

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

MATHEUS STIVAL BERLESI

Octools: Um agregador de redes sociais

Trabalho de Graduação.

Prof. Dr. Leandro Krug Wives
Orientador

Porto Alegre, julho de 2012.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Vice-Reitor: Prof. Rui Vicente Oppermann

Pró-Reitora de Graduação: Profa. Valquiria Link Bassani

Diretor do Instituto de Informática: Prof. Luis da Cunha Lamb

Coordenador do CIC: Prof. Raul Fernando Weber

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

Agradecimentos

Agradeço, em primeiro lugar, aos meus pais, que me mantiveram no caminho certo e me deram as condições para concluir esta importante etapa de minha vida. Aos meus irmãos e irmã, pela colaboração nos momentos difíceis, e pelo compartilhamento dos momentos felizes, pelo apoio que me deram e o incentivo para seguir em frente. À minha namorada, pela paciência e compreensão, e pela força que me deu para superar os obstáculos mais difíceis. Aos colegas, muitos dos quais responsáveis por grande parte do meu aprendizado.

Agradeço, também, aos professores da UFRGS, mestres na arte de ensinar, que não apenas transferem conhecimento aos alunos como também instigam-nos a buscar o conhecimento e a excelência, mesmo que fora da universidade, e o fazem com alegria. Em especial, agradeço ao meu orientador, por sempre encontrar um espaço em sua agenda cheia para me ajudar, mesmo quando parecia impossível concluir o trabalho em tempo hábil.

Por fim, agradeço aos amigos, que sempre compreenderam a importância dos estudos em minha vida e me ajudaram em muitos momentos, sem esperar nada em troca, sendo responsáveis por grande parte dessa conquista.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS	6
LISTA DE FIGURAS.....	7
LISTA DE TABELAS	8
RESUMO.....	9
ABSTRACT	10
1 INTRODUÇÃO	11
2 FUNDAMENTAÇÃO TEÓRICA.....	13
2.1 Redes Sociais	13
2.1.1 Flickr.....	14
2.1.2 YouTube	15
2.1.3 Twitter	17
2.1.4 Facebook.....	19
2.1.5 Vimeo	21
2.2 Agregadores de Redes Sociais	22
2.2.1 TweetDeck.....	23
2.2.2 Seismic	24
2.3 Mashups.....	25
2.3.1 Profilactic	26
2.3.2 Flavors	27
3 OCTOOLS: UM MISTO DE AGREGADOR E MASHUP	28
3.1.1 Critérios de inclusão	28
3.2 Glossário de termos	29
3.3 Requisitos do sistema	30
3.3.1 Requisitos funcionais.....	30
3.3.2 Requisitos não funcionais	31
3.4 Casos de uso preliminares.....	32
4 SOLUÇÃO PROPOSTA.....	33
4.1 Tecnologias utilizadas	33
4.1.1 A linguagem de modelagem UML	33
4.1.2 A linguagem PHP	33
4.1.3 Banco de dados MySQL.....	34
4.1.4 A linguagem de marcação HTML.....	34
4.1.5 Folhas de estilo CSS	34
4.1.6 A linguagem JavaScript.....	34
4.1.7 APIs	35
4.2 Casos de uso	36
4.2.1 Diagrama de casos de uso essenciais.....	36
4.2.2 Casos de uso essenciais expandidos	37

4.3	Modelagem do banco de dados.....	44
4.4	Classes do sistema.....	45
4.4.1	Padrões de projeto	45
4.4.2	Diagrama de classes.....	46
4.5	Metodologia de desenvolvimento	49
4.6	Testes	50
4.6.1	Teste de segurança.....	50
4.6.2	Testes de usabilidade	50
5	GUIA DE USO DO SISTEMA	51
6	CONCLUSÃO.....	54
	REFERÊNCIAS	56

LISTA DE ABREVIATURAS E SIGLAS

AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
CRUD	Create, Retrieve, Update and Delete
CSS	Cascading Style Sheet
DM	Direct Message
HTML	HyperText Markup Language
PHP	PHP Hypertext Preprocessor
REST	Representational State Transfer
RT	Retweet
SGBD	Sistema de Gerência de Banco de Dados
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
TT	Trending Topics
UML	Unified Modeling Language
XML	Extensible Markup Language
XSS	Cross-site Scripting

LISTA DE FIGURAS

Figura 2.1: Tela de visualização de uma foto no Flickr	14
Figura 2.2: Tela de visualização de um vídeo no YouTube	16
Figura 2.3: Timeline de um usuário do Twitter	18
Figura 2.4: Feed de notícias do Facebook	20
Figura 2.5: Página de visualização de vídeo no Vimeo	22
Figura 2.6: Interface web do TweetDeck	24
Figura 2.7: Interface do Seismic, na versão web	25
Figura 2.8: Lifestream de usuário do Profilactic	26
Figura 2.9: Página de usuário no Flavors	27
Figura 3.1: Diagrama de casos de uso preliminares	32
Figura 4.1: Diagrama de casos de uso essenciais	37
Figura 4.2: Diagrama do banco de dados	45
Figura 4.3: Diagrama de classes	47
Figura 4.4: Código do método getObject, implementado pela classe Servico	48
Figura 4.5: Código do método send() da classe Flickr	49
Figura 5.1: Tela inicial do sistema Octools	51
Figura 5.2: Tela de gerenciamento de perfis no sistema Octools	52
Figura 5.3: Página de usuário do Octools, acessada pelo próprio usuário	52
Figura 5.4: Página de composição do Octools, com perfil do Twitter selecionado	53

LISTA DE TABELAS

Tabela 3.1: Critérios de inclusão das redes sociais	29
Tabela 3.2: Glossário de termos usados no decorrer do projeto.....	29
Tabela 4.1: Caso de uso efetuar cadastro no sistema	38
Tabela 4.2: Caso de uso cancelar cadastro no sistema	38
Tabela 4.3: Caso de uso associar perfil de serviço	39
Tabela 4.4: Caso de uso desassociar perfil de serviço.....	40
Tabela 4.5: Caso de uso publicar conteúdo em um serviço.....	41
Tabela 4.6: Caso de uso consultar publicações de um perfil.....	41
Tabela 4.7: Caso de uso editar conteúdo de publicação em um serviço	42
Tabela 4.8: Caso de uso excluir publicação de um serviço	43
Tabela 4.9: Caso de uso consultar timeline de um perfil Twitter	44

RESUMO

Com a popularização da Internet, e o crescente número de redes sociais online disponíveis neste meio, é natural que os usuários acabem tendo contas em diversas delas. O gerenciamento de tais contas acaba se tornando uma tarefa trabalhosa, dada a necessidade de acesso a diferentes ambientes para realizá-la. Outro problema que surge com a proliferação das redes sociais online é a dispersão do conteúdo criado por um usuário nesses vários ambientes. Para solucionar ambos os problemas, surgiram dois novos tipos de aplicações: os agregadores de redes sociais online e os *mashups*.

Assim, este trabalho propõe o projeto de um sistema que una características desses dois novos tipos de aplicações, a fim de unificar o ambiente para o gerenciamento e visualização do conteúdo gerado por um mesmo usuário em diferentes redes sociais.

Para isso, foi feita uma análise das principais funcionalidades de algumas das mais populares redes sociais no momento, assim como de alguns serviços de *mashup* e agregadores de redes sociais. O texto aborda ainda todos os elementos presentes no decorrer do projeto, desde o planejamento inicial até os detalhes finais da implementação de um protótipo.

Palavras-Chave: redes sociais online, agregador, *mashup*, API, sistema Web.

Octools: A social network aggregator

ABSTRACT

With the Internet popularization and the growing number of online social networks available in this environment, it is natural that users end up having accounts in many of them. The management of such accounts becomes a laborious task, given the need to access different environments to do it. Another problem that arises from the proliferation of online social networking is the dispersal of content created by a user in these different environments. To solve both problems, two new types of applications have emerged: online social network aggregators and mashups.

Therefore, this paper proposes the project of a system that unites the characteristics of these two new types of applications, in order to unify the environment to the management and viewing of the content created by the same user in different social networks.

To do that, an analysis of the main features of some of the most popular social networks of the moment was done, as well as some mashup services and social networking aggregators. The text also addresses all the present elements during the project, from the initial planning to the final details of the prototype implementation.

Keywords: online social networks, aggregator, mashup, API, Web system.

1 INTRODUÇÃO

Desde a criação da Internet, diversas novas aplicações vêm surgindo nesse meio, como o e-mail, comércio eletrônico, além de vários outros serviços Web. Recentemente, houve uma proliferação de redes sociais online e um crescimento muito rápido do número de pessoas que os utilizam.

Nem todas as redes sociais online têm a mesma finalidade, sendo umas mais voltadas para o relacionamento profissional (ex., LinkedIn), outras para o relacionamento entre amigos (ex., Facebook, Orkut, Google+), e outras para o compartilhamento de conteúdos específicos, como mensagens curtas (ex., Twitter), fotos (ex., Flickr), vídeos (ex., YouTube, Vimeo) e áudio (ex., SoundCloud).

Com tantas redes sociais online para nichos específicos diferentes, os usuários acabam criando contas em diversos desses serviços, o que gera certa dificuldade para postar conteúdo em todas, já que cada uma está em um endereço diferente. Encontrar todo o conteúdo de um determinado usuário também se torna um problema, uma vez que a informação está distribuída em vários sites na Internet.

Para amenizar este primeiro problema é que surgiram os agregadores de redes sociais online, sistemas que permitem acesso a várias redes sociais online simultaneamente, através de um portal único (BENEVENUTO; ALMEIDA; SILVA, 2011). Os mashups, sites personalizados ou aplicações web que usam conteúdo de mais de uma fonte para criar um novo serviço completo (MAXIMILIEN et al., 2007), surgiram na tentativa de amenizar o problema da informação distribuída.

Mesmo com a criação desses tipos de aplicação, os usuários continuam tendo a necessidade de acessar aplicações diferentes, uma para postar e outra para visualizar todo o conteúdo gerado por determinado usuário.

O objetivo deste trabalho é o projeto de uma aplicação agregadora de redes sociais online, que contará também com algumas funcionalidades características de um mashup, para evitar a necessidade de acesso a duas aplicações diferentes para interagir com várias redes sociais. Tal aplicação chama-se Octools. O Octools é, portanto, uma ferramenta disponível na web, que permite ao usuário cadastrado associar suas contas de diversas redes sociais online, permitindo a esse usuário postar conteúdo nesses serviços através da interface do site. Ao mesmo tempo, o conteúdo postado pelo usuário em todas essas redes sociais, associadas por ele ao Octools, estará disponível em um único endereço web.

Assim, nos próximos capítulos, serão brevemente introduzidos alguns conceitos relacionados com este trabalho. Será feita uma análise superficial das principais funcionalidades de algumas das redes sociais online mais populares atualmente, além de aplicações que se assemelham ao sistema proposto. Após esta análise do estado da arte,

serão apresentados elementos do planejamento da solução, como os requisitos de sistema e a modelagem inicial.

Posteriormente, a solução proposta será detalhada, sendo apresentadas as tecnologias escolhidas para o desenvolvimento da solução, além da arquitetura do sistema, usando para isso os casos de uso, além do diagrama de classes e a modelagem do banco de dados.

Um guia de uso do sistema será apresentado no capítulo 5, onde estarão disponíveis capturas de tela do protótipo desenvolvido, além do detalhamento das funcionalidades do site. Por fim, serão então apresentadas as conclusões do projeto, junto com sugestões de trabalhos futuros e uma curta avaliação do protótipo implementado.

2 FUNDAMENTAÇÃO TEÓRICA

A seguir, são apresentados alguns conceitos relacionados diretamente com o projeto da ferramenta proposta.

2.1 Redes Sociais

Segundo Weaver e Morrison (2008), uma rede social é essencialmente uma estrutura social composta por três ou mais entidades que se comunicam e compartilham informações. Ainda de acordo com Weaver e Morrison (2008), uma coalizão de pesquisa, uma tropa de escoteiros, uma igreja, uma universidade, ou qualquer número de outras relações socialmente construídas configuram redes sociais.

O conceito não está, portanto, diretamente relacionado à Internet, tendo surgido muito antes da rede mundial de computadores (WEAVER; MORRISON, 2008). Neste trabalho, no entanto, o estudo é focado em redes sociais na Internet – as ditas redes sociais online.

Uma rede social online, conforme definido por Boyd e Ellison (2008), é um serviço Web que permite aos usuários (1) construir perfis públicos ou semipúblicos dentro de um sistema, (2) articular uma lista de outros usuários com os quais ele(a) compartilha conexões e (3) visualizar e percorrer suas listas de conexões, assim como outras listas criadas por outros usuários do sistema. Diante disso, este trabalho considera que uma rede social online é um serviço Web que provê a criação de uma rede social em um ambiente digital.

De acordo com essa definição, existem inúmeras redes sociais online disponíveis na Internet, algumas com propósito mais específico, focadas no compartilhamento de um determinado tipo de conteúdo, e outras com propósito mais geral, focadas no relacionamento interpessoal, seja ele profissional ou pessoal.

A seguir são listadas algumas dessas redes e suas principais funcionalidades.

2.1.1 Flickr

Flickr¹ é um serviço de armazenamento e compartilhamento de fotos com funcionalidades de rede social online, fundado em 2004 e adquirido em 2005 pelo *Yahoo!* (CHA; MISLOVE; GUMMADI, 2009). A principal funcionalidade disponibilizada pelo serviço é justamente o *upload* de imagens, que é feito através do site.

Além do envio de imagens, os usuários podem criar redes de amigos, juntar-se a grupos, trocar mensagens, comentar, adicionar *tags*, notas e marcar pessoas em fotos, bem como marcar fotos como favoritas. O usuário que faz o envio da imagem pode ainda informar o local onde a foto foi batida, escolher um entre vários tipos de licença disponíveis para cada foto e limitar sua visibilidade, deixando visível para todos os usuários da rede ou apenas para seus amigos. É possível ainda criar álbuns, coleções e exposições com as imagens disponíveis no site.

A maior parte dos recursos do Flickr está disponível apenas para membros registrados no site, sendo permitido aos visitantes não registrados apenas visualizar o conteúdo.



Leaving On A Jet Plane

Last night I went to Torrey Pines Glider Port to try and catch some of the para gliders flying around near the cliffs, unfortunately, they had all packed up!

Waited around for the sunset which slowly sank into the dreaded 'marine layer'.
en.wikipedia.org/wiki/Marine_layer

Comentários e favoritas

★ robert.molinarius, CharlieBrown8989, Deia Silva, *♥ šhaú m: ur pìx *♥* e 57 outras pessoas adicionaram esta foto a suas favoritas.

 **annfrau** *pro* (58 meses atrás)
 Hi, I'm an admin for a group called [The Photo Of The Week](#) (Limit 1 per day!), and we'd love to have your photo added to the group.
 Fantastic sunset!
 --
 Seen on my Flickr home page. (?)

 **Deia Silva** *pro* (58 meses atrás)
 Great shot!

 **Swamibu** *pro* (58 meses atrás)
 That sunset is remarkable!

Por peasap
 Nome real desconhecido + Adicionar contato

Esta foto foi tirada em 14 de setembro de 2007 em La Jolla Shores, La Jolla, CA, Estados Unidos da América, usando uma uma Canon EOS 30D.



2.334 50 61

Esta foto pertence a

Galeria de peasap (326)



Esta foto também aparece em

▶ (álbum: 29)

▶ [School of Digital Photography](#) (grupo: 254753)

Tags

automne • autumn • ca • california • ciel • coucher • coucher de soleil • deep • eau • Etats-Unis • fall • fog • la jolla • mare • marine layer • mer • ocean • océan pacifique • orange • Pacific • Pacific Ocean • pacifique • reflections • reflets • rouge • san diego • sea • september • septembre • setting sun • SOE • sol • soleil • sun • sunset • ...e 6 mais tags

Licença

ⓘ Alguns direitos reservados
 ⓘ Solicitação de licença das fotos de peasap via Getty Images

Privacidade

■ Esta foto está visível para todos

Figura 2.1: Tela de visualização de uma foto no Flickr

¹ Flickr. <http://www.flickr.com>. Acesso em: 08 de Junho de 2012.

Não é necessário pagar para ser usuário do Flickr, porém existe uma modalidade, chamada “Pro”, na qual os usuários pagam uma taxa anual para ter direito a espaço ilimitado para fotos e alguns outros benefícios.

Estima-se que, atualmente, o Flickr hospede mais de sete bilhões de imagens (Tech Crunch, 2012). De acordo com dados do site Alexa², o Flickr ocupa a 48ª posição na lista de sites com maior tráfego nos últimos três meses, e a estimativa é de que 1,75% dos usuários de internet no mundo todo visitam o site.

Para desenvolvedores externos, o Flickr disponibiliza uma extensa API³ (“*Application Programming Interface*”, ou Interface de Programação de Aplicativos), que conta com mais de 100 operações, através da qual é possível que uma aplicação de terceiros execute ações sobre o conteúdo disponível nos servidores da rede social online. De acordo com o “Guia do desenvolvedor do Flickr”⁴, através da API é possível executar quase todas as ações disponíveis no site www.flickr.com.

2.1.2 YouTube

Criado em Fevereiro de 2005, o YouTube⁵ é uma rede social online voltada para o envio e compartilhamento de vídeos. Os usuários podem navegar pelos vídeos de outros usuários, bem como enviar seus próprios vídeos para o site de graça (WEAVER; MORRISON, 2008). É possível, também, postar comentários em vídeos, enviar mensagens para outros usuários, criar listas de reprodução ou inscrever-se em canais de outros usuários para receber uma notificação por e-mail quando novos vídeos forem adicionados. Um vídeo pode ser assinalado como resposta a outro vídeo, criando uma ligação entre os dois.

Tanto os vídeos quanto os comentários contam com um sistema de avaliação, que é feita pelos próprios usuários da rede social. Um usuário pode avaliar positiva ou negativamente um vídeo ou comentário, e o total de votos positivos e negativos é exibido na tela de visualização do vídeo, conforme pode ser visto na Figura 2.2, a seguir.

Para utilização plena do sistema, é necessário possuir uma conta de usuário e estar registrado. Visitantes não registrados podem apenas navegar pelas listas e assistir aos vídeos públicos, porém sem nenhuma possibilidade de interação.

