

Uma Proposta de Sistema para Recomendação de Serviços Públicos

Leonardo Melo Costa da Silva

Orientador – Msc. Pablo Vieira Florentino

Grupo de Pesquisa em Sistemas Distribuídos, Otimização, Redes e Tempo-Real (GSORT)

Especialização em Computação Distribuída e Ubíqua

Instituto Federal da Bahia (IFBA)

e-mail: leonardomelo01@gmail.com, email orientador: pablovf@gmail.com

Resumo—The citizens have many rights and obligations in the society where they live, they are guaranteed by rules defined for everybody live in harmony and dignity. However, this is not enough to ensure they can be exercised when necessary, because in many cases it takes aid of public services, which have obstacles to their use, eg, the lack of information due to the deficiency in the disclosure of the same. Thus, a tool is proposed to analyze the needs of citizens, seek public services that solve their problems and make suggestions for better options. For this, it should be made a collection of messages posted by users of social networks on their own profiles as a source of data about the authors. These texts are extracted to pass through a filter to highlight the most important words, a process accomplished by creating words networks and thesaurus. The product of this process is submitted to an element of text mining, the classifier, which connects the services with the most important terms of the phrases analyzed. The solution then makes suggestions of appropriate services to citizens and improves upon receiving the assessment made by them.

Index Terms—Distributed Compute, Social Network, Text Mining, Desire Discovery

Resumo—Os cidadãos possuem diversos direitos e deveres na sociedade onde vivem, eles são garantidos por regras definidas visando promover uma convivência harmônica e digna a todos. No entanto, isso não basta para garantir que possam ser exercidos quando necessário, pois, em diversos casos é preciso auxílio de serviços públicos, os quais possuem obstáculos que impedem seus usos, por exemplo, a falta de informação devido à deficiência na divulgação dos mesmos. Assim, uma ferramenta é proposta para analisar necessidades dos cidadãos, buscar serviços públicos que resolvam seus problemas e faça sugestões com melhores opções. Para isso, deve ser feita a coleta das mensagens postadas pelos usuários das redes sociais em seus próprios perfis como fonte de dados a respeito dos seus autores. Esses textos extraídos têm de passar por uma filtragem para evidenciar as palavras de maior importância, processo esse realizado com a criação de redes de palavras e dicionário de sinônimos. O produto desse processamento é submetido a um elemento de mineração de textos, o classificador, que relaciona os serviços aos termos mais importantes das sentenças analisadas. A solução então faz as sugestões de serviços adequados aos cidadãos e se aperfeiçoa ao receber a avaliação realizada pelos mesmos.

Index Terms—Computação distribuída, Redes Sociais, Mineração de textos, descoberta de desejos

SUMÁRIO

I	Introdução	2
II	Redes sociais digitais	2

II-A	Explorando dados de redes sociais	3
III	Mineração de textos	3
III-A	Categorização de texto	3
III-B	Preparação dos textos	3
IV	Redes de palavras	4
V	Computação Ubíqua	4
VI	Trabalhos correlatos	4
VII	Documentação da Ferramenta	5
VII-A	Requisitos	6
VII-B	Diagramas	6
VIII	Módulos da Ferramenta	8
VIII-A	Rede Social	8
VIII-B	Módulo de Aplicação	9
VIII-C	Servidor	10
VIII-D	Módulo de Coleta	10
VIII-E	Módulo de Persistência	11
VIII-F	Módulo de Escolha de Termos	12
VIII-G	Módulo de Mineração de Textos	13
VIII-H	Módulo de Sugestão	15
VIII-I	Módulo de Avaliação	16
VIII-J	Validação	16
IX	Conclusão e Trabalhos futuros	16
	Referências	17
	Apêndice A: Coletor de interações	18
	Apêndice B: Conversor FreqToARFF	19
	Apêndice C: Notificador	20
	Apêndice D: CLASSES.arff	21
	Apêndice E: APIs do Facebook	23
	Apêndice F: Montagem de URL para gerar notificações no Facebook	26
	Apêndice G: Detalhes da implementação no Facebook	26

I. INTRODUÇÃO

Os cidadãos possuem diversos poderes perante a sociedade onde vivem, garantidos por regras definidas visando promover uma convivência harmônica e digna a todos. Para auxiliar o exercício desses direitos e deveres o estado disponibiliza instrumentos, os serviços públicos. No entanto, não basta somente existirem essas ferramentas quando a divulgação das mesmas não é feita de modo satisfatória, possibilitando a utilização apenas às poucas pessoas que possuem acesso a informação.

Apesar de alguns serviços públicos serem anunciados em meios de comunicação convencional, como a televisão e o rádio, ainda se conhece muito pouco sobre tais serviços.

Muito poderia ser feito no sentido de otimizar o tempo despendido na busca por estes serviços assim como os deslocamentos para o uso dos mesmos, informando, por exemplo, aqueles mais próximos ou acessíveis a cada pessoa.

Da mesma forma, muitas dessas informações poderiam ser pesquisadas em fontes como telefone ou sites na internet, porém, o tempo necessário e a abundância de informações envolvidas tendem a atrapalhar ou confundir nas buscas.

Dentro desse contexto de dificuldades, a detecção das necessidades é ponto chave do processo e pode interferir positivamente na resolução das tarefas cotidianas. Neste sentido, uma ferramenta capaz de analisar necessidades dos cidadãos, buscar serviços públicos que resolvam os problemas deles e faça-os sugestões com as melhores opções pode ser desenvolvida para automatizar esse processo.

Uma das possíveis maneiras para obtenção dos dados sobre necessidades pessoais seria coletar as interações que acontecem entre milhões de pessoas nas redes sociais. Essas interações incluem trocar mensagens privadas, realizar postagens na própria página ou fazer comentários relativos a publicações de outros usuários, por exemplo. Essa nova forma de comunicação abre espaço para exteriorização dos sentimentos, necessidades e pensamentos dos seres humanos que são expostos em formato digital. Assim, aproveitando essas características, na proposta da ferramenta é inclusa a coleta das constantes interações textuais dos usuários, as quais representam ações e fatos de seu cotidiano.

Levando em consideração o cenário atual dos serviços públicos, onde suas informações disponíveis não são suficientes para utilização de todos, faz-se necessário realizar um levantamento desses dados para serem repassados às pessoas quando forem detectadas suas necessidades.

Pretende-se obter como resultado a inversão do fluxo de busca da informação, fazendo com que o cidadão torne-se passivo neste processo, e que as alternativas de serviços públicos que mais se adequem às suas necessidades cheguem ao mesmo de forma automática. Esta ferramenta também pode auxiliar na apropriação do cidadão sobre direitos e serviços de uso coletivo, facilitando a divulgação dos mesmos e aproveitando o poder das redes sociais para atingir um número maior de pessoas no seu cotidiano.

Após coleta das postagens são obtidos textos em linguagem natural, que por característica própria possuem diversas

palavras com pouco conteúdo agregado e que podem induzir as detecções aos mais divergentes assuntos. Por esse motivo deve ser realizada uma filtragem para evidenciar as palavras de maior importância, possibilitando obter dessa forma o contexto envolvido na ideia descrita em cada sentença. Esse processo esse é realizado com a utilização da técnica de criação de redes de palavras e a substituição dos termos de mesmo significado, com o auxílio de um dicionário de sinônimos. O produto desse processamento é submetido a um elemento de mineração de textos, o classificador, que relaciona os serviços aos termos mais importantes das sentenças analisadas. A solução então faz as sugestões de serviços adequados aos cidadãos e se aperfeiçoa ao receber a avaliação realizada pelos mesmos, na prática sendo traduzida na atualização da base de dados do classificador, o qual faz uso das novas informações fornecidas pelos usuários.

Este trabalho está estruturado em oito seções, sendo iniciados pela introdução, onde é apresentado o tema, expressados os objetivos, o problema pesquisado, a metodologia empregada, as justificativas do estudo realizado e uma rápida descrição sobre as seções. Nas três próximas seções são explicitados os conceitos das tecnologias essenciais para o entendimento e modelagem da arquitetura proposta, organizados em: redes sociais digitais, mineração de textos e redes de palavras. Em seguida, é relacionado o trabalho com a vertente desse curso, a Computação Ubíqua. Na seção seguinte, os trabalhos correlatos existentes são apresentados.

Na sétima seção a arquitetura é explanada através abordagem da documentação da ferramenta. Na sequência, os módulos da solução são apresentados seguindo o fluxo de funcionamento, evidenciando as partes implementadas. Na conclusão estão os resultados e os trabalhos futuros.

II. REDES SOCIAIS DIGITAIS

Segundo [1], são os serviços WEB que permitem às pessoas criar um perfil com informações restritas ou não dentro de um sistema próprio, personalizar uma lista de conexões compartilhadas com outros usuários e ainda, visualizar e relacionar sua lista de conexões com as que foram feitas por outros usuários dentro do sistema. A natureza e nomenclatura dessas conexões podem variar de rede para rede.

Ainda sobre redes sociais, [2] afirma que as mesmas permitem que toda a sociedade interaja entre si de forma rápida e sem dificuldades utilizando aplicativos para isso.

O mesmo autor cita uma das grandes utilidades das redes sociais, lembrando uma tragédia que ocorreu no Rio de Janeiro, onde as mais diversas pessoas ajudaram na divulgação e batalha por recursos necessários aos que sofreram com o fato.

Além de ser voltada para esses fins, as redes sociais possuem uma vasta gama de informações disponíveis devido a quantidade de usuários interagindo constantemente. As interações dos usuários podem conter de desejos óbvios, como a requisição explícita de ajuda, ou um comentário a respeito de um acontecimento onde o mesmo possui como característica padrão a necessidade de algum serviço.

A. Explorando dados de redes sociais

O potencial das redes sociais já é conhecido há algum tempo e por isso já vem sendo explorado por diversas entidades que vem criando novas redes ou as utilizando como mais um recurso em seus sistemas.

Em [3] foi proposto uma solução de análise de redes sociais, como a plataforma Lattes¹, objetivando a visualização e avaliação da colaboração científica entre os pesquisadores. Neste artigo foram utilizadas técnicas de mineração de dados, inteligência artificial, aprendizagem de máquina e visualização interativa de informações no intuito de realizar análise e descoberta de conhecimento levando em consideração aspectos qualitativos e quantitativos. Esse trabalho foi justificado em razão da necessidade de melhor avaliar a produção científica de um pesquisador e de grupos de pesquisadores que é, baseada fortemente na análise de currículo. Os autores desse artigo citam algumas bases de dados científicos que armazenam dados interessantes que podem ser úteis em diversas aplicações de mineração de dados, como, por exemplo, quando pretende-se identificar padrões de comportamento de pesquisadores, nível de relação com coautores, consultas a históricos de produção científica, trajetória de carreira ou identificação de tendências.

Miyata [4] também fez um trabalho utilizando os dados da plataforma Lattes, detectando as áreas mas quais os autores atuaram.

Em [5] foi feito um estudo para prever usuários influenciadores do Twitter analisando para isso traços de personalidade. Como resultado foram obtidos modelos de previsão com acurácia aceitável.

Outro trabalho que também utilizou como fonte de dados o Twitter foi o de Albuquerque et al. [6]. Nele foram coletadas mensagens em que havia alguma referência a candidatos políticos de uma eleição em João Pessoa. Assim, conseguiu-se prever o eleito e o último colocado, resultado esse melhor que o apresentado pelas pesquisas de intenção de votos.

Na área de detecção de sentimentos foi realizado o estudo [7], onde foi possível identificar tendências e comportamentos típicos dos usuários em momentos e locais diferentes.

III. MINERAÇÃO DE TEXTOS

Mineração de dados é descrita como uma das fases do processo de descoberta de conhecimento em bases de dados e tem por função encontrar padrões nos dados analisados através de algoritmos [8], tradicionalmente os dados devem possuir alguma forma de estruturação para possibilitar sua mineração [9]. Já mineração de textos, conceitua-se como um campo voltado para extração de informações importantes presentes em textos escritos em linguagem natural. Em [10] é caracterizada como o processo de análise textual para extração de informação. A mesma também é definida por Tan [11] como, processo de extração de padrões interessantes e não-triviais ou conhecimento de textos de documentos não estruturados.

¹<http://lattes.cnpq.br>

A. Categorização de texto

Categorização ou classificação de texto, são atribuições de categorias feitas a documentos de linguagem natural de acordo com seus contextos [12]. Um dos diversos métodos existentes para minerar dados, a classificação segundo Chen [13], procura características em comum em objetos das bases de dados e os classifica em diferentes classes, de acordo um modelo de classificação.

