é associada com muitos registro do tipo Clipping, enquanto belongs_to um objeto do tipo Clipping é associado a um objeto do tipo Noticia.

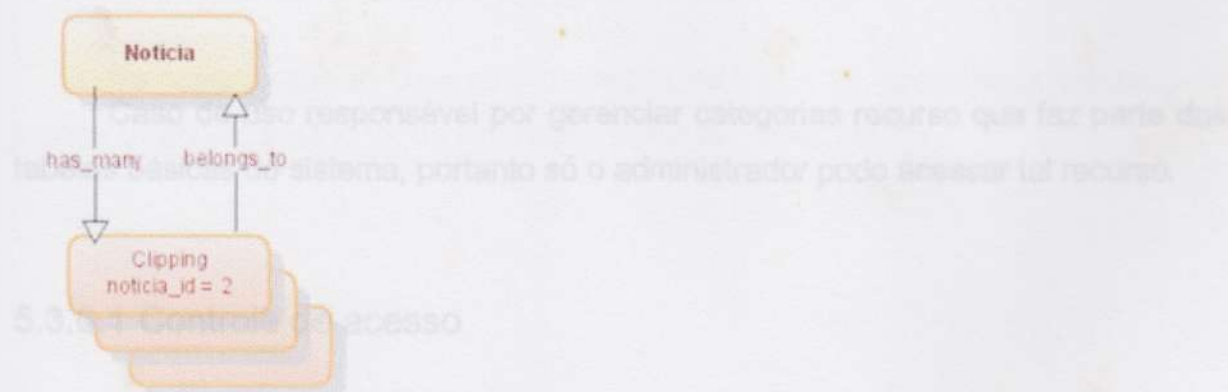


Figura 13: Relacionamento has_many \ belongs_to para Noticia e Clipping

Fonte: Dados da pesquisa(2008)..

Para que cada notícia adicionada inicialize com o clipping do editor, adicionaremos mais uma opção no cadastrado da notícia onde o usuário irá selecionar as opções Positivo, Negativo ou Neutro. Essa solução é viável para o clipping de quem estar adicionando a notícia, para os outros usuários, qualquer redirecionamento traria um problema de usabilidade. Então para prover o clipping das notícias um modo mais amigável, foi utilizado uma solução Ajax, os framework utilizados são o Prototype e Script.aculo.us que vem integrado o Rails e são usados de forma transparente.

5.3.5.1 Ajax

Para entender o uso de Ajax no clipping, vamos entender os Arquivos RJS: os arquivos com a extensão *.rjs* são usados para modificar partes de uma página existente. Quando usamos arquivos RJS a informação que é transferida para o *browser* do usuário é uma série de instruções Javascript que modifica, estende e aplica efeitos visuais nos elementos do HTML. Não é necessário adicionar nada de especial no controlador ou criar ações para usar RJS, simplesmente coloque o arquivo que tem a extensão *.rjs* na pasta que contém as views que deseja, e esta pronto para o uso.

5.3.6 Gerenciar Categorias

Caso de uso responsável por gerenciar categorias recurso que faz parte das tabelas básicas do sistema, portanto só o administrador pode acessar tal recurso.

5.3.6.1 Controle de acesso

Pare restringir o acesso as paginas. O código mostrado na figura 14 mostra parte da codificação do controlador.

```

class CategoriasController < ApplicationController
  before_filter :check_administrator_role, :except => [:index, :show]

  def index
    @categorias = Categoria.find(:all)
    respond_to do |wants|
      wants.html
      wants.xml { render :xml => @categorias.to_xml }
    end
  end

  def show
    @categoria = Categoria.find(params[:id])
    respond_to do |wants|
      wants.html { redirect_to categoria_news_url(:categoria_id => @cat
      wants.xml { render :xml => @categoria.to_xml }
    end
  end
end

```

Figura 14: Controlador da Categorias
 Fonte: Dados da pesquisa (2008).

O CategoriasController libera o acesso para qualquer usuário para as ações index e show, nas outras ações ele verifica se o mesmo tem perfil de administrador.

5.3.6.2 Validações

Validações, como o próprio nome indica, é um modo de garantir a integridade dos dados em uma aplicação. O modelo de validações que o Rails fornece é bastante completo e pode ser facilmente expandido pelo desenvolvedor caso ele necessite de algo não fornecido por padrão nas bibliotecas.