² Alexa. <http://www.alexa.com/siteinfo/flickr.com>. Acesso em 08 de Junho de 2012.

³ Flickr API. <http://www.flickr.com/services/api/>. Acesso em 10 de Junho de 2012.

⁴ Flickr Developer Guide. <http://www.flickr.com/services/developer/>. Acesso em 10 de Junho de 2012.

⁵ YouTube. <http://www.youtube.com>. Acesso em 11 de Junho de 2012.

The image shows a YouTube video player interface. At the top, there is a search bar and navigation links: "Procurar", "Enviar vídeos", "Criar conta", and "Fazer login". The video title is "Skylands Hardcore: Episode 2" by "gusmanak", with "62 vídeos" listed below. The video player shows a Minecraft-style scene with stone structures and a sign that says "LUSTIC QUAYSON". The video progress is at 00:14 / 16:46. Below the player, there are buttons for "Gostei" (4 likes), "Compartilhar", and "97 exibições". The description reads: "Publicado em 08/06/2012 por gusmanak. Are you ready for some more DEATH?! In this episode the trio test their skills in combat, as this episode marks their first attempt to roam at night." Below the description, there are two comments. The first comment is from "gusmanak" in response to "DoctorG27", mentioning "Optifine". The second comment is from "DoctorG27" asking how to zoom in the video. On the right side, there is a list of recommended videos, including "Skylands Hardcore: Episode 1", "How to Evade a Zombie in DayZ", "Arma II Combined Operations: DayZ Mod", "Technic Pack Co-Op: Episode 40", "[irpg Minecraft TV] - Skylands EP.1", "Technic Pack Co-Op: Episode 41 Finale", "2 Hour Happy Hardcore & Trance Mix", "Minecraft PE Lets Play episode 2 - I LOVE", "Wrackage Survival Map MP [Sk-Cz] - Part", and "Official Lucky13Clans Clan Wars video! ~".

Figura 2.2: Tela de visualização de um vídeo no YouTube

O YouTube disponibiliza também uma ferramenta que gera código para incorporação de vídeos em outras páginas da Web. Esta opção pode ser desabilitada pelo usuário proprietário do vídeo, se assim o desejar. Outra opção disponível para o usuário que fez o *upload* de um vídeo diz respeito à privacidade do conteúdo, podendo este ser público, de forma que seja visível por qualquer visitante do site, ou privado, limitando sua visibilidade a uma lista de até 50 usuários convidados pelo proprietário. Assim como no Flickr, ao enviar um arquivo de vídeo para o site do YouTube, o usuário pode definir uma licença para o conteúdo; no entanto, apenas duas opções estão disponíveis aqui – licença padrão do YouTube ou Creative Commons. O usuário pode desabilitar a interação de outros usuários em seus vídeos, removendo a opção de comentários, votação ou a marcação de vídeos como resposta, por exemplo, ou limitar essa interação, permitindo apenas a exibição de comentários ou respostas em vídeo que tenham sido previamente aprovadas por ele. Os vídeos devem ter um título, pertencer a uma categoria, e, opcionalmente, podem ter uma descrição e uma lista de *tags*, além de informações sobre a data e o local da gravação.

Segundo informações encontradas no site Alexa⁶, o YouTube é o terceiro site mais acessado do mundo, tendo sido visitado por 32,762% dos usuários de internet, em média, entre os meses de abril e junho de 2012.

No site de desenvolvedores do Google⁷, estão disponíveis algumas APIs para manipulação do YouTube, fornecidas e mantidas pelo próprio serviço, que permitem a interação de outras aplicações com o conteúdo dos usuários da rede social online. A Data API, uma das opções oferecidas, permite que um programa execute a maior parte das operações disponíveis no site do YouTube, como a busca de vídeos e consulta a vídeos relacionados, além de possibilitar a autenticação de um usuário para posterior *upload* de vídeos, edição de listas de reprodução, entre outras funcionalidades (Google Developers, 2012).

2.1.3 Twitter

Twitter⁸ é uma aplicação Web que desempenha tanto a função de *microblogging* como de rede social online (CHU et al., 2010). Como *microblog*, permite a postagem de mensagens de texto curtas, com até 140 caracteres, e mais recentemente também de imagens, de forma rápida e prática (RUFINO; TABOSA; NUNES, 2010). As mensagens postadas no Twitter, também chamadas de *tweets*, podem incluir, opcionalmente, a localização do autor no momento da postagem.

O Twitter mantém visível uma lista dos assuntos mais comentados na rede social online no momento. Trata-se de uma lista de termos, conhecida como *Trending Topics*, ou *TTs*, ordenada de forma decrescente pelo número de citações. É possível acessar a lista dos *TTs* mundial, ou de determinadas regiões, como Brasil ou, ainda, São Paulo.

Um usuário do Twitter pode seguir outros usuários. Considerando uma relação onde um usuário A adiciona um usuário B à sua lista de amigos, A é um *seguidor* de B, enquanto B é um *amigo* de A (CHU et al., 2010). Ao seguir alguém, o seguidor A recebe em sua página principal, ou *timeline*, as mensagens postadas pelo amigo seguido, B. B pode optar por adicionar A aos seus amigos também, seguindo A de volta, mas isso não é obrigatório. Por padrão, todas as mensagens postadas pelo usuário são públicas, podendo ser acessadas por qualquer pessoa que visite seu perfil, porém o autor pode optar por proteger seus *tweets*; fazendo isso, seus *tweets* ficam visíveis apenas aos usuários aprovados por ele.

A partir de sua *timeline*, o usuário do Twitter pode ver, além de suas próprias postagens, os *tweets* de seus amigos. É possível responder *tweets*, ou marcar uma mensagem como favorita, além de *retweetar* (*RT*), ou seja, repostar ou encaminhar um *tweet* de algum amigo (YANG et al., 2010), a fim de que seus seguidores também recebam a mensagem. Especificamente nos *tweets* cujo usuário é também o autor, existe a possibilidade de excluir a postagem. Não há forma de alterar o conteúdo de uma mensagem já postada. A Figura 2.3, exibida a seguir, mostra uma captura de tela com a *timeline* de um usuário do Twitter.

⁶ Alexa. <http://www.alexa.com/siteinfo/youtube.com/>. Acesso em 08 de Junho de 2012.

⁷ Google Developers. <https://developers.google.com/youtube/>. Acesso em 10 de Junho de 2012.

⁸ Twitter. <http://www.twitter.com>. Acesso em 11 de Junho de 2012.

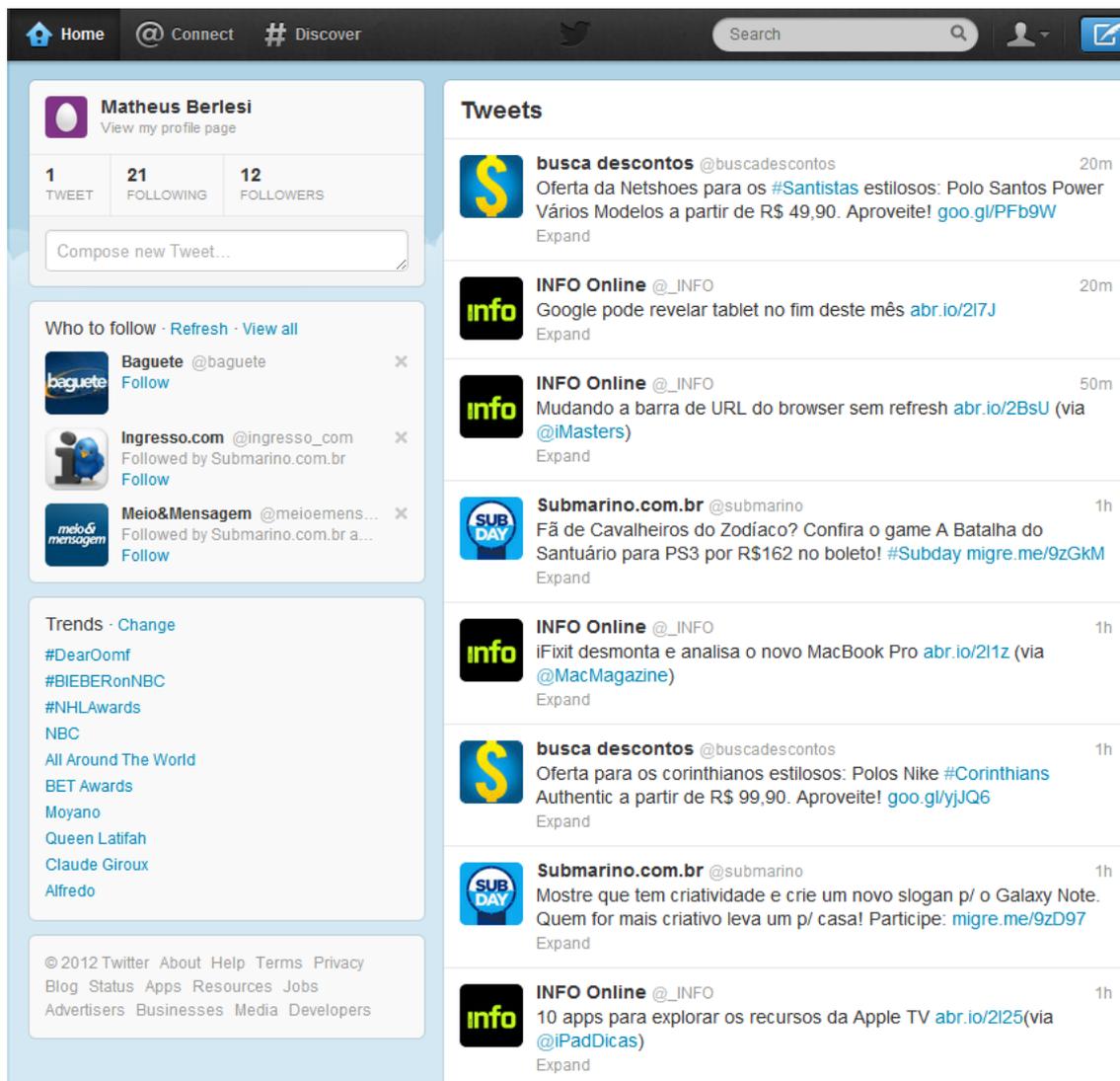


Figura 2.3: *Timeline* de um usuário do Twitter

Ao se cadastrar no Twitter, o usuário deve escolher um *username*, que deve ser único e servirá para identificá-lo na rede social online. Este *username* pode ser alterado a qualquer tempo pelo próprio usuário, desde que para outro *username* ainda não utilizado. No Twitter, um usuário é comumente referido pelo termo *@username*, onde *username* é justamente o nome de usuário escolhido por ele. Isso ocorre devido ao fato de que, ao escrever o termo *@username* em um *tweet*, a mensagem é entregue também ao usuário citado.

Através de abas na barra superior do site, o usuário pode alternar entre sua *timeline*, uma lista de *tweets* nos quais foi citado, ou as atividades de seus amigos, como marcar uma mensagem como favorita, passar a seguir alguém, entre outras.

Em um relacionamento onde dois usuários seguem um ao outro, é possível enviar mensagens privadas, chamadas de *Direct Messages*, ou simplesmente *DM*. Essas mensagens, ao contrário dos *tweets*, são sempre visíveis apenas pelos envolvidos.

Outra funcionalidade disponível no Twitter é a criação de listas de usuários. Ao acessar uma lista, os *tweets* de todos os usuários incluídos nela são exibidos, em ordem

cronológica de postagem, de maneira similar à *timeline*, com os mais recentes sendo exibidos primeiro.

O cadastro no Twitter, assim como no YouTube, é totalmente gratuito, não existindo uma modalidade paga, como no Flickr. Um visitante não registrado tem acesso a quase todo o conteúdo do *microblog*, com exceção dos *tweets* de autores que optaram por proteger suas mensagens. É necessário ser um usuário registrado para poder postar mensagens, seguir outros usuários, criar listas, marcar *tweets* como favoritos e todas as ações que dependam dessas para serem executadas, como *retweetar* ou enviar *DM*.

O site do Twitter, segundo dados do site Alexa⁹, foi visitado por aproximadamente 9,122% dos usuários de Internet, na média dos meses de abril, maio e junho de 2012, sendo o 8º mais visitado do mundo. Assim sendo, o Twitter é a rede social online de *microblogging* mais popular no momento (RUFINO; TABOSA; NUNES, 2010).

Assim como para o Flickr e o YouTube, existe uma API¹⁰ disponível para desenvolvedores que queiram criar aplicativos que interajam com o Twitter. Essa API possibilita que os aplicativos executem quase todas as ações possíveis de serem executadas por um usuário no próprio site do Twitter, como exibir, compor ou excluir *tweets*, seguir ou deixar de seguir usuários, recuperar a *timeline* de um usuário ou os TTs de qualquer região disponível, entre outras.

2.1.4 Facebook

O Facebook¹¹, diferente das demais redes sociais online citadas anteriormente, é uma rede social onde o foco principal é o relacionamento entre os usuários, e não o compartilhamento de um determinado tipo de conteúdo.

Um usuário do Facebook pode criar seu perfil pessoal, que pode incluir informações básicas como nome e idade, além de outras mais específicas, como livros, filmes, jogos e citações preferidas, e uma foto para o perfil (WEAVER; MORRISON, 2008). Cada uma dessas informações tem uma seleção de privacidade individual, que pode ser: público, significando que qualquer visitante do perfil pode ver a informação; somente amigos, significando que apenas aqueles que forem confirmados como amigos do usuário podem ver a informação; somente eu, significando que apenas o próprio usuário pode ver a informação; ou personalizado, onde o usuário pode definir uma lista de outros usuários que terão acesso à informação. É importante ressaltar que, ao contrário do que acontece no YouTube e no Twitter, no Facebook é necessário que um usuário A aprove um pedido de amizade de outro usuário B para que ambos sejam considerados amigos.

O Facebook permite a postagem, ou compartilhamento, de mensagens de texto, imagens e vídeos. Cada usuário conta com uma página, chamada de *Mural*, na qual consta sua atividade recente. A página principal do serviço, chamada de *Feed de Notícias*, exibe a atividade recente dos amigos do usuário na rede social, desde que a configuração de privacidade da atividade não restrinja sua visibilidade, de forma similar à *timeline* no Twitter. A ordenação padrão do Feed de Notícias coloca no topo as atividades mais populares, porém é possível alterar essa configuração para que a

⁹ Alexa. <http://www.alexa.com/siteinfo/twitter.com/>. Acesso em 08 de Junho de 2012.

¹⁰ Twitter Developers. <https://dev.twitter.com/docs/api>. Acesso em 12 de Junho de 2012.

¹¹ Facebook. <http://www.facebook.com>. Acesso em 16 de Junho de 2012.

ordenação seja feita por data, com as atividades mais recentes aparecendo primeiro. Ao compartilhar um conteúdo, o usuário pode optar por publicar a informação no mural de um amigo, ao invés de publicar no seu próprio. O Facebook conta ainda com recurso de criação de álbuns, com a possibilidade de marcar usuários nas fotos.

Todo o conteúdo compartilhado pode receber comentários de outros usuários. Além disso, um usuário pode “Curtir”, ou seja, avaliar positivamente, uma postagem. Outra funcionalidade presente no sistema é o compartilhamento de um conteúdo que já foi compartilhado por outro usuário, de maneira análoga ao *retweet* no Twitter.

Há um sistema de troca de mensagens privadas, e um indicativo de quais amigos estão conectados, em tempo real.

Usuários do Facebook podem anunciar eventos, convidar amigos e solicitar confirmação através da rede social online. Um usuário que tenha sido convidado para um evento pode confirmar presença ou recusar o convite.

Uma característica interessante no Facebook é a possibilidade de aplicativos de terceiros serem executados dentro da plataforma, acessando informações cadastradas na base de dados da rede social, desde que o usuário permita. Outra funcionalidade importante é a possibilidade de criação de grupos, locais onde usuários se reúnem para troca de informações a respeito de assuntos específicos, definidos pelo criador do grupo. A Figura 2.4 mostra a interface do Facebook, onde podem ser vistas algumas das funcionalidades citadas.

The image shows a screenshot of the Facebook news feed interface. At the top, there is a search bar and the user's name, Matheus Berlesi. The left sidebar contains navigation options like 'Feed de notícias', 'Mensagens', 'Eventos', and 'Localizar amigos'. The main content area displays several posts:

- A status update from Matheus Berlesi with the text "No que você está pensando?".
- A post from CartolaFC with the text "Confira as dicas do Programa do Cartola FC e escale a sua equipe para a sexta rodada do Brasileiro. Compartilhe!".
- A sponsored post from SPOR TV titled "André, Dagoberto e Julio Cesar... as dicas do Programa do Cartola#6".
- A post from Formula 1 titled "Who said what after Europe GP Friday Practices".
- A post from Gabriela Rezende titled "Obsessão | Julia Petit – Petiscos".

The right sidebar shows options like 'Criar evento', '30 solicitação para CityVillesolicitaç...', and '10 outras solicitações de aplicativos pendentes'. At the bottom, there is a chat window titled 'Bate-papo - (5)'.

Figura 2.4: Feed de notícias do Facebook

O cadastro no Facebook é gratuito, não existindo nenhuma versão na qual seja necessário pagar para obter determinados benefícios. Usuários não registrados podem ver apenas a informação marcada como pública no perfil de um usuário ou grupo. Não é possível ver o conteúdo compartilhado pelos usuários, mesmo que as configurações de privacidade definam o conteúdo como público.

Com 44,56% dos usuários da Internet tendo acessado o site, na média dos meses de abril, maio e junho de 2012, o Facebook é o segundo site mais visitado do mundo¹². Segundo informações divulgadas pelo site Facebook Newsroom (2012), ao final do mês de março de 2012, o Facebook contava com 901 milhões de usuários ativos.

Uma série de APIs é disponibilizada para desenvolvedores¹³, tanto para o desenvolvimento de aplicativos a serem executados dentro da plataforma da rede social, como para a integração de aplicativos externos à rede social.