Essa técnica é classificada como um tipo de aprendizado supervisionado onde as categorias são determinadas e conhecidas antecipadamente para cada documento do treinamento. Diferentemente do agrupamento de documentos (*clustering document*), que é um aprendizado não-supervisionado na qual não existe categoria pré-definida ou classe, e sim grupos de documentos que devem estar juntos quando procurados. A abordagem predominante tem usado máquinas de aprendizado para inferir categorias automaticamente oriundas de um conjunto de treinamento de documentos pré-classificados. No momento da criação das categorias (concebidas previamente pelo técnico), são definidos apenas rótulos simbólicos sem significado adicional, pois, toda a informação usada na classificação de um documento é a presente em seu conteúdo [10].

Os dados de treinamento compreendem um número substancial de amostras de documentos para cada categoria. Cada documento é usado como uma instância positiva para o rótulo da categoria a qual está associado e uma instância negativa para todas as outras categorias. Abordagens típicas extraem atributos de cada documento e usa os vetores de atributos como entrada para um modelo que aprende como classificar documentos [10]. Em razão de existirem muitas palavras, problemas surgem com alguns métodos de aprendizado de máquina, e um processo de seleção é frequentemente usado para identificar somente alguns atributos aparentes [12].

Em [14] foram realizados experimentos com algumas bases de dados e os seguintes algoritmos de classificação: J48, REPTree, RandomTree e Logistic Model Trees(LMT). Os resultados obtidos apontaram variação nas porcentagens de acerto a depender da base utilizada e o algoritmo mais preciso foi o pior no quesito tempo, obtendo uma diferença sensível em relação aos outros. Esse resultado demonstra que além da taxa de acerto, deve ser observado o tempo necessário para a técnica gerar as previsões.

A classificação pode ser aplicada em uma vasta gama de aplicações, tais como: identificação de linguagem; em alguns casos onde a autoria de textos é incerta pode ser deduzida através do texto do documento, sendo frequentemente tratada como um problema de categorização e na recuperação de informação em bibliotecas, onde já é uma técnica tradicional antiga. [10]

B. Preparação dos textos

Ao lidar com linguagem natural, algumas decisões podem afetar o resultado apresentado. Separar os dados de entrada em palavras - ou *tokens*, conjunto de letras que não necessariamente formam palavras de algum idioma - constitui um passo determinante que necessita análise do idioma para escolher

corretamente quais os caracteres delimitarão cada *token*. Os *tokens* podem ser padronizados utilizando um dicionário para mapear as diferentes ocorrências de um termo para uma forma canônica [10].

Após a entrada ser transformada em *tokens*, algum nível de processamento sintático é requerido. A mais simples das operações realizadas é a remoção das *stop-words*, palavras que executam papéis sintáticos bem definidos mas não carregam informação relevante [10]. Muitos sistemas descartam essas palavras comuns, primariamente por razões de eficiência, embora técnicas apropriadas de compressão eliminam a necessidade disso [10].

Há duas formas de atribuir pesos a palavras que costumam ser mais efetivas: primeiro, de acordo o número de documentos em que elas aparecem (frequência de documento), normalmente as palavras frequentes carregam menos informação do que as raras; segundo, pelo número de vezes que elas aparecem em cada documento (frequência do termo) [10].

Um passo posterior a realizar é a análise semântica e marcar cada palavra com suas partes do discurso. Isto ajuda a tirar ambiguidades de uma palavra e eliminar análises incorretas causadas por palavras de sentidos incomuns [10].

IV. REDES DE PALAVRAS

No trabalho de [15], foram criadas redes de palavras com os discursos orais de estudantes coletados durante a fase de pesquisa. Após capturados, os discursos sofreram conversão para um formato digital e passaram por processamentos visando retirar palavras gramaticais - palavras que carregam consigo pouca semântica. Esse processo de limpeza dos textos foi uma aprimoração àquela feita por [16], que criou algumas ferramentas para complementar o produto gerado pelas ferramentas de tratamento de textos em língua humana, Unitex. Esse conjunto de programas é disponibilizado pelos laboratórios RELEX e, assim como as criações de [16] e suas modificações feitas por [15], podem ser livremente distribuídos. Apesar do trabalho de [15] fazer a limpeza para outros objetivos diferentes da mineração de textos, ambos possuem as mesmas dificuldades iniciais ao lidar com textos escritos, os ruídos advindos das palavras e outros caracteres utilizados por cada idioma na formação de sentenças. Em razão dessa dificuldade em comum, boa parte da extração de impurezas dos textos que seria feita na mineração de textos torna-se dispensável, pois, o conjunto de ferramentas citadas acima já o fizeram.

V. COMPUTAÇÃO UBÍQUA

[17] define como a união de duas áreas da computação, utilizando a mobilidade dos serviços computacionais da computação móvel aliado à capacidade da computação pervasiva de através de uma computação embarcada no ambiente de forma invisível, detectar as variáveis do ambiente e modificar características do conteúdo fornecido aos componentes presentes no ambiente, sem que haja solicitação humana.

Em [18], é descrita como a computação que move-se para fora das estações de trabalho e computadores pessoais (PCs) e torna-se pervasiva em nossa vida cotidiana. Foi previsto

em [19] que os objetos mais triviais teriam computadores embarcados imperceptíveis ao usuário e com o potencial de transformar o modo de utilização, não sendo mais necessário ir ao encontro das máquinas para ter acesso as suas funcionalidades. Assim, o principal objetivo de acordo [20] é disponibilizar a computação por todo o ambiente físico sem que o usuário a enxergue.

Dentre as características relacionadas à computação ubíqua, algumas podem ser encontradas nesse trabalho como: sensibilidade à posição - no momento que a ferramenta detecta o local onde o usuário se encontra e propõe o serviço mais próximo -, consciência de contexto - com a interpretação das mensagens coletadas das redes sociais - e a computação invisível - o usuário interage normalmente na rede social sem perceber a computação ser realizada e recebe a notificação somente quando há algo interessante para ele, dessa forma há uma mudança da relação de interação homem-máquina, tornando o homem passivo.

VI. TRABALHOS CORRELATOS

Não foram encontrados artigos na área envolvendo serviços públicos e mineração de redes sociais, no entanto, existem bastante trabalhos focados em minerar redes sociais com objetivos particulares voltados a outras áreas, como o varejo. Existem dois serviços disponíveis no mercado que se assemelham a proposta apresentada nesse trabalho no que se refere a execução de ações ao ocorrer determinados eventos, sejam eles a ocorrência de determinado texto ou uma outra ação. A área de detecção de desejos e sentimentos ainda possui campo para estudo, visto que os trabalhos existentes não possuem precisão suficiente para serem utilizadas com total confiança na veracidade dos dados obtidos.

O Google possui um sistema para veiculação de anúncios nos seus produtos, como por exemplo: Youtube, Gmail e na página de pesquisa. A definição do público-alvo é feita através do relacionamento das palavras-chave ou tópicos referentes aos anúncios, com o tema central ou termos presentes nas páginas. Em [21] são definidos, Palavras-chave são palavras individuais, enquanto tópicos são os conceitos ou o tema central de uma página da Web, e não seus termos individuais. Assim, para realizar a análise do conteúdo das páginas WEB consideram-se características como: texto, idioma, estrutura de links e a estrutura da página. Já para exibir os anúncios são levados em consideração fatores como idioma, local de destino e histórico de navegação recente da pessoa que está visualizando a página.

No Gmail, os anúncios estão relacionados ao conteúdo da Conta do Google do usuário. Onde é usada uma abordagem com critérios semelhantes aos usados para detectar quais mensagens podem ser importantes [22]. Segundo o site [23], para classificar a importância das mensagens são verificadas: quais mensagens são abertas, respondidas e os destinatários mais frequentes, palavras-chave que despertam o interesse, e as mensagens que recebem estrelas, são arquivadas e excluídas. Este tipo de verificação é automática e é o modo como muitos

serviços de e-mail fornecem recursos como filtragem de spam e verificação ortográfica.

Dentre as metas que as empresas que tem anúncios veiculados pelo Google estão: o aumento das reservas financeiras, diminuição do custo por conversão, segmentar nichos de público alvo, aumentar o conhecimento e engajamento da marca, reconquistar compradores, aumentar a taxa de conversão, vender mais produtos ou serviços, estimular a lealdade de clientes, interagir com clientes e aumentar o conhecimento da marca [24].

Outra ferramenta com características parecidas com a proposta desse trabalho é o Google Now, um assistente pessoal desenvolvido para ser executado em Smartphones e está disponível para os sistemas operacionais IOS e Android. Utilizando interfaces para se conectar a diversos WEB Services, obtém informações com potencial para serem determinantes para os usuários realizarem suas tarefas cientes das melhores opções disponíveis e ainda estarem bem informados com os conteúdos de seus interesses.

Através das experiências vivenciadas com o uso do celular e ainda com as pesquisas realizadas no buscador da Google, as informações são filtradas e direcionadas de acordo às preferências.

O site [25] o conceitua como um serviço que tem função de organizar a rotina do usuário, mostrando previsões do tempo, trânsito e notícias de acordo com cada perfil. Além disso, o Now também responde às perguntas feitas pelo usuário com informações completas e antecipa algumas atividades, como shows e viagens para ajudar o dono do aparelho.

Em [4] é descrito um estudo para a detecção automática das áreas às quais os autores de artigos pertencem, utilizando para isso as informações contidas na base de dados de currículos da plataforma Lattes. Para alcançar o objetivo do mesmo é utilizada a combinação das técnicas de mineração de textos e análise de redes sociais para detectar a qual área cada autor pertence.

Já em [7] foram utilizadas mensagens do Twitter para detectar os sentimentos das pessoas no momento da postagem. A estratégia descrita consiste em, através de expressões regulares, realizar buscas por referências feitas aos sentimentos levando em consideração variações de escrita, como exemplo: "feliiz". Para melhorar a acuracidade dos dados coletados foram também pesquisados nas frases, a existência de termos que evidenciam o relato do estado do usuário, "tô" e "fiquei" são palavras que transmitem essa ideia, neste caso específico.

Os dados coletados foram enriquecidos com informações de geolocalização e condições meteorológicas. Com os resultados foram feitas análises estatísticas relacionando os sentimentos a eventos importantes, ao clima no exato momento e o estado onde foi feita a postagem propiciando diferentes estudos e formas de agrupamento dos dados.

Em [6] foram utilizadas postagens do Twitter que continham palavras referentes aos candidatos - palavras-chave (*hashtag*), nome da conta, id e termos de campanha - de João Pessoa. Com dados do Censo, foi possível definir os bairros com maior poder aquisitivo e assim maior acesso às redes sociais.

Após a captura foram utilizadas métricas de redes sociais para descobrir os candidatos mais citados na rede. Também foram coletados dados de localização das postagens na tentativa de visualizar os votos em cada bairro mas, a maioria das mensagens não continham informações suficientes para isso, o que impediu a descoberta das áreas mais influentes. Finalmente, obteve como resultado o acerto do vencedor, porém o terceiro lugar real foi o segundo nas pesquisas e vice versa. A provável justificativa do erro foi a falta de uma análise mais detalhada das mensagens para verificar se eram menções positivas ou não.

Gadelha et al [5] coletou, mediu e relacionou traços de personalidade presentes nos tweets e indicadores do Twitter desses mesmos usuários (mensagens encaminhadas, menções e respostas) afim de encontrar relação desses valores com os usuários influenciadores e posteriormente utilizar modelos matemáticos para prever os valores. Pôde estabelecer relação entre traços de personalidade com as pessoas influenciadoras. Para extrair as informações do Twitter utilizou a API oficial da rede social, onde através dela foram coletados dois conjuntos de dados com informações dos perfis de usuário e suas últimas 200 postagens. Em ambos os conjuntos, filtrou-se pelo idioma (inglês), pela privacidade da conta (pública) e número de tweets postados (mais de 200). Durante o agrupamento de dados de usuários comuns, escolheu-se um único usuário para obter todos os seus amigos, repetindo-se esta ação iterativamente para cada amigo do amigo, até obter um milhão de usuários (somente os identificadores de cada usuário). Além do filtro comum foram excluídos usuários que seguiam celebridades, sendo estas últimas caracterizadas por número de seguidores maior que quinze mil.

