

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
FACULDADE DE INFORMÁTICA
CURSO DE BACHARELADO EM CIÊNCIAS DA COMPUTAÇÃO

FERRAMENTA DE AUTOMAÇÃO PARA REDES SOCIAIS

TRABALHO DE CONCLUSÃO II

Sergio Nogueira de Sá Filho

Orientador: Professor Alexandre Agustini

Porto Alegre, 17 de Junho de 2013.

RESUMO

Este trabalho descreve o desenvolvimento de uma ferramenta online utilizada por usuários da rede social *Twitter* com a finalidade de realizar ações automáticas no perfil desta rede com o intuito de automatizar a conta do mesmo. O objetivo geral é realizar ações pré-determinadas possibilitando a atualização constante da conta sem que haja interação do usuário.

Palavras-chave: tratar mensagens *Twitter*; envio automático de mensagens *Twitter*; *retweet* automático; automatizar *Twitter*; ferramenta para *Twitter*.

ABSTRACT

This paper describes the development of an online tool that is used by users of the Twitter's social network performing automatic actions on the profile of the user in order to automate its account. The overall goal is to perform predetermined actions enabling constant updating on the account without user interaction.

Keywords: treat Twitter messages; automatically send Twitter messages; *retweet* automatic; automate Twitter; tool for Twitter.

LISTA DE ILUSTRAÇÕES

FIGURA 1 - CENÁRIO REDUZIDO.....	18
FIGURA 2 - CENÁRIO TOTAL.....	19
FIGURA 3 - MODELO DE DADOS.....	34
FIGURA 4 - CONEXÕES DA ARQUITETURA.....	37
FIGURA 5 - RELAÇÃO DA CLASSE SERVICESSINGLETON.....	39
FIGURA 6 - EXEMPLO DE MODIFICAÇÃO DE REGRAS NA FERRAMENTA	41

LISTA DE TABELAS

TABELA 1 - ANÁLISE DE FRASES.....	16
TABELA 2 - REQUISITOS FUNCIONAIS.....	20
TABELA 3 - USABILIDADES	21
TABELA 4 - LICENÇAS.....	21
TABELA 5 - DISPONIBILIDADE.....	21
TABELA 6 - SEGURANÇA	21
TABELA 7 - ALGORITMO ENABLEAUTORETWEET.....	39
TABELA 8 - ALGORITMO VERIFICAÇÃO DE PERFIS CADASTRADOS	40
TABELA 9 - ALGORITMO DE VERIFICAÇÃO DE UMA MENTION	40
TABELA 10 - SUGESTÃO DE NOVAS REGRAS (PALAVRAS EM COMUM)	42
TABELA 11 - SUGESTÃO DE NOVAS REGRAS	42
TABELA 12 - COMPARATIVO DAS FERRAMENTAS.....	45
TABELA 13 - UTILIZAÇÃO DAS CHAVES DE SEGURANÇA DO PERFIL (TWITTER4J).....	46
TABELA 14 - UTILIZAÇÃO DAS CHAVES DE SEGURANÇA DA APLICAÇÃO (TWITTER4J)	47
TABELA 15 - VERIFICAÇÃO DE MENTIONS (TWITTER4J)	47
TABELA 16 - ENVIO DE MENSAGEM PARA TIMELINE (TWITTER4J).....	48
TABELA 17 - ENVIO DE MENSAGEM PRIVADA (TWITTER4J)	49
TABELA 18 - LEITURA DE MENSAGEM PRIVADA (TWITTER4J)	49
TABELA 19 - BLOQUEIO DE UM PERFIL (TWITTER4J)	49
TABELA 20 - DESBLOQUEIO DE UM PERFIL (TWITTER4J)	50
TABELA 21 - BUSCA DE HASHTAGS (TWITTER4J).....	50

SUMÁRIO

RESUMO	2
ABSTRACT.....	3
1. INTRODUÇÃO.....	9
1.1 MOTIVAÇÃO	10
1.2 OBJETIVO GERAL.....	10
1.3 OBJETIVO ESPECÍFICO	11
1.4 METODOLOGIA	11
1.5 ORGANIZAÇÃO DO TEXTO.....	12
2. SISTEMA	13
2.1 VISÃO GERAL	13
2.1.1 MENTIONS	13
2.1.2 HASHTAGS.....	13
2.1.3 FRASES ESPECÍFICAS.....	14
2.2 RETWEET DE MENTIONS E HASHTAGS.....	14
2.3 DEFINIÇÃO DE REGRAS	15
2.4 ENVIO AUTOMÁTICO DE MENSAGENS	16
2.5 BLOQUEIO DE USUÁRIOS	16
2.6 ESTATÍSTICAS.....	16
2.7 SUGESTÃO DE NOVAS REGRAS	17
2.8 TRATAMENTO DE MENSAGENS PRIVADAS	17
3. ANÁLISE E IDENTIFICAÇÃO DE REQUISITOS	18
3.1 ESCOPO.....	18
3.2 REQUISITOS FUNCIONAIS	19
3.3 REQUISITOS NÃO FUNCIONAIS	20
3.3.1 USABILIDADES.....	20
3.3.2 LICENÇAS	21
3.3.3 DISPONIBILIDADE	21
3.3.4 SEGURANÇA.....	21
3.4 DETALHAMENTO DOS CASOS DE USO	21
3.4.1 CADASTRO DE USUÁRIO.....	21
3.4.2 MODIFICAÇÃO DE CADASTRO DE USUÁRIO	22
3.4.3 EXCLUSÃO DO CADASTRO DO USUÁRIO.....	23
3.4.4 ENTRAR NA FERRAMENTA	23

3.4.5	SAIR DA FERRAMENTA.....	24
3.4.6	INCLUSÃO DE UMA CONTA DO TWITTER.....	25
3.4.7	EXCLUSÃO DE UMA CONTA DO TWITTER	25
3.4.8	ATIVAÇÃO DOS SERVIÇOS.....	26
3.4.9	DESATIVAÇÃO DOS SERVIÇOS	27
3.4.10	INSERÇÃO DE REGRAS	27
3.4.11	MODIFICAÇÃO DE REGRAS	28
3.4.12	EXCLUSÃO DE REGRAS.....	29
3.4.13	CONSULTA DE REGRAS.....	30
3.4.14	CONSULTA DE REGRAS SUGERIDAS.....	30
3.4.15	HABILITAR INTEGRAÇÃO COM OUTRA APLICAÇÃO	31
3.4.16	DESABILITAR INTEGRAÇÃO COM OUTRA APLICAÇÃO	32
3.4.17	CONSULTAR INTEGRAÇÕES DISPONÍVEIS.....	32
3.5	MODELO DE DADOS	34
4.	ARQUITETURA.....	36
4.1	VISÃO GERAL	36
4.2	PACOTES SIGNIFICATIVOS	36
4.3	VISÃO DE PROCESSOS	36
4.4	VISÃO DA IMPLEMENTAÇÃO	36
4.5	PLATAFORMA E GERENCIAMENTO DE DADOS	37
5.	PRINCIPAIS ALGORITMOS	38
5.1	<i>RETWEET</i>	38
5.1.1	<i>SERVICESSINGLETON</i>	38
5.1.2	<i>AUTORETWEETTHREAD</i>	40
5.2	MODIFICAÇÃO DE REGRAS.....	41
5.3	SUGESTÃO DE NOVAS REGRAS	42
5.4	ESTATÍSTICAS GERAIS.....	42
6.	TRABALHOS RELACIONADOS	44
6.1	<i>ROUNDTEAM</i>	44
6.2	<i>RETRANSTWITTER</i>	44
6.3	<i>TWEETADDER</i>	44
6.4	COMPARATIVOS DAS FERRAMENTAS	45
7.	BIBLIOTECA <i>TWITTER4J</i>	46
7.1	CONEXÃO COM O <i>TWITTER</i>	46
7.2	VERIFICANDO <i>MENTIONS</i>	47
7.3	ENVIANDO MENSAGEM PARA <i>TIMELINE</i>	48

7.4	ENVIO E RECEBIMENTO DE MENSAGENS	48
7.5	BLOQUEIO E DESBLOQUEIO DE UM PERFIL	49
7.6	BUSCA DE <i>HASHTAGS</i>	50
8.	CONCLUSÃO	51
8.1	TRABALHOS FUTUROS	51
	BIBLIOGRAFIA	52

1. INTRODUÇÃO

O crescimento de usuários nas redes sociais vem sendo cada vez maior, requerendo o desenvolvimento de ferramentas que trabalhem independente da iteração do usuário, para automatizar determinadas funcionalidades que a rede oferece. Deste modo, é possível realizar atualizações em seu perfil automaticamente, sem a necessidade de se preocupar em monitorar para que ela seja atualizada constantemente.

Ao criar uma conta em uma rede social, é gerado um perfil que pode ser visualizado publicamente por todos os usuários desta rede. Este perfil pode ter como finalidade realizar propagandas na rede, disseminar informações, entre outros que diferem de usuários comuns que desejam somente ter o seu perfil para manter-se conectado com seus amigos e familiares. No caso do perfil pessoal, é possível encontrar informações sobre o usuário como: nome, sobrenome, idade, sexo, onde trabalha e outras informações que o mesmo disponibiliza, e para outros tipos de perfis estas informações são preenchidas com o nome da empresa, locais onde existe, quando foi criada, dentre outras.

A grande propagação que uma rede social oferece vem sendo um lado positivo para as empresas que desejam realizar marketing ou fornecer informações dos mais variados tipos. Tendo em vista que o cadastro de contas ocorre gratuitamente, este se torna um meio de propaganda barato que abrange significativamente diversos tipos de público. Além do marketing, existem também perfis que se tornam de utilidade pública, como informação sobre o trânsito, previsão do tempo e notícias importantes. Todos estes fatores proporcionam informação ao usuário como o principal objetivo, sendo ela pessoal ou não.

O *Twitter* [Twitter, 2012], uma rede social e servidor para *microblogging*¹, permite que seus usuários enviem e recebam atualizações online de seus contatos em forma de mensagens (limitada a 140 caracteres por mensagem enviada) a partir do perfil da sua página da internet, por meio de SMS² provenientes do celular ou por programas específicos que utilizam a API³ do *Twitter*. O recebimento destas atualizações ocorre somente quando um usuário se registra a um perfil para receber notificações de novas mensagens. Este registro é nomeado como seguidor, ou seja, perfis que seguem outros perfis para se manter atualizado quando ocorrer novas mensagens, no qual é enviado somente para os seus seguidores.

Logo, para se manter informado, ter conhecimento de promoções, novidades e outros fatos, acaba-se seguindo diversos perfis. Estes perfis são diferenciados de usuários comuns, pois foram criados com o intuito de enviar mensagens que não são pessoais ou provindas de uma só pessoa. Atualmente, empresas especializadas para a criação deste tipo de conta e manutenção das mesmas, realizam todo o trabalho de marketing para fazer com que fique em constante movimentação (sempre enviado novas mensagens) e atualização, tendo como consequência um maior número de seguidores.

¹ Termo adotado para especificar um diário da internet que permite atualização rápida e breve de texto.

² Termo adotado para especificar serviço de mensagem curta, no inglês: *Short Message Service (SMS)*

³ Termo adotado para especificar interface de programação do aplicativo, no inglês: *Application Programming Interface (API)*

Para os perfis que utilizam informação dos usuários para campanhas de marketing, é inviável lidar com o grande número de informações recebidas. Para este tipo de situação, é adequado que haja uma ferramenta que auxilie nas respostas e ações que devem ser tomadas, na qual será abordada a seguir.

1.1 MOTIVAÇÃO

Ferramentas online que realizam o repasse de informações automaticamente na rede social *Twitter* não possuem os serviços necessários para os tipos de perfis citados anteriormente. A integração com outras aplicações, bloqueio de usuários que realizam *mentions*⁴ indesejados, estatísticas gerais, sugestão de novas definições para o *retweet*⁵ automático, dentre outras funções que serão abordadas neste trabalho são atualmente inexistentes em uma só ferramenta.

Hoje, existem grupos de usuários do *Twitter* que criaram, por exemplo, perfis para prover informações do trânsito avisando onde ocorre engarrafamento, acidente e outras informações. Estes usuários devem ficar constantemente verificando o seu perfil para efetuar o repasse das mensagens para os seus seguidores.

Visando melhorar estes perfis do *Twitter* para re-enviar mensagens recebidas de seus seguidores para os seus contatos, surgiu a motivação de explorar o desenvolvimento de uma aplicação que viabiliza o repasse automático dessas mensagens de acordo com regras e definições que serão descritas a seguir, mantendo sempre a informação atualizada.

1.2 OBJETIVO GERAL

Este trabalho apresenta o desenvolvimento de um sistema automatizado para realizar ações pré-determinadas de um usuário em sua conta da rede social *Twitter*, possibilitando a atualização constante sem que haja a interação do mesmo. Com o intuito de tornar uma ferramenta mais completa em relação às existentes hoje e focada nos perfis descritos anteriormente, foi realizado um estudo para verificar as necessidades atuais.

A integração com outras aplicações também foi vista, para fornecer uma ferramenta com informações adicionais que ajudam o usuário a interagir de uma forma mais real com as mensagens enviadas/recebidas. Como exemplo, a utilização do *GoogleMaps*, para mensagens que sejam mencionadas com conteúdo de endereço, estado, cidade ou país para poder visualizar no mapa do *Google* de uma forma mais clara e precisa. Esta questão será descrita nos trabalhos futuros, no qual o usuário deverá prover de uma aplicação onde possa inserir junto à mensagem enviada a sua localização atual de acordo com as coordenadas do *GPS*⁶.