2.1.5 Vimeo

O Vimeo¹⁴ é uma rede social online cujo foco principal é o compartilhamento de vídeos, funcionando de forma muito similar ao YouTube. Usuários fazem o envio de vídeos para o sistema, podendo cada vídeo ser então marcado como público ou privado. O cadastro pode ser feito gratuitamente, sendo impostos alguns limites, como o número de vídeos em alta definição que podem ser enviados para o site por semana ou o espaço em disco disponível para *uploads* semanalmente. Uma das grandes diferenças desse sistema para o YouTube é a existência de uma modalidade de conta de usuário paga, chamada “*Plus*”, na qual são expandidos ou removidos os limites existentes na conta gratuita, chamada “*Basic*”. Existe ainda um terceiro tipo de conta, “*Pro*”, para usuários comerciais.

Os vídeos enviados para o Vimeo devem ser de autoria do usuário, sendo vetados vídeos comerciais, a não ser para usuários *Pro*. Assim, é disponibilizada uma opção para o usuário informar o seu papel na produção do vídeo enviado. Além disso, é possível creditar outras pessoas, usuários ou não do site, nos vídeos, adicionando sua função no desenvolvimento do material.

Usuários podem seguir outros usuários, de forma similar ao que ocorre no Twitter, recebendo informações sobre suas atualizações. As opções de privacidade dos vídeos também são um pouco diferentes das presentes no YouTube, existindo opções que permitem que qualquer um assista ao vídeo (opção padrão), apenas o autor, apenas as pessoas as quais o autor segue, apenas pessoas indicadas pelo autor, ou ainda uma proteção por senha. O autor ainda pode limitar os comentários do vídeo, permitindo comentários de qualquer usuário, apenas dos usuários seguidos por ele ou ainda desabilitar completamente os comentários.

Assim como no Flickr, o usuário pode atribuir uma dentre várias licenças de uso para cada vídeo enviado. O Vimeo permite ainda que o usuário disponibilize, além da incorporação em outras páginas, a opção de download do vídeo. A interface de visualização de um vídeo no Vimeo pode ser vista na Figura 2.5, a seguir.

¹² Alexa. <http://www.alexa.com/siteinfo/facebook.com/>. Acesso em 08 de Junho de 2012.

¹³ Facebook Developers. <http://developers.facebook.com>. Acesso em 08 de Junho de 2012.

¹⁴ Vimeo (<http://www.vimeo.com>). Acesso em 08 de Julho de 2012.

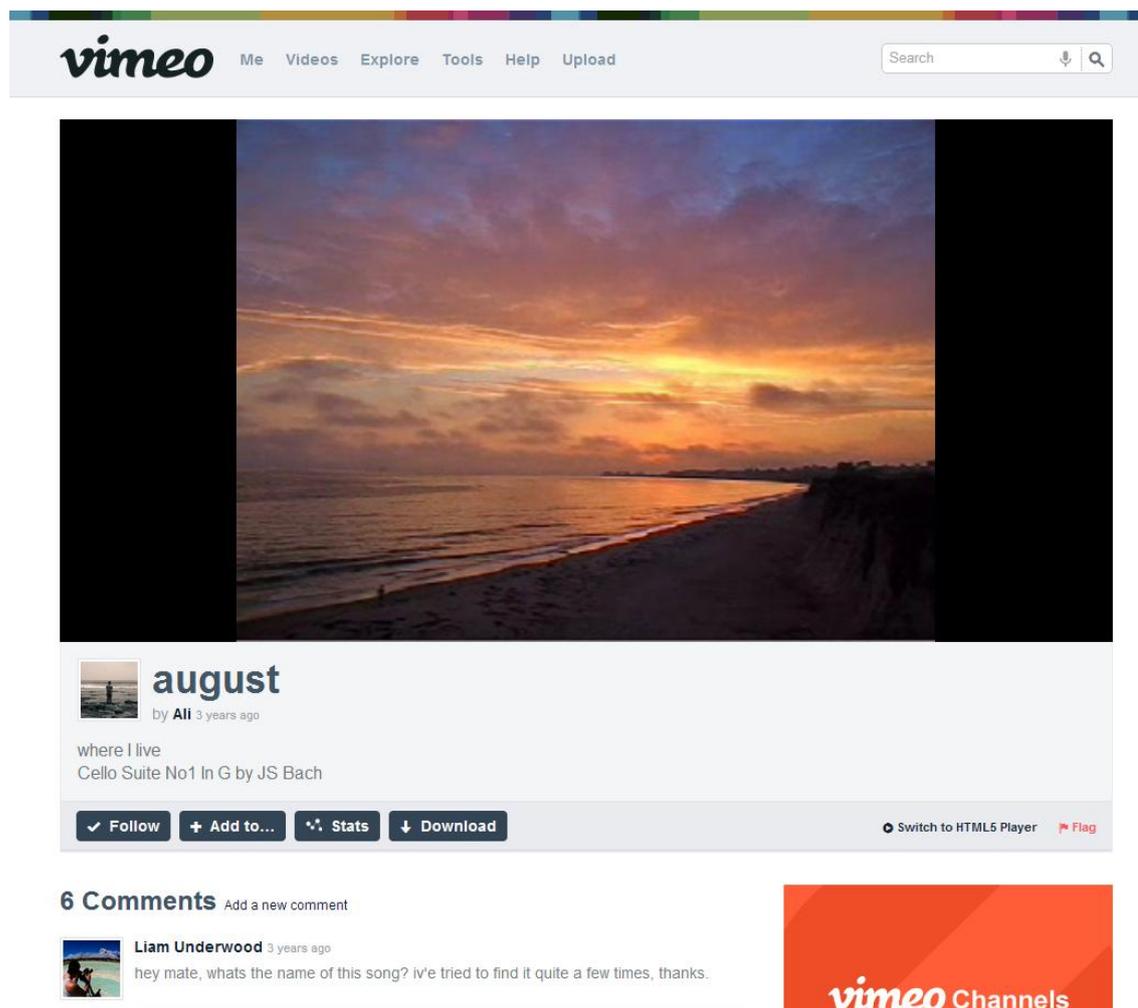


Figura 2.5: Página de visualização de vídeo no Vimeo

São disponibilizadas, na página de desenvolvedores do Vimeo¹⁵, três APIs para interação de programas de terceiros com o sistema. Uma das opções oferecidas, a “Advanced API”, permite a autenticação de usuários para efetuar ações possíveis apenas para usuários autenticados, sendo possível efetuar através dessa API as principais ações disponíveis no site.

2.2 Agregadores de Redes Sociais

Um agregador de redes sociais é um portal que permite acesso a várias redes sociais online simultaneamente (BENEVENUTO; ALMEIDA; SILVA, 2011). A ideia é tornar mais simples para os usuários de múltiplas redes sociais a tarefa de gerenciar seus perfis online, centralizando o processo, evitando assim a necessidade de acessar cada rede social separadamente. O usuário acessa o sistema agregador, tipicamente um site da internet ou aplicativo, que em tempo real se comunica com cada uma das redes sociais disponíveis. Essa comunicação entre o agregador e as redes sociais é feito, normalmente, através de APIs. Através da interface do sistema agregador, o usuário pode utilizar várias funcionalidades das redes sociais às quais está conectado, como

¹⁵ Vimeo Developers. <https://developer.vimeo.com/apis>. Acesso em 09 de Julho de 2012.

compartilhar fotos ou vídeos, enviar mensagens ou verificar a atividade recente de seus amigos, entre outras.

A seguir, são listados alguns agregadores existentes, com uma breve descrição de suas principais funcionalidades.

2.2.1 TweetDeck

O TweetDeck¹⁶ é um aplicativo que agrega as redes sociais Twitter e Facebook. Está disponível como um programa que pode ser instalado em computadores com sistema operacional Windows XP (ou posterior) ou Mac OS X 10.6, assim como em versão Web, acessível através de um browser, e ainda em versão *mobile*, podendo ser instalado em celulares com sistema operacional *Android* ou *iOS*.

No TweetDeck, é permitido a um usuário conectar seus perfis de ambas as redes sociais. Após conectar ao menos um perfil, uma coluna com as atualizações recentes dos amigos do usuário nas redes sociais conectadas surge na interface do aplicativo. As atualizações de todos os perfis do usuário, de ambas as redes, aparecem mescladas nessa coluna, que é chamada de “Início”, e é ordenada pela data de postagem, com as mais recentes no topo. É possível adicionar outras colunas com conteúdos diversos, como *tweets* de um único amigo, resultados da busca por um determinado termo no Twitter, ou as mensagens diretas recebidas. Para o Facebook, há apenas a opção de exibir o *feed* de notícias ou as notificações. O conteúdo das colunas é atualizado em tempo real, mantendo-se a ordenação inicial, com as atualizações mais recentes no topo. Todas essas informações ficam visíveis apenas para o usuário registrado no sistema agregador, sendo necessário efetuar o *login* no aplicativo ou site com o e-mail e senha cadastrados para ter acesso a elas. As configurações feitas em um ambiente são sincronizadas com todos os outros, ou seja, qualquer alteração feita no aplicativo *mobile*, como remoção de uma coluna ou adição de um novo perfil, estará disponível também nos ambientes web e desktop, e vice-versa.

O usuário tem a possibilidade de postar conteúdo usando qualquer um dos perfis associados ao agregador, ou mesmo com múltiplos perfis ao mesmo tempo, através da interface do programa. Para isso, no momento em que for escrever a mensagem, o usuário deve selecionar através de quais perfis deseja fazer o envio. Uma das contas associadas deve ser marcada como principal, estando sempre selecionada inicialmente no momento do envio de mensagens. Devido à limitação no número de caracteres para mensagens postadas no Twitter, ao selecionar uma conta dessa rede social para o envio de uma mensagem, o TweetDeck limita em 140 caracteres o tamanho da mensagem; caso apenas perfis do Facebook sejam selecionados, o limite passa a ser de 420 caracteres. O usuário pode ainda anexar uma imagem à mensagem, que será enviada para os respectivos serviços aos quais pertencem os perfis selecionados, e exibida em forma de *hiperlink*. Outro recurso interessante presente no TweetDeck é o agendamento do envio de mensagens. É possível escrever uma mensagem e selecionar uma data e hora para sua publicação, ao invés de fazer o envio imediato; na data marcada, a mensagem será enviada sem nenhuma interação adicional do usuário. O envio de mensagens privadas está disponível apenas para perfis da rede social Twitter. Existe ainda um alerta sonoro, que é acionado em determinadas ocasiões, tais como um dos

¹⁶ TweetDeck. <http://www.tweetdeck.com>. Acesso em 22 de Junho de 2012.

perfis conectados ser citado em uma mensagem ou receber uma mensagem em alguma das redes sociais.

A Figura 2.6, apresentada a seguir, mostra a interface do TweetDeck, em sua versão web, com 4 colunas configuradas, e apenas 3 visíveis.

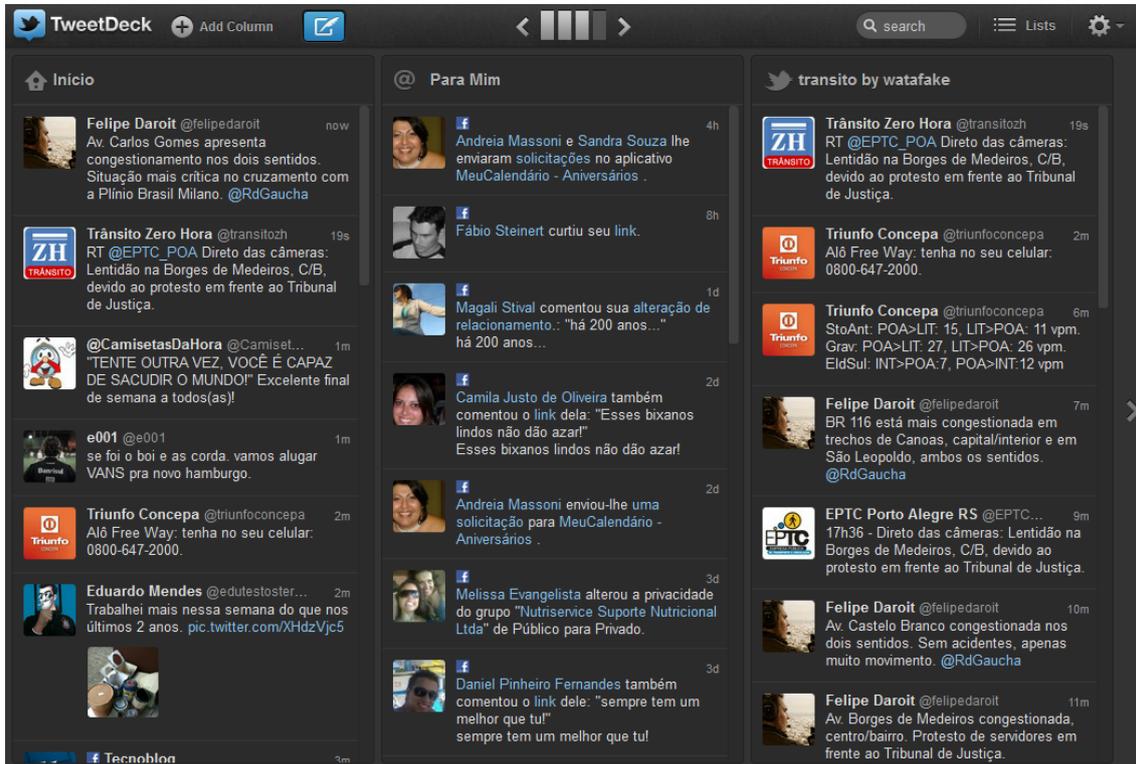


Figura 2.6: Interface web do TweetDeck

19,7% das mensagens postadas no Twitter em 2009 foram publicadas através do TweetDeck (Sysomos, 2012), sendo assim o aplicativo agregador mais popular do segmento. Em maio de 2011, o TweetDeck foi adquirido pela empresa responsável pelo Twitter (Twitter Blog, 2012).

2.2.2 Seesmic

O Seesmic¹⁷ é outro agregador de redes sociais, com aplicativos disponíveis tanto para desktops com sistema operacional Windows ou Mac OS, como para dispositivos móveis com sistema operacional Android, iOS ou Windows Phone. Conta ainda com uma versão web, que pode ser acessada através de um *browser*.

As redes sociais online suportadas pelo Seesmic são o Facebook, Twitter e LinkedIn. Assim como o TweetDeck, múltiplos perfis são permitidos para cada uma das redes, e o usuário pode optar por postar conteúdo em todas elas simultaneamente. Também de forma similar ao TweetDeck, o Seesmic divide as atividades das diversas redes sociais em colunas, porém não existe a opção de mesclar as atualizações de diferentes serviços, ou mesmo diferentes perfis de uma mesma rede social, em uma única coluna. A interface do agregador pode ser vista na Figura 2.7, apresentada a seguir.

¹⁷ Seesmic. <http://www.seesmic.com>. Acesso em 26 de Junho de 2012.

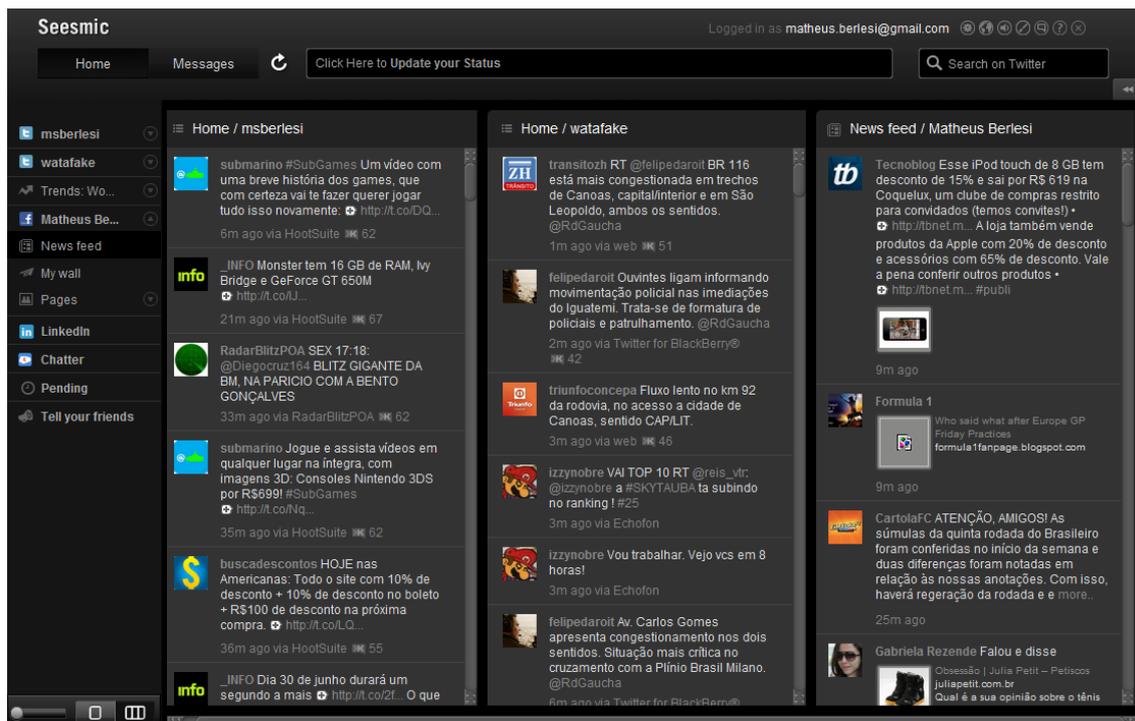


Figura 2.7: Interface do Seesmic, na versão web

Uma funcionalidade interessante do Seesmic, não existente no TweetDeck, é o modo de mensagens. Através de abas na parte superior da interface do agregador, o usuário pode alternar entre o modo “Home”, onde ficam todas as atividades das redes sociais do usuário, e o modo “Messages”, onde as mensagens diretas, recebidas e enviadas pelos perfis do Twitter conectados, ficam organizadas por amigo, e não mescladas, como acontece no site do Twitter. Nesse modo, é possível criar uma coluna para cada usuário com o qual já se trocou mensagens diretas no Twitter. Cada coluna exibe todas as mensagens trocadas com o usuário selecionado, em ordem cronológica, com as mais recentes no topo.