VII. DOCUMENTAÇÃO DA FERRAMENTA

Nesta seção serão apresentados os requisitos para construção da aplicação e diagramas sob diversas visões para facilitar a compreensão da proposta.

A escolha da rede social é de extrema importância, pois, cada uma possui suas particularidades, necessitando que sejam realizadas modificações no projeto para se adequar às necessidades trazidas com a rede. Assim, em meio a diversidade de redes sociais existentes, um dos itens verificados foi a popularidade das redes, utilizando para isso a ferramenta online Alexa², onde são mantidas informações estatísticas de acesso aos sites no mundo e disponibilizados rankings de acordo diversos critérios. Ao realizar uma consulta à lista das páginas mais acessadas mundialmente, verifica-se que o segundo colocado é o Facebook, perdendo apenas para o Google. Confirmando a rede como a mais popular, [26] expõe os resultados de uma pesquisa feita por uma ferramenta de inteligência da Serasa Experian, a Hitwise, onde mais uma vez o Facebook aparece em primeiro colocado dentre a preferência brasileira em relação às redes sociais. As pesquisas

²Alexa internet é uma empresa da Amazon que faz listas ordenadas calculadas usando uma metodologia própria que combina a média diária de visitantes e estima o número de visualizações das páginas em um intervalo de 3 meses. Fonte: www.alexa.com

demonstram que o primeiro colocado do ranking é o mesmo desde Janeiro de 2012. As etapas abaixo descritas serão influenciadas pelo modo como a empresa permite acesso aos dados dos seus usuários. Já se tratando das possibilidades trazidas com a rede, com pouco tempo de utilização, pode-se verificar que a mesma permite utilizar diversos recursos de comunicação, seja através de figuras, vídeos, endereços ou textos. Além disso, é possível utilizar os tipos de mídias de forma flexível, deixando o usuário interagir da forma que achar melhor. Todas essas opções de expressão oferecidas, dão oportunidade a diversas abordagens para minerar informações. Após a exposição dessas características e ainda levando em consideração a possibilidade de extração de dados através de diversas linguagens de programação, a rede social adotada foi o Facebook.

A. Requisitos

Alguns cuidados devem ser tomados para que a aplicação se torne útil aos usuários e ainda seja executada sem modificar drasticamente o fluxo de utilização da rede nem atrapalhar as ações do usuário. Foram levantados requisitos primordiais e classificados de acordo como obrigatórios - funcionais - e desejados - não funcionais, tabela I.

Tipo	Requisito
Funcional	O cidadão deve ter uma conta no Facebook
Funcional	Deve acessar a aplicação conectado à internet
Funcional	Deve acessar a aplicação através da rede social
Funcional	Deve permitir acesso a seus dados
Funcional	Os dados devem ser extraídos das redes automaticamente
Funcional	Devem ser feitas sugestões de serviços públicos com base nas interações feitas pelos usuários
Não Funcional	O sistema deve aprimorar-se com o tempo
Não Funcional	O sistema deve fazer sugestões sem interferir no fluxo normal de utilização da rede, permitindo ao usuário escolher se aceita a sugestão

Tabela I
LISTA DE REQUISITOS

B. Diagramas

Para uma melhor compreensão do funcionamento do sistema foram feitos diagramas que demonstram o fluxo, as atividades e os usuários agindo durante todo o ciclo de funcionamento da aplicação.

1) *Diagramas de caso de uso:* Através de diagramas de caso de uso são demonstradas ações executadas pelos atores do sistema. São eles: administrador e usuário. O ator usuário é essencial no fluxo do sistema, assim como a permissão de acesso a seus dados. Na Figura 1 são exibidas as ações desse ator que serão logo em seguida descritas.

A primeira das ações realizadas para que o sistema tenha acesso aos dados, possa enviar notificações e assim fazer sugestões, é **vincular a rede ao sistema**. Essa ação é essencial devido ao sistema utilizar essas informações restritas somente a aplicações autorizadas. **Postar conteúdo**, apesar de ser comum a todos os usuários da rede, é essencial para que a ferramenta tenha utilidade, é a partir do conteúdo gerado que

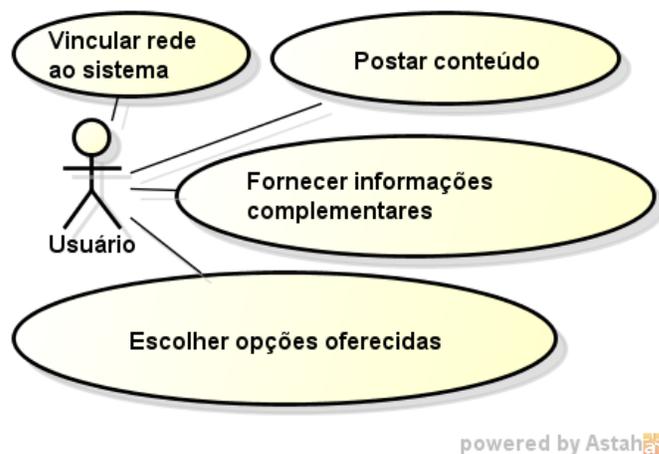


Figura 1. Caso de uso - Usuário

serão classificadas as ideias contidas nas mensagens e assim o usuário poderá ser ajudado. No entanto, essas não são as únicas ações necessárias, devido a dificuldade em detectar a melhor opção para o usuário, o mesmo precisa primeiro, **fornecer informações complementares**. Após a detecção de novas interações o sistema irá eleger serviços candidatos que devem ter alguns testes relacionados a eles para que sejam apresentados ao usuário ou descartados. Esses testes serão perguntas que devem ser respondidos pelos usuários com respostas pré-determinadas pelo administrador do sistema. Finalmente, após o sistema verificar quais são as opções mais adequadas, exibirá as sugestões para o usuário **escolher dentre as opções oferecidas**. Essa ação engloba o aceite, caso as informações apresentadas sejam úteis ou rejeição, caso contrário. O descarte das sugestões poderá ser benéfico para a ambos quando o usuário dá um retorno, sugerindo um serviço que melhor atenda a sua necessidade, pois, ajudará também no refinamento do sistema.

O usuário administrador possui um papel tão importante quanto o vínculo da aplicação, ele configura o sistema para gerar sugestões e aprimorá-las. Antes de qualquer contato com o público, o administrador deve **adicionar serviços públicos** ao sistema juntamente com suas informações relevantes para que o usuário ao receber a sugestão já possa se beneficiar sem consultar nenhum outro meio e, também, **relacionar esse serviços a termos** da língua portuguesa. Outra função do ator é coletar a avaliação dos usuários e periodicamente **atualizar o classificador**. Na Figura 2 é demonstrada como o Administrador do sistema inicia e mantém o sistema em funcionamento.

Após o usuário postar uma nova mensagem, é executada uma sequência de tarefas automaticamente. Com esse gatilho o sistema começa a **interpretar a interação**, ou seja, armazenar as mensagem e filtrar as palavras mais relevantes. Após terem sido selecionados os termos, são usados para **localizar serviços públicos** relacionados para **fazer sugestões** aos usuários. A figura 3 demonstra como as ações automáticas



Figura 2. Caso de uso - Técnico

são realizadas.

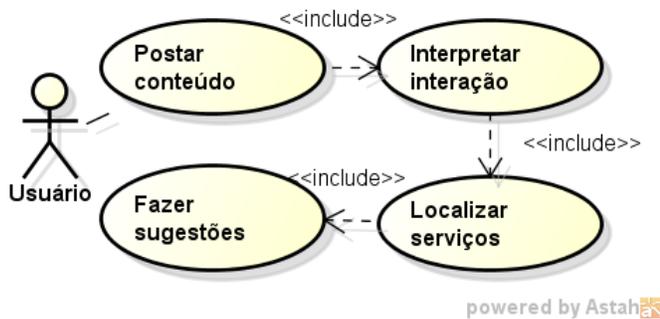


Figura 3. Caso de uso - Tarefas automáticas

2) *Diagrama de arquitetura em blocos didáticos*: Para facilitar a compreensão das conexões e das tarefas executadas pelo sistema um diagrama de arquitetura dividido em blocos didáticos foi desenvolvido. O ciclo de execução da ferramenta começa após o usuário permitir acesso aos seus dados no módulo aplicação. A partir daí, as novas postagens serão capturadas pelo módulo de coleta e logo após armazenado em um banco de dados pelo módulo de persistência. Na sequência, o módulo de filtro de palavras reduz o conjunto de palavras possíveis e filtra as mais importantes. Após selecionadas, as palavras serão enviadas ao módulo de mineração de textos para que seja feita uma consulta em busca de serviços relacionados, para que sejam sugeridos ao usuário autor da postagem pelo módulo de sugestões. Posteriormente, o usuário poderá validar confirmando ou não a sugestão oferecida pela aplicação. Caso não seja, uma lista de serviços é disponibilizada ao usuário

para a escolha do serviço que mais se adequa, para que o módulo de avaliação possa com base na resposta do usuário, ajudar no aperfeiçoamento do módulo de mineração de textos. Esse fluxo é demonstrado através do diagrama apresentado na Figura 4, cujos módulos serão descritos na seção VIII, visando melhor entendimento da solução.

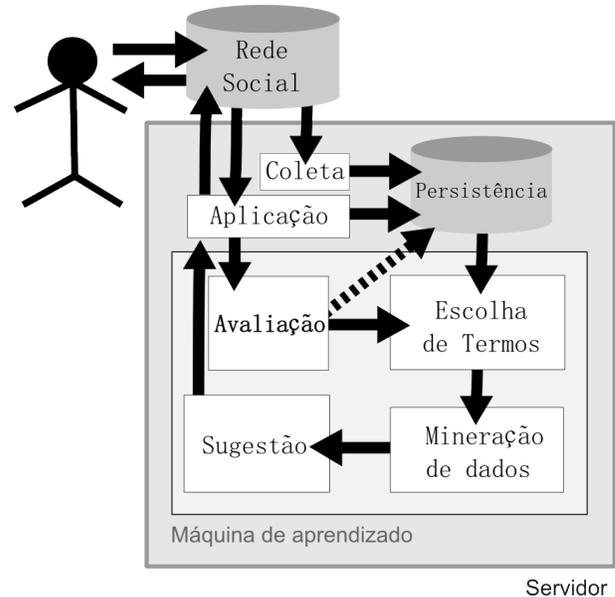


Figura 4. Diagrama de Arquitetura em Blocos Didáticos

3) *Fluxograma de exemplos*: Um fluxograma foi feito para demonstrar através de exemplos, como a utilização de perguntas pode ser feita para direcionar o serviço correto a uma pessoa que realizou interação na rede. Essa visão expõe algumas possibilidades de como o usuário poderá agir na aplicação e os rumos que podem tomar o processo ao variar as respostas. Na Figura 5 é possível visualizá-lo.

4) *Diagrama estrutural*: Seguindo a ideia de modularidade do diagrama presente na Figura 4, foi escolhido o padrão arquitetural *Three-tier* ou três camadas³ pela popularidade e por atender às necessidades do projeto, proporcionando independência entre as camadas e melhora na construção por permitir que o desenvolvedor trabalhe pensando sob visões diferentes em cada camada. Na Figura 6 é exibida estrutura.

5) *Diagrama de implantação*: No intuito de estimar a configuração física das máquinas necessária para a utilização do sistema, foi feito uma diagrama de implantação onde são demonstradas a forma de conexão entre os componentes e os requisitos para cada um. É possível verificá-lo na Figura 7.

6) *Diagrama de sequência*: O fluxo da aplicação pode ser visto através do diagrama de sequência que exhibe a ordem e quais componentes interagem entre si. Para facilitar o entendimento foram criados quatro atores: usuário, rede social,

³Arquitetura composta das camadas de interface do usuário, lógica da aplicação e banco de dados. Seu objetivo é resolver um número de problemas recorrentes de design e desenvolvimento, consequentemente torna o trabalho de desenvolvimento da aplicação mais fácil e eficiente [27].