⁴ Termo adotado para especificar quando um usuário mencionou o seu perfil, no inglês: *mention*

⁵ Termo adotado para especificar quando um usuário re-envia uma mensagem, no inglês: *retweet*

⁶ Termo em inglês adotado para especificar o sistema de posicionamento global, no inglês: *Global Positioning System*

1.3 OJETIVO ESPECÍFICO

Para entender os objetivos específicos, devemos nos familiarizar com o funcionamento do *Twitter*. Esta rede social dispõe como principal produto uma página da internet onde inicialmente visualizamos mensagens que foram enviadas de seus seguidores⁷, ou seja, sempre que um perfil que está sendo seguido postar uma mensagem no *Twitter*, todos os seguidores deste perfil recebe em sua *timeline*⁸ esta mensagem.

Para enviar uma mensagem como *mention*, basta colocar uma arroba seguida do nome do perfil, por exemplo, o perfil de nome FACIN é mencionado em uma mensagem de outro usuário da seguinte forma: “Mensagem exemplo na qual menciono a @FACIN”. Qualquer usuário no *Twitter* que postar uma mensagem em sua *timeline* e utilizar “@FACIN” na mensagem, o usuário FACIN poderá receber em sua seção que este usuário fez uma *mention* do seu perfil. Caso o usuário FACIN deseje que esta *mention* seja vista por todos os perfis que o seguem, o mesmo irá realizar o *retweet* desta mensagem na qual irá aparecer em sua *timeline* e todos os seus seguidores serão capazes de visualizar.

Sabendo destas funcionalidades do *Twitter*, é possível criar perfis que realizem o repasse de informações quando ocorrer uma *mention* do perfil e quando lhes convêm. Desta forma, viu-se a necessidade de automatizar uma conta que possui milhares de seguidores no qual a leitura de todos os *mentions* que ocorrem é inviável, sendo necessária uma ferramenta que realize o *retweet* automático destas informações para os seguidores do perfil.

Mediante esta realidade, torna-se evidente o uso de ferramentas capazes de tratar estes *mentions* e, de acordo com regras estabelecidas, realizar o *retweet* em sua *timeline* de mensagens verdadeiras e importantes para que todos os seus seguidores obtivessem estas informações.

Assim, os objetivos específicos podem ser enumerados da seguinte forma:

1. *Retweet* automático de acordo com regras
2. Definições de regras pelo usuário para ocorrer o *retweet*
3. Ferramenta de fácil uso
4. Integração com outras ferramentas
5. Fácil alteração de regras

1.4 METODOLOGIA

Para a construção deste projeto, foi utilizado o UP⁹, visto que este orienta no desenvolvimento de modo iterativo e incremental. Deste modo, realizou-se uma seqüência de versões executáveis a cada iteração e uma integração contínua entre as diversas versões do sistema, onde cada nova versão incorpora novos aperfeiçoamentos incrementais em relação à anterior pela incorporação do produto

⁷ Termo adotado para especificar que um perfil segue outro perfil, no inglês: *following*

⁸ Termo adotado para especificar que uma mensagem apareceu em sua tela inicial, no inglês: *timeline*

⁹ Termo adotado para especificar o Processo Unificado, no inglês: *Unified Process* [Jacobson, 1999]

resultante da última interação [Hunt, 2000]. O *UP* também utiliza o *UML*¹⁰, na qual será utilizada como ferramenta para a elaboração de todos os diagramas e artefatos.

Seguindo o modelo proposto pelo *UP*, teremos quatro fases de desenvolvimento [Hunt, 2000]:

- Conceção: análise de negócio, viabilidade e escopo do projeto;
- Elaboração: análise do problema e especificação da arquitetura do sistema;
- Construção: elaboração do sistema completo a partir da arquitetura desenvolvida;
- Transição: implantação do sistema viabilizando a utilização por usuários;

A proposta foi realizada a partir de pesquisas na internet de aplicações já existentes [RoundTeam, 2012] [RetransTwitter, 2012] [TweetAdder, 2012], leitura de artigos, revisões bibliográficas através de livros e na necessidade de uma ferramenta que possibilitasse realizar as ações descritas anteriormente.

1.5 ORGANIZAÇÃO DO TEXTO

Este relatório está dividido em 7 capítulos. No capítulo 1 é introduzido o assunto deste projeto. No capítulo 2 é apresentada uma visão geral da ferramenta. No capítulo 3 é especificado o sistema e as funcionalidades. No capítulo 4 é abordada a arquitetura do sistema. No capítulo 5 são detalhados os principais algoritmos do sistema. No capítulo 6 são descritas outras ferramentas que foram vistas para comparação com a mesma. No capítulo 7 é vista a biblioteca na qual a ferramenta utiliza para realizar as funcionalidades do *Twitter*. Por fim, no capítulo 8, é apresentada a conclusão e os trabalhos futuros.

¹⁰ Termo em inglês adotado para especificar Linguagem de Modelagem Unificada, no inglês: *Unified Modeling Language*

2. SISTEMA

2.1 VISÃO GERAL

O sistema busca facilitar ações realizadas na conta de um usuário do *Twitter* de maneira automática de acordo com regras definidas. Estas ações e regras, que serão abordadas detalhadamente nos tópicos a seguir, são:

- *Retweets* de *mentions* que estejam de acordo com as regras;
- Definição de regras;
- Envio de mensagens na *timeline* de acordo com o definido;
- Bloqueio de usuários;
- Estatísticas gerais;
- Sugestão de novas regras;
- Tratamento de mensagens privadas de perfis diferenciados;

Para habilitar tais serviços em seu perfil do *Twitter*, o usuário deve criar uma conta na aplicação desenvolvida para que possa fornecer os dados da conta o *Twitter* possibilitando a integração entre os dois sistemas. Esta é realizada a partir da *API Twitter4J*, na qual oferece segurança, estabilidade e todas as ações que podem ser realizadas no perfil, como o envio de mensagens em sua *timeline*, leitura dos *mentions*, bloqueio de usuários para não receber mais *mentions* do mesmo e outras funções necessárias para o funcionamento adequado da aplicação.

Abaixo um breve detalhamento de como ocorrem os *mentions*, *hashtags* e frases específicas, para podermos entender melhor como estas regras são tratadas e definidas a seguir nas funcionalidades da ferramenta proposta.

2.1.1 MENTIONS

Ao registrar como regra o *mention* de um determinado perfil, quaisquer *mention* que seja feito para este perfil será realizado o *retweet* em sua conta cadastrada. Exemplo:

1. Cadastra-se o perfil “@Teste” na ferramenta;
2. Adiciona uma regra de *mention* para o perfil: “@Perfila”;
3. Perfil “@Outro” envia uma mensagem para a sua *timeline*: “Teste de mensagem. @Perfila”;
4. Ferramenta verifica que o perfil “@Outro” fez uma *mention* do “@Perfila”;
5. Ferramenta realiza o *retweet* da mensagem;
6. Perfil “@Teste” terá a mensagem “Teste de mensagem. @Perfila” em sua *timeline*.

2.1.2 HASHTAGS

Ao registrar como regra um *hashtag*, quaisquer *hashtag* que seja feito será realizado o *retweet* em sua conta cadastrada. Exemplo:

1. Cadastra-se o perfil “@Teste” na ferramenta;
2. Adiciona uma regra de *hashtag*: “#hashtag”;
3. Perfil “@Outro” envia uma mensagem para a sua *timeline*: “Teste de mensagem. #hashtag”;
4. Ferramenta verifica que o perfil “@Outro” mencionou a *hashtag* “#hashtag”;
5. Ferramenta realiza o *retweet* da mensagem;
6. Perfil “@Teste” terá a mensagem “Teste de mensagem. #hashtag” em sua *timeline*.

2.1.3 FRASES ESPECÍFICAS

Ao registrar como regra uma frase específica para um determinado perfil ou uma lista de perfis, caso esta frase seja enviada para a *timeline* deste perfil, será realizado o *retweet* em sua conta cadastrada. Exemplo:

1. Cadastra-se o perfil “@Teste” na ferramenta;
2. Adiciona uma frase específica para o perfil “@Outro”: “Teste de frase”;
3. Perfil “@Outro” envia uma mensagem para a sua *timeline*: “Teste de frase”;
4. Ferramenta verifica que o perfil “@Outro” mencionou a frase “Teste de frase”;
5. Ferramenta realiza o *retweet* da mensagem;
6. Perfil “@Teste” terá a mensagem “Teste de frase” em sua *timeline*.

2.2 RETWEET DE MENTIONS E HASHTAGS

A partir da verificação automática de *mentions* recebidos de outros usuários, ocorre o *retweet* ou não em sua *timeline* de acordo com regras definidas. Tomando como exemplo a situação atual do trânsito em Porto Alegre, um usuário do *Twitter* que segue o perfil fictício @TransitoPOA deseja compartilhar a informação de que na rua Anita Garibaldi com a Dom Pedro II o sinal de trânsito está queimado, ocorrendo congestionamento no local. Deste modo, este usuário irá enviar uma mensagem para o perfil @TransitoPOA da seguinte maneira: “Congestionamento na Anita Garibaldi com a Dom Pedro II, sinaleira queimada. @TransitoPOA”.

Pelo fato do perfil @TransitoPOA ter uma conta no sistema proposto ocorrerá uma verificação de usuários que o mencionaram (*mentions*) e o mesmo irá receber esta mensagem, realizando uma análise em suas regras de acordo com o recebido para averiguar se deve ocorrer o *retweet*. Ocorrendo o *retweet*, todos os perfis que seguem o @TransitoPOA receberam esta mensagem, e a partir de então, saberão que há trânsito no local informado.

Para o caso do *hashtag*, esta funciona da mesma maneira que um *mention*, porém, ao invés do nome na mensagem enviada para a *timeline* ser com uma arroba na frente, está é enviada com um sustenido. Desta forma, para o exemplo acima, o @TransitoPOA seria escrito como #TransitoPOA.

2.3 DEFINIÇÃO DE REGRAS

Para o funcionamento adequado das regras definidas, é utilizado a *stop list* [Van Rijsbergen, 1975]. Uma *stop list* é uma lista de palavras negativas na qual utilizamos para verificar quando uma palavra está contida na mensagem a qual queremos examinar. Nos casos das regras que serão utilizadas, teremos dois tipos de *stop list*: destinado a palavras (aceitas e negadas) e destinados a usuários (diferenciado e bloqueado). Separando por seções, estas serão divididas em quatro, sendo elas:

- Palavras aceitas;
- Palavras negadas;
- Perfil diferenciado;
- Perfil bloqueado;

Assim, o conjunto de palavras aceitas irá conter as palavras que serviram como regra para ocorrer o *retweet* de um *mention* verificado. Deste modo, ao realizar a verificação de uma mensagem, caso haja uma ou mais palavras nesta regra e não haja nenhuma palavra negada ou um perfil bloqueado, esta mensagem será re-enviada em sua *timeline*.

O conjunto de palavras negadas irá servir como palavra chave para não ocorrer o *retweet* de um *mention*. Neste caso, sempre que uma mensagem verificada conter uma ou mais palavras negadas, não ocorrerá o *retweet*.

O conjunto de perfis diferenciados são perfis ditos como administradores do sistema. Estes serão capazes de alterar regras remotamente a partir de outro perfil pelo envio de mensagens privadas, podendo realizar a adição ou remoção de palavras aceitas ou negadas e de perfis bloqueados.

Por fim, o conjunto de perfis bloqueados são usuários do *Twitter* que por algum motivo foram adicionados a este para que independente do *mention* realizado não seja tomada nenhuma ação. Esta regra é útil para casos de propagandas indevidas que ocorrem com frequência nas redes sociais.

Abaixo, um conjunto inicial de palavras aceitas e palavras negadas foi definido para exemplificar como funciona a ferramenta conforme a definição acima (o algoritmo responsável pelas verificações da exemplificação abaixo será vista no capítulo: 5. Principais Algoritmos do Sistema).

- Palavras aceitas
 - Trânsito
 - Congestionamento
 - Acidente
 - Trancado
- Palavras negadas
 - Celular
 - Festa
 - Animação
 - Rápido

A partir destes conjuntos podemos analisar a exemplificação abaixo na Tabela 1 – Análise de frases as frases que foram feitas *retweet* ou não e o seu motivo (no Capítulo 5 – Principais Algoritmos, é apresentada em detalhe como é realizada esta verificação).

Frase	Aceita	Motivo
Acidente na Anita Garibaldi com a Perimetral	Sim	Contém a palavra aceita: Acidente.
Tudo trancado ... Vou falar no celular!	Não	Apesar de conter a palavra “trancado”, a frase contém a palavra negada: celular.
A festa na Padre Chagas está muito boa!!!	Não	Contém a palavra negada: festa.
Esta tudo trancado na Freeway indo para a praia	Sim	Contém a palavra aceita: trancado
O fim de semana na praia estava muito bom	Não	Não contém nenhuma palavra aceita.

Tabela 1 - Análise de frases

2.4 ENVIO AUTOMÁTICO DE MENSAGENS

O envio de mensagens automático possui a finalidade de informar ou até mesmo realizar propagandas. Para o exemplo do trânsito citado anteriormente, podemos definir algumas mensagens como, por exemplo, “Use o cinto de segurança” para enviar todas as segundas e sextas feiras às sete horas da tarde. Há também a possibilidade da integração com outros serviços de trânsito no qual a partir de um determinado fato, envia uma mensagem específica.