Os tipos de conteúdo que podem ser enviados através do Seesmic são limitados a texto, imagem e link. É possível ainda adicionar localização às mensagens. Assim como ocorre no TweetDeck, as mensagens são limitadas a 140 caracteres quando um perfil Twitter é selecionado no momento da composição da mensagem; caso contrário, o limite é de 63000 caracteres. Também existe a possibilidade de agendamento do envio das mensagens. O envio de mensagens privadas está disponível apenas para perfis do Twitter, e deve ser feito através da *dashboard* de mensagens.

2.3 Mashups

Mashups são, essencialmente, aplicações Web que agregam múltiplos serviços, a fim de atingir um novo propósito através do reaproveitamento de recursos e serviços Web já existentes (MAXIMILIEN et al., 2007).

Atualmente, cada vez mais aplicações Web disponibilizam APIs, permitindo, assim, que desenvolvedores integrem dados e funcionalidades, sem a necessidade de criá-las novamente. Este trabalho considera que o grande diferencial desse tipo de aplicação está na possibilidade de combinar dados resultantes de computação em vários pontos, para obter o resultado final do *mashup*.

Alguns *mashups* existentes, bem como suas principais características, são listados a seguir.

2.3.1 Profilactic

O Profilactic¹⁸ é um site onde o usuário pode cadastrar contas de diversos serviços, tipicamente redes sociais online, e recebe um endereço Web, onde todo o conteúdo publicamente disponível dos perfis cadastrados fica visível, em ordem cronológica decrescente; esta página é chamada de *lifestream*. É possível, no entanto, selecionar quais perfis devem apresentar seu conteúdo no *lifestream*, e quais não. Em um menu, situado à direita do *lifestream*, aparecem links para os diversos perfis associados, nos sites dos respectivos serviços. Existe ainda a possibilidade de se utilizar filtros para a *lifestream*, tanto por palavras-chave como por serviço; aplicando estes filtros, apenas o conteúdo que satisfaz as condições estipuladas é exibido. A Figura 2.8 mostra a interface da aplicação, onde algumas dessas características podem ser vistas.

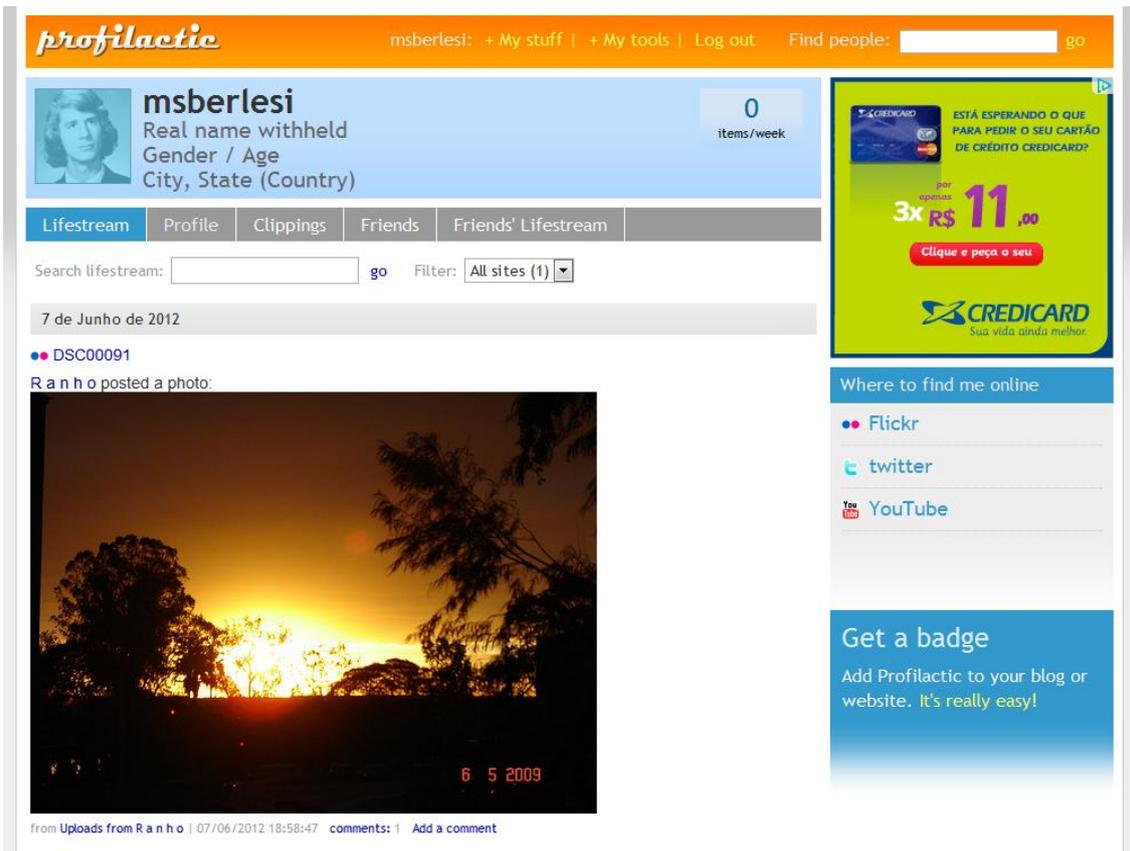


Figura 2.8: *Lifestream* de usuário do Profilactic

Além de ser um *mashup*, o site possui funcionalidades de rede social: é possível criar um perfil, adicionar outros usuários como amigos e enviar comentários para os conteúdos postados. Apesar de buscar o conteúdo de outros sistemas que permitem esse tipo de interação entre os usuários, os comentários e ligações de amizade não são passados adiante, ficando armazenados no site do próprio *mashup*. Existe, ainda, uma área do site onde são exibidas as *lifestreams* de todos os amigos do usuário, mescladas e ordenadas cronologicamente, ficando a atividade mais recente no topo.

¹⁸ Profilactic. <http://www.profilactic.com>. Acesso em 05 de Julho de 2012.

Visitantes não autenticados podem ver tanto a *lifestream* de um usuário quanto a de seus amigos. O cadastro só se faz necessário para a criação de seu próprio *lifestream*, e para a utilização das funcionalidades de rede social providas pela aplicação.

2.3.2 Flavors

Assim como o Profilactic, o Flavors¹⁹ é um *mashup* onde o usuário pode associar contas de vários serviços disponíveis na Web, não apresentando, entretanto, as mesmas características de rede social. Outra diferença entre ambos os sistemas é que, no Flavors, os conteúdos não ficam mesclados em uma única linha de tempo. Os perfis ficam separados, na forma de *hyperlinks* em um menu. Ao clicar em um desses *hyperlinks*, o conteúdo do perfil selecionado é carregado na página principal. A seguir, a Figura 2.9 apresenta a interface do sistema, exibindo uma página de usuário.

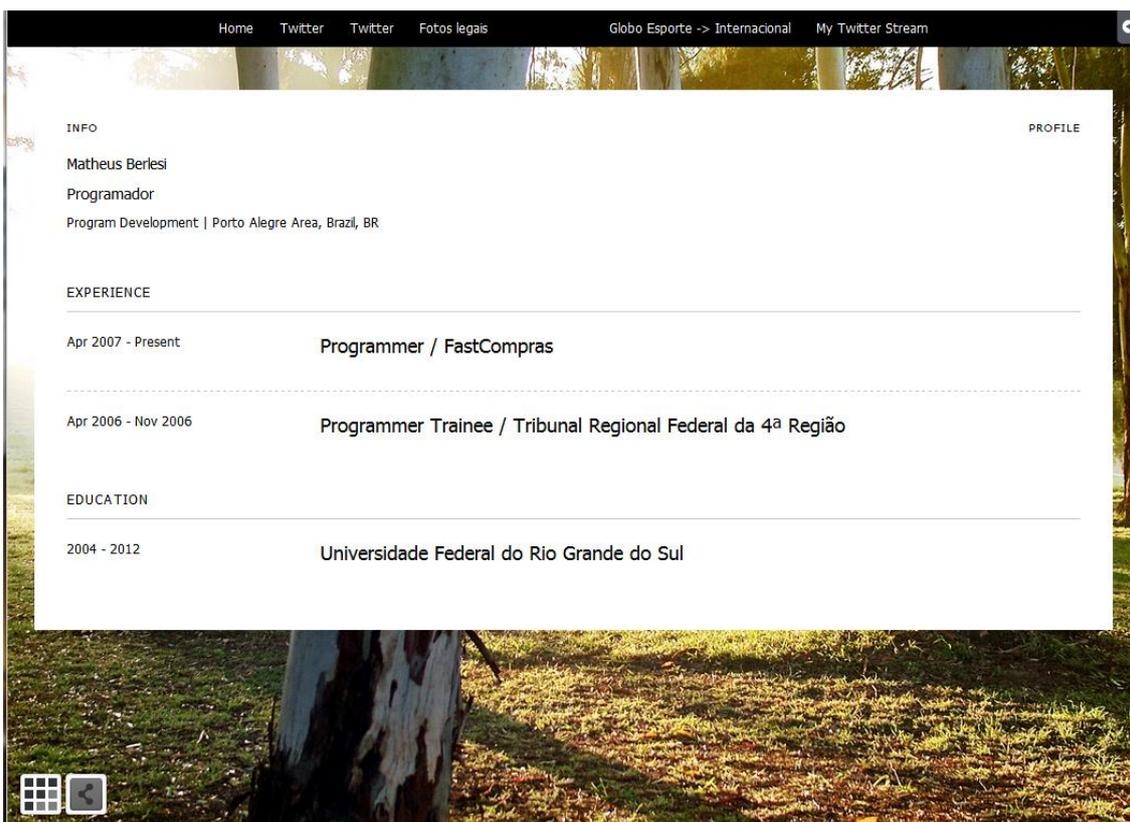


Figura 2.9: Página de usuário no Flavors

Um dos diferenciais do Flavors é a possibilidade de o usuário customizar a aparência de sua página. Alguns *layouts* e fundos de tela pré-definidos estão disponíveis, além de ser possível enviar um arquivo de imagem do computador do usuário e alterar formas, fontes e cores, de acordo com a preferência do usuário.

O cadastro só é necessário para usuários que pretendem criar sua própria página no sistema. Todo o conteúdo associado é exibido publicamente, não havendo distinção entre usuários autenticados ou visitantes. Algumas funcionalidades só estão disponíveis para usuários que pagam para utilizar o sistema, adquirindo uma conta *Premium*, como acesso a estatísticas de visitação ou a adição de um formulário de contato.

¹⁹ Flavors. <http://www.flavors.me>. Acesso em 05 de Julho de 2012.

3 OCTOOLS: UM MISTO DE AGREGADOR E MASHUP

O sistema proposto neste trabalho, denominado Octools, tem a finalidade de unir as facilidades propiciadas pelos sistemas agregadores de redes sociais com a funcionalidade de *mashups*. Para atingir este objetivo, é necessário que usuários cadastrados possam associar seus perfis de diferentes redes sociais, possibilitando a interação com todas elas através da própria interface do sistema proposto. Além disso, deve haver uma página, acessível a outros usuários ou visitantes, onde o conteúdo submetido às redes sociais através dos perfis associados pelo usuário seja exibido.

Nessa seção serão apresentados os critérios de inclusão utilizados neste trabalho para a escolha das redes sociais online a serem implementadas no sistema. Um glossário com alguns termos importantes utilizados durante o projeto também é apresentado. A seguir é feita uma análise de requisitos do sistema Octools, levando em consideração as peculiaridades das redes sociais online escolhidas e os sistemas já existentes. Por fim, é apresentado um diagrama de casos de uso preliminares, onde são evidenciadas as principais funcionalidades do sistema.

3.1.1 Critérios de inclusão

Para a realização deste trabalho, optou-se pela escolha de três redes sociais online, a serem integradas e oferecidas como opção aos usuários da ferramenta proposta. Uma exigência técnica para a escolha dessas redes sociais é a existência de uma API que permita não só a recuperação de informações postadas pelos usuários, como também o envio de conteúdo, sem a necessidade de acesso ao site do serviço. Além do critério técnico, optou-se por escolher redes sociais online cujo foco seja o compartilhamento de algum tipo específico de conteúdo, deixando de fora aqueles serviços onde o relacionamento interpessoal é o foco principal. Ainda, dentre essas redes sociais, teve-se o cuidado de não incluir duas com foco no compartilhamento do mesmo tipo de conteúdo. Para decidir quais redes sociais incluir, entre aquelas cujo foco é o compartilhamento de um mesmo tipo de conteúdo, usou-se o critério de popularidade. Como não há informação disponível sobre o número de usuários para todas as redes sociais online, a popularidade foi medida com base nos dados do site Alexa²⁰, que estima o número de visitantes e o alcance de cada site entre os usuários da Internet, e disponibiliza um ranking dos sites mais populares, baseado nesta estimativa.

Assim, a Tabela 3.1 aponta algumas redes sociais e os critérios citados nessa seção para cada uma delas, bem como a informação de quais foram utilizadas neste trabalho ou não.

²⁰ Alexa. <http://www.alexa.com/>. Acesso em 08 de Junho de 2012.

Tabela 3.1: Critérios de inclusão das redes sociais

Serviço	Possui API	Foco principal	Alcance	Posição	Utilizado no trabalho
Facebook	Sim	Relacionamento interpessoal	44,556%	2°	Não
YouTube	Sim	Compartilhamento de vídeos	32,641%	3°	Sim
Twitter	Sim	Compartilhamento de mensagens curtas	9,122%	8°	Sim
Flickr	Sim	Compartilhamento de imagens	1,750%	48°	Sim
Vimeo	Sim	Compartilhamento de vídeos	0,958%	116°	Não

Pela tabela 2.1, vê-se que o Facebook, apesar de possuir uma API que permite tanto o envio quanto a recuperação de informações compartilhadas pelos seus usuários, e ser o mais popular serviço de rede social online do mundo, acabou não fazendo parte deste trabalho, pois seu foco principal é no relacionamento interpessoal. Por outro lado, YouTube, Twitter e Flickr são os serviços mais populares em seus respectivos nichos, dispõem de uma API com as capacidades necessárias para a realização deste trabalho e, portanto, foram incluídos, em detrimento de outros serviços do mesmo tipo com menor popularidade, como o Vimeo, uma rede social cujo foco também é o compartilhamento de vídeos.

3.2 Glossário de termos

Para reduzir problemas de comunicação e ambiguidades no projeto, foi definida uma lista de termos e suas definições, chamada aqui de glossário de termos. Nesta lista constam alguns termos importantes no decorrer do projeto, específicos do domínio da aplicação ou que poderiam ser utilizados de modo diferente.

Tabela 3.2: Glossário de termos usados no decorrer do projeto

Termo	Definição e Informações
Usuário	Visitante, registrado e autenticado no website, que possui um nome de usuário e senha. Pode associar à sua conta perfis de redes sociais online, para utilizá-las a partir da interface do sistema.
Visitante	Qualquer pessoa não autenticada que acesse o website do sistema proposto.
Serviço	Rede social online agregada pelo website.

Página de usuário	Uma página do website cujo conteúdo é controlado por um usuário. É nessa página que o conteúdo dos serviços integrados por um usuário é exibida. É também o único ponto por onde é possível postar conteúdo para os Serviços.
Perfil associado	Um perfil ou conta de usuário em um dos Serviços, o qual tenha sido associado ao sistema agregador, de forma a poder ser utilizado para o gerenciamento do conteúdo na rede social online.

3.3 Requisitos do sistema

Os requisitos do sistema são uma parte importante da documentação, uma vez que definem o que o sistema deve fazer (HAZAN; LEITE, 2003). São classificados em dois grupos, requisitos funcionais e requisitos não funcionais.

Os requisitos funcionais referem-se aos aspectos da funcionalidade do software, isto é, descrevem funções que deverão ser incorporadas, como o sistema deve reagir a determinadas entradas, e como o sistema deve se comportar em situações específicas (SOMMERVILLE, 2011).

Já os requisitos não funcionais são restrições relacionadas ao uso da aplicação, em termos de desempenho, usabilidade, confiabilidade, tecnologias envolvidas, entre outros aspectos.

3.3.1 Requisitos funcionais

Durante a definição dos requisitos funcionais do sistema, um conjunto de funções frequentemente utilizadas é o CRUD, acrônimo da expressão em língua inglesa “*Create, Retrieve, Update and Delete*” (KILOV, 1998). Esta expressão refere-se às operações básicas, tanto no escopo de bancos de dados como no de interfaces para usuários, de criação, recuperação ou consulta, atualização e exclusão de dados. A seguir, estão listados os requisitos funcionais do sistema:

- O sistema deve prover no website, aos visitantes não autenticados, uma forma para que efetuem a autenticação, informando nome de usuário e senha.
- O sistema deve prover no website, aos visitantes não autenticados, uma forma para que efetuem um novo cadastro, no qual devem obrigatoriamente fornecer um endereço de e-mail, escolher um nome de usuário único e uma senha, os quais serão utilizados posteriormente durante o processo de autenticação.
- Os usuários autenticados devem poder associar e desassociar perfis das redes sociais online Twitter, Flickr e YouTube à sua conta no sistema.
- Todos os usuários com perfis da rede social online Flickr associados devem ter a possibilidade de realizar o CRUD de suas fotos do Flickr, uma vez que essas são as principais funcionalidades para esse serviço.
- Todos os usuários com perfis da rede social online YouTube associados devem ter a possibilidade de realizar o CRUD de seus vídeos do YouTube, uma vez que essas são as principais funcionalidades para esse serviço.
- Todos os usuários com perfis da rede social online Twitter associados devem ter a possibilidade de realizar o CRD, ou seja, criação, recuperação e deleção, de

suas mensagens de texto do Twitter, uma vez que essas são as principais funcionalidades para este serviço.

- Todos os usuários com perfis da rede social online Twitter associados devem ter a possibilidade de visualizar sua timeline do Twitter, já que essa é uma importante atividade para essa rede social online.
- Todos os usuários com perfis da rede social online Twitter associados devem ter a possibilidade de responder mensagens de sua timeline do Twitter, já que essa é também uma importante função do serviço.
- Todos os usuários com perfis da rede social online Twitter associados devem ter a possibilidade de retweetar mensagens de sua timeline do Twitter, já que essa é também uma importante função do serviço.
- O sistema deve prover uma forma de o usuário desativar sua conta, cancelando suas credenciais de acesso.
- Todos os visitantes devem ter acesso às páginas dos usuários, e poder visualizar o conteúdo postado pelos perfis associados à conta do usuário em questão.