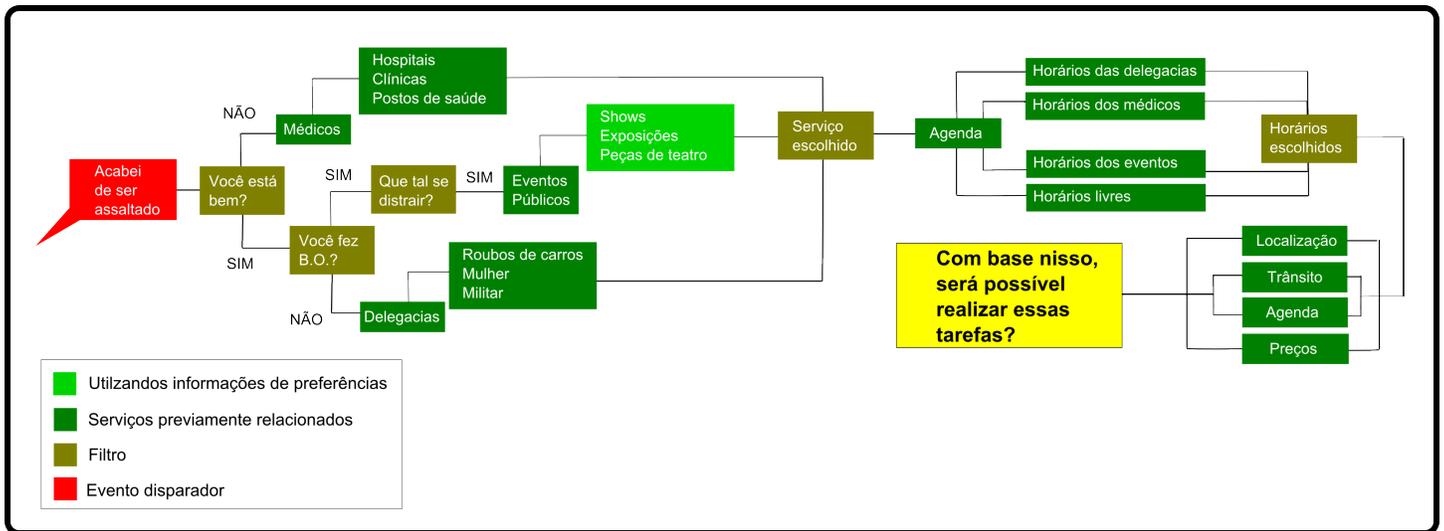


Figura 5. Fluxograma de exemplos

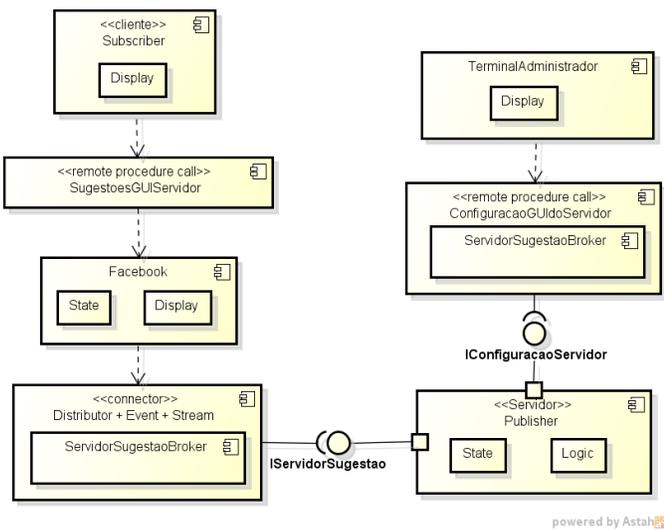


Figura 6. Diagrama estrutural de arquitetura

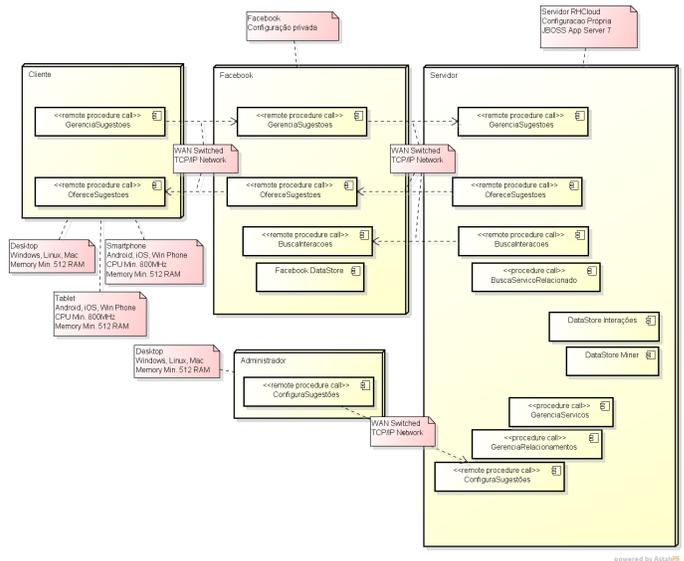


Figura 7. Diagrama de implantação de arquitetura

coletor (JOB) e o gerador de sugestões. Devido ao seu tamanho foi dividido nas Figuras, 8 a) e 8 b).

VIII. MÓDULOS DA FERRAMENTA

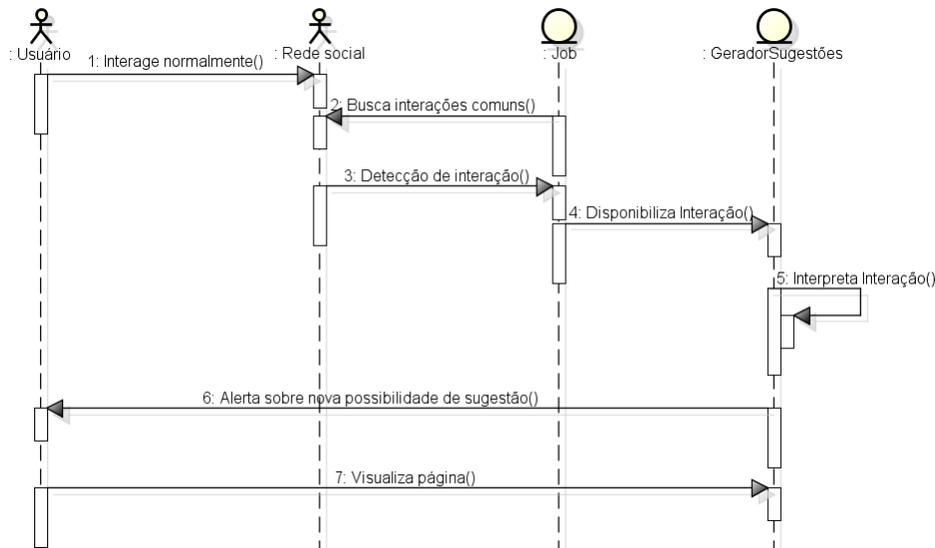
Nesta seção são explanados os detalhes dos módulos da Figura 4 citados superficialmente na seção VII. Embora não seja o objetivo deste trabalho, algumas partes dele foram implementadas, o que proporcionou um maior conhecimento a respeito das tecnologias, comprovação da viabilidade de colocar em prática a solução e caso alguém resolva seguir essa proposta, terá menos trabalho a fazer. Nos módulos onde houve implementação há uma seção onde é apresentado o fluxo de funcionamento e configurações necessárias sob a perspectiva das funcionalidades presentes em cada módulo. Adicionalmente os blocos funcionais são classificados em: im-

plementados por terceiros (bloco branco), a implementar (bloco cinza) e implementados na para a solução (bloco preto).

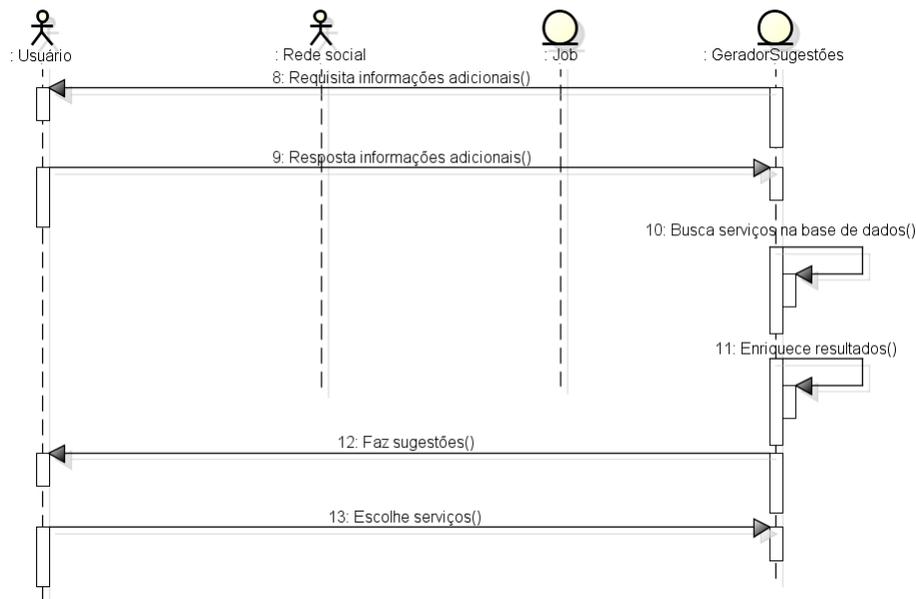
A. Rede Social

A rede social possui dois papéis no projeto. O primeiro é o de base de dados, onde serão coletados dados periodicamente através de uma aplicação criada para esse fim. O segundo papel é o de interface com o usuário, permitindo que os usuários sejam comunicados sobre a existência de novas sugestões e interajam com a aplicação.

1) *Aplicações do Facebook:* Na página de documentação do Facebook [28], as páginas canvas representam o contexto onde as aplicações Web são carregadas. É possível desenvolver essas aplicações usando qualquer linguagem ou conjunto



(a) Diagrama de Sequencia - Parte 1



(b) Diagrama de Sequencia - Parte 2

Figura 8. Diagrama de sequência: Na Figura a) é exibido o fluxo em ordem, partindo da interação na rede até o momento em que o usuário acessa a aplicação. Já na Figura b), são exibidas as ações realizadas entre o usuário e aplicação.

de ferramentas que suporte programação Web, como PHP, Python, Java ou C#.

Ainda segundo [28], o primeiro passo para criar uma aplicação é entrar na página destinada a desenvolvedores e fazer a solicitação. Em seguida, serão solicitadas algumas informações pertinentes à nova aplicação a ser criada.

A aplicação é aberta dentro de uma janela interna do site do Facebook (Canvas) - área hachurada da Figura 9 - e mostrará as opções de sugestões.

No Apêndice G estão as informações relativas a implementação no Facebook.

B. Módulo de Aplicação

Esse módulo tem a função de interagir com o usuário, oferecendo sugestões, recebendo as respostas para melhor direcionar os serviços aos usuários e coletando a avaliação feita por cada indivíduo insatisfeito - pode ser simplesmente aceitar uma sugestão ou indicar uma diferente. As pessoas só poderão usar a aplicação após vinculá-la ao seu perfil.

1) *Vínculo do aplicativo*: O aplicativo fará uso de dados dos usuários relativos a cadastro e utilização das redes sociais, por isso o usuário deve acessar o endereço do aplicativo, onde será informado quanto ao vínculo da aplicação ao seu perfil e

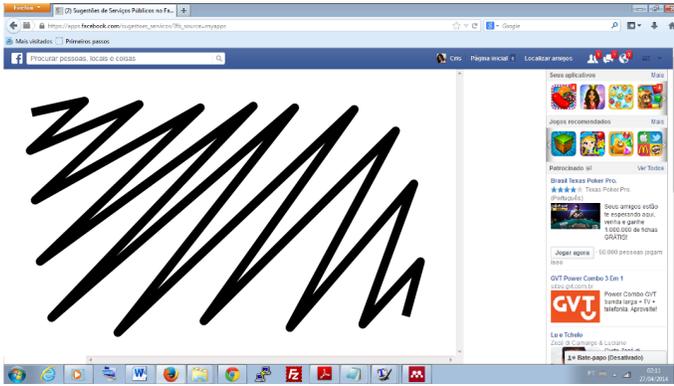


Figura 9. Canvas

quais tipos de dados o aplicativo terá acesso e ainda, inquirido quanto a permissão para que a ação seja concluída, um fator que torna tão importante essa solicitação é o de muitos dados estarem sujeitos a políticas de privacidade.

Há diversos tipos de dados pessoais disponíveis no Facebook, com isso, aumentam-se as possibilidades de utilização dessas informações e também as responsabilidades com o conteúdo disponibilizado. Para reduzir os riscos de os dados dos usuários ficarem acessíveis a pessoas mal intencionadas, o Facebook impõe restrições para liberar as aplicações acessarem determinados dados, sendo necessário explicar como serão usados. Somente é permitido à aplicação solicitar acesso ao perfil público, lista de amigos e endereço de e-mail de uma pessoa sem que seja necessário solicitar revisão da empresa [29].