Nota-se que o envio automático de mensagens e as integrações disponíveis para o usuário, deverão ser ativados pelo mesmo. Para as integrações existentes, sempre existiram novas ferramentas e formas de integrar com a ferramenta existente. Logo, as integrações serão desenvolvidas sempre que surgir novas necessidades para assim abranger diversos tipos de perfis.

2.5 BLOQUEIO DE USUÁRIOS

O bloqueio de usuários é realizado somente por um perfil diferenciado. A partir do envio de uma mensagem privada, é informada a palavra chave “*block*” seguido do nome do perfil que se deseja bloquear. Assim, este perfil não terá o seu *mention* tratado. Isto significa que não irá ocorrer o *retweet* de quaisquer mensagens enviada para o perfil, mesmo que esta esteja de acordo com as regras de aceitação. Ao realizar um bloqueio, este será permanente até que seja removido desta lista, enviando uma mensagem privada com a palavra chave “*unblock*” seguido do nome do perfil.

2.6 ESTATÍSTICAS

Todas as ações do sistema são armazenadas em um banco de dados para gerar estatísticas que possam a ser utilizadas para novas definições ou alterações de regras existentes e para fins de controle. Abaixo, uma lista destas estatísticas que serão tratadas:

- Total de *mentions* tratados;
- Total de *mentions* aceites;
- Total de *mentions* negados;
- Total de usuários bloqueados;

2.7 SUGESTÃO DE NOVAS REGRAS

Tendo como base as estatísticas realizadas, serão sugeridas para palavras aceites palavras que venham a ser citadas com freqüência nos *mentions* tratados. Uma vez ocorrida a sugestão de uma determinada palavra, esta não será mais sugerida depois que ocorrer a verificação por parte do usuário, removendo a mesma da lista de sugestão.

2.8 TRATAMENTO DE MENSAGENS PRIVADAS

Usuários ditos como administradores são adicionados na lista de perfis diferenciados. Estes podem enviar mensagens privadas para o perfil realizando alterações de definições de regras e ajustes nas configurações do sistema, tais como:

- Adição ou remoção de uma nova palavra aceite;
- Adição ou remoção de uma nova palavra negada;
- Adição ou remoção de um usuário bloqueado;

Desta forma, estará sempre atualizado por não só um usuário, e sim, por um grupo de usuários que visa manter atualizadas as regras, mantendo as informações que devem ser passadas a todos os perfis que o seguem corretas.

3. ANÁLISE E IDENTIFICAÇÃO DE REQUISITOS

Para compreender as atividades realizadas pela ferramenta que visa à automatização em uma rede social, e seguir o modelo proposto em UML, serão definidos a seguir os requisitos, possibilitando a compreensão do que o usuário terá ao seu dispor com a ferramenta. Serão abordados também os equipamentos e tecnologias utilizadas para o funcionamento da mesma.

3.1 ESCOPO

O sistema desenvolvido possui três atores, sendo eles: usuário da ferramenta, cliente final e administrador da ferramenta. O usuário da ferramenta é responsável pela criação das regras e da ativação do serviço na qual a mesma disponibiliza, iniciando ou parando o mesmo conforme necessidade. Para isto, deve ocorrer o preenchimento de diversas regras que serão tratadas a seguir para que ocorra o funcionamento desejado em seu perfil do *Twitter*.

O cliente final é um usuário que busca e precisa de informações a partir de um perfil do *Twitter*. Este irá acessar a página da internet do *Twitter* em que irá conter todas as informações que serão enviadas pela ferramenta automaticamente conforme os serviços ativados que foram definidos para este perfil.

Para realizar modificações de definições da ferramenta sem precisar acessar a página da internet da ferramenta, existe a funcionalidade de enviar uma mensagem privada ao perfil do *Twitter* que será tratada de forma adequada e realizará as alterações recebidas. Para ocorrer tal cenário, o usuário irá definir nas regras da ferramenta uma lista de cliente final que serão tratados como administradores. Desta maneira, quando um cliente final que esteja na lista de administrador enviar uma mensagem privada para o perfil, ocorrerá um tratamento adequado para realizar as mudanças recebidas.

Um exemplo desta funcionalidade é o envio de uma nova palavra para que seja adicionada à *stop list* de palavras negadas. Ao adicioná-la, novos *mentions* que forem tratados pela ferramenta não irão realizar o *retweet* em mensagens que contem tal palavra.

Na Figura 1 – Cenário Reduzido abaixo, esta representado o cenário reduzido de como ocorre à relação entre os atores, a ferramenta e o *Twitter*.

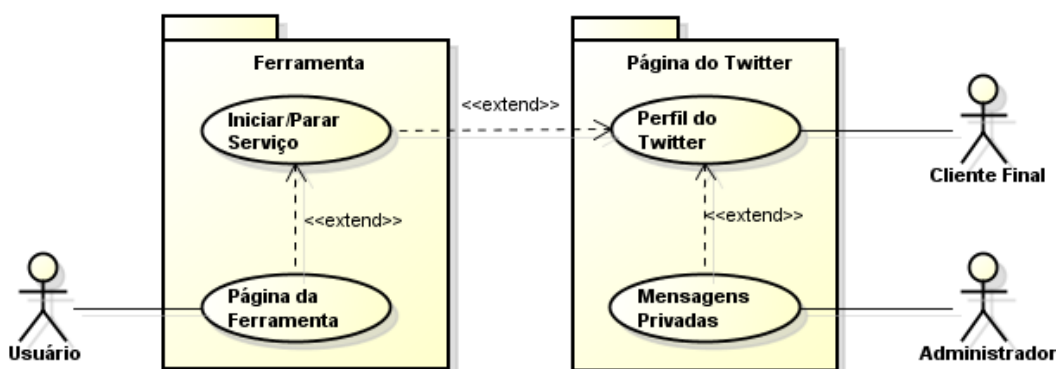


Figura 1 - Cenário Reduzido

Como podemos ver, o usuário da ferramenta é o único ator no qual pode realizar modificações diretamente na ferramenta de como ocorre o *retweet* tem seu perfil do *Twitter* cadastrado, adição e remoção de usuários administradores e todas as funcionalidades que a ferramenta dispõe. Já o cliente final, a partir da ativação do serviço pelo usuário, poderá usufruir das informações contidas na *timeline* deste perfil, e enviar *mentions* para que este seja tratado e, se estiver de acordo com as regras definidas, ocorrer o *retweet* para a *timeline* do perfil. O administrador pode alterar por mensagem privada algumas regras, nas quais serão detalhadas posteriormente.

Na Figura 2 – Cenário Total abaixo, podemos verificar de uma maneira mais completa o papel de cada ator no cenário total.

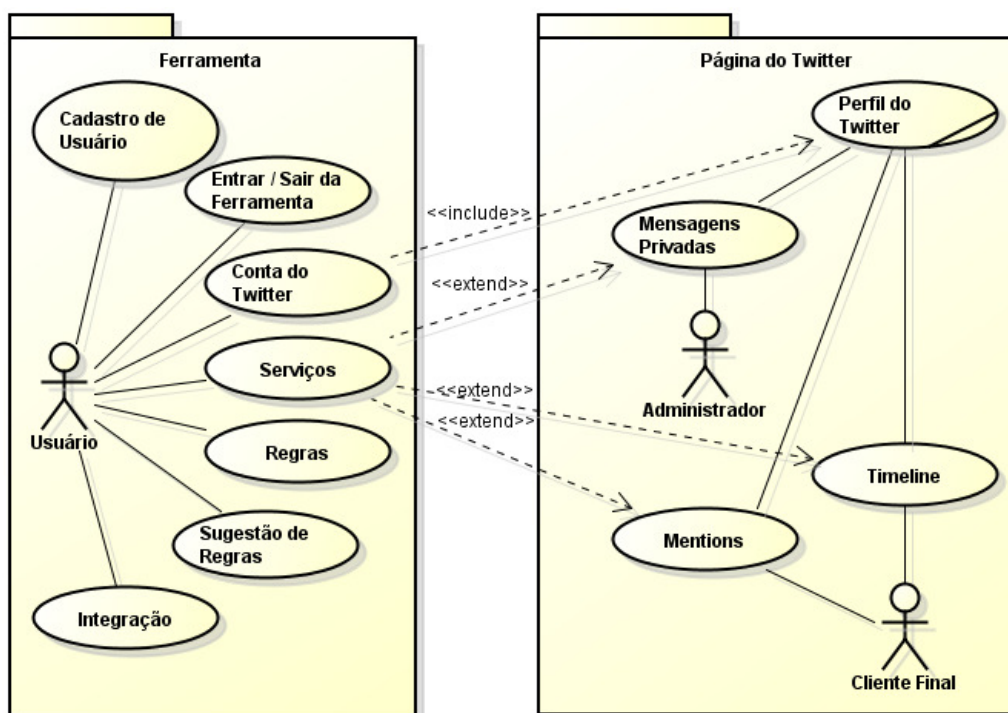


Figura 2 - Cenário Total

3.2 REQUISITOS FUNCIONAIS

Os requisitos funcionais referem-se aos requisitos que estão relacionados com a maneira que o sistema deve operar, onde se especificam as entradas e saídas do sistema e o relacionamento comportamental entre elas, assim como a iteração com o usuário. Desta forma, os requisitos encontrados estão descritos abaixo na Tabela 2 – Requisitos Funcionais.

ID	Requisito	Descrição
RF01	Cadastro de Usuário.	Permite o registro do usuário no sistema.
RF02	Modificação de Cadastro de Usuário.	Permite a modificação das informações do usuário do sistema (nome, email, senha...)
RF03	Exclusão do Cadastro	Permite a exclusão do usuário do sistema.

	de Usuário.	
RF04	Entrar na Ferramenta	Permite o usuário a entrar na ferramenta fornecendo o seu endereço eletrônico e senha.
RF05	Sair da Ferramenta	Permite o usuário sair da sessão atual a qual entrou na ferramenta pela página da internet.
RF06	Inclusão de uma Conta do <i>Twitter</i>	Permite o usuário a vincular a sua conta do <i>Twitter</i> para habilitar os serviços na mesma.
RF07	Exclusão de uma Conta do <i>Twitter</i>	Permite o usuário a desvincular a sua conta do <i>Twitter</i> e desabilita os serviços na mesma.
RF08	Ativação dos Serviços	Permite o usuário a ativar os serviços da ferramenta em determinada conta cadastrada do <i>Twitter</i> .
RF09	Desativação dos Serviços	Permite o usuário a desativar os serviços da ferramenta em determinada conta cadastrada do <i>Twitter</i> .
RF10	Inserção de Regras	Permite incluir regras nas configurações da conta de seu usuário.
RF11	Modificação de Regras	Permite modificar regras nas configurações da conta de seu usuário.
RF12	Exclusão de Regras	Permite excluir regras nas configurações da conta de seu usuário.
RF13	Consulta de Regras	Permite consultar as regras existentes na conta de seu usuário.
RF14	Consulta de Regras Sugeridas	Permite consultar regras sugeridas para a conta de seu usuário.
RF15	Habilitar Integração	Permite habilitar a integração da conta do usuário com outras aplicações terceirizadas.
RF16	Desabilitar Integração	Permite desabilitar a integração da conta do usuário com outras aplicações terceirizadas.
RF17	Consultar Integrações Disponíveis	Permite consultar as integrações disponibilizadas pela ferramenta.

Tabela 2 - Requisitos Funcionais

3.3 REQUISITOS NÃO FUNCIONAIS

Os requisitos não funcionais são aqueles que não são especificamente relacionados com a funcionalidade do sistema. Eles impõem restrições na ferramenta a ser desenvolvida e/ou processo de desenvolvimento do sistema como também especificam restrições externas as quais a ferramenta precisa atender. Exemplos destes requisitos são: segurança, confiabilidade, usabilidade, performance, entre outros. Deste modo, os requisitos não funcionais encontrados para o primeiro ciclo do projeto são:

3.3.1 USABILIDADES

ID	Descrição
RNF01	Facilidade de Registro: o usuário da ferramenta deverá ter facilidade de realizar o registro no sistema fornecendo o mínimo de dados necessário para realizar a criação de uma conta, tais como: nome, email e senha.
RNF02	Facilidade de Uso: o usuário da ferramenta deverá ter a facilidade de navegar pelo sistema de maneira clara e simples de modo que fiquem visíveis sempre as funcionalidades mais importantes para o funcionamento da conta.
RNF03	Interface Web: o usuário utilizará a ferramenta através de uma web <i>browser</i> que deve estar

	disponível <i>online</i> a partir de uma <i>URL</i> .
--	---

Tabela 3 - Usabilidades

3.3.2 LICENÇAS

ID	Descrição
RNF04	Biblioteca do <i>Twitter</i>: a ferramenta irá utilizar a biblioteca <i>Twitter4J</i> para realizar as requisições do servidor da conta do usuário do <i>Twitter</i> cadastrada. Esta biblioteca é uma das sugeridas pelo <i>Twitter</i> , na qual é pública e pode ser utilizada livremente.
RNF05	<i>TinyURL</i>: a ferramenta irá utilizar a <i>TinyURL</i> para compactar <i>URL</i> e realizar o redirecionamento para outros página da internet. Este é um serviço de graça que pode ser utilizado livremente.

Tabela 4 - Licenças

3.3.3 DISPONIBILIDADE

ID	Descrição
RNF06	Ferramenta <i>Online</i>: a ferramenta deverá estar sempre disponível. Caso ocorra alguma interrupção, deverá ser restaurado o mais rápido possível.
RNF07	Servidor de Hospedagem: a ferramenta será hospedada em um servidor de página da internet que deverá ser definido posteriormente.
RNF08	Serviço Automático: o sistema terá um serviço que será ativado pelo usuário onde ficará ativo no servidor de hospedagem da aplicação.