3.3.2 Requisitos não funcionais

Os requisitos não funcionais, apesar de não expressarem nenhuma função a ser realizada pelo sistema, são muito importantes para o desenvolvimento do projeto, pois definem restrições cruciais para o funcionamento e a eficiência do sistema final. Assim, os requisitos não funcionais para o sistema Octools estão listados a seguir:

- O sistema deve tratar os erros ocorridos, evitando que mensagens técnicas sejam exibidas para o usuário.
- O sistema deve coibir a ação de visitantes mal-intencionados, tratando os dados entrados em formulários a fim de manter a segurança do sistema.
- O usuário que se autentica na tela inicial do sistema deve ser direcionado para sua página de usuário após a autenticação, visto que essa é a principal funcionalidade prevista para o sistema.
- O usuário que se autentica em uma tela diferente da inicial deve ser mantido nessa mesma tela após a autenticação.
- Apenas o usuário autenticado deve poder fazer alterações em sua conta de usuário, devendo ser vetada a opção de adição ou remoção de perfis, bem como qualquer interação diferente da visualização do conteúdo, a qualquer outro visitante ou usuário do sistema.
- O sistema deve solicitar confirmação do usuário para ações de exclusão de conteúdo dos perfis associados, uma vez que essa ação é irreversível.
- Devem estar disponíveis, em todas as telas do sistema, os campos para autenticação dos usuários que ainda não estejam autenticados, facilitando o acesso dos usuários.
- No caso de usuários autenticados, deve estar disponível em todas as telas do sistema, com exceção de eventuais janelas auxiliares, uma opção para o usuário sair do sistema.

- Sempre que houver ações disponíveis para o usuário, o sistema deve exibir botões, links ou outros elementos iniciadores de ações que permitam ao usuário efetuar tais ações, permitindo assim ao usuário identificá-las.
- O sistema deve evitar a exibição de elementos iniciadores de ações, como botões ou links, para ações às quais o usuário não tenha permissão de executar, a fim de manter a clareza do sistema.

3.4 Casos de uso preliminares

O objetivo dos casos de uso preliminares é sintetizar as principais funcionalidades do sistema, de forma que sejam facilmente compreendidas, especialmente nas fases iniciais do projeto. Assim, de forma simplificada, os principais serviços oferecidos pelo sistema são representados em um diagrama de casos de uso (Figura 3.1), onde ficam evidenciados também os papéis desempenhados por cada ator²¹ na utilização do sistema.

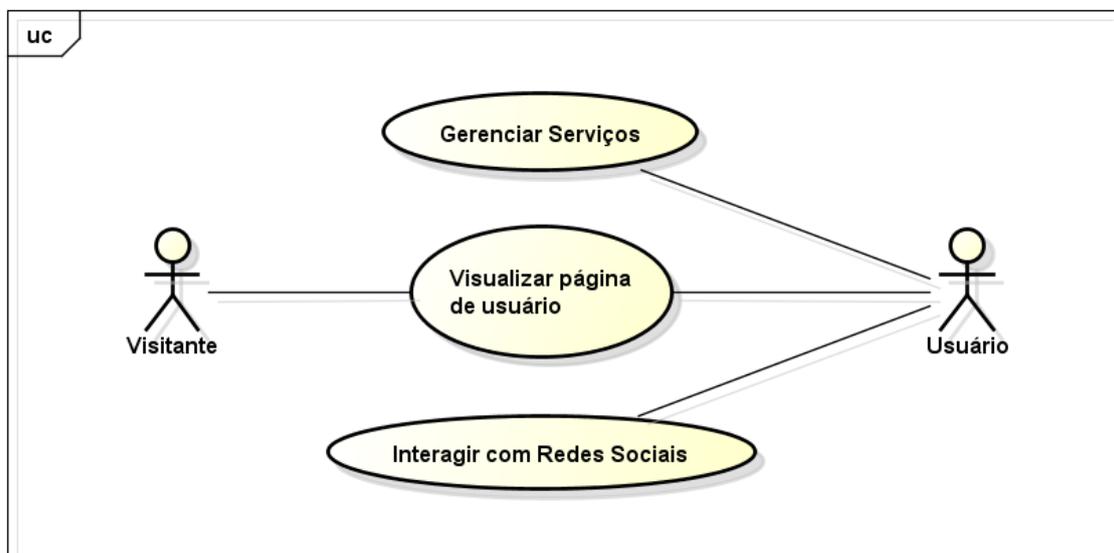


Figura 3.1: Diagrama de casos de uso preliminares

O diagrama da Figura 3.1 apresenta dois atores, o visitante e o usuário. O primeiro deles tem acesso apenas à visualização das páginas de usuários. O outro, por sua vez, tem a possibilidade de gerenciar serviços e interagir com as redes sociais online disponíveis.

²¹ Ator: qualquer coisa com um comportamento, inclusive o próprio sistema quando invoca os serviços de outros sistemas (LARMAN,2007).

4 SOLUÇÃO PROPOSTA

Nesta seção é apresentada a solução proposta por este trabalho. São discutidas as tecnologias utilizadas na implementação da ferramenta e alguns padrões de projeto que fizeram parte do desenvolvimento. São mostrados também alguns artefatos usados na fase de projeto da ferramenta proposta, como o diagrama de casos de uso e a modelagem do banco de dados.

4.1 Tecnologias utilizadas

Para a especificação do sistema, adotou-se a Linguagem de Modelagem Unificada (UML), largamente utilizada para modelagem de sistemas orientados a objetos (OMG, 2005). Por se tratar de um sistema projetado para rodar em ambiente Web, a linguagem de marcação HTML (NEGRINO; SMITH, 2009) acaba se tornando uma escolha natural para o desenvolvimento da interface do sistema, com o auxílio de folhas de estilo CSS e Javascript. Foi utilizada no desenvolvimento das funcionalidades do site a linguagem PHP, em conjunto com banco de dados MySQL. Algumas páginas utilizam a tecnologia AJAX para comunicação assíncrona.

4.1.1 A linguagem de modelagem UML

A UML, acrônimo para o termo em língua inglesa “*Unified Modeling Language*”, é uma linguagem visual para modelagem de sistemas. Permite representar sistemas de software através de diagramas padronizados, sendo considerada uma das linguagens mais expressivas para modelagem de sistemas orientados a objetos (SILVA, 2001), o que justifica sua escolha para este projeto.

4.1.2 A linguagem PHP

O PHP, acrônimo recursivo para o termo “*PHP Hypertext Preprocessor*”, é uma poderosa linguagem de programação *open source*, largamente utilizada em ambiente Web. Ao contrário de linguagens que devem ser compiladas para executar, o PHP é uma linguagem interpretada, em tempo real, por um servidor (*server-side*), instalado no dispositivo onde a aplicação está rodando. Uma de suas principais características é capacidade de se misturar ao HTML, o que torna mais fácil a geração de páginas Web dinâmicas (SOARES, 2004). Além disso, oferece velocidade, robustez e portabilidade. Estas características foram determinantes na escolha dessa linguagem para a implementação do sistema. Outro aspecto positivo dessa linguagem é sua extensa e ativa comunidade, sendo possível encontrar em repositórios diversas bibliotecas para as mais variadas finalidades, possibilitando um ganho de produtividade através do reuso de tais recursos.

4.1.3 Banco de dados MySQL

O sistema de gerenciamento de banco de dados (SGBD) MySQL oferece alta confiabilidade, desempenho e facilidade de uso. A linguagem SQL (Linguagem de Consulta Estruturada, do inglês *Structured Query Language*) é utilizada para a comunicação e manipulação dos dados armazenados. É, atualmente, o SGBD de código-fonte aberto mais popular do mundo²², sendo, portanto, justificada sua escolha para este trabalho.

4.1.4 A linguagem de marcação HTML

HTML (acrônimo para a expressão inglesa *HyperText Markup Language*, que significa Linguagem de Marcação de Hipertexto) é uma linguagem de marcação amplamente utilizada para produzir páginas na Web. Arquivos codificados em HTML são interpretados por navegadores para fornecer, assim, a exibição desejada do conteúdo do arquivo (CASTRO, 2007).

A versão da linguagem HTML utilizada neste trabalho é a quarta, pois mesmo já existindo a especificação da quinta versão, nem mesmo as versões mais recentes dos navegadores interpretam em sua totalidade a versão mais recente do HTML.

4.1.5 Folhas de estilo CSS

Para a formatação da aparência do sistema, foi utilizada a linguagem de estilo CSS, acrônimo para o termo em inglês “*Cascading Style Sheet*”, ou “Folha de Estilos em Cascata”, em português. Trata-se de uma linguagem utilizada para definir a apresentação de documentos escritos em uma linguagem de marcação, como HTML ou XML. O principal benefício atingido com o uso dessa linguagem é a separação do código de formatação do conteúdo do documento.

Um arquivo contendo apenas os estilos é criado, e incluído por cada uma das páginas do sistema através de uma *tag* HTML. Assim, para alterar a aparência do sistema, não é necessário efetuar alterações em cada uma das páginas, mas sim em um único arquivo (CASTRO, 2007).

4.1.6 A linguagem JavaScript

Assim como PHP, JavaScript é uma linguagem de programação interpretada, e não compilada. No entanto, nesse caso o responsável por interpretar a linguagem é o próprio navegador (*client-side*), e não o servidor onde a aplicação está rodando (GOODMAN, 2001). Com essa linguagem, é possível adicionar interatividade às páginas Web, sendo por isso bastante útil para a validação de formulários (NEGRINO; SMITH, 2009). A escolha do JavaScript permite ainda a utilização de tecnologias que ajudam a aprimorar a interatividade com o usuário, como o AJAX, descrito a seguir. Em seguida, é apresentada a biblioteca jQuery, utilizada para simplificar a programação nessa linguagem.

4.1.6.1 A tecnologia de desenvolvimento AJAX

O JavaScript permite a utilização de AJAX (*Asynchronous JavaScript and XML*, ou JavaScript e XML Assíncronos, em português) no desenvolvimento de páginas Web. Por AJAX, entende-se o uso metodológico de algumas tecnologias providas por

²² MySQL. <http://www.mysql.com/why-mysql>. Acesso em 28 de junho de 2012.

navegadores, como JavaScript e XML, a fim de tornar as páginas Web mais interativas com o usuário, utilizando-se para isso de requisições assíncronas de informações.

AJAX pode ser entendido, ainda, como um conjunto de tecnologias trabalhando juntas, cada qual fazendo sua parte, e, assim, oferecendo novas funcionalidades. Nesse conjunto de tecnologias estão, por exemplo, os padrões HTML e CSS, responsáveis pelo modelo de apresentação. O intercâmbio e manipulação de dados pode ser feito utilizando-se XML, e a recuperação assíncrona de dados pode ser feita usando o objeto JavaScript XMLHttpRequest (STAMEY; RICHARDSON, 2006). A união de todas essas tecnologias anteriormente mencionadas é feita através de Javascript (NEGRINO; SMITH, 2009).

No sistema proposto, o AJAX é bastante utilizado para atualizar partes de determinadas páginas sem a necessidade do recarregamento da página toda. Outro ponto onde o AJAX faz grande diferença é na validação do cadastro de um novo usuário. Por ser acionado através do JavaScript, é possível detectar quando o visitante terminou de preencher o campo *username* e, antes mesmo de o formulário ser submetido, acionar uma chamada assíncrona a uma outra página, que verificará a disponibilidade dos dados preenchidos no banco de dados e responderá à requisição. Ao receber a informação, a página de cadastro exibe ao visitante, sem que haja necessidade de submeter o formulário e aguardar o recarregamento da página. Nesse caso, apesar de uma simples consulta ao banco de dados resolver o problema, seria necessário recarregar todo o conteúdo da página caso não se utilizasse de AJAX para fazer a solicitação assíncrona, tornando o sistema menos eficiente.

4.1.6.2 A biblioteca jQuery

Para facilitar tanto o tratamento de eventos quanto as interações AJAX, utilizou-se jQuery²³, uma biblioteca JavaScript, *open source*, que fornece uma forma simplificada para a manipulação de HTML e JavaScript. Uma das vantagens do uso dessa biblioteca é que suas funções são compatíveis com todos os navegadores, evitando que o programador precise se preocupar com diferenças entre a sintaxe de um comando JavaScript em cada um dos navegadores. Além disso, fornece uma maneira simples de adicionar *event listeners*, rotinas que devem ser executadas quando determinados eventos ocorrem, sem modificações no código HTML, deixando assim o código mais limpo e com carregamento mais rápido.

4.1.7 APIs

A comunicação entre o Octools e as redes sociais online implementadas só foi possível pela existência de APIs, disponibilizadas pelos responsáveis por tais redes sociais justamente para que aplicações externas possam trocar informações com seus sistemas de forma simples. Todas as redes sociais disponíveis neste trabalho disponibilizam, ao menos, uma API em formato REST (acrônimo para “Representational State Transfer”, ou “Transferência de Estado Representacional”, em português). O termo REST, para este trabalho, refere-se a qualquer interface Web simples, que utilize apenas o protocolo HTTP como forma de transmissão das informações, sejam as solicitações feitas a um servidor, ou as respostas a essas solicitações, em oposição à interfaces que utilizam protocolos baseados em padrões de troca de mensagens, como é caso do protocolo de *web services* SOAP (acrônimo de

²³ jQuery. <http://www.jquery.com>. Acesso em 05 de Julho de 2012.

“Protocolo Simples de Acesso a Objetos”, em inglês “*Simple Object Access Protocol*”). Assim, as APIs dos serviços integrados pelo agregador proposto recebem solicitações através de requisições HTTP, do tipo GET ou POST, e respondem também através de requisições desses tipos.

Como descrito no capítulo 2, algumas ações em tais redes sociais estão disponíveis apenas para usuários autenticados. Assim, seria necessário enviar, através de uma requisição HTTP GET, por exemplo, os dados de autenticação do usuário, o que é extremamente inseguro. Visando evitar o roubo dessas informações sensíveis, além de abusos por parte de aplicações de terceiros, foi desenvolvido um protocolo aberto para possibilitar a autenticação de usuários sem o compartilhamento das credenciais, chamado OAuth²⁴, também utilizado por todas as redes sociais utilizadas no projeto. O OAuth permite ao usuário ter um controle maior sobre as ações que um aplicativo pode executar em seu nome, ou mesmo por quanto tempo é válida essa autorização. O aplicativo deve solicitar um *token* à rede social, que deverá ser autorizado pelo usuário no site da própria rede social. A partir da autorização por parte do usuário no site da rede social, o aplicativo pode agir em nome do usuário, enviando junto de cada solicitação à API o *token* fornecido. A qualquer momento, o usuário pode voltar ao site da rede social e cancelar a permissão ao aplicativo, invalidando o *token*. O aplicativo não terá posse das credenciais do usuário, não podendo, assim, agir em seu nome.

4.2 Casos de uso

Os casos de uso descrevem as principais interações que podem ser feitas com o sistema, seja por uma pessoa ou um sistema externo (SOMMERVILLE, 2011). O objetivo é ter cada funcionalidade do sistema bem definida e especificada. Para tal, um diagrama de casos de uso essenciais será apresentado, seguido por uma descrição mais detalhada dos principais casos de uso, os casos de uso expandidos.

4.2.1 Diagrama de casos de uso essenciais

Os casos de uso essenciais do sistema proposto neste trabalho, baseados no refinamento dos casos de uso preliminares, apresentados anteriormente na Figura 3.1, estão expressos na Figura 4.1, a seguir, através de um diagrama de casos de uso.

²⁴ OAuth. <http://www.oauth.net>. Acesso em 03 de Julho de 2012.

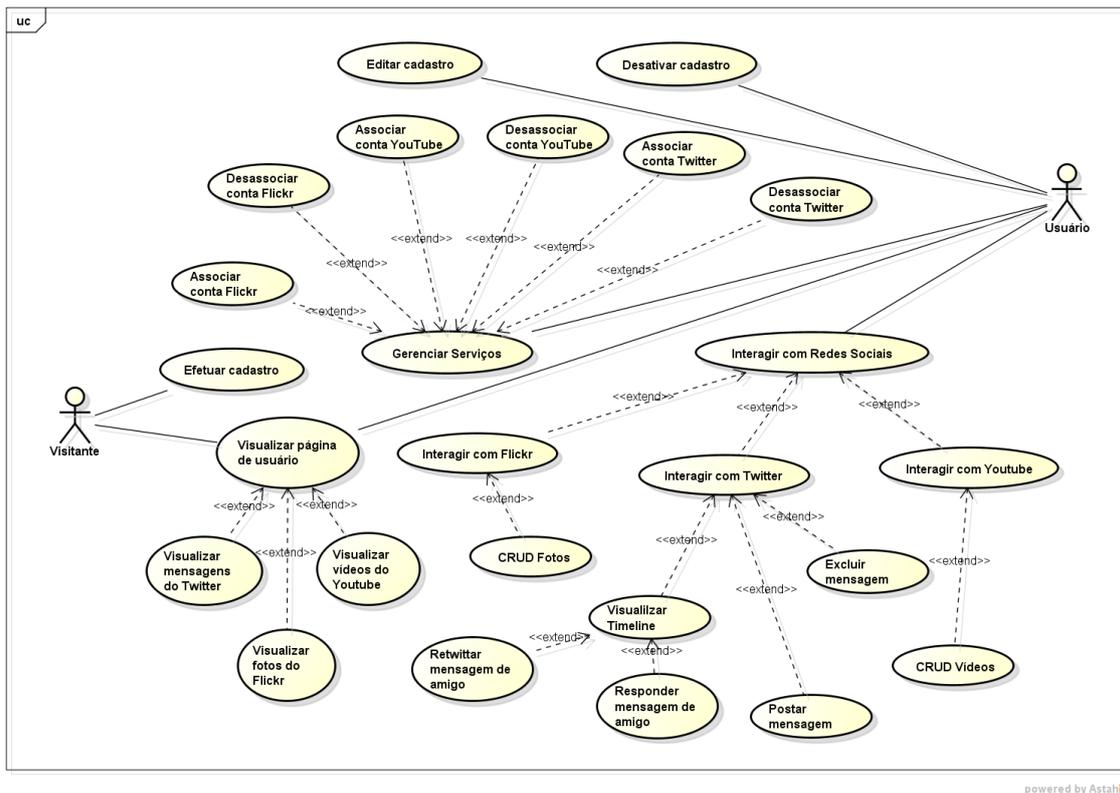


Figura 4.1: Diagrama de casos de uso essenciais

O diagrama de casos de uso essenciais da Figura 4.1 mostra as principais funcionalidades do sistema, e quem pode usufruir delas. Percebe-se que a ação de um visitante é bastante limitada, sendo permitido apenas o cadastro e a visualização das páginas dos usuários.