2) *APIs do Facebook:* Antes de começar a desenvolver uma aplicação do Facebook é imprescindível conhecer as ferramentas disponíveis para manipular os dados contidos na rede. No apêndice E, foi feita uma abordagem sobre algumas das opções que podem ser utilizadas.

3) *Implementação:* Dentre as funcionalidades presentes nesse módulo estão as relacionadas à parte de autorização, onde é verificado se o usuário deu permissão de acesso a seus dados, caso não tenha dado, é solicitado que o mesmo autorize o acesso e a utilização dessas informações. Por fim, é gerado um *Token*, o qual é armazenado no banco de dados da ferramenta. Essas funcionalidades são exibidas na Figura 10 dentro dos retângulos na cor preta.

C. Servidor

Após cadastrar na rede social os endereços (*links*) usados para acessar a aplicação e escolher as ferramentas para desenvolvê-la, é necessário configurar um computador para hospedar as páginas da mesma e disponibilizar acesso a elas via internet. Esse passo é devido ao Facebook não possibilitar que o conteúdo dos aplicativos de terceiros seja armazenado em seus servidores. Outra razão da hospedagem do aplicativo acontecer em um servidor separado é não limitar a solução a uma rede social específica. Além de persistir os arquivos, é necessário processar o conteúdo do lado servidor, para

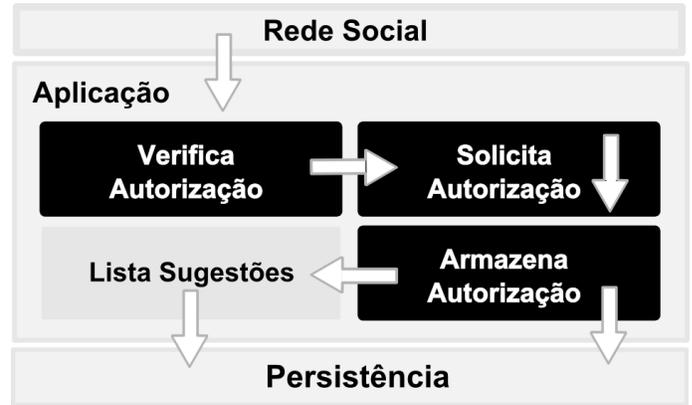


Figura 10. Funcionalidades Implementadas do Módulo de Aplicação - Verificar, solicitar e armazenar autorização

isso, deve haver algum servidor WEB, como Apache, caso a linguagem escolhida para desenvolver a aplicação seja PHP ou JBoss ou GlassFish, caso seja utilizado Java WEB. Uma plataforma que atende a solução é o RHCloud, uma opção gratuita de servidor WEB, que pode processar as duas linguagens citadas e permite com poucos cliques escolher os recursos a utilizar. Outro requisito para a aplicação do Facebook obter o conteúdo a ser exibido no Canvas é que a página possa ser acessada externamente através de um endereço público, citado no Apêndice G, o Canvas URL.

A hospedagem da aplicação em servidores externos ainda prevê independência da rede, ou seja, podem ser extraídos do Facebook e Twitter por exemplo, armazenar no mesmo banco de dados e na mesma base de mineração de dados.

D. Módulo de Coleta

Após ter vinculado a aplicação ao seu perfil e, assim, autorizado acesso aos seus dados, o módulo de coleta começa a coletar as interações do usuário para que sejam submetidas a análise.

O funcionamento da ferramenta consiste de uma rotina executada periodicamente (*JOB*), realizando consultas aos servidores do Facebook em busca de novas interações dos usuários. Inicialmente, o *JOB* somente procurará por postagens dos usuários na sua própria linha do tempo, caso seja necessário pode-se aproveitar qualquer outra interação através de texto, como mensagens privadas e na linha do tempo de outros usuários. Dentro da rotina serão feitas chamadas a Graph API do Facebook para extrair os dados da rede, em razão de que em breve será a única com suporte oficial.

Após escolher a ferramenta de extração faz-se necessário definir como será a estratégia de coleta, que visando poupar recursos, buscará apenas as interações ainda não armazenadas no banco de dados do sistema aqui proposto. Para isso, será necessário guardar a data da última coleta e implementar um filtro que permita solicitar ao Facebook somente as postagens realizadas após este dia. Assim, a funcionalidade de Paginação Temporal do Facebook, abordada no Apêndice E, será usada para trazer esses dados.

Outra definição que é necessária ser feita é o método de obtenção da localização do usuário, já que, a mesma servirá como referência para indicar ao usuário a localização do órgão mais próximo que preste o serviço sugerido. Para atendê-lo é necessário obter como referência a posição global do usuário. Um recurso que o Facebook possui para auxiliar a recuperação desse dado é a possibilidade de atrelar informações de geolocalização às postagens. No entanto, a utilização de somente essa estratégia para determinar o ponto de referência não é suficiente, pois, muitas vezes são enviadas mensagens sem marcar uma localização e ainda que seja atribuída pode não ser a melhor opção para o usuário. Assim, é proposto que seja verificado se está disponível essa informação em cada interação, checando o campo *place*⁴, e no momento da sugestão possibilitar ao usuário escolher outro ponto de referência caso deseje.

Para criação do modelo de classificação optamos por utilizar um conjunto de exemplos empíricos que representam postagens cotidianas informando acontecimentos e problemas pessoais dos usuários das redes. O mesmo é apresentado no Apêndice D. Com esta base, serão selecionados e classificados termos a fim de extrair alguma necessidade de seus autores, para em seguida, serem organizadas de acordo os serviços públicos relacionados a elas. Esse trabalho manual de seleção e organização, servirá como entrada para criar a base de dados do classificador, presente no módulo de mineração de textos. A utilização desses dados ajudará os usuários que utilizem a aplicação e também contribuirá para o aperfeiçoamento do sistema.

1) *Implementação*: Das funcionalidades presentes nesse módulo, foi implementada a coleta de interações dos usuários nas redes sociais - o retângulo preto visto na Figura 11. Para completar o desenvolvimento dessa parte da ferramenta proposta, resta apenas ser feita a rotina para buscar os *Tokens* salvos no banco de dados e assim, capturar automaticamente o conteúdo postado por cada usuário que permitiu acesso a seus dados.

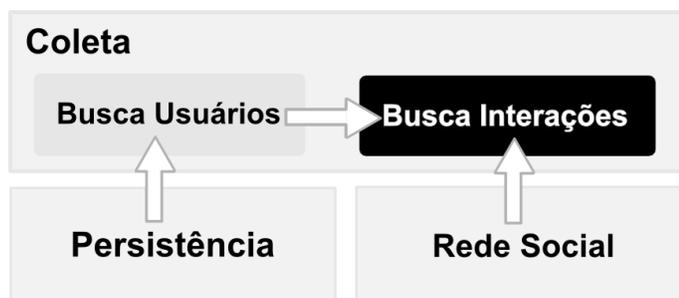


Figura 11. Funcionalidades Implementadas do Módulo de Coleta - Buscar interações

A coleta é realizada por uma rotina desenvolvida em java que periodicamente é executada seguindo um agendamento feito

⁴Mais informações: <https://developers.facebook.com/docs/graph-api/reference/v2.0/user/feed>

com o CRON⁵.

O funcionamento da rotina consiste em buscar no banco de dados os tokens armazenados para obter acesso às informações do usuário na rede, em seguida, é feita uma pesquisa por novas interações do mesmo e caso existam, são armazenadas no banco de dados do sistema.

E. Módulo de Persistência

Cada interação do usuário é armazenada em um banco de dados modelado para persistir essas informações. Adicionalmente, nesse mesmo banco são registradas as credenciais dos usuários, assim como dados de escolhas realizadas pelos usuários anteriormente. A modelagem de dados é exibida na Figura 12, enquanto o dicionário de dados é descrito abaixo e através das tabelas II, III, IV, V, VI, VII, VIII e IX.

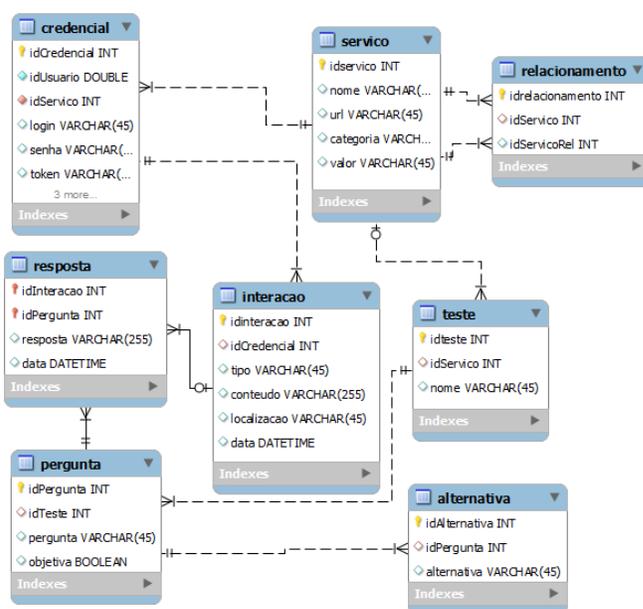


Figura 12. Modelo de dados do banco

1) *Resposta* : São registradas as respostas dadas pelos usuários para as perguntas feitas nos testes. Na Tabela II são descritos os campos.

Nome	Tipo	Descrição
idInteração	Inteiro	Identificador único da resposta
idPergunta	Inteiro	Referência para a pergunta
resposta	Varchar(255)	Texto da resposta
data	DATETIME	Data de resposta à pergunta

Tabela II
RESPOSTA

2) *Interação*: As interações de cada usuário nas redes sociais são registradas e informações como tipo e o próprio conteúdo são armazenadas. Na Tabela III são descritos os campos.

⁵<http://www.infowester.com/linuxcron.php>

Nome	Tipo	Descrição
idInteracao	Inteiro	Identificador único da interação
idCredencial	Inteiro	Referencia para credencial
tipo	Varchar(45)	Tipo da interação, o valores que pertencem ao domínio são: 'FOTO', 'VIDEO', 'MENSAGEM INBOX' e 'MENSAGEM TIMELINE'
conteudo	Varchar(500)	A informação textual importante
localizacao	Varchar(255)	Localização do usuário no momento da postagem
data	DATETIME	Data da interação

Tabela III
INTERAÇÃO

3) *Pergunta*: São armazenadas as perguntas que são feitas aos usuários quando são os serviços são escolhidos. Na Tabela IV são descritos os campos.

Nome	Tipo	Descrição
idPergunta	Inteiro	Identificador único da pergunta
idTeste	Inteiro	Referencia para o teste relacionado
pergunta	Varchar(45)	Texto da pergunta
objetiva	Booleano	Flag que define se é objetiva ou subjetiva

Tabela IV
PERGUNTA

4) *Alternativa*: As perguntas que são feitas aos usuários podem ser objetivas ou subjetivas, no primeiro caso, são armazenadas nessa tabela as possíveis alternativas que os usuários podem escolher. Na Tabela V são descritos os campos.

Nome	Tipo	Descrição
idAlternativa	Inteiro	Identificador único da alternativa
idPergunta	Inteiro	Referencia para a pergunta relacionada
alternativa	Varchar(45)	Texto da alternativa

Tabela V
ALTERNATIVA

5) *Serviço*: Nessa tabela são armazenadas as informações que viabilizam que cada usuário tenha acesso aos serviços disponibilizados. Dentre as informações estão dados de acesso, nome de serviço e algum valor a ser salvo como resultado de uma consulta ao serviço. Na Tabela VI são descritos os campos.

Nome	Tipo	Descrição
idServico	Inteiro	Identificador único do serviço
nome	VARCHAR(45)	Descrição do serviço
url	VARCHAR(45)	Endereço eletrônico do serviço
categoria	VARCHAR(45)	Categoria do serviço
valor	VARCHAR(45)	Valor do serviço, caso haja

Tabela VI
SERVIÇO

6) *Relacionamento*: Essa tabela permite armazenar os relacionamentos entre os serviços. Na Tabela VII são descritos os campos.

Nome	Tipo	Descrição
idRelacionamento	Inteiro	Identificador único do relacionamento
idServiço	Inteiro	Referência para o serviço
idServiçoRel	Inteiro	Referência para um serviço relacionado

Tabela VII
RELACIONAMENTO

7) *Teste*: Essa tabela armazena os testes aplicados aos usuários quando um usuário adere a um serviço. Na Tabela VIII são descritos os campos.