Tabela 5 - Disponibilidade

3.3.4 SEGURANÇA

ID	Descrição
RNF09	Integração com <i>Twitter</i>: como a ferramenta será via Web, a mesma deverá ser a mais segura possível. Desta forma, será disponibilizado o acesso integrado ao <i>Twitter</i> , no qual o usuário irá utilizando sua conta do <i>Twitter</i> para entrar na ferramenta e adicionar novas contas. Esta funcionalidade é disponibilizada pelo <i>Twitter</i> de forma gratuita.

Tabela 6 - Segurança

3.4 DETALHAMENTO DOS CASOS DE USO

3.4.1 CADASTRO DE USUÁRIO

Ponto de ativação: Este caso de uso começa quando um usuário entra na página da internet e seleciona o cadastro de novo usuário para se registrar na ferramenta.

Ator: Usuário.

Objetivo: Realizar o cadastro de uma nova conta na ferramenta.

Pré-condição: Ferramenta estar disponível.

Pós-condições: Registro realizado com sucesso.

Fluxo normal:

1. O usuário acessa a página da internet da ferramenta
2. O usuário pressiona o registro de novo usuário
3. O usuário informa o seu endereço eletrônico
4. O usuário informa a sua senha
5. O usuário redigita a sua senha para confirmação
6. O usuário pressiona o registro de usuário
7. O sistema processa o registro do usuário
8. Redireciona para a inclusão de uma conta do *Twitter* (RF06)

Fluxo alternativo:

1. O usuário acessa a página da internet da ferramenta
2. O usuário pressiona o registro de novo usuário
3. O usuário informa o seu endereço eletrônico
4. O usuário informa a sua senha
5. O usuário redigita a sua senha para confirmação
6. O usuário pressiona o registro de usuário
7. O sistema processa o registro do usuário
8. Exibe a mensagem de endereço eletrônico já cadastrado

Requisito funcional: RF01

Requisito não funcional: RNF01, RFN03 e RFN06

3.4.2 MODIFICAÇÃO DE CADASTRO DE USUÁRIO

Ponto de ativação: Este caso de uso começa quando o usuário deseja modificar informações de sua conta.

Ator: Usuário.

Objetivo: Alterar informações da conta.

Pré-condição: Estar conectado na ferramenta.

Pós-condições: Realizar alterações com sucesso.

Fluxo normal:

1. Usuário acessa as suas informações
2. Usuário pressiona editar
3. Usuário altera as informações desejadas
4. Usuário pressiona salvar
5. O sistema salva as alterações

Fluxo alternativo:

1. Usuário acessa as suas informações
2. Usuário pressiona editar
3. Usuário altera as informações desejadas
4. Usuário pressiona salvar
5. O sistema exibe a mensagem de erro

Requisito funcional: RF02

Requisito não funcional: RNF02 e RNF03

3.4.3 EXCLUSÃO DO CADASTRO DO USUÁRIO

Ponto de ativação: Este caso de uso começa quando o usuário deseja excluir a sua conta do banco de dados da ferramenta.

Ator: Usuário

Objetivo: Excluir a conta do usuário nos registros de dados.

Pré-condição: Estar conectado na ferramenta.

Pós-condições: Excluir todas as informações do usuário do banco de dados.

Fluxo normal:

1. Usuário acessa as suas informações
2. Usuário pressiona excluir conta
3. O sistema requisita a senha atual do usuário para confirmação
4. O sistema exclui do banco de dados a conta do usuário
5. Redireciona para a inclusão de uma conta do *Twitter* (RF06)

Fluxo alternativo:

1. Usuário acessa as suas informações
2. Usuário pressiona excluir conta
3. O sistema requisita a senha atual do usuário para confirmação
4. O sistema exibe mensagem de erro caso senha seja inválida

Requisito funcional: RF03

Requisito não funcional: RNF03

3.4.4 ENTRAR NA FERRAMENTA

Ponto de ativação: Este caso de uso começa quando o usuário deseja entrar na sua conta cadastrada na ferramenta.

Ator: Usuário.

Objetivo: Acessar a conta do usuário cadastrada na ferramenta.

Pré-condição: Ferramenta estar disponível.

Pós-condições: Acessar a conta do usuário na ferramenta.

Fluxo normal:

1. Usuário acessa a página da internet da ferramenta
2. Usuário digita o seu endereço eletrônico e senha no campo de acesso
3. O sistema valida as informações
4. Redireciona para a página da internet inicial do usuário

Fluxo alternativo:

1. Usuário acessa a página da internet da ferramenta
2. Usuário digita o seu endereço eletrônico e senha no campo de acesso
3. O sistema exibe a mensagem de usuário ou senha inválido

Requisito funcional: RF04

Requisito não funcional: RNF03, RNF06 e RNF09

3.4.5 SAIR DA FERRAMENTA

Ponto de ativação: Este caso de uso começa quando o usuário deseja finalizar a sua sessão.

Ator: Usuário.

Objetivo: Finalizar a sessão do usuário na ferramenta.

Pré-condição: Estar conectado na ferramenta.

Pós-condições: Finalizar a sessão da ferramenta.

Fluxo normal:

1. Usuário pressiona sair
2. O sistema finaliza a sessão do usuário
3. Redireciona para a página da internet inicial

Fluxo alternativo:

1. Usuário pressiona sair
2. O sistema exibe a mensagem de erro

Requisito funcional: RF05

Requisito não funcional: RNF03 e RNF06

3.4.6 INCLUSÃO DE UMA CONTA DO TWITTER

Ponto de ativação: Este caso de uso começa quando o usuário deseja vincular o seu perfil do *Twitter* com a ferramenta.

Ator: Usuário

Objetivo: Vincular o perfil do *Twitter* do usuário com a sua conta da ferramenta.

Pré-condição: Estar conectado na ferramenta.

Pós-condições: Realizar a vinculação do perfil do *Twitter* com sucesso.

Fluxo normal:

1. Usuário pressionar vincular novo perfil do *Twitter*
2. Usuário fornece os dados do perfil do *Twitter*
3. O sistema verifica os dados informados
4. O sistema salva os dados necessários
5. Redireciona para inserção de regras para este perfil (RF10)

Fluxo alternativo:

1. Usuário pressionar vincular novo perfil do *Twitter*
2. Usuário fornece os dados do perfil do *Twitter*
3. O sistema exibe mensagem de usuário ou senha inválido

Requisito funcional: RF06

Requisito não funcional: RNF02, RNF03, RNF04, RNF06 e RNF09

3.4.7 EXCLUSÃO DE UMA CONTA DO TWITTER

Ponto de ativação: Este caso de uso começa quando o usuário deseja desvincular o seu perfil do *Twitter* com a ferramenta.

Ator: Usuário

Objetivo: Desvincular o perfil do *Twitter* do usuário com a sua conta da ferramenta.

Pré-condição: Estar conectado na ferramenta.

Pós-condições: Realizar a desvinculação do perfil do *Twitter* com sucesso.

Fluxo normal:

1. Usuário seleciona o perfil do *Twitter*
2. Usuário pressiona desvincular o perfil do *Twitter*
3. O sistema requisita a senha do perfil para confirmação
4. O sistema verifica os dados informados
5. O sistema desvincula o perfil do *Twitter*
6. Redireciona para a página da internet inicial

Fluxo alternativo:

1. Usuário seleciona o perfil do *Twitter*
2. Usuário pressiona desvincular o perfil do *Twitter*
3. O sistema requisita a senha do perfil para confirmação
4. O sistema verifica os dados informados
5. O sistema exibe mensagem de usuário ou senha inválido

Requisito funcional: RF07

Requisito não funcional: RNF02, RNF03, RNF04, RNF06 e RNF09

3.4.8 ATIVAÇÃO DOS SERVIÇOS

Ponto de ativação: Este caso de uso começa quando o usuário deseja ativar os serviços que a ferramenta oferece para o *retweet* automático em seu perfil específico do *Twitter*.

Ator: Usuário.

Objetivo: Ativar os serviços no perfil específico do *Twitter*.

Pré-condição: Estar conectado na ferramenta.

Pós-condições: Ativar os serviços automáticos no perfil específico do *Twitter*.

Fluxo normal:

1. Usuário entra no perfil vinculado do *Twitter* desejado
2. Usuário pressiona ativar funções automáticas
3. O sistema inicia o serviço automático para o perfil do *Twitter*
4. O sistema informa que foi ativado com sucesso

Fluxo alternativo:

1. Usuário entra no perfil vinculado do *Twitter* desejado
2. Usuário pressiona ativar funções automáticas
3. O sistema exibe mensagem de erro

Requisito funcional: RF08

Requisito não funcional: RNF08

3.4.9 DESATIVAÇÃO DOS SERVIÇOS

Ponto de ativação: Este caso de uso começa quando o usuário deseja desativar os serviços que a ferramenta oferece para o *retweet* automático em seu perfil específico do *Twitter*.

Ator: Usuário.

Objetivo: Desativar os serviços no perfil específico do *Twitter*.

Pré-condição: Estar conectado na ferramenta.

Pós-condições: Desativar os serviços automáticos no perfil específico do *Twitter*.

Fluxo normal:

1. Usuário entra no perfil vinculado do *Twitter* desejado
2. Usuário pressiona desativar funções automáticas
3. O sistema para o serviço automático para o perfil do *Twitter*
4. O sistema informa que foi desativado com sucesso

Fluxo alternativo:

1. Usuário entra no perfil vinculado do *Twitter* desejado
2. Usuário pressiona desativar funções automáticas
3. O sistema exibe mensagem de erro

Requisito funcional: RF09

3.4.10 INSERÇÃO DE REGRAS

Ponto de ativação: Este caso de uso começa quando o usuário deseja inserir novas regras para o *retweet* automático do seu perfil específico do *Twitter*.

Ator: Usuário

Objetivo: Definir regras para o *retweet* automático do perfil específico do *Twitter*.

Pré-condição: Estar conectado na ferramenta.

Pós-condições: Salvar as regras inseridas para o perfil específico do *Twitter*.

Fluxo normal:

1. Usuário entra no perfil vinculado do *Twitter* desejado
2. Usuário entra nas regras deste perfil
3. Usuário seleciona o tipo de regra a ser criada
4. Usuário especifica detalhes da regra a ser criada
5. Usuário pressiona adicionar regra
6. O sistema salva a regra adicionada

Fluxo alternativo:

1. Usuário entra no perfil vinculado do *Twitter* desejado
2. Usuário entra nas regras deste perfil
3. Usuário seleciona o tipo de regra a ser criada
4. Usuário especifica detalhes da regra a ser criada
5. Usuário pressiona adicionar regra
6. O sistema exibe mensagem de erro

Requisito funcional: RF10

Requisito não funcional: RNF02 e RNF08

3.4.11 MODIFICAÇÃO DE REGRAS

Ponto de ativação: Este caso de uso começa quando o usuário deseja modificar as regras para o *retweet* automático do seu perfil específico do *Twitter*.

Ator: Usuário

Objetivo: Modificar regras para o *retweet* automático do perfil específico do *Twitter*.

Pré-condição: Estar conectado na ferramenta.

Pós-condições: Salvar as regras modificadas para o perfil específico do *Twitter*.

Fluxo normal:

1. Usuário entra no perfil vinculado do *Twitter* desejado
2. Usuário entra nas regras deste perfil
3. Usuário seleciona a regra desejada
4. Usuário pressiona modificar regra
5. Usuário especifica detalhes da regra a ser modificada

6. Usuário pressiona salvar regra
7. O sistema modifica a regra

Fluxo alternativo:

1. Usuário entra no perfil vinculado do *Twitter* desejado
2. Usuário entra nas regras deste perfil
3. Usuário seleciona a regra desejada
4. Usuário pressiona modificar regra
5. Usuário especifica detalhes da regra a ser modificada
6. Usuário pressiona salvar regra
7. O sistema exibe mensagem de erro

Requisito funcional: RF11

Requisito não funcional: RNF02 e RNF08

3.4.12 EXCLUSÃO DE REGRAS

Ponto de ativação: Este caso de uso começa quando o usuário deseja excluir as regras para o *retweet* automático do seu perfil específico do *Twitter*.

Ator: Usuário

Objetivo: Excluir regras para o *retweet* automático do perfil específico do *Twitter*.

Pré-condição: Estar conectado na ferramenta.

Pós-condições: Excluir as regras para o perfil específico do *Twitter*.

Fluxo normal:

1. Usuário entra no perfil vinculado do *Twitter* desejado
2. Usuário entra nas regras deste perfil
3. Usuário seleciona a regra desejada
4. Usuário pressiona excluir regra
5. O sistema exibe mensagem de confirmação de exclusão
6. Usuário confirma a exclusão da regra
7. O sistema exclui a regra

Fluxo alternativo:

1. Usuário entra no perfil vinculado do *Twitter* desejado
2. Usuário entra nas regras deste perfil
3. Usuário seleciona a regra desejada
4. Usuário pressiona excluir regra

5. O sistema exibe mensagem de confirmação de exclusão
6. Usuário confirma a exclusão da regra
7. O sistema exibe mensagem de erro

Requisito funcional: RF12

Requisito não funcional: RNF02 e RNF08

3.4.13 CONSULTA DE REGRAS

Ponto de ativação: Este caso de uso começa quando o usuário deseja visualizar as regras para o *retweet* automático do seu perfil específico do *Twitter*.

Ator: Usuário

Objetivo: Visualizar regras para o *retweet* automático do perfil específico do *Twitter*.

Pré-condição: Estar conectado na ferramenta.