Um usuário, por sua vez, tem a possibilidade de gerenciar os seus perfis dos serviços no sistema, sendo permitido tanto associar como desassociar perfis de qualquer uma das redes sociais online agregadas. Além disso, com perfis associados, o usuário pode interagir com as redes sociais online, efetuando o envio, edição ou remoção de conteúdo.

A seguir, são apresentados os principais casos de uso do sistema de forma expandida, a fim de esclarecer o diagrama de casos de uso apresentado anteriormente na Figura 4.1.

4.2.2 Casos de uso essenciais expandidos

Os casos de uso expandidos são narrativas em texto, que tem o objetivo de descobrir e registrar os requisitos do sistema. O estilo de escrita é dito essencial porque ele evita detalhes da interface com o usuário e focaliza as reais intenções do usuário (LARMAN, 2007). Assim, é possível que as principais funcionalidades do sistema sejam especificadas de forma mais abrangente, independente da tecnologia utilizada e dos detalhes dos mecanismos relacionados com a interface.

Ainda de acordo com Larman (2007), não há um padrão definido para o formato de escrita dos casos de uso expandidos. Optou-se, portanto, pela escolha de um formato com duas colunas, por propiciar uma clara separação visual, enfatizando a ocorrência de uma interação entre atores e sistema.

Tabela 4.1: Caso de uso efetuar cadastro no sistema

Ação – Efetuar cadastro no sistema	
Ator:	Visitante
Pré-Condição:	Visitante não está autenticado no sistema.
Pós-Condição:	Visitante tem uma conta de usuário e está autenticado no sistema, ou seja, passa a ser um Usuário.
Entrada:	Dados cadastrais.
Saída:	-
Fluxo de Eventos	
Ator	Sistema
1. Informa os dados exigidos para o cadastro.	
	2. Valida os dados de entrada. Caso estejam... A) <i>Corretos</i> Registra os dados, criando uma conta de usuário. B) <i>Incorretos</i> Volta para (1), informando o erro.

A ação de efetuar o cadastro, detalhada na tabela 4.1, pode ser considerada a porta de entrada para o sistema, uma vez que a maioria das funcionalidades está disponível apenas para usuários. Apenas visitantes podem efetuar o cadastro, visto que usuários já possuem uma conta no sistema.

Analogamente, a ação de desativar o cadastro está disponível apenas para usuários, e deve solicitar uma confirmação antes de efetivamente cancelar o cadastro do usuário. Esse caso de uso está detalhado na tabela 4.2. Após cancelar o cadastro, o usuário passa a ser novamente considerado um visitante, até que se autentique com outra conta de usuário.

Tabela 4.2: Caso de uso cancelar cadastro no sistema

Ação – Cancelar cadastro no sistema	
Ator:	Usuário
Pré-Condição:	Usuário está autenticado no sistema.
Pós-Condição:	Conta de usuário deixa de existir no sistema.
Entrada:	-
Saída:	-
Fluxo de Eventos	

Ator	Sistema
1. Solicita o cancelamento de sua conta de usuário.	
	2. Solicita ao ator que confirme a exclusão.
3. Ator confirma ou não a exclusão.	
	4. Avalia resposta do ator. Caso seja... A) <i>Positiva</i> Exclui conta de usuário. B) <i>Negativa</i> Volta para (1), informando o erro.

Os casos de uso expandidos apresentados até aqui tratam do cadastro do usuário no próprio sistema. Uma vez cadastrado no sistema, o usuário tem a possibilidade de associar perfis de diferentes redes sociais online. O processo de associação de perfis, para qualquer um dos serviços disponíveis, é bastante semelhante e, portanto, optou-se por explicar um caso de uso de associação de perfil de forma genérica, o qual aplica-se para cada um dos serviços, como pode ser visto na tabela 4.3.

Tabela 4.3: Caso de uso associar perfil de serviço

Ação – Associar perfil de serviço	
Ator:	Usuário
Pré-Condição:	Ator possui ao menos um perfil, em ao menos uma das redes sociais online atendidas pelo sistema.
Pós-Condição:	Perfil é associado à conta de usuário do ator.
Entrada:	Serviço ao qual o perfil a ser associado pertence.
Saída:	-
Fluxo de Eventos	
Ator	Sistema
1. Informa que deseja associar um perfil de algum dos serviços disponibilizados pelo sistema, especificando qual.	
	2. Direciona o ator para uma página dentro da rede social especificada.
3. Autentica-se na rede social em questão, fornecendo os dados do perfil que deseja associar, e autoriza ou não o seu uso pelo sistema.	
	4. Recebe retorno sobre a permissão para usar o perfil. Caso... A) <i>Concedida</i>

	<p>Associa perfil à conta de usuário do ator e retorna para (1), informando que obteve sucesso.</p> <p><i>B) Negada</i> Retorna para (1), informando que a ação não foi concluída.</p>
--	--

Podemos ver no caso de uso apresentado na tabela 4.3 que é necessário que o usuário possua previamente ao menos um perfil que possa ser associado ao sistema. O caso de uso que trata da exclusão de perfis associados, de maneira similar à associação de perfis, está explicado de forma genérica na tabela 4.4, a seguir.

Tabela 4.4: Caso de uso desassociar perfil de serviço

Ação – Desassociar perfil de serviço	
Ator:	Usuário
Pré-Condição:	Ator possui ao menos um perfil associado ao sistema.
Pós-Condição:	Perfil é desassociado da conta de usuário do ator.
Entrada:	Perfil a ser desassociado.
Saída:	-
Fluxo de Eventos	
Ator	Sistema
1. Informa que deseja desassociar um perfil do sistema, especificando qual.	
	2. Solicita confirmação da ação.
3. Responde à solicitação de confirmação, positiva ou negativamente.	
	<p>4. Avalia a resposta do ator. Caso seja...</p> <p><i>A) Positiva</i> Desassocia perfil da conta de usuário do ator, informando que obteve sucesso.</p> <p><i>B) Negativa</i> Retorna para (1), informando que a ação não foi concluída.</p>

No caso de uso evidenciado na tabela 4.4, o usuário deve ter, obviamente, ao menos um perfil associado à sua conta de usuário no sistema. Assim como no caso de uso “Cancelar cadastro no sistema”, é solicitada uma confirmação, uma vez que a ação é irreversível, ainda que o usuário possa associar novamente o perfil recém-desassociado, se preferir.

As ações do tipo CRUD, disponíveis para publicações dos diferentes tipos de conteúdo nos serviços, também são apresentadas de forma genérica, sendo explicadas

conceitualmente cada uma das operações, em casos de uso distintos. É importante destacar, no entanto, que para o Twitter, particularmente, não é permitida a edição das mensagens já publicadas, conforme foi visto no diagrama de casos de uso essenciais, exibido na Figura 4.1. Assim, a tabela 4.5 detalha a primeira das quatro operações, equivalente à letra “C” da sigla CRUD, a publicação de conteúdo em um serviço.

Tabela 4.5: Caso de uso publicar conteúdo em um serviço

Ação – Publicar conteúdo em um serviço	
Ator:	Usuário
Pré-Condição:	Usuário possui ao menos um perfil associado ao sistema.
Pós-Condição:	Conteúdo é enviado para rede social.
Entrada:	Perfil a ser utilizado, além dos dados a serem enviados.
Saída:	-
Fluxo de Eventos	
Ator	Sistema
1. Informa a partir de qual perfil deseja postar o conteúdo, bem como o conteúdo a ser publicado.	
	2. Valida os dados a serem enviados. Caso estejam... A) <i>Corretos</i> Envia os dados para o serviço ao qual pertence o perfil especificado, para serem publicados, informando que obteve sucesso. B) <i>Incorretos</i> Retorna para (1), informando o erro.

A validação feita pelo sistema, conforme indicado no passo (2) do fluxo de eventos exibido na tabela 4.5, depende do serviço para o qual se está fazendo o envio dos dados. Por exemplo, para o Twitter, é feita a validação do tamanho da mensagem, que pode ter entre 1 e 140 caracteres. Esta validação é diferente para cada tipo de conteúdo, porém os passos do caso de uso são os mesmos.

A segunda operação de CRUD, detalhada na tabela 4.6, é o equivalente a letra “R” da sigla.

Tabela 4.6: Caso de uso consultar publicações de um perfil

Ação – Consultar publicações de um perfil	
Ator:	Visitante, Usuário
Pré-Condição:	Perfil está associado ao sistema.

Pós-Condição:	
Entrada:	Identificador do perfil.
Saída:	Lista de publicações e conteúdo do perfil.

Fluxo de Eventos	
Ator	Sistema
1. Informa o identificador do perfil a ser consultado.	
	2. Verifica se perfil está associado à alguma conta de usuário ativa. Caso... A) <i>Sim</i> Retorna todas as publicações do perfil, bem como seu conteúdo. B) <i>Não</i> Retorna para (1), informando o erro.

A consulta a publicações de um perfil é uma das únicas ações disponíveis também para visitantes. Para que isso seja possível, é necessário que o perfil indicado esteja associado a alguma conta de usuário do sistema.

A operação de atualização de uma publicação, representada pela letra “U” na sigla CRUD, é detalhada na tabela 4.7, a seguir.

Tabela 4.7: Caso de uso editar conteúdo de publicação em um serviço

Ação – Editar conteúdo de publicação em um serviço	
Ator:	Usuário
Pré-Condição:	Publicação a ser editada pertence a um perfil associado à conta do usuário no sistema.
Pós-Condição:	Conteúdo da publicação é alterado na rede social.
Entrada:	Identificador da publicação a ser alterada, e conteúdo alterado para a publicação.
Saída:	-
Fluxo de Eventos	
Ator	Sistema
1. Informa o identificador da publicação a ser alterada, e o novo conteúdo a ser publicado.	
	2. Valida os dados a serem enviados. Caso estejam... A) <i>Corretos</i> Envia os dados para o serviço ao qual pertence o perfil especificado,

	para serem editados, e informa que obteve sucesso. B) <i>Incorretos</i> Retorna para (1), informando o erro.
--	--

O usuário apenas pode editar uma publicação feita por um perfil associado à sua conta de usuário no sistema. Esta operação não é permitida para publicações feitas no Twitter, por uma limitação da própria rede social online.

Por fim, a quarta e última operação CRUD, ou seja, a exclusão de publicações de serviços, representada pela letra “D” no acrônimo, é detalhada na tabela 4.8.

Tabela 4.8: Caso de uso excluir publicação de um serviço

Ação – Excluir publicação de um serviço	
Ator:	Usuário
Pré-Condição:	Publicação a ser editada pertence a um perfil associado à conta do usuário no sistema.
Pós-Condição:	Conteúdo da publicação é removido do serviço.
Entrada:	Identificador da publicação a ser removida.
Saída:	-

Fluxo de Eventos	
Ator	Sistema
1. Informa o identificador da publicação a ser excluída.	
	2. Solicita confirmação da ação.
3. Responde à solicitação de confirmação, positiva ou negativamente.	
	4. Avalia a resposta do ator. Caso seja... A) <i>Positiva</i> Envia solicitação de exclusão da publicação informada ao serviço adequado, e informa que obteve sucesso. B) <i>Negativa</i> Retorna para (1), informando que a ação não foi concluída.

Assim como na edição de publicações, é necessário que o conteúdo a ser excluído pertença a um perfil associado à conta do usuário no sistema, impossibilitando que um usuário A apague publicações de um usuário B. De forma similar a outros procedimentos de exclusão existentes no sistema, é solicitada uma confirmação do

usuário, sendo possível cancelar a ação mesmo depois de ela ter sido iniciada, uma vez que se trata de uma ação irreversível.

Para a visualização da *timeline* no Twitter, optou-se por fazer um caso de uso específico, visto que o procedimento é um pouco diferente daqueles presentes em outros serviços. A tabela 4.9 detalha esse caso de uso.

Tabela 4.9: Caso de uso consultar *timeline* de um perfil Twitter

Ação – Consultar <i>timeline</i> de um perfil Twitter	
Ator:	Usuário
Pré-Condição:	Perfil está associado à conta do usuário no sistema.
Pós-Condição:	
Entrada:	Identificador do perfil.
Saída:	Lista de publicações e conteúdo da <i>timeline</i> do perfil.
Fluxo de Eventos	
Ator	Sistema
1. Informa o identificador do perfil a ser consultado.	
	2. Verifica se perfil está associado à conta de usuário do ator. Caso... A) <i>Sim</i> Retorna todas as publicações na <i>timeline</i> do perfil, bem como seu conteúdo. B) <i>Não</i> Retorna para (1), informando o erro.

A *timeline* de um perfil no Twitter exibe, além das mensagens publicadas pelo próprio perfil, as mensagens publicadas pelos perfis seguidos pelo usuário. Uma explicação mais detalhada dessa funcionalidade pode ser encontrada na seção 2.1.3.

4.3 Modelagem do banco de dados

Considerando que o conteúdo postado e visualizado pelos usuários e visitantes não ficam armazenados no banco de dados do sistema proposto, mas sim nas redes sociais online agregadas por ele, o banco de dados é bastante simples, conforme mostra o diagrama ilustrado na Figura 4.2. Optou-se por apresentar aqui o modelo físico, construído com a ferramenta MySQL Workbench especificamente para o SGBD escolhido para a implementação deste trabalho, uma vez que assim podem ser exibidas algumas otimizações feitas através da criação de índices, visando reduzir o tempo de resposta do sistema à interação do usuário.

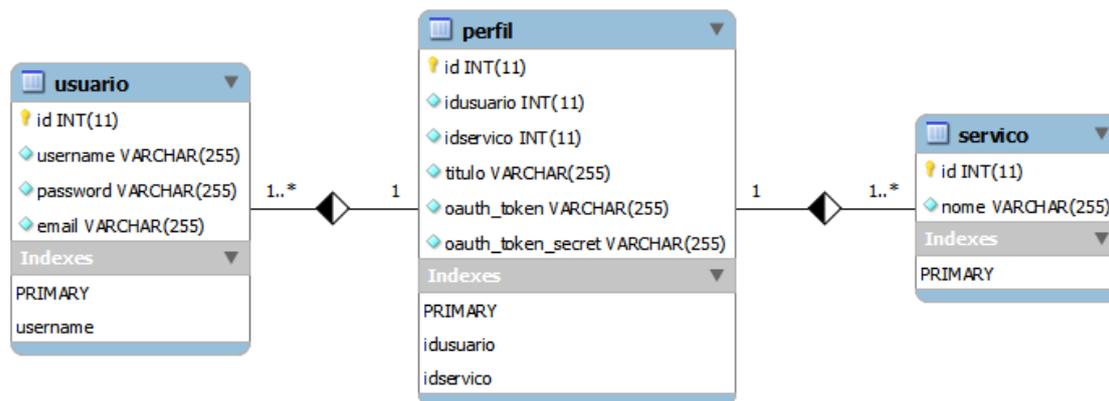


Figura 4.2: Diagrama do banco de dados

Apenas três tabelas foram criadas para o armazenamento das informações de usuários e perfis associados ao agregador. A tabela *usuario* guarda as informações sobre os usuários que se cadastram para utilizar o sistema, como nome de usuário, endereço de e-mail e senha, enquanto a tabela *servico* contém as informações sobre as redes sociais online, ou serviços, implementados pelo agregador. A terceira e última tabela, *perfil*, grava as informações dos perfis associados pelos usuários. Os campos *idusuario* e *idservico* são chaves estrangeiras, respectivamente, das tabelas *usuario* e *servico*. O campo *titulo* é utilizado para exibição no site. Por fim, os campos *oauth_token* e *oauth_token_secret* fazem parte do processo de autenticação da aplicação nas redes sociais online integradas. Optou-se por gravar as informações de autenticação na tabela *perfil*, ao invés de criar uma tabela para cada serviço, pois todos os serviços incluídos utilizam o mesmo procedimento de autenticação através da biblioteca OAuth, como explicado na seção 4.1.7, sobre as APIs.

A tabela *usuario*, além da chave primária, conta com um índice por nome de usuário, uma vez que consultas por esse campo tendem a ser muito frequentes na aplicação. Também a tabela *perfil* tem índices para os campos *idusuario* e *idservico*, pelos mesmos motivos citados para a tabela *usuario*.

4.4 Classes do sistema

Para a implementação do sistema, criou-se uma classe para representar cada uma das entidades envolvidas. Assim, uma classe *User* foi criada para representar um usuário do sistema. Para a representação de um perfil associado ao sistema, criou-se também uma classe chamada *Perfil*. No entanto, para a representação de serviços, criou-se uma classe para cada uma das redes sociais, utilizando-se de alguns padrões de projeto. Esta solução é explicada com mais detalhes na próxima seção deste trabalho.

4.4.1 Padrões de projeto

Padrões de projeto apresentam soluções genéricas reutilizáveis para problemas recorrentes no desenvolvimento de sistemas orientados a objetos (GAMMA et al., 1995). Um padrão de projeto descreve o problema, a solução, quando aplicar tal solução, e suas consequências. Assim, para este trabalho, foram identificadas situações onde o uso de algumas dessas soluções genéricas poderiam ser aplicadas para resolver um problema ou melhorar a manutenibilidade do sistema. Os padrões utilizados neste projeto estão descritos a seguir.

4.4.1.1 Padrão *Factory Method*

As operações que estarão disponíveis para as três redes sociais online escolhidas são fundamentalmente idênticas, porém a forma como devem ser implementadas difere para cada serviço. Identificou-se, assim, um cenário favorável à utilização do padrão de projeto *Factory Method*.