No caso da classe Categoria mostrado na figura 15, precisamos validar pelo menos o fato do usuário ter de informar o nome da categoria.

```

class User < ActiveRecord::Base
  has_many :friendships
  friends, through => :friendships, class_name => 'User'
end

```

```
class Categoria < ActiveRecord::Base

  has_many :noticias
  validates_length_of :nome, :maximum => 80
  validates_presence_of :email, :message => 'é obrigatório'
end
```

Figura 15: Classe Categoria

Fonte: Dados da pesquisa(2008).

Múltiplas validações podem ser efetuadas em um mesmo campo, como caso de validar o tamanho do campo *validates_length_of*. Por padrão as mensagens estão em inglês, mas atribuindo uma mensagem ao atributo *:message* o desenvolvedor poderá customizar as mensagens.

5.3.7.2 Microformato XFN

5.3.7 Gerenciar Amigos

Caso de uso responsável pelo gerenciamento de amigos. Para fazer o gerenciamento de amigos, precisaremos criar e modificar lista de amigos. Faremos isso criando um recurso chamado *friends* (amigos). Cada usuário poderá ter apenas uma lista de amigos. Registraremos a ultima atividade e a data das mesma, para adicionar amigos a lista, o usuário deve entrar nos detalhes do amigo.

5.3.7.1 Relacionamentos entre os modelos

O modelo *Friendship*(Amizade) pertence a duas instâncias de *User* (Usuário) : uma como usuário e outra como amigo. A listagem 3 mostra o código das duas classes, evidenciando seus relacionamentos *has_many*, *through*.

```
class User < ActiveRecord::Base....
  has_many :friendships
  :friends, :through => :friendships, :class_name => 'User'
```

```

.....
end

class Friendship < ActiveRecord::Base
  belongs_to :user
  belongs_to :friend, :class_name => 'User',
  :foreign_key => 'friend_id'
  ...
end

```

Listagem 3 – Relacionamento User / Friendship
 Fonte: Dados da pesquisa(2008).

5.3.7.2 Microformato XFN

Quando mostramos um link para um usuário, seja amigo ou você mesmo, nós podemos adicionar informações extra sobre este relacionamento usando um microformato chamado *XHTML Friends Network* (XFN). O XFN rede de amigos em XHTML, é uma maneira de representar relações entre pessoas através de links, afim de tentar mostrar quais as relações pessoais entre essas pessoas.

Existem classificações em termos de amizade, físicas, geográficas, profissionais, parentesco e romance. E essas classificações, mostradas na listagem 4, podem ser colocadas nos links através do atributo *rel* que cada um possui.

- Identidade
 - *me* para si mesmo
- Amizade
 - *contact* para contatos em geral
 - *acquaintance* para conhecidos
 - *friend* para amigos
- Encontro físico
 - *met* para pessoas que você conhece pessoalmente
- Profissional
 - *co-worker* para pessoas da mesma profissão

- *colleague* para colegas de trabalho
- Geográfico
 - *co-resident* para pessoas que habitam o mesmo lugar que você
 - *neighbor* para vizinhos
- Parentesco
 - *child* para filhos
 - *parent* para pai ou mãe
 - *sibling* para irmão ou irmã
 - *spouse* para cônjuge (*spouse* é um falso amigo no inglês, embora pareça esposa serve para ambos)
 - *kin* para parentes em geral
- Romance
 - *muse* para musa
 - *crush* para quem você tem uma queda
 - *date* para encontros, ou “ficantes” aqui no Brasil
 - *sweetheart* para namorada(o)

Listagem 4 – Classificações XFN

Fonte: JOHN(2007)

Estas informações podem ser mostradas paralelamente com os links usando CSS, ou podem ser usadas por outras aplicações para mapear seus relacionamentos no clippingweb ou em até mesmo com outras redes sociais.

5.3.7.3 Estabelecendo e Editando relacionamento com amigo

Para criar as relações, foi implementada uma pagina em que o usuário classifica através de *checkboxes* e *radio buttons* todas as opções relacionadas. Na lista de amigos, cada amigo mostrado em forma link e uma uma imagem que indica que tipo de relacionamento.