Pós-condições: Visualizar as regras para o perfil específico do *Twitter*.

Fluxo normal:

1. Usuário entra no perfil vinculado do *Twitter* desejado
2. Usuário entra nas regras deste perfil
3. O sistema exibe as regras deste perfil

Fluxo alternativo:

1. Usuário entra no perfil vinculado do *Twitter* desejado
2. Usuário entra nas regras deste perfil
3. O sistema exibe mensagem de erro

Requisito funcional: RF13

Requisito não funcional: RNF02 e RNF08

3.4.14 CONSULTA DE REGRAS SUGERIDAS

Ponto de ativação: Este caso de uso começa quando o usuário deseja visualizar as regras sugeridas para o *retweet* automático do seu perfil específico do *Twitter*.

Ator: Usuário

Objetivo: Visualizar regras sugeridas para o *retweet* automático do perfil específico do *Twitter*.

Pré-condição: Estar conectado na ferramenta.

Pós-condições: Visualizar as regras sugeridas para o perfil específico do *Twitter*.

Fluxo normal:

1. Usuário entra no perfil vinculado do *Twitter* desejado
2. Usuário entra nas regras deste perfil
3. Usuário pressiona regras sugeridas
4. O sistema exibe as regras sugeridas deste perfil

Fluxo alternativo:

1. Usuário entra no perfil vinculado do *Twitter* desejado
2. Usuário entra nas regras deste perfil
3. Usuário pressiona regras sugeridas
4. O sistema exibe mensagem de erro

Requisito funcional: RF14

Requisito não funcional: RNF02 e RNF08

3.4.15 HABILITAR INTEGRAÇÃO COM OUTRA APLICAÇÃO

Ponto de ativação: Este caso de uso começa quando o usuário deseja habilitar a integração de um perfil específico do *Twitter* com outra aplicação disponibilizada pela ferramenta.

Ator: Usuário

Objetivo: Habilitar integração com outra aplicação.

Pré-condição: Estar conectado na ferramenta.

Pós-condições: Ativar o serviço de integração disponível com outra aplicação para o perfil específico do *Twitter*.

Fluxo normal:

1. Usuário entra no perfil vinculado do *Twitter* desejado
2. Usuário entra na integração de aplicações
3. O sistema exibe uma lista de aplicações de integração disponível
4. Usuário selecionar a aplicação de integração que deseja ativar
5. O sistema ativa o serviço de integração disponível
6. O sistema exibe uma confirmação que o serviço de integração foi ativado

Fluxo alternativo:

1. Usuário entra no perfil vinculado do *Twitter* desejado
2. Usuário entra na integração de aplicações

3. O sistema exibe uma lista de aplicações de integração disponível
4. Usuário selecionar a aplicação de integração que deseja ativar
5. O sistema exibe uma mensagem de erro

Requisito funcional: RF15

Requisito não funcional: RNF02, RNF05 e RNF08

3.4.16 DESABILITAR INTEGRAÇÃO COM OUTRA APLICAÇÃO

Ponto de ativação: Este caso de uso começa quando o usuário deseja desabilitar a integração de um perfil específico do *Twitter* com outra aplicação disponibilizada pela ferramenta.

Ator: Usuário

Objetivo: Desabilitar integração com outra aplicação.

Pré-condição: Estar conectado na ferramenta.

Pós-condições: Desativar o serviço de integração disponível com outra aplicação para o perfil específico do *Twitter*.

Fluxo normal:

1. Usuário entra no perfil vinculado do *Twitter* desejado
2. Usuário entra na integração de aplicações
3. O sistema exibe uma lista de aplicações de integração disponível
4. Usuário selecionar a aplicação de integração que deseja desativar
5. O sistema desativa o serviço de integração disponível
6. O sistema exibe uma confirmação que o serviço de integração foi desativado

Fluxo alternativo:

1. Usuário entra no perfil vinculado do *Twitter* desejado
2. Usuário entra na integração de aplicações
3. O sistema exibe uma lista de aplicações de integração disponível
4. Usuário selecionar a aplicação de integração que deseja desativar
5. O sistema exibe uma mensagem de erro

Requisito funcional: RF16

Requisito não funcional: RNF02, RNF05 e RNF08

3.4.17 CONSULTAR INTEGRAÇÕES DISPONÍVEIS

Ponto de ativação: Este caso de uso começa quando o usuário deseja visualizar as integrações disponíveis para um perfil específico do *Twitter* na qual é disponibilizada pela ferramenta.

Ator: Usuário

Objetivo: Visualizar integrações disponíveis com outras aplicações.

Pré-condição: Estar conectado na ferramenta.

Pós-condições: Exibir as integrações disponíveis para o perfil específico do *Twitter*.

Fluxo normal:

1. Usuário entra no perfil vinculado do *Twitter* desejado
2. Usuário entra na integração de aplicações
3. O sistema exibe uma lista de aplicações de integração disponível

Fluxo alternativo:

1. Usuário entra no perfil vinculado do *Twitter* desejado
2. Usuário entra na integração de aplicações
3. O sistema exibe uma mensagem de erro

Requisito funcional: RF17

Requisito não funcional: RNF02, RNF05 e RNF08

3.5 MODELO DE DADOS

O modelo de dados da ferramenta possui sete tabelas, conforme podemos visualizar abaixo na Figura 3 – Modelo de Dados.

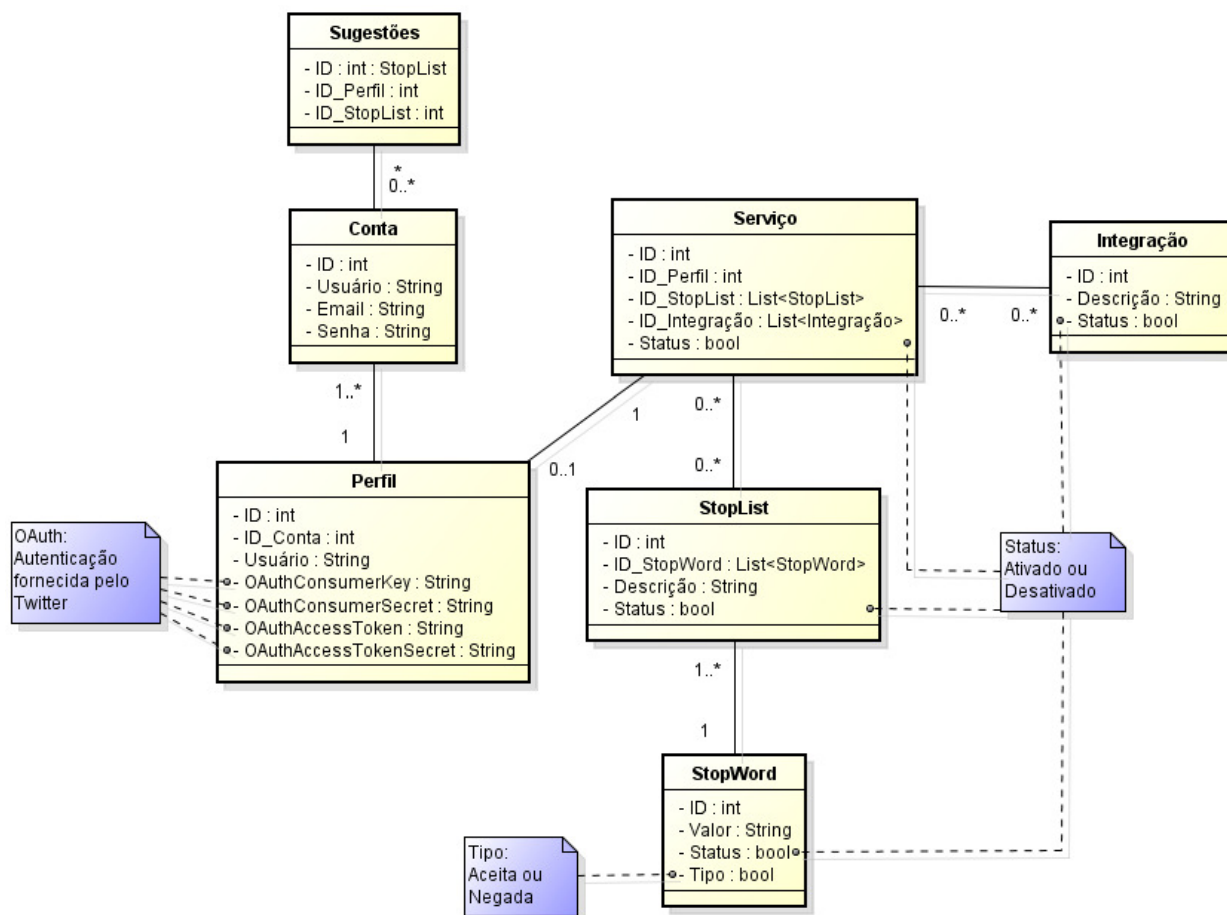


Figura 3 - Modelo de Dados

Estas possuem as seguintes funções na ferramenta:

- **Perfil:** responsável pela geração da autenticação com o *Twitter*. Para que seja obtida a chave de autenticação (“*OAuth*”, descrito no capítulo 7.1, *Conexão com o Twitter*), o usuário irá realizar uma comunicação direta com o *Twitter* no qual o mesmo irá efetuar uma verificação de usuário e senha e retornar a chave de autenticação para possibilitar a modificação da conta, tais como: enviar mensagens, realizar *retweet*, ler mensagens privadas, entre outras.
- **Conta:** contém uma lista de perfis do *Twitter* para que o usuário possa ter mais de uma conta do *Twitter* cadastrada na ferramenta e dados essenciais para utilizar a ferramenta, sendo eles: usuário, email e senha.

- **Serviço:** detém uma relação entre os perfis do *Twitter* cadastrados com as integrações disponíveis e as regras determinadas pelo usuário, possibilitando o mesmo ativar ou desativar o serviço do perfil do *Twitter* específico.
- **Integração:** lista de integrações disponíveis que será fornecida pela ferramenta, onde cada integração pode ser ativada ou desativada pelo usuário.
- **Stop List:** contém uma lista de *stop words* na qual o usuário pode criar ao seu critério e também listas públicas fornecidas pela ferramenta em que o usuário pode ativar ou desativar. Pode-se dizer que esta *stop list* é um conjunto de palavras (*stop words*) na qual a ferramenta utiliza para realizar ou não um *retweet*.
- **Stop Word:** lista de palavras que é utilizada pela *stop list*.
- **Sugestões:** responsável pela geração automática de novas sugestões de regras para um determinado perfil do *Twitter* cadastrado na ferramenta que sugere novas regras que podem ser ativadas ou desativadas como serviço.

4. ARQUITETURA

A arquitetura adotada no sistema, com separação em camadas, busca dar mais produtividade às implementações e otimizar a manutenção do sistema. Nesta seção, serão descritas as camadas e componentes, seus objetivos e restrições, organização das classes em pacotes e outros padrões de implementação na qual foram utilizados para o desenvolvimento. Também serão apresentadas as ferramentas utilizadas para o desenvolvimento e o funcionamento adequado da ferramenta..

4.1 VISÃO GERAL

A arquitetura é distribuída em pacotes que se subdivide por finalidade de utilização das classes, visando atender uma fácil criação, manutenção e separação das camadas do sistema. O padrão adotado para esta divisão foi o *MVC*¹¹ [2003, George Reese], no qual isola a lógica da aplicação com a interface do usuário, permitindo o desenvolvedor alterar, editar e testar cada parte do sistema separadamente.

4.2 PACOTES SIGNIFICATIVOS

A estrutura está dividida em dois pacotes significativos, sendo eles: *Java Resources* e *WebContent*. Esta divisão comporta as classes de negócio, contendo as regras do funcionamento do sistema, e as classes responsáveis pela interface do usuário.

4.3 VISÃO DE PROCESSOS

O sistema, com base na arquitetura proposta, tanto os processamentos leves quanto os processamentos pesados, são acionados por *threads* controlados pelo próprio *container*, no qual foi utilizado o *Apache Tomcat 7* [2013, Tomcat]. O padrão *Singleton* foi utilizado para realizar o controle adequado das *threads* dos serviços que a aplicação disponibiliza para o usuário nas verificações para realizar o *retweet*. Este será visto no próximo capítulo: 5. Principais Algoritmos.

4.4 VISÃO DA IMPLEMENTAÇÃO

A implementação da arquitetura segue o padrão de projeto *MVC* [2003, George Reese] no qual é dividido em três camadas: Visão, Controle e Persistência. Deste modo, temos as camadas abaixo e o respectivo local no pacote da aplicação:

- Visão (*WebContent*)
 - Camada responsável pela apresentação da interface da aplicação ao usuário. As páginas implementadas utilizam: *JSP*¹², *JSTL*¹³, *JavaScript*, *CSS*¹⁴ e *Tags*.
- Controle (*Java Resources: Services e Util*)

¹¹ Termo adotado para o padrão de projeto Model-view-controller (MVC) [2003, George Reese]

¹² Tecnologia utilizada para desenvolvimento de aplicações Web, no inglês: *Java Server Pages*

¹³ Biblioteca disponibilizada pelo *Java*, no inglês: *Java Server Tag Library* [2013, JSTL]

¹⁴ Linguagem de estilo utilizada para definir a apresentação das páginas, no inglês: *Cascading Style Sheets*

- Camada responsável pela comunicação entre as camadas de Visão e Persistência. Implementa também as regras de negócio pertinentes a requisitos funcionais.
- Persistência (*Java Resources: Model*)
 - Camada responsável pelo acesso a base de dados. É utilizado o padrão de projeto *DAO*¹⁵ para a implementação das classes e o banco de dados utilizado foi o *MySQL 5*.