Factory Method define uma interface para a criação de um objeto, delegando a escolha da classe a ser instanciada para as subclasses (GAMMA et al., 1995). Criou-se, então, uma classe *Servico*, a qual serve de interface para a instanciamento das subclasses *Twitter*, *YouTube* e *Flickr*. No momento da instanciamento do objeto, o sistema passa um parâmetro à classe *Servico* e, a partir desse parâmetro, a subclasse a ser instanciada é escolhida. Assim, mesmo que mais serviços sejam adicionados no futuro, poucas alterações terão de ser feitas no código.

4.4.1.2 Padrão *Facade*

Para interação com as APIs, utilizou-se bibliotecas já existentes, disponibilizadas na Internet, que permitem essa interação de uma forma simplificada para programas em PHP. No entanto, para garantir que todas as funcionalidades de cada API possam ser utilizadas, essas bibliotecas acabam tendo um grau de complexidade maior do que o necessário para este trabalho. Assim, optou-se por utilizar o padrão de projeto *Facade*, a fim de criar uma interface ainda mais simplificada, visando apenas as operações utilizadas neste trabalho.

O objetivo do padrão *Facade* é, justamente, definir uma interface em um nível mais alto, tornando um subsistema mais fácil de utilizar (GAMMA, 1995).

4.4.2 Diagrama de classes

A fim de esclarecer a forma como o sistema foi implementado, um diagrama de classes é apresentado na Figura 4.3. Apesar de ter sido utilizada a notação UML para a confecção do diagrama, algumas adaptações foram feitas. Entendeu-se que tais modificações não prejudicam a compreensão do diagrama, nem tampouco sua representatividade do sistema. Ainda assim, é dada uma breve explicação dessas adaptações, bem como do diagrama como um todo, a seguir.

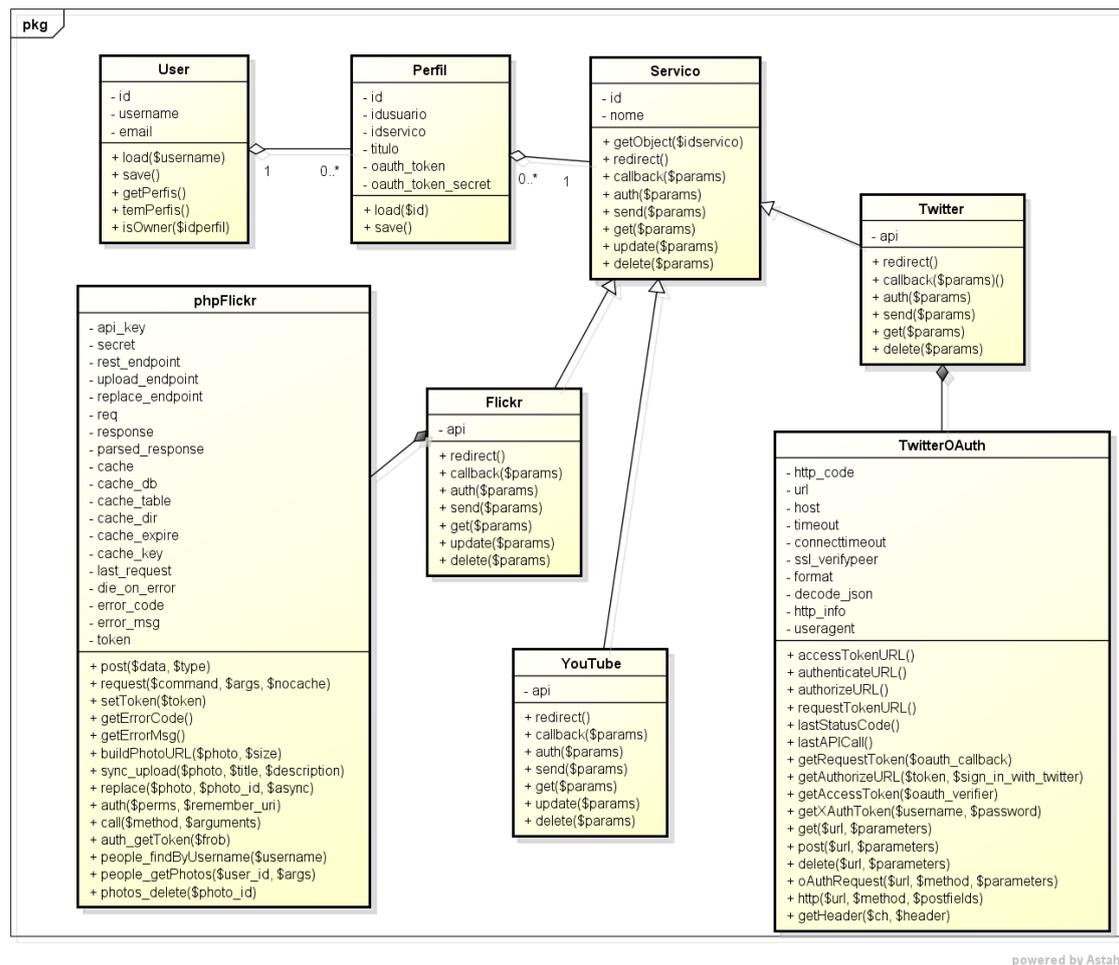


Figura 4.3: Diagrama de classes

Apesar de não estarem representados no diagrama, todas as classes possuem métodos `get()` e `set()`, responsáveis por retornar o valor de um atributo e colocar um valor recebido por parâmetro em um atributo, respectivamente. Por simplicidade, optou-se por ocultar tais métodos.

Uma vez que a linguagem PHP não possui tipagem forte, ou seja, variáveis podem assumir tipos diferentes durante sua vida no sistema, optou-se por não exibir os tipos de atributos, parâmetros e retorno de funções.

Pode-se perceber no diagrama ilustrado na Figura 4.3 a presença das classes representativas de usuários (classe `User`) e perfis (classe `Perfil`), citadas anteriormente. Estas classes possuem métodos específicos para a persistência com o banco de dados, `load()` e `save()`. O primeiro recebe como parâmetro o identificador do usuário ou perfil a ser instanciado, busca no banco de dados todas as informações do registro relacionado a esse identificador e instancia um objeto com essas informações. O método `save()`, por sua vez, faz o processo inverso, atualizando o banco de dados com as informações carregadas pelo objeto instanciado. Um usuário pode ter inúmeros perfis associados à sua conta no sistema, sendo necessário, portanto, a implementação de um método que retorna uma lista desses perfis, chamado `getPerfis()`, além de um método para especificar se um perfil está associado ou não à conta de um usuário; este último método foi chamado de `isOwner()`, e recebe como parâmetro o identificador único do

perfil. O método *temPerfis()*, como o nome sugere, simplesmente informa da existência de perfis associados à conta do usuário ou não.

A seguir, então, tem-se a aplicação do padrão *Factory Method* na classe *Servico*, tendo como subclasses as representações de cada um dos serviços implementados: *Twitter*, *Flickr* e *YouTube*. Essas três subclasses são especializações da classe *Servico*, herdando suas propriedades e métodos. O método *getObject()*, no entanto, é implementado apenas pela classe *Servico*, tratando-se de um método do tipo *final*. Além disso, é um método dito *static*, podendo ser chamado sem a necessidade de instanciação de um objeto. Assim, é utilizado para instanciar as subclasses, sendo o parâmetro *idservico* o responsável por definir qual das classes será instanciada. Dessa forma, outros serviços podem ser facilmente incorporados ao sistema, bastando a atualização desse método para que as novas classes possam ser utilizadas. O código para esse método pode ser visto na Figura 4.4.

```
// factory method
static final function getObject($idservico) {
    switch($idservico){
        case ID_TWITTER:
            return new Twitter();
        break;
        case ID_FLICKR:
            return new Flickr();
        break;
        case ID_YOUTUBE:
            return new Youtube();
        break;
        default:
            return false;
    }
}
```

Figura 4.4: Código do método *getObject*, implementado pela classe *Servico*

Todos os demais métodos da classe *Servico* retornam o valor booleano *false*, uma vez que devem ser implementados pelas subclasses.

Ainda na Figura 4.3, nota-se que a subclasse *Twitter*, assim como a subclasse *Flickr*, possuem um atributo adicional, chamado *api*. Para a classe *Twitter*, este atributo é uma instância da classe *TwitterOAuth*²⁵, a qual é responsável pela interação com a API do serviço. A classe *Flickr*, no entanto, tem no atributo *api* um objeto da classe *phpFlickr*²⁶ instanciado. Isso ocorre justamente devido à aplicação do padrão *Facade*, explicado anteriormente. As classes específicas para comunicação com as APIs são muito mais complexas que as classes utilizadas no sistema, pois implementam muito mais possibilidades. No diagrama exibido na Figura 4.3, apenas os métodos utilizados pela classe *Flickr* foram listados na classe *phpFlickr*; a listagem completa conta com mais de 100 métodos, sendo um para cada função da API do *Flickr*. A classe *YouTube*, apesar de ser exibida no diagrama, não chegou a ser implementada, sendo sugerida para um trabalho futuro, como será explicado mais adiante.

²⁵ *TwitterOAuth*. <https://github.com/abraham/twitteroauth>. Acesso em 08 de Julho de 2012.

²⁶ *phpFlickr*. <http://phpflickr.com/>. Acesso em 08 de Julho de 2012.

Os métodos *redirect()* e *callback()* são utilizados durante a fase de associação de um perfil ao sistema. O método *redirect()* envia o usuário para a página de autenticação na rede social escolhida, informando os dados necessários, enquanto o método *callback()* recebe e trata a resposta, positiva ou negativa, vinda dessa página. Em caso de autorização concedida, o perfil é associado a conta do usuário.

O método *auth()* é responsável pela autenticação de um perfil, através da API do serviço correspondente, necessária para a execução de determinadas ações, como o envio ou exclusão de postagens. Os demais métodos das subclasses de *Servico* – *send()*, *get()*, *update()* e *delete()* – são os responsáveis pelas operações CRUD de publicações, as quais foram descritas nos casos de uso essenciais expandidos, na seção 4.2.2. A Figura 4.5 mostra a implementação do método *send()* feita pela classe *Flickr*.

```
public function send($params=array()){

    if(!isset($this->api)) { //nao autenticou
        return false;
    }

    if(!$params['foto']) { //nao tem a foto (obrigatorio)
        return false;
    } elseif(!file_exists($params['foto'])){ //arquivo informado não existe
        return false;
    }

    $result = $this->api->sync_upload(
        $params['foto'], //obrigatorio
        $params['title'], //opcional
        $params['description'] //opcional
    );

    return $result;
}
```

Figura 4.5: Código do método *send()* da classe *Flickr*

Na Figura 4.5, é importante perceber que o atributo *api* deve conter uma instância da classe *phpFlickr*, para que a comunicação através da API do Flickr seja possível. O upload da imagem a ser enviada é feito primeiro no servidor local, sendo informado o endereço do recurso através do parâmetro *\$params['foto']*. O método ainda verifica a existência do arquivo no servidor, antes de acionar a API para fazer o envio da publicação.

4.5 Metodologia de desenvolvimento

A metodologia de desenvolvimento adotada para este projeto conta com algumas práticas do desenvolvimento ágil de *software*, como a implementação prioritária de funcionalidades consideradas mais importantes para o sistema e a utilização de padrões de codificação. No entanto, alguns conceitos do modelo de desenvolvimento em cascata, como a finalização de uma parte do projeto antes do início da próxima fase, também foram adotadas. A refatoração do código, outra característica dos métodos ágeis, também foi bastante utilizada durante o desenvolvimento, sendo atualizados os diagramas para se adequarem às mudanças no código.

4.6 Testes

Após a implementação do protótipo, foram efetuados alguns testes simples, a fim de verificar a aceitação do sistema junto aos usuários e a segurança dos dados.

4.6.1 Teste de segurança

Para o teste de segurança, foram verificadas algumas das principais técnicas de ataque a sistemas PHP. Os testes foram feitos manualmente, e o escopo englobou as técnicas de injeção de SQL, *Cross-site scripting* (XSS) e a validação do tipo de arquivo a ser enviado em formulários de *upload*. O sistema se mostrou imune aos três tipos de ataques, sendo para este trabalho considerado um resultado satisfatório. Outra medida adotada para garantir a segurança das contas de usuário foi a utilização de um algoritmo de criptografia de 512 bits para o armazenamento da senha.

4.6.2 Testes de usabilidade

Foram convidadas 10 pessoas para os testes de usabilidade do sistema. Um pequeno roteiro foi desenvolvido, o qual compreendia a criação de uma conta no sistema, a associação de pelo menos um perfil à conta de usuário, a utilização do perfil associado para fazer o CRUD de ao menos um conteúdo e, por fim, a navegação pelas páginas de usuários existentes. Todos os convidados são leigos em programação, no entanto tem certa intimidade com o manuseio de redes sociais online. Das 10 pessoas selecionadas, apenas uma relatou já ter usado agregadores de redes sociais anteriormente. Para esse teste, foram solicitadas as impressões dos usuários quanto ao sistema, bem como sugestões de melhorias. Os testes foram aplicados individualmente, estando um avaliador presente.

Após finalizados os testes, 7 pessoas indicaram que o sistema tem um grande potencial no mercado. 9 pessoas sugeriram que o sistema deveria ser expandido para acolher outras redes sociais online (em especial o Facebook), e 5 sentiram falta de outras funcionalidades das redes sociais já existentes, como envio de mensagens privadas ou a possibilidade de adicionar comentários através da interface do sistema. 3 usuários sugeriram mudança na interface do programa, além da possibilidade de personalização da página de usuário, e duas pessoas sugeriram a possibilidade de postagem de conteúdo através de dois perfis ao mesmo tempo.

Os resultados desse teste mostram que o sistema tende a ser bem aceito no mercado, desde que algumas melhorias sejam feitas. Todas as sugestões coletadas no teste são sugeridas como trabalhos futuros, além de um estudo mais amplo, com um grupo maior de usuários e abrangendo mais variáveis a serem analisadas.

5 GUIA DE USO DO SISTEMA

Ao acessar a página inicial do sistema, apresentada na Figura 5.1, o visitante tem a opção de efetuar o cadastro, preenchendo os campos “*email*”, “*username*” e “*password*”, na caixa à direita, e submetendo o formulário através do botão “Cadastrar-se”. O sistema verifica a existência do e-mail informado no banco de dados, não permitindo que sejam cadastrados dois iguais. O *username* escolhido também deve ser único no sistema, e uma verificação similar é feita. Ao concluir o cadastro, o visitante já estará devidamente autenticado com os dados recém-criados. Visitantes que já possuem uma conta no sistema podem fazer a autenticação também a partir da página inicial, preenchendo os campos “*username*” e “*password*” da caixa à esquerda, e submetendo o formulário através do botão “Login”.

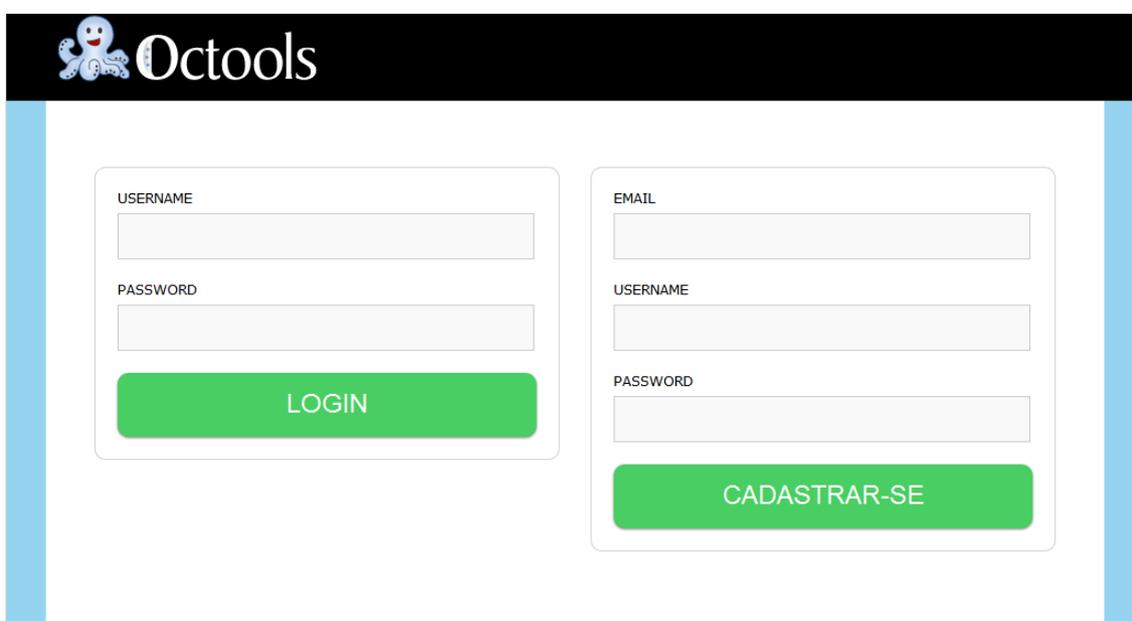
A imagem mostra a interface de usuário do sistema Octools. No topo, há um cabeçalho preto com o logo Octools (um polvo azul) e o nome 'Octools' em branco. Abaixo, há duas caixas de formulário brancas com bordas cinzas, cada uma com um botão verde. A caixa da esquerda contém campos para 'USERNAME' e 'PASSWORD', com o botão 'LOGIN' abaixo. A caixa da direita contém campos para 'EMAIL', 'USERNAME' e 'PASSWORD', com o botão 'CADASTRAR-SE' abaixo. O fundo da página é branco com duas barras verticais azuis nas laterais.

Figura 5.1: Tela inicial do sistema Octools

Uma vez autenticado, o usuário é direcionado para sua página de usuário, onde é exibida no menu lateral a opção de “Gerenciar perfis”. Ao clicar nesse link, uma tela se abre, permitindo ao usuário associar perfis dos serviços implementados no sistema à sua conta. Para fazer essa associação, é necessário que o usuário clique sobre o ícone do serviço ao qual pertence o perfil a ser associado, sendo então enviado ao site da rede social online, em uma nova janela, onde são solicitadas suas credenciais, caso não esteja autenticado na rede social ainda, e um pedido de autorização para a aplicação acessar seus dados. Após autorizar o acesso, o usuário volta para o site do agregador, já com esse perfil associado. Nessa mesma tela de gerenciamento de perfis, uma lista com os

perfis já associados à conta é exibida no lado direito, sendo permitido ao usuário desassociá-las clicando no botão “X” ao lado do respectivo perfil. A tela de gerenciamento de perfis é exibida na Figura 5.2, a seguir.