A figura 17 mostra a tela de cadastrado / edição de um relacionamento

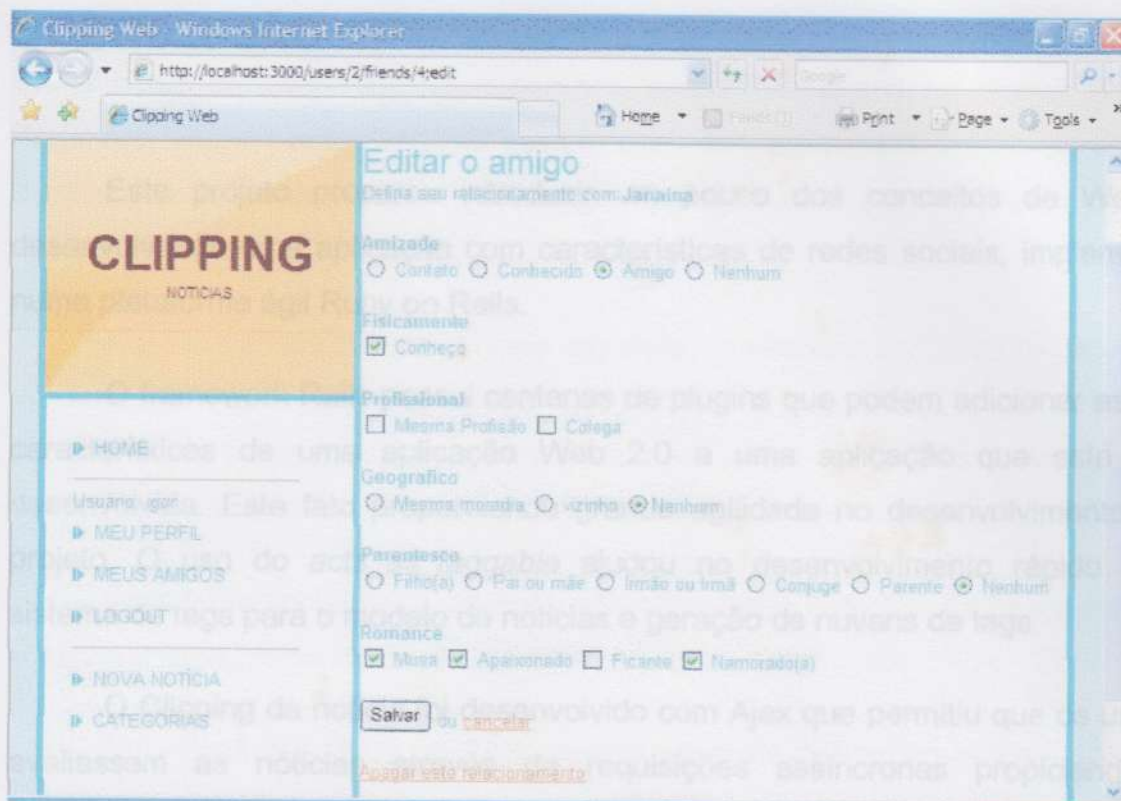


Figura 16: Tela Editando relacionamento com o amigo

Fonte: Dados da Pesquisa

Sabendo que as relações se alteram com o decorrer do tempo, o sistema suporta a edição do relacionamento ou até mesmo a exclusão.

Relacionamento dos usuários é reforçado com a edição de metadados baseado no micro formato XFN, que permite aos usuários e aplicações facilmente compreender os relacionamentos entre os usuários e os seus amigos.

O sistema de controle de usuários e perfis (roles), permitiu adicionar os usuários e gerenciar suas permissões. Usuários já cadastrados podem fazer alterações nas suas informações pessoais.

A IDE Netbeans mostrou grande flexibilidade e integração com o Ruby on Rails, que junto com MySQL trabalham de forma integrada. O tempo economizado no desenvolvimento deve ser utilizado para melhorar o desempenho do SGBD e do servidor Web além de deixar a aplicação Ruby on Rails mais rápida.

Alguns pontos de melhoria no sistema devem ser observados, como a integração da rede social, pois uma das grandes problemas das redes sociais é o isolamento que cada usuário promove. Não no sentido de ficar longe das pessoas que

6 CONCLUSÃO

Este projeto procurou introduzir um pouco dos conceitos de Web 2.0, desenvolvendo uma aplicação com características de redes sociais, implementado numa plataforma ágil Ruby on Rails.

O framework Rails possui centenas de plugins que podem adicionar as várias características de uma aplicação Web 2.0 a uma aplicação que está sendo desenvolvida. Este fato proporcionou grande agilidade no desenvolvimento deste projeto. O uso do *acts_as_taggable* ajudou no desenvolvimento rápido de um sistema de tags para o modelo de notícias e geração de nuvens de tags.

O Clipping da notícia foi desenvolvido com Ajax que permitiu que os usuários avaliassem as notícias através de requisições assíncronas propiciando boa usabilidade a funcionalidade.