Abaixo, podemos visualizar as conexões entre os pacotes descritos na Figura 4 – Conexões da arquitetura.

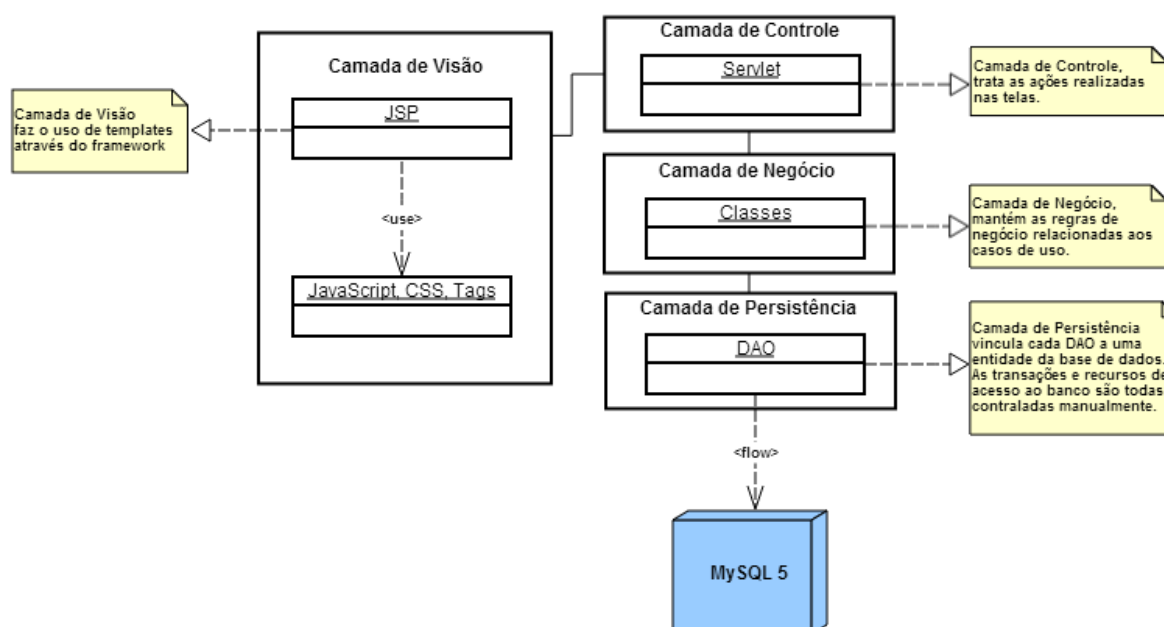


Figura 4 - Conexões da arquitetura

4.5 PLATAFORMA E GERENCIAMENTO DE DADOS

A plataforma utilizada foi o *JAVA EE*¹⁶ no qual é programada na linguagem *Java* e executado em um servidor propriamente configurado. Para este projeto, foi utilizado como *IDE*¹⁷ o software *Eclipse JEE Europa* que possibilita a configuração de um servidor *Tomcat 7* que simula a aplicação em um servidor.

O gerenciamento dos dados para a persistência das informações do usuário, detalhes da conta do *Twitter*, regras aceitas e negadas e outras informações necessárias para o bom funcionamento do sistema é administrado pela ferramenta *MySQL 5*. O mesmo utiliza linguagem *SQL*¹⁸ que facilita este gerenciamento de dados para a requisição das informações utilizadas.

¹⁵ Padrão de persistência de dado, no inglês: *Data Access Object* [2003, George Reese]

¹⁶ Plataforma de programação, no inglês: *Java Platform, Enterprise Edition*

¹⁷ Ambiente de desenvolvimento de software, no inglês: *Integrated Development Environment*

¹⁸ Linguagem de consulta estruturada, no inglês: *Structured Query Language*

5. PRINCIPAIS ALGORITMOS

O sistema possui alguns algoritmos principais, no qual possui as funcionalidades de realizar o *retweet*, modificação de regras, sugestão de novas regras e estatísticas gerais. Abaixo, será detalhado o funcionamento de cada para melhor entender o funcionamento da ferramenta.

5.1 RETWEET

O algoritmo de *retweet* possui algumas configurações que são realizadas conforme a especificação do usuário. A definição das palavras aceitas e negadas, perfis diferenciados e bloqueados, verificação de *mentions* e a integração com outras ferramentas são alguns dos dados que serão definidos e adquiridos conforme o usuário modifica o banco de dados utilizando a ferramenta a partir da página da internet.

Para realizar as verificações de *retweet*, é necessário realizar requisições para verificar novas *mentions*. Estas requisições são limitadas de acordo com a especificação do *Twitter* [Twitter, 2013] para ferramentas que utilizam a *API 1.1*. Desta forma, a biblioteca *Twitter4J*, na qual utiliza a *API 1.1*, é capaz de realizar somente 15 requisições de verificação dos *mentions* para o servidor do *Twitter* a cada 15 minutos. Com esta limitação, verificamos que só é possível realizar aproximadamente 1 requisição por minuto. Logo, foi definido que o máximo de requisições que pode ser realizado para a verificação de *mentions* é de uma a cada um minuto e meio.

Tendo como base a verificação de *mentions* de um perfil do *Twitter* sendo realizada a cada um minuto e meio, é utilizada uma *thread* de verificação destinada a 40 perfis. Desta forma, uma *thread* da ferramenta é capaz de verificar a cada um minuto, 40 perfis cadastrados na ferramenta. Conseqüentemente, à medida que necessário é criado uma nova *thread* para suportar um maior número de perfis a serem verificados (será detalhado posteriormente na seção 5.1.1. *ServicesSingleton* como é realizado esta alocação de *threads*).

As classes utilizadas no algoritmo de verificação do *retweet* são: *ServicesSingleton* e *AutoRetweetThread*. Estas classes serão descritas abaixo.

5.1.1 SERVICESSINGLETON

Esta classe é responsável por gerenciar as *threads* da ferramenta, realizando a alocação de novas *threads* sempre que for necessário. Conforme mencionado anteriormente, cada *thread* é capaz de lidar com 40 perfis, sendo alocado sempre uma nova *thread* à medida que perfis ativem o serviço de *retweet* e esta esteja com o máximo de alocações.

O gerenciamento destas alocações ocorre com a relação de um identificador e a sua respectiva *thread*. À medida que é ativada uma verificação de *retweet* de um perfil qualquer, este é adicionado a *thread* que possui o menor número de alocação, e esta percorre os perfis realizando a verificação dos *mentions* realizando ou não o *retweet* de acordo com as regras definidas. A forma que ocorre o *retweet* será exemplificada na próxima seção.

A codificação da relação do identificador com a *thread* existente foi desenvolvida com um *HashMap* sendo o primeiro atributo um inteiro e o segundo a respectiva classe da *thread* no qual é responsável pelo processo do *retweet*. Pelo fato de ser uma classe *Singleton*, para ativar uma nova verificação é chamada uma função para habilitar a verificação e esta função verifica qual *thread* possui menos serviços para verificar a ativa na mesma. Abaixo uma ilustração para melhor entender (Figura 5 – Relação da classe *ServicesSingleton*):

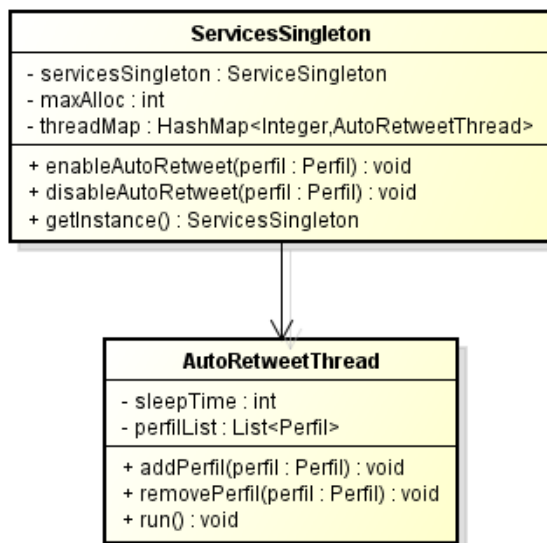


Figura 5 - Relação da classe *ServicesSingleton*

A variável *maxAlloc* possui um valor fixo igual a 40. Este é o máximo de alocação por *thread*, conforme já definido. A variável *sleepTime* tem o valor inicial de 90000, para realizar uma verificação a cada um minuto e trinta segundos (90 000 milissegundos), e conforme é adicionado um perfil para verificar, esta variável é atualizada para 90 000 sobre o número de perfis com o serviço ativado nesta *thread*. Assim, para dois perfis ativos teremos pausas de 45 000 milissegundos (45 segundos) entre cada verificação.

A função *enableAutoRetweet* realiza a seguinte verificação:

1. Verifica qual *AutoRetweetThread* no *hashMap* possui a lista de perfis (*perfilList*) com o menor número de perfis (*perfilList.size()*);
2. Caso exista uma *AutoRetweetThread* com o tamanho do *perfilList* menor que 30:
 - a. Adiciona o perfil nesta *thread*;
 - b. Salva o inteiro de onde esta localizada a *thread* no *hashMap* para melhor identificar onde esta alocada a *thread* futuramente.
3. Caso não exista uma *AutoRetweetThread* com o tamanho do *perfilList* menor que 30:
 - a. Adiciona uma nova *thread* no *hashMap*;
 - b. Segue para o passo 2.a.

Tabela 7 - Algoritmo *enableAutoRetweet*

Com esta implementação, é possível realizar o gerenciamento de alocações de *threads* nos processadores dos servidores a qual a ferramenta é instalada, permitindo que os processos sejam executados em núcleos diferentes do processador eliminando problemas de processamento. Em servidores que realizam tarefas que exigem grandes demandas, é necessário prever acontecimentos como: maximizar a produtividade¹⁹, minimizar o tempo de resposta da tarefa o máximo possível, minimizar a sobrecarga²⁰, balancear o uso de recursos dentre outros fatores que podem levar ao servidor não obter o máximo de desempenho que o mesmo pode prover.

5.1.2 AUTORETWEETTHREAD

Conforme descrito na seção anterior, esta classe é responsável por realizar o *retweet* conforme regras de um perfil que tenha o serviço ativado. Internamente, a função *run* desta classe (na qual é ativada como *thread*), possui o seguinte funcionamento:

1. Percorre a lista de perfis;
2. Para cada perfil, realiza as seguintes verificações:
 - a. Recebe do *Twitter* os *mentions* do perfil;
 - b. Verifica se o *mention* já foi analisado;
 - i. Caso não tenha sido, verifica se deve ocorrer o *retweet* de acordo com as regras definidas pelo usuário.
 - ii. Caso tenha sido verificado, desconsidera o *mention*.

Tabela 8 - Algoritmo verificação de perfis cadastrados

A verificação das regras para o caso de *retweet* possui uma hierarquia de prioridade. Esta foi estipulada para melhor atender a necessidade de que ocorra o *retweet*, pois caso a ordem desta verificação seja alterada, pode não ocorrer. Como por exemplo, se um perfil diferenciado enviar uma *mention* que contem uma palavra bloqueada e a verificação de palavras bloqueadas for realizado primeiro que perfil diferenciado não ocorrerá *retweet* desta *mention*. Tendo isto em vista, foi programada a seguinte hierarquia de verificação:

1. Verifica se a *mention* é de um perfil diferenciado;
 - a. Caso afirmativo, realiza o *retweet*.
 - b. Caso negativo segue para o passo 2.
2. Verifica se a *mention* contém palavra(s) negada(s);
 - a. Caso afirmativo, descarta a *mention*.
 - b. Caso negativo segue para o passo 3.
3. Verifica se a *mention* contém palavra(s) aceita(s);
 - a. Caso afirmativo, realiza o *retweet*.
 - b. Caso negativo descarta a *mention*.

Tabela 9 - Algoritmo de verificação de uma *mention*

¹⁹ Conhecido, no inglês, como: *throughput*.

²⁰ Conhecido, no inglês, como *overhead*.

Ao ser realizado o *retweet*, há também o tratamento da mensagem que será enviada para a *timeline*. Este é feito de acordo com a integração com outras aplicações, que estará disponível na ferramenta para a ativação conforme disponibilidade de compatibilidade com a ferramenta. Um exemplo é a abreviação de endereços eletrônicos, convertendo uma *URL* de tamanho relativamente grande tornando-a uma *URL* compacta, possibilitando o maior número de caracteres a ser enviado na *timeline*. Note que uma mensagem enviada para a *timeline* é limitada a 140 caracteres.

5.2 MODIFICAÇÃO DE REGRAS

A modificação de novas regras ocorre através da ferramenta online. Acessando o perfil do *Twitter* cadastrado pelo usuário, acessa-se o *menu* de Modificações de Regras. Este exibe uma tela com as opções de remover, adicionar e editar os conjuntos de Regras Aceitas e Regras Negadas.

A Figura 6 – Exemplo de modificação de regras na ferramenta abaixo ilustra este funcionamento. Percebe-se que as regras são agrupadas em conjuntos para ficar mais fácil ao usuário o gerenciamento das palavras.