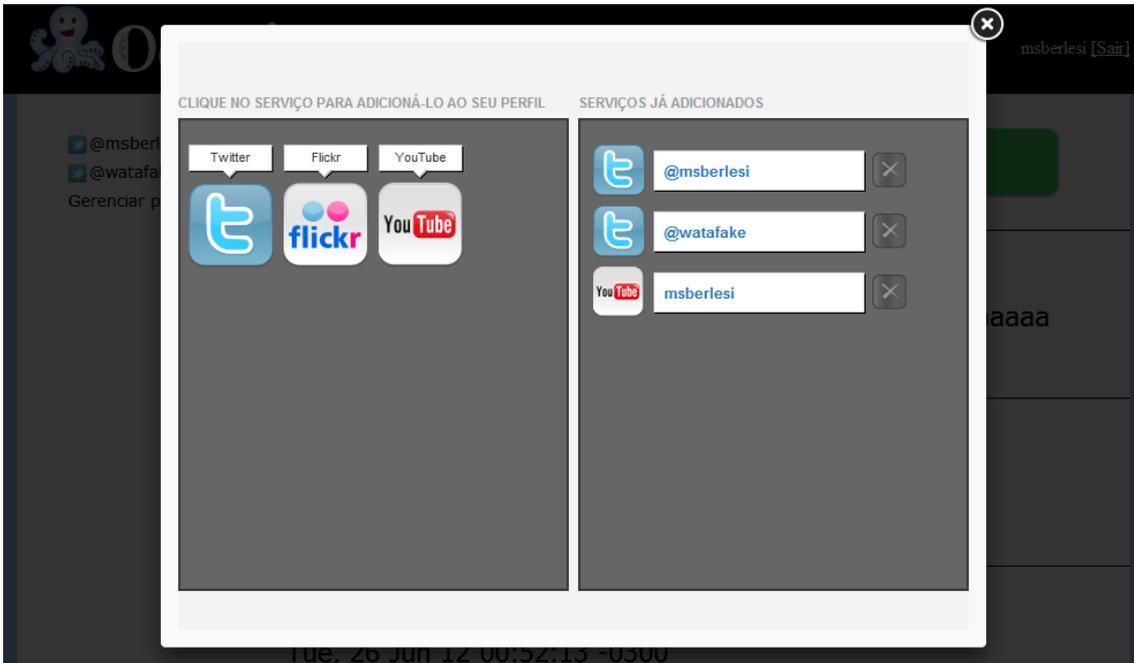


Figura 5.2: Tela de gerenciamento de perfis no sistema Octools

A página de usuário, quando houver perfis associados, exibirá por padrão o conteúdo do primeiro perfil associado. No menu à esquerda, são exibidos todos os perfis já associados pelo usuário. Ao clicar em um desses itens do menu, o sistema carrega na página principal o conteúdo desse perfil. Caso o item acessado seja uma conta do Twitter, o usuário poderá ver, acima das mensagens, os botões “Tweets de @username”, selecionado por padrão, e “Sua Timeline”, podendo alternar entre o conteúdo criado por ele próprio e o conteúdo criado por aqueles que ele segue, como pode ser visto na Figura 5.3. O usuário pode, também, excluir ou editar seus conteúdos através da interface do agregador. Essas opções são exibidas apenas quando as ações são possíveis.



Figura 5.3: Página de usuário do Octools, acessada pelo próprio usuário

Um usuário A, assim como um visitante não autenticado, acessando a página de um usuário B, consegue ver os perfis associados pelo usuário B no menu à esquerda, porém a opção “Gerenciar perfis” não é exibida. Especificamente para o Twitter, a opção “Sua *timeline*” também aparece apenas para o usuário B, uma vez que não é permitida aos demais a visualização da *timeline* de um usuário. Caso a página do usuário B não tenha nenhum perfil associado, uma mensagem informando que “Usuário B ainda não associou nenhum perfil” é exibida.

No topo do site, os visitantes tem a opção de entrar no sistema, através de um formulário de *login*, que é exibido ao lado de um link para a página de cadastro. No caso de um usuário autenticado, o nome de usuário da conta Octools autenticada no momento é exibido, junto a uma opção para sair do sistema. Ao clicar sobre o nome de usuário, uma tela se abre oferecendo ao usuário a opção de alterar seus dados cadastrais. A tela de alteração dos dados é similar ao formulário de cadastro no sistema. Optou-se por não permitir a alteração do nome de usuário, sendo então possível alterar apenas o e-mail fornecido e a senha. Nessa tela de alteração de cadastro é exibido ainda um link para o usuário que deseje desativar sua conta no sistema. Ao clicar no link, é solicitada uma confirmação ao usuário, antes de excluir definitivamente o cadastro do sistema. Todos os perfis associados a uma conta são apagados no momento em que a conta é desativada, evitando assim perfis ditos órfãos no sistema. Outra opção sempre disponível no topo do site, para usuários autenticados, é a de “compor”, representada por um ícone. Ao clicar neste ícone, surge uma tela onde há um campo para o usuário escolher a partir de qual perfil será feita a interação, no qual constam todos os perfis associados pelo usuário ao sistema; além disso, um campo para redigir a mensagem ou escolher o arquivo a ser enviado é exibido, dependendo do tipo de perfil selecionado. No caso do Twitter, um campo de texto é exibido, enquanto no caso do Flickr é disponibilizado um campo para envio de arquivos. A tela de composição é exibida na Figura 5.4, estando um perfil do Twitter selecionado e, portanto, com uma caixa de texto sendo exibida.

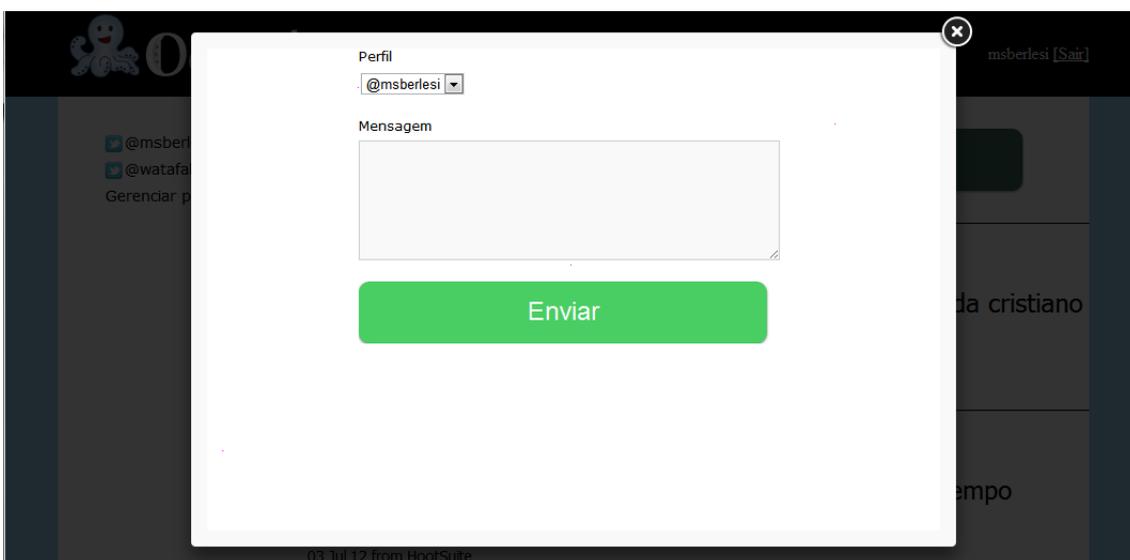


Figura 5.4: Página de composição do Octools, com perfil do Twitter selecionado

6 CONCLUSÃO

Este trabalho propôs o projeto de um sistema agregador de redes sociais, baseado em ambiente Web, que conta ainda com funcionalidades de *mashup*. Como pôde ser visto nas seções anteriores, este objetivo principal foi atingido, através do desenvolvimento de um sistema chamado de Octools. O objetivo principal deste trabalho pode ser dividido em duas partes, o agregador de redes sociais e o *mashup*, com o conteúdo de vários perfis do usuário. A primeira parte deste objetivo principal consistia em permitir que usuários cadastrados pudessem interagir com redes sociais online, dentre as quais foram escolhidas três para fazer parte do sistema inicialmente, através da interface do portal. A segunda parte, por sua vez, consistia em reunir o conteúdo postado através dos perfis associados pelo usuário em uma única página Web. Durante este projeto, foram pesquisados alguns sistemas já existentes, com funcionalidades similares às do Octools, porém nenhum deles reunia estas duas faces em um único sistema.

Assim, o Octools cumpre com seu objetivo principal, visando facilitar a interação de usuários com redes sociais online, propiciando uma interface única para acesso a várias redes sociais. No entanto, o protótipo produzido não atingiu todos os requisitos funcionais propostos, uma vez que a integração com o serviço YouTube não chegou a ser implementada. Assim, é impossível associar perfis YouTube à conta de usuário no sistema, ficando incompleto o terceiro requisito funcional, o qual diz que "os usuários autenticados devem poder associar e desassociar perfis das redes sociais online Twitter, Flickr e YouTube à sua conta no sistema". A integração do YouTube é, então, proposta como trabalho futuro. Outro requisito funcional, que diz que "todos os usuários com perfis da rede social online YouTube associados devem ter a possibilidade de realizar o CRUD de seus vídeos do YouTube", é também considerado inatingido por este trabalho, apesar de ser possível uma interpretação de que se não houverem perfis YouTube associados no sistema, o requisito poderia ser considerado como atingido, mesmo sem ter sido implementado. Esse requisito fica, portanto, proposto também como um trabalho futuro. Os demais requisitos funcionais foram atingidos, assim como todos os requisitos não funcionais.

Naturalmente, várias melhorias podem ser feitas ao sistema, sendo uma delas o acréscimo de outros serviços ao agregador. Desde o primeiro momento do projeto, passando pelo desenvolvimento da interface, até a implementação do código final, pensou-se na possibilidade de adição de novos serviços, estando, portanto, o sistema preparado para receber tal melhoria. Ainda no contexto de agregador de redes sociais, outra possibilidade de melhoria no sistema é a implementação de outras funcionalidades inerentes às redes sociais, como o envio de mensagens privadas, comentários, entre outras. Embora o sistema não esteja preparado para tais funcionalidades, a flexibilidade da linguagem de programação escolhida neste projeto facilitaria sua adição. Além das funcionalidades presentes nos sites das próprias redes sociais, há ainda outras,

encontradas em outros sistemas agregadores, que poderiam ser incorporadas ao Octools, como, por exemplo, o agendamento do envio de mensagens, ou o envio através de vários perfis simultaneamente.

Por outro lado, várias mudanças também podem ser sugeridas ao *mashup* implementado, como a possibilidade de o usuário personalizar sua página de usuário, escolhendo as fontes, cores, imagens de fundo e formatos para a exibição de seu conteúdo, tornando o ambiente ainda mais pessoal, ou ainda com a sugestão de diferentes modelos de *layout* prontos.

O protótipo desenvolvido foi exposto a um pequeno grupo de usuários selecionados, os quais foram encorajados a utilizar e criticar o sistema. As avaliações, em sua maioria, citam o grande potencial do sistema, apesar da falta de várias opções de interação nas redes sociais no protótipo desenvolvido inicialmente. As sugestões de melhoria para o sistema incluíram várias das já citadas anteriormente nessa seção. Foram aplicados também alguns testes relacionados à segurança dos dados armazenados no sistema, obtendo-se um resultado satisfatório.

Antes da abertura do Octools ao grande público, entretanto, várias melhorias devem ser implementadas, testes mais rigorosos de segurança devem ser aplicados, e avaliações com grupos maiores de usuários reais, embora ainda fechados, devem ser feitas, a fim de coletar dados sobre a usabilidade do sistema e de sua interface.

REFERÊNCIAS

Alexa. Disponível em: <<http://www.alexa.com>>. Acesso em: Jun. 2012.

BENEVENUTO, F.; ALMEIDA, J. M.; SILVA, A. S. Explorando Redes Sociais Online: Da Coleta e Análise de Grandes Bases de Dados às Aplicações. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES E SISTEMAS DISTRIBUÍDOS, 29., 2011, Campo Grande. **Anais...** Campo Grande: UFMS: SBRC, 2011. p. 63-102.

BOYD, D. M.; ELISSON, N. B. **Social Network Sites: Definition, History, and Scholarship**. Journal of Computer-Mediated Communication, vol 13, n.1, p. 210-230, 2008.

CASTRO, E. **HTML, XHTML, and CSS**. 6th ed. Berkeley: Peachpit Press, 2007.

CHA, M.; MISLOVE, A.; GUMMADI, K. P. A Measurement-driven Analysis of Information Propagation in the Flickr Social Network. In: INTERNATIONAL WORLD WIDE WEB CONFERENCE, 18., 2009, Madri. **Proceedings...** Madri: ACM, 2009. p. 721-730.

CHU, Z.; GIANVECCHIO, S.; WANG, H.; JAJODIA, S. Who is tweeting on Twitter: human, bot, or cyborg?. In: ANNUAL COMPUTER SECURITY APPLICATIONS CONFERENCE, 26., 2010, Austin. **Proceedings...** Austin: ACM, 2010. p. 21-30.

Facebook Newsroom. Disponível em <<http://newsroom.fb.com/content/default.aspx?NewsAreaId=22>>. Acesso em: Jun. 2012.

Facebook. Disponível em: <<http://www.facebook.com>>. Acesso em: Jun. 2012.

Flavors. Disponível em <<http://www.flavors.me>>. Acesso em: Jul. 2012.

Flickr. Disponível em: <<http://www.flickr.com>>. Acesso em: Jun. 2012.

GAMMA, E.; HELM, R.; JOHNSON, R.; VLISSIDES, J. **Design Patterns: Elements of Reusable Object-Oriented Software**. Boston: Addison-Wesley, 1995.

GOODMAN, D. **Javascript: a Bíblia**. 3. ed. [s.l.]:Campus, 2001.

Google Developers. Disponível em <https://developers.google.com/youtube/getting_started>. Acesso em: Jun. 2012.

HAZAN, C.; LEITE J. C. S. Indicadores para a Gerência de Requisitos. In: WORKSHOP EM ENGENHARIA DE REQUISITOS, 6., 2003, Piracicaba. **Anais...** Piracicaba: UFRJ: 2003. p. 285-301.

jQuery. Disponível em <<http://www.jquery.com>>. Acesso em: Jul. 2012.

KILOV, H. **Business Specifications: The Key to Successful Software Engineering**. [s.l.]: Prentice Hall, 1998.

LARMAN, C. **Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo**. 3. ed. Porto Alegre: Bookman, 2007.

MAXIMILIEN, E. M.; WILKINSON, H.; DESAI, N.; TAI, S. A Domain-Specific Language for Web APIs and Services Mashups. In: INTERNATIONAL CONFERENCE ON SERVICE-ORIENTED COMPUTING, 5., 2007, Viena. **Proceedings...** Viena: Springer Verlag: ICSOC, 2007. p. 13-26.

NEGRINO, T.; SMITH, D. **JavaScript and Ajax for the Web**. 7th ed. Berkeley: Peachpit Press. 2009.

OAuth. Disponível em <<http://www.oauth.net>>. Acesso em: Jul. 2012.

OBJECT MANAGEMENT GROUP. **OMG: UML 2.0 Superstructure Specification**. 2005. Disponível em: <<http://www.omg.org/spec/UML/2.0/>>. Acesso em: jul. 2012.

phpFlickr. Disponível em <<http://phpflickr.com/>>. Acesso em: Jul. 2012.

Profilactic. Disponível em <<http://www.profilactic.com>>. Acesso em: Jul. 2012.

RUFINO, A.; TABOSA, H. R.; NUNES, J. V. Redes Sociais: Surgimento e Desenvolvimento dos Micro-Bloggings. In: INFOBRASIL TI & TELECOM, 36., 2010, Fortaleza. **Proceedings...** Fortaleza: UFC, 2010.

Seismic. Disponível em <<http://www.seismic.com>>. Acesso em: Jun. 2012.

SILVA, D. M. **UML: Guia de Consulta Rápida**. São Paulo: Novatec, 2001.

SOARES, W. **PHP 5: Conceitos, Programação e Integração com Banco de Dados**. São Paulo: Érica, 2004.

SOMMERVILLE, I. **Software Engineering**. 9th ed. Boston: Addison-Wesley, 2011.

STAMEY, J.; RICHARDSON, T. Middleware Development with AJAX. **Journal of Computing Sciences in Colleges**, v.22, n.2, p. 281-287, Dez. 2006.

Sysomos. Disponível em <<http://www.sysomos.com/insidetwitter/>>. Acesso em: Jun. 2012.

Tech Crunch. Disponível em: <<http://techcrunch.com/2012/04/25/with-7b-photos-flickr-debuts-new-speedy-html5-image-uploader-drag-and-drop-interface-and-more/>>. Acesso em: Jun. 2012.

TweetDeck. Disponível em <<http://www.tweetdeck.com>>. Acesso em: Jun. 2012.

Twitter Blog. Disponível em <<http://blog.twitter.com/2011/05/all-decked-out.html>>. Acesso em: Jun. 2012.

Twitter. Disponível em: <<http://www.twitter.com>>. Acesso em: Jun. 2012.

TwitterOAuth. Disponível em <<https://github.com/abraham/twitteroauth/>>. Acesso em: Jul. 2012.

Vimeo. Disponível em <<https://www.vimeo.com>>. Acesso em: Jul. 2012.

WEAVER, A. C.; MORRISON B. B. Social Networking. **Computer**, v.41, n.2, p. 97-100, Fev. 2008.

YANG, Z.; GUO, J.; CAI, K.; TANG, J.; LI, J.; ZHANG, L.; SU, Z. Understanding Retweeting Behaviors in Social Networks. In: ACM INTERNATIONAL CONFERENCE ON INFORMATION AND KNOWLEDGE MANAGEMENT, 19., 2010, Toronto. **Proceedings...** Toronto: ACM, 2010. p. 1633-1636.

YouTube. Disponível em: <<http://www.youtube.com>>. Acesso em: Jun. 2012.