A Lista de Amigos desenvolvida permite que usuário adicione outros usuários a suas listas. A partir desta lista, o usuário pode rapidamente ver a última atividade do seus amigos, assim ele sabe se um amigo adicionou uma nova notícia. O Relacionamento dos usuários é reforçado com a adição de metadados baseado no micro formato XFN, que permite aos usuários e aplicações facilmente compreender os relacionamentos entre os usuários e os seus amigos.

O sistema de controle de usuários e perfis(*roles*), permitiu adicionar os usuários e gerenciar suas permissões. Usuários já cadastrados podem fazer alterações nas suas informações pessoais.

A IDE Netbeans mostrou grande flexibilidade e integração com o Ruby on Rails, que junto com MySQL trabalham de forma integrada. O tempo economizado no desenvolvimento deve ser utilizado para melhorar o desempenho do SGDB e do servidor Web afim de deixar a aplicação Ruby on Rails mais rápida.

Alguns pontos de melhoria no sistema devem ser observados, como a integração da rede social, pois uns dos grandes problemas das redes sociais é o isolamento que cada uma promove. Não no sentido de ficar longe das pessoas que

you know and yes, if you are part of the social network X and its friends, of Y, it does not exist as talking one with the other. Initiatives with Open Social Google and Facebook came to fill this gap, that just as the RSS aggregates its readings, it makes sense to have a system that aggregates its social profiles throughout the Web.

In the management of friends some common functionalities should be added, such as: notify user when he is added to some list of friends, as well as asking for permission, adding a method to visualize mutual friends and visualization of the last updates of the same.

For the management of users adding means for which it is possible to recover the password and login, as well as a system of *newsletter*.

In relation to Ruby on Rails, it seems to be more indicated the use in applications oriented to access to data and Web communication, or in other words, in which the code of the application has a very small weight in the overall performance, since the majority of the work is done by a database system or Web server.

A good news is that Sun is hosting a project called JRuby in which it implements the Ruby language on the JVM, attacking the problems in two ways: with an interpreter implemented in Java and a compiler that converts Ruby source code to Java bytecode. In both cases, the JVM is responsible, in the last instance, for executing the Ruby code. Thus, the Ruby program benefits from all the sophistication of the Java platform.

REFERÊNCIAS

- BARABASI, A.-L. **Linked: The new science of networks**. Cambridge: Perseus. 2002
- BROWN J. S. & DUGUID, P. **The Social Life of Information**. Boston: Harvard Press. 2000..
- DEITEL, H. M. **Java Como Programar**, 3. 3d. São Paulo: Bookman. 2001.
- FOWLER, Martin. **Patterns of Enterprise Application Architecture**. U.S: Addison Wesley. 2002
- FLICKR. Disponível em: < <http://www.flickr.com/photos/tags/>>. Acesso em 18 jan. 2008.
- GARRET, Jesse. **The Elements of User Experience** . Boston: AIGA, 2007.
- JOHN, Allsopp. **Microformats: Empowering Your Markup for Web 2.0**. Berkeley: Apress. 2007.
- MARTINS, José C. **Gerenciando projetos de desenvolvimento de software com PMI, RUP e UML**, 2 ed. Rio de Janeiro: Brasport.2005.
- MANIFESTO Ágil, Disponível em : < <http://agilemanifesto.org/> >, Acesso em 18 jan. 2008.
- O'REILLY, Tim. **What's is Web 2.0**. Disponível em: <http://www.oreilly.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>. Acesso em 18 jan. 2008. O'reilly Media. 2005
- PORTENEUVE, Christophe. **Prototype and script.aculo.us**. Dalas: The Pragmatic Bookshelf . 2007.
- SYDIK, Jeremy J. **Design Accessible WebSites**. 2007. Dalas: The Pragmatic Bookshelf . 2007.
- THE OPEN Network for Bussines People, Disponível em : <<http://www.spoke.com/>>. Acesso em: 20 fev. 2008.
- THE WORLD map of social networks, Disponível em : <<http://valleywag.com/tech/data-junkie/the-world-map-of-social-networks-273201.php>>. Acesso em: 20 fev. 2008.
- THOMAS, Dave. **Agile Web Development with Rails**. 2. ed. 2006. Dalas: The Pragmatic Bookshelf . 2007.
- LENZ, Patrick. **Build Your Own Ruby On Rails Web Applications**. 1. Ed. SitePoint. 2007.

VOSSEN, Gottfried. **Unleashing Web 2.0**. Burlington: Morgia Kauffmann. 2007.

SEGARAN, Toby. **Programming Collective Intelligence**. U.S: Oreilly Media. 2007.