Nome	Tipo	Descrição
idTeste	Inteiro	Identificador único do teste
idServico	Inteiro	Referencia para serviço
nome	Varchar(45)	Nome do teste

Tabela VIII
TESTE

8) *Credencial*: Essa tabela armazena as informações dos usuários em cada serviço, como: *login*, senha(usada somente se não houver outro modo de acessar o serviço, por exemplo, utilização de token), *token* e data de coleta. A descrição dos campos encontra-se na Tabela IX.

Nome	Tipo	Descrição
idCredencial	Inteiro	Identificador único da credencial
idServico	Inteiro	Referência para o serviço relacionado
idUsuario	Double	Referência do usuário no serviço
nome	Varchar(45)	Nome do usuário no serviço
login	Varchar(45)	Nome de acesso ao serviço
senha	Varchar(45)	Senha de acesso ao serviço
token	Varchar(500)	Token de acesso ao serviço
dtValidade	DATETIME	Data de validade do token de acesso
dtUltimaColeta	DATETIME	Data da última coleta

Tabela IX
CREDENCIAL

F. Módulo de Escolha de Termos

Devido a grande quantidade de palavras e possibilidades de combinações existentes no idioma português, é possível que os mesmos desejos sejam expressos com frases contendo palavras e combinações diferentes, em contextos diferentes, tendo em comum apenas a necessidade do usuário. Isto dificulta a interpretação da ideia intrínseca ao texto. Para isso será utilizado um dicionário de sinônimos e em seguida um filtro de palavras. Esse módulo realiza sua tarefa após o armazenamento na base de dados das postagens dos usuários que o acesso a seus dados.

1) *Dicionário de sinônimos*: Através da utilização de um dicionário de sinônimos, será feita a substituição das palavras de mesma semântica por uma única pertencente ao mesmo grupo sinonímico, reduzindo a necessidade de utilização de recursos computacionais para processar elementos que, para o objetivo do projeto são considerados duplicidades. A base de dados do dicionário será a TeP 2.0 [30], resultado de alguns

trabalhos acadêmicos que construíram um Thesaurus⁶ em português. Uma aplicação capaz de ler os arquivos contendo os sinônimos deve ser feita, a mesma deve receber como entrada um texto e passar todas as palavras para uma forma unificada, que pode ser uma palavra de fato ou a identificação numérica do grupo do grupo o qual ela pertence.

2) *Filtro de palavras*: Após ter transformado as palavras de mesmo significado, ocorrerá a filtragem das mais importantes. O processo consiste da compilação em um arquivo texto, de diferentes sentenças que tem em comum a necessidade do mesmo serviço. Após essa junção, o arquivo passa por um conjunto de aplicativos que leem seu conteúdo e formam uma rede de palavras com informações estatísticas quanto a incidência de cada par de palavras. Dentre os arquivos gerados na formação da rede, um será lido pelo sistema proposto nesse trabalho e extrairá as duplas e suas frequências de ocorrência para posterior armazenamento na base do classificador.

Para a construção do filtro será utilizada a solução apresentada por [15]. A ferramenta disponibilizada recebe como entrada textos na língua portuguesa, faz transformações e gera alguns tipos de arquivos diferentes. Do produto, será utilizado apenas o arquivo de frequências, onde são combinadas todas as duplas de palavras importantes e são mensuradas frequências individuais, das duplas, etc. A estratégia adotada será extrair a frequência da dupla - coluna em negrito da tabela X - e as duas palavras, colocar em uma fila ordenada em ordem decrescente, para que sejam usadas as mais frequentes como entrada para o módulo de mineração de textos. A estrutura do arquivo é demonstrada na tabela X.

Sentencas	26							
Vocabulário	47							
Voc/Sen	1							
Par	1	2	3	4	5	6	7	8
ser-roubar	3	5	1	0.03846	0	0.1428	0.02082	0
ser-carteira	3	4	1	0.03846	0	0.1667	0.0419	0
roubar-carteira	5	4	2	0.03615	0	0.125	0	0
roubar-bolsa	5	1	3	0.03845	0	0.2	0.08543	0
roubar-carro	5	2	1	0.03845	0	0.1667	0.0419	0

Tabela X

ESTRUTURA DO ARQUIVO DE FREQUÊNCIAS GERADO POR [15]

A atualização das redes para cada serviço deve ser feita a cada nova sentença adicionada ao arquivo de texto utilizado como base para a criação da rede de palavras.

3) *Implementação*: Esse módulo é constituído em sua maioria por rotinas desenvolvidas por outros autores - os blocos brancos da Figura 13. Para a definição dos termos mais importantes das frases analisadas é utilizada a rotina implementada em [15]. E para a substituição dos sinônimos deve ser desenvolvida outra rotina que consulte a base de dados do *TeP 2.0*.

G. Módulo de Mineração de Textos

Esse módulo é o elemento encarregado de interligar os termos importantes dos textos aos serviços públicos relacionados

⁶<http://thesaurus.com>



Figura 13. Funcionalidades Implementadas do Módulo de Escolha de Termos - (Teixeira, 2007) e TeP2.0

e provavelmente úteis para os autores das interações. Após passar pelo filtro de palavras e pelo dicionário de sinônimos, as palavras são utilizadas pelo classificador para realizar a busca na sua base de dados e entregar o resultado encontrado. Para essa parte do projeto é utilizado um componente da mineração de dados, chamado classificador.

Para construir a base de classificação - também chamada de modelo de classificação - e ainda atribuir uma classe às palavras inexistentes na solução deve ser usada a ferramenta gratuita de mineração WEKA. Os motivos da escolha são: amplo material didático disponível, grande variedade de filtros de pré-processamento e classificadores e, maturidade da ferramenta.

Para se criar um modelo de classificação no WEKA⁷, é necessário primeiro carregar as palavras e as classes as quais elas pertencem. Para isso, deve-se criar um arquivo texto com extensão ARFF⁸ e seguir a estrutura exibida no código 1.

```

1 @relation classes-servicos
2
3 @attribute spamclass {ligar-ANATEL,ligar-IBAMA}
4 @attribute text String
5
6 @data
7 ligar-ANATEL,'TIM ruim'
8 ligar-ANATEL,'internet lenta'
9 ligar-IBAMA,'achar carro'
```

Código 1. Arquivo usado para criar modelo de classificação no WEKA

Após a construção do modelo, o módulo estará habilitado a exercer a tarefa para o qual foi concebido, a classificação das palavras contidas nas consultas solicitadas. Semelhante ao modo de criação da base de dados, é necessário construir um arquivo no formato *ARFF* para realizar consultas no classificador. A estrutura é a mesma apresentada no código 1 exceto pelo valor das classes, que em razão de serem desconhecidas são substituídas pelo sinal de interrogação. No Código 2 é exibida a estrutura usada para atribuir classes a termos sem classificação.

⁷Weka é uma coleção de algoritmos para mineração de dados. Fonte: www.cs.waikato.ac.nz/ml/weka/

⁸Attribute-Relation File Format

```

1 @relation consulta
2
3 @attribute spamclass {COELBA, EMBASA, Policia}
4 @attribute text String
5
6 @data
7
8 ?, 'roubar celular'
9 ?, 'roubar carro'
10 ?, 'faltar energia'

```

Código 2. Arquivo .ARFF para classificar termos

Esse processo acontece após o módulo de escolha de termos enviar as duplas de palavras mais importantes da interação para o módulo de mineração de textos, gerando com o conteúdo recebido o arquivo. Em seguida, o WEKA fará uma varredura em sua base e retornará uma classificação para cada atributo sem classe definida para que o módulo de sugestões possa fazer o que é proposto.

1) *Implementação:* As funcionalidades desenvolvidas foram divididas em três partes: Classificação dos Termos, a Avaliação da Sugestão e a Configuração do Módulo aqui abordado. As explicações serão realizadas a seguir, obedecendo a ordem de citação. A primeira parte pode ser vista na Figura 13. O fluxo tem início após as principais palavras das interações dos usuários serem extraídas pelo módulo de escolha de termos. Esses termos chegam ao módulo de mineração de textos organizados em duplas, sendo usadas para pesquisas de serviços relacionados no Modelo de Classificação. O conversor de duplas não foi implementado, seguindo o padrão aqui apresentado, é exibido em um bloco de cor cinza, já o Modelo de Classificação, já construído, é apresentado em um bloco na cor preta.

Após a sugestão ser ofertada ao usuário, o mesmo pode recusá-la e em seguida, receberá como sugestão outros serviços e com base na serventia apresentada fará uma avaliação para que o Classificador possa ser atualizado por uma rotina que ainda será desenvolvida. Esse fluxo é demonstrado na Figura 14.

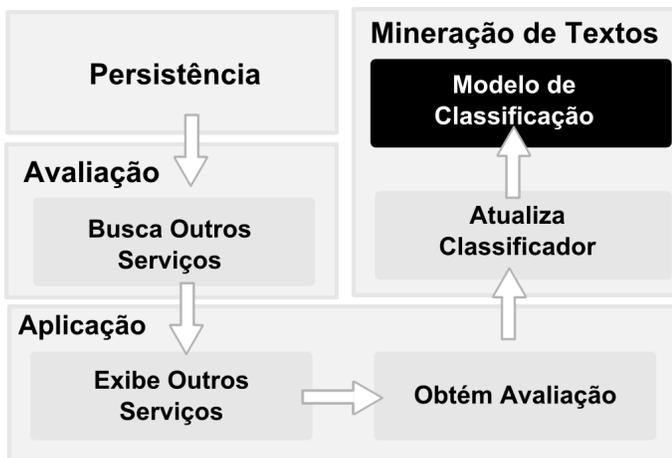


Figura 14. Funcionalidades Implementadas do Módulo de Mineração de Textos - Modelo de Classificação

Após ter sido descrito o fluxo de funcionamento da solução é apresentada a Configuração da Ferramenta, que inicia com o módulo de Escolha de Termos realizando a filtragem das mensagens coletadas das redes sociais. Em seguida, os termos são convertidos em um arquivo a ser usado para a criação do Modelo de Classificação e posteriormente, são executadas rotinas para que o WEKA crie e treine esse modelo. Na Figura 15 pode ser visto que o conversor do arquivo já foi implementado e o mesmo é representado por um bloco preto.

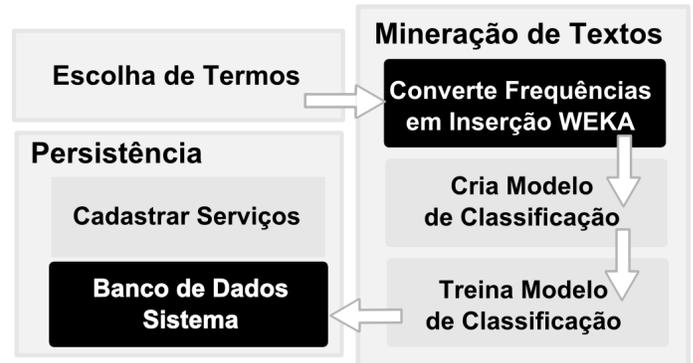


Figura 15. Funcionalidades Implementadas do Módulo de Mineração de Textos - Converter Duplas de Palavras em Arquivo WEKA(Fluxo de Configuração da Aplicação)

Os dados utilizados para construir o modelo de classificação foram extraídos de frases selecionadas empiricamente que são relacionadas a algum serviço público. O processo ocorre da seguinte maneira: após inserir as frases no módulo de escolha de termos, é executada a rotina para gerar a rede de palavras, a qual cria um arquivo de extensão *freq* contendo duplas de palavras com informações de suas respectivas frequências de ocorrência nos textos analisados. Assim, de posse da fonte das palavras mais importantes, surgiu a necessidade de transformá-la em um formato que fosse inteligível ao WEKA, então, foi criado um conversor para extrair o conteúdo dela e inserir em um arquivo texto novo com a extensão *.arff*. Para ter acesso ao código do conversor e o arquivo para construir o modelo deve-se ver os Apêndices B e D, respectivamente.

Após criação do arquivo *.arff*, foi necessário utilizar o filtro do WEKA chamado *StringToWordVector*, para extrair as palavras das duplas e passá-las para caixa baixa (todas as letras minúsculas). Na sequência, foram realizados testes com diferentes técnicas para gerar o modelo de classificação, na Tabela XI os mesmos são descritos. Devido à maior taxa de classificações corretas, foi escolhida a técnica de Naive Bayes para construir o modelo.