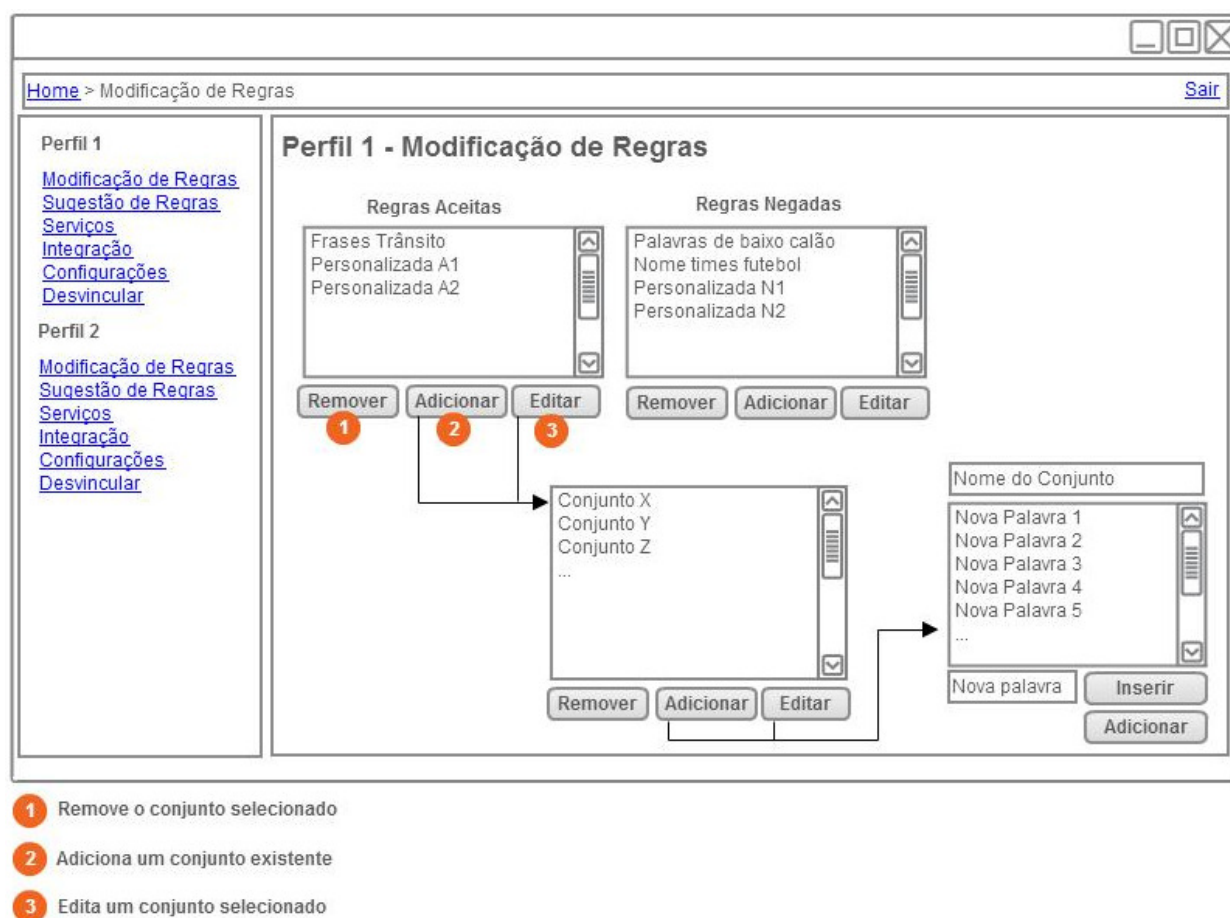


Figura 6 - Exemplo de modificação de regras na ferramenta

Os conjuntos de palavras são criados no banco de dados da ferramenta e vinculados ao perfil corrente. Nas configurações da conta de cada perfil existe uma opção para poder tornar estes conjuntos

públicos ou não, possibilitando outros usuários a usufruir destes. A existência de conjuntos pré-definidos da ferramenta também irá existir, para que palavras comuns sejam adicionadas as regras sem a necessidade de criar conjuntos facilitando a configuração da conta. Um exemplo são as palavras de baixo calão.

5.3 SUGESTÃO DE NOVAS REGRAS

A sugestão de novas regras irá ocorrer pela análise dos *mentions* verificados pela ferramenta sempre que um *mention* não estiver dentro de alguma das regras definidas. Logo, ao passar pela verificação de perfis diferenciados, palavras negas ou aceitas sem tomar nenhuma ação, este é um *mention* que não foi realizado o *retweet* e pode existir alguma palavra que poderia estar em algum dos dois conjuntos (aceito ou negado).

A Tabela 10 – Sugestão de novas regras (palavras em comum) abaixo exemplifica uma lista de 5 *mentions*, onde se pode perceber uma palavra incomum entre eles. Assim, torna-se capaz de criar uma sugestão para usuário para adicionar esta palavra nas regras de palavras aceitas ou negadas, de acordo com o critério do usuário. Para este exemplo, a palavra “buracos” não esta contida em nenhum dos conjuntos definidos e não existe nenhuma palavra aceita ou negada nas *mentions*.

Mention recebido
Rua Ipiranga em frente a PUC com muitos buracos .
Lentidão na saída de POA devido a buracos .
Muitos buracos na estrada para o litoral.
Cuidado com os buracos na Rua Salvador França em frente ao Jardim Botânico.
Há muitos buracos por causa das obras na Rua Anita Garibaldi com a Perimetral.

Tabela 10 - Sugestão de novas regras (palavras em comum)

Desta forma foi elaborada a Tabela 11 – Sugestão de novas regras com palavras e um contador de repetição para a mesma. Após criar esta tabela, teremos uma lista com palavras para sugestão, sendo sugeridas as cinco primeiras palavras que tiver o maior contador. Para o exemplo acima, teríamos as seguintes sugestões descritas na tabela abaixo:

Contador	Palavras
5	Buracos
3	Muitos
1	Lentidão

Tabela 11 - Sugestão de novas regras

5.4 ESTATÍSTICAS GERAIS

Para fins de estatísticas, controle do usuário e dados do processamento realizado pela ferramenta referente ao perfil, é apresentado uma tabela com os resultados das verificações que foram aceitas ou não no momento da verificação. Os dados apresentados são:

- Total de *mentions* tratados;

- Total de *mentions* aceites;
- Total de *mentions* negados;
- Total de usuários bloqueados;

Estes dados são gerados pelo algoritmo de *retweet*. Sempre que uma *mention* que ainda não foi verificada for tratada, irá ocorrer uma atualização dos contadores no banco de dados conforme o tratamento, tornando possível a apresentação dos dados para o usuário.

6. TRABALHOS RELACIONADOS

Foram analisadas outras três ferramentas que possuem funcionalidades similares, sendo elas: *RoundTeam* [RoundTeam, 2012], *RetransTwitter* [RetransTwitter, 2012] e *TweetAdder* [TweetAdder, 2012]. Serão abordadas as funcionalidades de cada uma e analisada as diferenças para a ferramenta desenvolvida. As regras de *mentions*, *hashtags* e frases específicas, conforme já exemplificado anteriormente, ocorrem da mesma maneira nestas ferramentas.

6.1 ROUNDTEAM

Esta é uma ferramenta online que possui a funcionalidade de realizar um *retweet* a partir de *mentions*, *hashtags* ou frases que ocorrem conforme especificado em um perfil ou uma lista de perfis do *Twitter*. Funciona como serviço, logo, o usuário não precisa se preocupar em deixar o computador ligado ou mesmo adquirir meios para que a mesma funcione sempre.

Uma funcionalidade na qual a *RoundTeam* oferece e que não será disponibilizada na ferramenta proposta é o *retweet* automático de todas as novas mensagens da *timeline* de um determinado perfil do *Twitter*. Um ponto negativo é a limitação de no máximo 15 regras de *mentions* e *hashtags*.

6.2 RETRANSTWITTER

Esta ferramenta possui a funcionalidade de realizar um *retweet* a partir de *mentions* e *hashtags* que ocorrem conforme especificado em um perfil ou uma lista de perfis do *Twitter*. Entre as ferramentas estudadas, está é a que menos possui funcionalidades conforme podemos verificar na Tabela 12 – Comparativo das ferramentas.

A simplicidade da *RetransTwitter* dificulta a utilização por parte do usuário visto que a definição de regras são realizadas por padrões especificados pela mesma, não tendo uma parte visual robusta para torná-la mais amigável ao usuário.

6.3 TWEETADDER

Esta ferramenta possui a funcionalidade de realizar um *retweet* a partir de *mentions* e regras que ocorrem conforme especificado em um perfil ou uma lista de perfis do *Twitter*. Também, possui a função de enviar mensagens para a *timeline* conforme frases definidas e regras elaboradas de forma automática.

Entre as ferramentas, esta é a única que não é acessada através de uma página da internet. Para utilizá-la, é necessário comprar e baixar o programa *TweetAdder*. Apesar de se aproximar bastante da ferramenta proposta com as funcionalidades existentes, é necessário que o usuário deixe o seu computador pessoal ligado sempre que desejar usufruir dos serviços.

Uma integração que esta ferramenta possui no qual será visto para a ferramenta proposta é o *retweet* de RSS. Esta funcionalidade é muito útil quando o usuário necessita manter o seu perfil

atualizado com conteúdos que são recebidos pelo RSS e que são disponibilizados em diversos meios, como páginas da internet, programas de computador entre outros.

6.4 COMPARATIVOS DAS FERRAMENTAS

A tabela abaixo representa uma comparação de todas as ferramentas e suas principais funcionalidades.

Funcionalidade	Ferramenta	<i>RoundTeam</i>	<i>Retranstwitter</i>	<i>TweetAdder</i>
Ferramenta online	Sim	Sim	Sim	Não
Retweet automático	Sim	Sim	Sim	Sim
Retweet mentions	Sim	Sim	Sim	Sim
Retweet hashtag	Sim	Sim	Sim	Não
Retweet palavras específicas	Sim	Sim	Não	Sim
Integração outras ferramentas	Sim	Não	Não	Sim
Sugestão novas regras	Sim	Não	Não	Não
Modificação regra por mensagens privadas	Sim	Não	Não	Não
Usuário administradores	Sim	Não	Não	Sim
Bloqueio de usuário	Sim	Não	Não	Sim
Customização do retweet	Sim	Não	Não	Não
Vinculação de mais de um perfil do Twitter	Sim	Sim	Não	Sim
Serviço gratuito	Sim	Sim	Sim	Não
Limite de serviços	Não	Sim	Não	Não
Retweet de outro perfil	Não	Sim	Não	Não
Retweet de RSS	Não	Não	Não	Sim

Tabela 12 - Comparativo das ferramentas

7. BIBLIOTECA TWITTER4J

Para realizar as funcionalidades propostas da ferramenta com a rede social do *Twitter* é utilizada a biblioteca *Twitter4J* [Twitter4J, 2012]. Esta biblioteca é destinada a usuário que desejam utilizar a linguagem de programação *Java*, possibilitando o usuário codificar todas as funções que podem ser utilizadas na página da internet do *Twitter*, tais como: enviar mensagens para a sua *timeline*, realizar um *retweet*, receber e enviar mensagens particulares, verificar *mentions* do seu perfil e outras que o *Twitter* possui.

Existem outras bibliotecas [Twitter, 2012] em diversas linguagens disponibilizadas na página da internet do *Twitter* que são ditas como oficiais, na qual segue as regras de autenticação fornecida pelo *Twitter*. O *Twitter4J* foi utilizado por proporcionar um maior número de funcionalidades na linguagem *Java* na qual a ferramenta proposta necessita para realizar todas as funcionalidades especificadas e por ser uma biblioteca gratuita e de fácil uso.

Esta biblioteca dispõe em sua página da internet [Twitter4J, 2012] um exemplo de todas as funções que podem ser utilizadas, tornando fácil a programação e também por existir uma documentação detalhada e eficiente.

A seguir, será detalhado e exemplificado o estudo realizado sobre a biblioteca para ser capaz de desenvolver todas as funcionalidades que serão codificadas na ferramenta. Este estudo foi elaborado a partir dos exemplos contidos na página da internet do *Twitter4J* [Twitter4J, 2012] e por um programa prático criado para realizar testes desta biblioteca.

7.1 CONEXÃO COM O TWITTER

Para realizar a conexão com o perfil do usuário que irá se registrar na ferramenta é necessário primeiramente que se obtenha o registro da aplicação realizada na página da internet de desenvolvedores do *Twitter*. Acessando a página <https://dev.twitter.com/apps/new>, efetuamos a autenticação do usuário proprietário da ferramenta proposta, para obtermos as credenciais necessárias para utilizarmos a aplicação.

Estas credenciais são necessárias para fins de segurança do *Twitter*. Desta maneira, pode haver um controle das aplicações que utilizam o mesmo, podendo a qualquer momento ser bloqueada, caso não esteja de acordo com as regras.

Após fornecer os dados a qual a ferramenta será vinculada ao perfil, são geradas duas chaves, denominadas: *consumerKey* e *consumerSecret*. Com estas duas chaves, pode-se realizar a conexão da ferramenta a partir da biblioteca *Twitter4J* com o *Twitter*, conforme exemplo:

```
Twitter twitter = new Twitter();
twitter.setOAuthConsumer("consumerkey", "consumersSecret");
RequestToken twitterRequestToken = twitter.getOAuthRequestToken();
```

Tabela 13 - Utilização das chaves de segurança do perfil (Twitter4J)

Com as chaves de autenticação, realizamos a conexão com o usuário e senha do perfil informado na ferramenta. No exemplo abaixo, demonstrado pela função *loginAccount*, podemos verificar que é passado como parâmetro o usuário e senha fornecidos do perfil, utilizando juntamente as chaves da ferramenta que foi cadastrada no site e as chaves de autenticação recebidas pelo exemplo acima.

```
private TwitterFactory tf;
private ConfigurationBuilder cb;

public void loginAccount(String User, String Password){
    cb = new ConfigurationBuilder();
    cb.setDebugEnabled(true)
    .setOAuthConsumerKey("consumerkey") // key da aplicação
    .setOAuthConsumerSecret("consumerSecret") // secret da aplicação
    .setOAuthAccessToken("consumerKeyReceived") // key recebida da aplicação
    .setOAuthAccessTokenSecret("consumerSecretReceived") // secret recebida da aplicação
    .setUser(User) // perfil do usuário do Twitter
    .setPassword(Password); // senha do perfil do usuário do Twitter

    tf = new TwitterFactory(cb.build());
}
```

Tabela 14 - Utilização das chaves de segurança da aplicação (Twitter4J)

partir da instância da classe *TwitterFactory* fornecida pela biblioteca *Twitter4J*. A seguir, podemos ver exemplos das funcionalidades que a ferramenta irá utilizar.