Na sequência, foi utilizado um arquivo de formato *.arff*, que continha duplas de palavras sem classificação, para que o modelo criado atribuísse classes às mesmas. Com o arquivo já pronto, foi realizado o teste e todas as duplas avaliadas obtiveram foram corretamente classificadas. O Código 3 exibe o conteúdo do arquivo utilizado.

Técnica	Instâncias classificadas corretamente (%)
Locally weighted learning (LWO)	92,4138 %
SMO	92,7586 %
Naive Bayes	93,4483 %
J48 pruned tree	69,3103 %

Tabela XI
TESTES DE TÉCNICAS DE CLASSIFICAÇÃO

```

1 @relation teste
2
3 @attribute spamclass {ANATEL,COELBA,EMBASA,
  Policia}
4 @attribute text String
5
6 @data
7
8 ?, 'tim sinal'
9 ?, 'sem sinal'
10 ?, 'curto energia'
11 ?, 'roubar tablet'

```

Código 3. Arquivo .ARFF usado para testar o modelo de classificação

H. Módulo de Sugestão

Em sequência à descoberta das duplas de palavras mais importantes da interação, é feita uma pesquisa na base do classificador em busca de serviços relacionados às duplas. Ao achar serviços candidatos, o módulo irá gerar uma notificação ao usuário para que o mesmo tome conhecimento e acesse a aplicação para ver os detalhes. Caso haja algum teste vinculado ao serviço, o usuário deverá responder a uma ou mais perguntas para que sejam filtrados e eleitos os serviços que serão sugeridos. Nessa etapa serão feitas perguntas como "Você foi assaltado?", "Você já prestou queixa em uma delegacia?" ou ainda solicitar ao usuário que indique a localização onde ele gostaria de obter o serviço. Em seguida, deverá escolher dentre as opções a que mais se adequa à sua necessidade. A intervenção feita pelo aplicativo será via notificação do Facebook, da mesma forma que o usuário é alertado, por exemplo, quando um novo comentário é feito em uma foto sua. No Apêndice F é exibido como a URL para requisitar a notificação é montada.

1) *Levantamento de frases:* Fazendo uso de empirismo pode-se levantar diversas frases que indiquem uma possível necessidade do usuário que fez as suas inclusões na rede social e que podem ser sanadas com a utilização de serviços públicos. Na Tabela XII está uma lista com algumas das sentenças possíveis e os órgãos que podem cuidar sanar os problemas.

2) *Levantamento de órgãos e serviços públicos prestados:* Para que a sugestão seja efetiva, é necessário fornecer informações que facilitem a utilização como, o nome e também quais órgãos públicos prestam o serviço desejado e ainda estejam mais próximos do local onde o usuário está no momento. Uma fonte importante que terá bastante serventia para a ferramenta, é o site [31], onde existe um cadastro nacional de serviços públicos que contém descrição, valor, localização e contato das unidades. Na Tabela XIII estão elencados alguns dos serviços

órgão	Sentença
Operadora/ANATEL	"Tim me fez uma cobrança ilegal"
Operadora/ANATEL	"Vivo é péssima"
Operadora/ANATEL	"A Oi não vale nada"
Operadora/ANATEL	"Meu oi está fora"
Operadora/ANATEL	"Celular está fora de área"
Operadora/ANATEL	"Tim está sem sinal a 3 dias"
Operadora/ANATEL	"Não consigo fazer ligação com a Tim"
Operadora/ANATEL	"A Tim continua sem sinal."
Operadora/ANATEL	"Meu celular está sem serviço desde o início da manhã. A Oi está fora?"
Operadora/ANATEL	"Internet da Claro, uma bosta."
Operadora/ANATEL	"O 3G da Oi não serve nem para What's App."
Operadora/ANATEL	"A ligação da Vivo cai toda hora."
Operadora/ANATEL	"Por que minha ligação nunca completa quando eu preciso?"
Operadora/ANATEL	"Tenho internet no celular para nada! :("
Operadora/ANATEL	"Minha GVT está uma carroça."
Operadora/ANATEL	"A velox não serve para nada, socorro!"
Prestadora de energia/ANEEL	"Estou sem energia."
Prestadora de energia/ANEEL	"Blackout aqui na rua."
Prestadora de energia/ANEEL	"A rua onde moro está um breu total."
Prestadora de energia/ANEEL	"Como faço para saber se tem gato em minha casa?"
Prestadora de energia/ANEEL	"É só aqui que faltou energia?"
Prestadora de energia/ANEEL	"Alguém mais na escuridão total?"
Prestadora de energia/ANEEL	"Os postes da Avenida Sete estão todos apagados."
Prestadora de energia/ANEEL	"Faltou luz aqui em casa, logo na véspera da prova de matemática."
Abastecedora de água/ANA	"Estou a 3 dias sem água."
Abastecedora de água/ANA	"Embasa cobrando valores exorbitantes na conta, se eu não reclamasse ficaria por isso mesmo."
Abastecedora de água/ANA	"Há como eu gostaria de ter água para tomar banho."
Abastecedora de água/ANA	"Alguém tem uma sugestão de como verificar se há vazamento de água no vaso sanitário?"
Abastecedora de água/ANA	"Falta de água diversos bairros."
Polícia/Ministério Público	"Acabei de ser roubado na passarela do extra. Levaram meu celular e carteira mas estou bem."
Polícia/Ministério Público	"O dono levou meu celular."
Polícia/Ministério Público	"Perdi meu celular."
Polícia/Ministério Público	"Estou sem celular por tempo indeterminado."
Polícia/Ministério Público	"Celta de placa JQY-1145 acabou de ser roubado na Lagoa do Abaeté."
Polícia/Ministério Público	"Triste, por um instante que deixei meu celular na pia do banheiro do shopping iguatemi, sumiu."
Polícia/Ministério Público	"A violência está demais, 10 minutos no engarrafamento foram suficientes para ser assaltado."

Tabela XII
SERVIÇO X SENTENÇA

e suas instituições prestadoras encontrados no site e também levantados empiricamente.

3) *Implementação:* Na Figura 16 foram representadas as funcionalidade do módulo e o fluxo envolvido, também foi destacado o bloco de cor preta que representa a rotina de notificação do usuário. Para desenvolvê-la foi utilizada a linguagem

órgão	Serviço
ANATEL - Agência Nacional de Telecomunicações	- Reclamação de prestação de serviço de operadora de telefonia
ANA - Agência Nacional de Águas	- Cadastro de uso de recursos hídricos
ANS - Agência Nacional de Saúde Suplementar	- Reclamação de prestação de serviço de operadora de plano de saúde
ANTT - Agência Nacional de Transportes Terrestres	- Consulta de Linhas que Fazem Ligação entre Duas Localidades
MDS - Ministério do Desenvolvimento Social e Combate à Fome	- Cadastro Único para Programas Sociais do Governo Federal
MTE - Ministério do Trabalho e Emprego	- Vagas de Emprego – SINE e Portal Mais Emprego - Seguro-Desemprego
RFB - Receita Federal	- Restituição e Situação da Declaração do Imposto de Renda Pessoa Física - IRPF
MEC - Ministério da Educação	- Fies — Programa de Financiamento Estudantil
MPS - Ministério da Previdência Social	- Consulta Situação do Requerimento de Benefício Previdenciário
MS - Ministério da Saúde	- Auxílio-Doença - Aqui tem Farmácia Popular
BACEN - Banco Central do Brasil	- Calculadora do Cidadão
MJ - Ministério da Justiça	- Assistência jurídica integral e gratuita - Atendimento a acidentes de trânsito/ DPRF - Autorização para comprar arma de fogo

Tabela XIII
LISTA DE ÓRGÃOS



Figura 17. Notificação de nova sugestão no Facebook

oferecidos deverão indicar sua opinião com o clique em um botão pré-definido para que sua escolha seja armazenada no banco de dados. O usuário também poderá escolher entre uma lista de serviços caso a opção sugerida não seja compatível ou satisfatória, nesse caso além do texto da postagem e da escolha feita, será regerada a rede de palavras e as associações entre pares de palavras e serviços oferecidos para o serviço.

J. Validação

Os testes para realizar a validação da aplicação quanto ao que foi proposto serão feitos com o auxílio de voluntários que irão vincular a aplicação ao seu perfil do Facebook e irão interagir normalmente como se não estivessem realizando o teste, até o momento em que serão notificados sobre uma nova sugestão disponível na aplicação. A partir daí, no momento que cada usuário achar mais conveniente, irá visualizar a sugestão e opinar quanto a sua utilidade. Visando o controle sobre o perfil dos usuários que irão testar aplicação, serão feitos os convites a amigos e conhecidos do IFBA e ainda pessoas que participam dos grupos de onde serão extraídos os textos.

Java, realizando requisições do tipo POST⁹ à Graph API do Facebook.

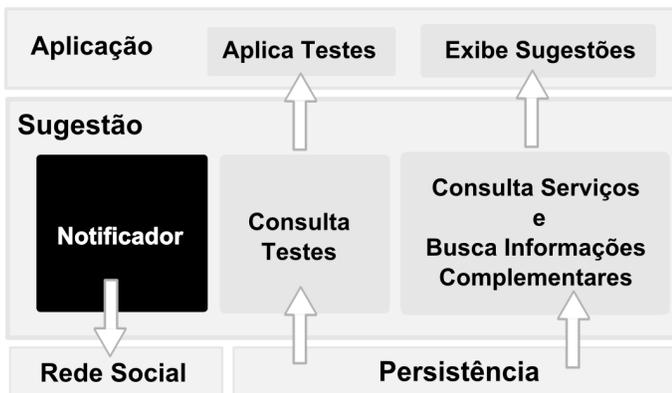


Figura 16. Funcionalidades Implementadas do Módulo de Sugestão - Notificador de Sugestões

O resultado apresentado pela rotina é exibido na Figura 17 e o código fonte da mesma está disponível no Apêndice C.

I. Módulo de Avaliação

Ao receber as sugestões, os usuários poderão aceitá-las ou não. Caso os mesmos mostrem interesse nos serviços

IX. CONCLUSÃO E TRABALHOS FUTUROS

A velocidade das grandes metrópoles exige que um maior número de tarefas sejam executadas em períodos mais curtos de tempo, sendo que, grande parte destas poderiam ser resolvidas usufruindo de informações providas de maneira automática ao cidadão. No entanto, a má divulgação de serviços de utilidade pública e a escassez de tempo das pessoas podem dificultar a utilização dos mesmos. Em decorrência dos fatos apresentados, foi desenvolvida uma proposta de arquitetura de sistema que auxilie a descobri-los e localizar órgãos prestadores de tais serviços observando as interações em redes sociais de larga utilização pela população. Foram usados para isso o potencial das redes sociais, recursos de processamento e mineração de texto. Através dessas estratégias buscou-se prever serviços que possam ser úteis aos usuários, antecipando-se às requisições ou necessidades dos mesmos .

Esta proposta de arquitetura já aparece como uma contribuição, podendo ser continuada com a devida implementação, realização de testes e, caso necessário, ajustes, estimando o nível de eficácia da ferramenta. Outra contribuição dada é na divulgação dos serviços de utilidade pública, fornecendo uma interface entre os beneficiários e os prestadores possibilitando informar a respeito da existência, utilidade e local mais próximo de prestação. Embora não fosse o objetivo,

⁹<http://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html>

também foram implementados módulos e realizados testes de alguns módulos, que mostram um alto grau de viabilidade de implementação e utilização das tecnologias indicadas de forma integrada.

Logo, através de técnicas de mineração de texto, redes de palavras e extração de informações de redes sociais, buscou-se antecipar às necessidades dos cidadãos no que tange serviços de utilidade pública.

A implementação completa dessa ferramenta é o trabalho futuro principal, embora estudos práticos e implementação de parte da mesma tenham sido realizados e também apareçam como uma contribuição. Junto com a sua materialização, novas dificuldades e ideias podem ser identificadas, requerendo adaptações para aperfeiçoar a solução.

AGRADECIMENTOS

Esse curso para mim foi como uma corrida de muitos desafios e obstáculos que foram vencidos durante esses longos dois anos. A essa minha conquista gostaria de agradecer imensamente a equipe de profissionais competentes e comprometidos, na qual, cada um contribuiu com sua competência e conhecimentos, para o meu crescimento, desenvolvimento e finalização desse processo acadêmico. Ao meu orientador Pablo Florentino gostaria de expressar o meu profundo e sincero obrigado, ele que esteve sempre do meu lado, me orientando, incentivando e ajudando durante todos os difíceis momentos, e ainda, foi o autor da ideia desse trabalho.