7.2 VERIFICANDO MENTIONS

Para verificar os *mentions* do perfil do usuário conectado, utilizamos a classe *TwitterFactory* conforme exemplificado anteriormente e chamamos o método *getMentions*, como podemos observar abaixo.

```
Twitter twitter = tf.getInstance();
List<Status> timeline;
try {
    timeline = twitter.getMentions();
    for (Status s : mentions) {
        System.out.println("Timeline: " + s.getText());
    }
} catch (TwitterException ex) {
    Logger.getLogger(TwitterConnection.class.getName()).log(Level.SEVERE, null, ex);
}
```

Tabela 15 - Verificação de *mentions* (Twitter4J)

A partir dos *mentions*, podemos realizar verificações de palavras chaves que se enquadrem nas regras definidas pelo usuário. Caso um *mention* esteja de acordo com uma regra estabelecida para ocorrer o *retweet*, enviamos o mesmo para a *timeline* do perfil, conforme será visto no próximo tópico. Nota-se que, sempre que é realizado a função *getMentions*, são retornados os últimos dez *mentions* do perfil a que se está conectado. Assim, é necessário que haja uma verificação de novos *mentions*, ou seja, um algoritmo capaz de tratar um *mention* e saber se o mesmo já não foi verificado pelas regras da ferramenta.

Somente *mentions* que sejam novos podem passar pela verificação das regras e sofrer um *retweet*. O funcionamento deste algoritmo será visto posteriormente.

7.3 ENVIANDO MENSAGEM PARA *TIMELINE*

Para realizar um *retweet* ou enviar uma mensagem para a *timeline* do perfil conectado na ferramenta, utilizamos a função *updateStatus*. Abaixo um exemplo de como a mesma é utilizada.

```
Twitter twitter = tf.getInstance();
try {
    String newStatus = "Exemplo de mensagem para timeline";
    twitter.updateStatus(newStatus);
} catch (TwitterException ex) {
    Logger.getLogger(TwitterConnection.class.getName()).log(Level.SEVERE, null, ex);
}
```

Tabela 16 - Envio de mensagem para *timeline* (Twitter4J)

Como podemos ver no exemplo acima, será enviado para a *timeline* do perfil a mensagem “Exemplo de mensagem para *timeline*”. Se ocorrer erro de conexão ou quaisquer outros erros quando é realizado o envio de mensagem para a *timeline* ou outra função utilizada pela biblioteca *Twitter4J*, é gerada uma exceção. Por este motivo, sempre é utilizado o bloco de *try* e *catch*.

7.4 ENVIO E RECEBIMENTO DE MENSAGENS

As mensagens são enviadas diretamente de um perfil para outro. Funciona semelhante a um correio eletrônico, para que possa realizar troca de mensagens privadas. Estas serão utilizadas pelos administrados para poderem realizar alterações nas regras da ferramenta. Por segurança, sempre que uma mensagem é recebida, caso se enquadre em uma nova definição ou alteração de regras, será verificado se o usuário pertence ao grupo de administradores.

Abaixo, um exemplo de como enviar uma mensagem para um perfil específico. Neste caso, uma mensagem para o perfil @FACIN está sendo enviada com o seguinte conteúdo: “Teste de mensagem privada”.


```

Twitter twitter = tf.getInstance();
try {
    String toUser = "@FACIN"
    String message = "Teste de mensagem privada"
    DirectMessage message = twitter.sendDirectMessage(toUser, message);
} catch (TwitterException ex) {
    Logger.getLogger(TwitterConnection.class.getName()).log(Level.SEVERE, null, ex);
}

```

Tabela 17 - Envio de mensagem privada (Twitter4J)

Para ler as mensagens recebidas, realizamos a seguinte codificação, conforme abaixo. Nota-se que as mensagens recebidas são retornadas como uma lista somente com as primeiras dez mensagens.

```

Twitter twitter = tf.getInstance();
try {
    List<DirectMessage> message = twitter.getDirectMessage(toUser, message);
    for (DirectMessage m : message) {
        System.out.println("Message: " + m.getText());
    }
} catch (TwitterException ex) {
    Logger.getLogger(TwitterConnection.class.getName()).log(Level.SEVERE, null, ex);
}

```

Tabela 18 - Leitura de mensagem privada (Twitter4J)

7.5 BLOQUEIO E DESBLOQUEIO DE UM PERFIL

O bloqueio de um perfil é necessário para que não seja recebida nenhuma *mention* realizada pelo mesmo. Desta maneira, pode-se ajudar a aplicação a não realizar verificações de perfis que estejam tentando burlar as regras da ferramenta para fazer com que ocorra o *retweet*.

Os principais casos de perfis que tentam realizar *mentions* para que ocorra o *retweet* são de usuários que tentam realizar propagandas. Estes são capazes de obter um conhecimento sobre as regras da ferramenta e a melhor solução para que não seja feito o *retweet* é banindo o mesmo. Abaixo um exemplo de como é realizado o bloqueio de um usuário.

```

Twitter twitter = tf.getInstance();
try {
    String userScreenName = "MarketingSA";
    twitter.createBlock(userScreenName);
} catch (TwitterException ex) {
    Logger.getLogger(TwitterConnection.class.getName()).log(Level.SEVERE, null, ex);
}

```

Tabela 19 - Bloqueio de um perfil (Twitter4J)

No exemplo acima foi realizado o bloqueio do perfil *@MarketingSA*. Para o desbloqueio deste usuário, teremos a seguinte chamada:

```
Twitter twitter = tf.getInstance();
try {
    String userScreenName = "MarketingSA";
    twitter.destroyBlock(userScreenName);
} catch (TwitterException ex) {
    Logger.getLogger(TwitterConnection.class.getName()).log(Level.SEVERE, null, ex);
}
```

Tabela 20 - Desbloqueio de um perfil (Twitter4J)

7.6 BUSCA DE HASHTAGS

Esta funcionalidade disponibilizada pela biblioteca *Twitter4J* possibilita buscar por palavras e *hashtags*. Para realizar esta busca, utilizamos a função *search* na qual verifica com o bando de dados do *Twitter* todas as ocorrências de palavras ou *hashtags* existentes. O retorno desta busca é fornecido o perfil a qual existe a ocorrência e a mensagem por completo. Abaixo, o exemplo de como utilizar.

```
Query query = new Query("#exemplo");
QueryResult result = twitter.search(query);
for (Tweet tweet : result.getTweets()) {
    System.out.println(tweet.getFromUser() + ":" + tweet.getText());
}
```

Tabela 21 - Busca de *hashtags* (Twitter4J)

Neste exemplo, foi realizada uma procura pela *hashtag*: "#exemplo". Todos os perfis do *Twitter* que enviaram para a sua *timeline* esta *hashtag* irão ser retornados nesta busca, logicamente, serão retornadas as últimas dez ocorrências.

8. CONCLUSÃO

Neste trabalho foi apresentada uma ferramenta que possibilita a automatização da rede social *Twitter* e, com o decorrer do desenvolvimento, foram verificadas novas possibilidades de inovações que serão tratadas na sessão 8.1 Trabalhos Futuros. Foi visto também que a mídia vem utilizando cada vez mais estas redes para propagar a informação e propagandas, tornando esta ferramenta uma opção para automatizar o trabalho que hoje, na maioria dos casos, é realizado por pessoas. Desta forma, o objetivo do trabalho foi atingido, tornando disponível um programa que pode ser utilizado por diversos nichos e propagando a informação de uma forma rápida e eficiente.

8.1 TRABALHOS FUTUROS

Ao decorrer do desenvolvimento, foi verificada algumas inovações que podem enriquecer a aplicação possibilitando o usuário final interagir melhor com as mensagens lidas. Como a manutenção e a atualização de um programa são contínuas, segue abaixo as integrações que podem ser desenvolvidas e realizações futuras que podem agregar mais valor ao sistema e ainda torná-lo mais robusto.

- Desenvolvimento de uma ferramenta *Mobile* para enviar as coordenadas, de acordo com o *GPS*, na mensagem enviada, possibilitando assim enviar o local com uma maior precisão no momento do envio e também uma visualização pelo *GoogleMaps* por parte do usuário.
- As credenciais necessárias para a autenticação da conta do *Twitter* na aplicação para a verificação da *timeline* (realizada pelo *OAuth*) podem ser realizada automaticamente. Na ferramenta desenvolvida, estas credenciais foram colocadas manualmente no banco de dados e utilizadas quando ocorre comunicação com o perfil. Conforme o suporte do *Twitter* [Twitter, 2013], esta autenticação esta disponível com a funcionalidade conhecida como “*Twitter Integration*”, no qual disponibiliza um botão na ferramenta (“*Sign in with Twitter*”) e retorna as chaves de *OAuth* que são salvas no banco de dados.
- Para perfis que proporcionam informações de trânsito, é interessante o desenvolvimento de uma aplicação *Mobile* que a partir de mensagens enviadas por usuários verifique rotas alternativas para desviar de ruas que possuem trânsito intenso por motivos diversos como: acidente, obras, manifestações, etc.
- Integração com empresas meteorológicas para informar quando conveniente a previsão do tempo, ou até mesmo para casos como tornados, furacões e outros desastres naturais para avisar ou alertar usuários.
- O algoritmo de *Retweet* (Capítulo 5.1 *Retweet*) pode ser otimizado na questão de verificação das *mentions* que ocorrem de 90 em 90 segundos. Uma forma de otimizar é desenvolver um algoritmo capaz de analisar a frequência que o perfil recebe *mentions*, assim, somente fazer a verificação a partir desta análise ao invés de 90 em 90 segundos. Um exemplo é um perfil que recebe *mentions* de uma em uma hora como média. Este pode ser verificado de uma em uma hora, economizando 39 verificações/requisições em uma hora.
- Suporte para outras redes sociais como: *Facebook*, *Orkut*, *MySpace*...
- Verificação de *SPAM* [Smith, 2007], para banir mensagens indesejadas.

BIBLIOGRAFIA

Twitter.com Site Info. AboutTwitter. 2012. Disponível em: <<https://twitter.com/about>>. Acessado em: 05 de ago. 2012.

Twitter.com Site Info. FAQs About Following. 2012. Disponível em: <<https://support.twitter.com/articles/14019-faqs-about-following>>. Acessado em: 05 de ago. 2012.

Twitter.com Site Info. What is Twitter Timeline. 2012. Disponível em: <<https://support.twitter.com/articles/164083-what-is-a-twitter-timeline>>. Acessado em: 05 de ago. 2012.

Twitter.com Site Info. What are replies and mentions. 2012. Disponível em: <<https://support.twitter.com/articles/14023-what-are-replies-and-mentions>>. Acessado em: 05 de ago. 2012.

Twitter.com Site Info. Twitter Libraries. 2012. Disponível em: <<https://dev.twitter.com/docs/twitter-libraries#java>>. Acessado em: 20 out. 2012.

Twitter.com Site Info. FAQs About Retweet RT. 2012. Disponível em: <<https://support.twitter.com/articles/77606-what-is-retweet-rt>>. Acessado em: 05 de ago. 2012.

Twitter.com Site Info. Rest API Rate Limiting. 2013. Disponível em: <<https://dev.twitter.com/docs/rate-limiting/1.1>>. Acessado em 07 de Abril de 2013

Twitter4J – A Java library for the Twitter API. Main page. 2012. Disponível em: <<http://twitter4j.org/en/index.html>>. Acessado em: 20 out. 2012.

Jacobson, Ivar. The unified software development process. Addison-Wesley, 1999. 463 p.

Hunt, John. The unified process for practitioners: object-oriented design, UML and Java. London : Springer, 2000. 280 p.

RoundTeam Site. Main Page. 2012. Disponível em: <<http://roundteam.co/>>. Acessado em: 05 de ago. 2012.

RetransTwitter Site. About Page. 2012. Disponível em: <<http://retranstwitter.com/en>>. Acessado em: 05 de ago. 2012.

TweetAdder Site. HelpDesk. 2012. Disponível em: <<http://www.tweetadder.com/>>. Acessado em: 05 de ago. 2012.

Van Rijsbergen, C.J., Information Retrieval, Butterworths, 1975.

Smith, Marcia. Internet Privacy. 2007. 1 p.

Steven Abney. Understanding the Yarowsky algorithm. Computational Linguistics. 2004. 30: 365–395.

George Reese. Java Database Best Practices. 2003: 16-17.

Tomcat.apache.org. Apache Tomcat Welcome. 2013. Disponível em: <<http://tomcat.apache.org>>. Acessado em: 26 de maio 2013.

JSTL. Stander Tag Libraries for Java Server Pages. 2013. Disponível em: <<https://jstl.java.net>>. Acessado em: 26 de maio 2013.

Twitter.com Site Info. Sign in with Twitter. 2013. Disponível em: <<https://dev.twitter.com/docs/auth/sign-twitter>>. Acessado em: 09 de jun. 2013.