REFERÊNCIAS

- [1] D. M. Boyd and N. B. Ellison, "Social Network Sites: Definition, History, and Scholarship," *Journal of Computer-Mediated Communication*, vol. 13, no. 1, pp. 210–230, 2008. [Online]. Available: <http://onlinelibrary.wiley.com/doi/10.1111/j.1083-6101.2007.00393.x/pdf>
- [2] A. Mendes, "As redes sociais e sua influência na sociedade," p. 4, 2011. [Online]. Available: <http://imasters.com.br/artigo/19889/redes-sociais/as-redes-sociais-e-sua-influencia-na-sociedade/>
- [3] C. M. D. S. Freitas, R. G. Luciana P. Nedel, L. C. Lamb, A. S. Spritzer, J. P. M. d. O. Sérgio Fujii, R. M. Araújo, and M. M. Moro, "Extração de Conhecimento e Análise Visual de Redes Sociais," *Seminário Integrado de Software e Hardware*, pp. 106–120, 2008.
- [4] B. K. O. Miyata, V. Y. Kano, and L. A. Digiampietri, "Combinando Mineração de Textos e Análise de Redes Sociais para Identificação das Áreas de Atuação de Pesquisadores," in *Brasnam 2013*, 2013.
- [5] R. Gadelha, R. Prudêncio, R. Lima, and C. Souza, "Predizendo Influenciadores no Twitter por meio de Traços de Personalidade," *II Brazilian Workshop on Social Network Analysis and Mining*, p. 12, 2013.
- [6] D. W. Albuquerque, A. V. Brito, J. B. Pereira, M. L. de Mendonça, and A. N. Duarte, "Estudo do uso do Twitter como Ferramenta de Análise de Opinião durante as Eleições Municipais de João Pessoa," in *Brasnam 2013*. Maceió, Alagoas: Universidade Federal da Paraíba, 2013, p. 12.
- [7] A. O. Rodrigues, R. C. D. Melo-minardi, and W. M. Jr, "Como nos Sentimos : Uma Ferramenta de Mineração Visual de Sentimentos no Twitter," in *Brasnam 2013*, 2013.
- [8] U. Fayyad, G. Piatetsky-shapiro, and P. Smyth, "From Data Mining to Knowledge Discovery in databases," *American Association for Artificial Intelligence*, pp. 37–51, 1996.
- [9] R. Feldman and I. Dagan, "Knowledge Discovery in Textual Databases (KDT)," in *KDD-95*, Bar-Ilan University, Math and Computer Science Dept. Ramat-Gan, ISRAEL: Bar-Ilan University, 1995, p. 6.
- [10] I. H. Witten, "Text mining," in *Practical handbook of internet computing*, M. Singh, Ed. Boca Raton, Florida: Chapman & Hall/CRC Press, 2002, ch. 14, pp. 1–22.
- [11] A.-h. Tan, "Text Mining : The state of the art and the challenges," in *PAKDD 1999 Workshop on Knowledge Discovery from Advanced Databases*. Singapore: Kent Ridge Digital Labs, 1999, p. 5.
- [12] F. Sebastiani, "Machine learning in automated text categorization," *ACM Computing Surveys*, vol. 34, no. 1, pp. 1–47, Mar. 2002. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=505282.505283>
- [13] M.-S. Chen, J. Han, and P. S. Yu, "Data mining: An overview from a database perspective," *IEEE Transactions on knowledge and data engineering*, vol. 8, no. 6, p. 16, 1996.
- [14] P. Ozer, "Data Mining Algorithms for Classification," Ph.D. dissertation, Radboud University Nijmegen, 2008.
- [15] G. M. Teixeira, "Redes Semânticas em Discurso Oral: Uma Proposta Metodológica Baseada na Psicologia Cognitiva Utilizando Redes Complexas," Ph.D. dissertation, FACULDADE VISCONDE DE CAIRU, 2007.
- [16] S. M. G. Caldeira, "Caracterização da Rede de Signos Lingüísticos: Um Modelo Baseado no Aparelho Psíquico de Freud," Ph.D. dissertation, FACULDADE VISCONDE DE CAIRU, 2005.
- [17] Y. Y. Kalle Lyytinen, "Issues and Challenges in Ubiquitous Computing," *Communications of the ACM*, vol. 45, no. 12, pp. 63–65, 2002.
- [18] R. B. D. Araujo, "Computação Ubíqua : Princípios , Tecnologias e Desafios," in *XXI Simpósio Brasileiro de Redes de Computadores*, 2003, pp. 45–115.
- [19] M. Weiser, "The Computer for the 21st Century," *Scientific American*, vol. 265, no. 3, pp. 94–104, 1991. [Online]. Available: <http://www.ubiq.com/hypertext/weiser/SciAmDraft3.html>
- [20] M. Weiser and X. Corporation, "Ubiquitous computing," *IEEE Computer Society*, vol. 26, no. 10, pp. 71–72, 1993.
- [21] Google, "Sobre a segmentação contextual," p. 4, 2014. [Online]. Available: <https://support.google.com/adwords/answer/2404186>
- [22] —, "Anúncios no Gmail e seus dados pessoais," p. 2, 2014. [Online]. Available: <https://support.google.com/mail/answer/6603?hl=pt-BR>
- [23] —, "Classificação e marcadores de importância," p. 3, 2014. [Online]. Available: <https://support.google.com/mail/answer/186543?hl=pt-BR>
- [24] —, "Sobre a Rede de Display do Google," p. 8, 2014. [Online]. Available: https://support.google.com/adwords/answer/2404190?hl=pt-BR&ref_topic=3121944
- [25] TechTudo, "Google Now," p. 1, 2013. [Online]. Available: <http://www.techtudo.com.br/tudo-sobre/google-now.html>
- [26] S. Experian, "Facebook é líder nas visitas a redes sociais em dezembro no Brasil, de acordo com Hitwise," p. 3, 2014. [Online]. Available: <http://noticias.serasaexperian.com.br/facebook-e-lider-nas-visitas-a-redes-sociais-em-dezembro-no-brasil-de-acordo-com-hitwise/>
- [27] X. Huang, L. Luo, N. Morisseau-leroy, S.-c. Chen, S. Gulati, M. D. Powell, C. Zhan, and C. Zhang, "A Three-Tier System Architecture Design and Development for Hurricane Occurrence Simulation," in *International Conference on Information Technology: Research and Education*. Neward, NJ, USA: IEEE, 2003, pp. 113–117.
- [28] Facebook, "Product Docs: Canvas," p. 7, 2014. [Online]. Available: <https://developers.facebook.com/docs/games/canvas>
- [29] —, "Login Review," p. 6, 2014. [Online]. Available: <https://developers.facebook.com/docs/apps/review/login>
- [30] E. G. Maziero, T. A. S. Pardo, A. D. Felippo, and B. C. Dias-da silva, "A Base de Dados Lexical e a Interface Web do TeP 2 . 0 – Thesaurus Eletrônico para o Português do Brasil," *VI Workshop em Tecnologia da Informação e da Linguagem Humana(TIL)*, pp. 390–392, 2008.
- [31] G. Federal, "Guia de Serviços Públicos do Governo Federal," 2014. [Online]. Available: <http://www.servicos.gov.br/>
- [32] H. R. D. Moraes and B. C. Dias-da silva, "A Construção de um Thesaurus Eletrônico para o Português do Brasil," *ALFA*, vol. 47, no. 2, pp. 101–115, 2003.
- [33] C. D. Barros, "Antonímia nos Adjetivos Descritivos do Português do Brasil : Uma Proposta de Análise e Representação," Ph.D. dissertation, Universidade Federal de São Carlos, 2010.
- [34] M. Célia, F. Rocha, P. V. Florentino, and G. C. Pereira, "City as Commons : Study of Shared Visions by Communities on Facebook," *Springer International Publishing Switzerland*, pp. 486–501, 2014.
- [35] B. Dias-da silva, H. Moraes, M. Oliveira, R. Hasegawa, D. Amorim, C. Paschoalino, and A. Nascimento, "Construção de um Thesaurus Eletrônico para o Português do Brasil," *Processamento Computacional do Português Escrito e Falado (PROPOR)*, vol. 4, pp. 1–10, 2000.
- [36] H. G. Oliveira, D. Santos, and P. Gomes, "Extração de Relações Semânticas entre Palavras a Partir de um Dicionário : Primeira Avaliação," 2008.

- [37] Facebook, "Facebook Platform Changelog," p. 6, 2014. [Online]. Available: <https://developers.facebook.com/docs/apps/changelog>
- [38] —, "Facebook SDK for JavaScript," p. 1, 2014. [Online]. Available: <https://developers.facebook.com/docs/javascript>
- [39] —, "Getting started with the Facebook SDK for PHP," p. 3, 2014. [Online]. Available: <https://developers.facebook.com/docs/php/gettingstarted/4.0.0>
- [40] —, "Graph API: Quickstart," p. 5, 2014. [Online]. Available: <https://developers.facebook.com/docs/graph-api/quickstart/v2.0>
- [41] S. Paumier and English, "Manual Unix 3.1 BETA," p. 361, 2003. [Online]. Available: <http://www-igm.univ-mlv.fr/unitex>
- [42] F. java api, "OldHomePage facebook-java-api," 2014. [Online]. Available: <https://code.google.com/p/facebook-java-api/wiki/OldHomePage>
- [43] Facebook, "Product Docs: access-tokens," p. 16, 2014. [Online]. Available: <https://developers.facebook.com/docs/facebook-login/access-tokens>

APÊNDICE A COLETOR DE INTERAÇÕES

```

1 package com.streamReader;
2
3
4 import java.io.FileInputStream;
5 import java.sql.Connection;
6 import java.sql.DriverManager;
7 import java.sql.PreparedStatement;
8 import java.sql.ResultSet;
9 import java.sql.SQLException;
10 import java.sql.Statement;
11 import java.util.Calendar;
12 import java.util.Date;
13 import java.util.List;
14 import java.util.Properties;
15 import com.restfb.DefaultFacebookClient;
16 import com.restfb.FacebookClient;
17 import com.restfb.types.Message;
18 import javax.naming.Context;
19 import javax.naming.InitialContext;
20 import javax.naming.NamingException;
21 import javax.sql.DataSource;
22
23
24
25 class ConnectionFactory {
26
27     private static Connection conn;
28
29     public static java.sql.Connection
30         getConnection(){
31
32         //MYSQL
33         Connection conn = null;
34         try {
35
36             String driverName = "com.mysql.jdbc.
37                 Driver";
38             String url = "jdbc:mysql
39                 ://127.9.61.30:3306/
40                 publicservices";
41             String username = "admin665K1JVYV";
42             String password = "BDwtVS3432Whl--H";
43
44             // Load the JDBC driver
45             Class.forName(driverName);
46             // Create a connection to the
47             database
48             conn = DriverManager.getConnection(
49                 url, username, password);
50
51         } catch (SQLException f) {
52             // Could not connect to the database
53         } catch (ClassNotFoundException e) {
54             // TODO Auto-generated catch block
55             e.printStackTrace();
56         }
57         return conn;
58     }
59 }
60
61
62
63
64
65 class StreamReader2 {
66     static String MY_APP_SECRET = "
67         fe5662ed34cb8062f46c29e48b5bbb88",
68         MY_APP_ID = "291802777521539", MY_ID = "
69         100001341139185";
70
71     // AccessToken accessToken = new
72     // DefaultFacebookClient().obtainAppAccessToken
73     // (MY_APP_ID,MY_APP_SECRET);
74     // token=accessToken.getAccessToken();
75     // out.print(token);
76
77     public static void main(String[] args) {
78
79         if (args.length < 2) {
80
81             System.out
82                 .println("Sintaxe correta:
83                 StreamReader2 [UID] [SERVICO]");
84             return;
85         }
86
87         FileInputStream stream;
88         try {
89             Connection con = ConnectionFactory.
90                 getConnection();
91             Statement comando = con.createStatement();
92
93             String uid = args[0];
94             if (uid == null) {
95                 System.out
96                     .println("Sintaxe correta:
97                     StreamReader2 [UID] [SERVICO]");
98                 return;
99             }
100
101             System.out
102                 .println("\n" + Calendar.getInstance().
103                 toString() + "\n\n");
104
105             String idServico = args[1];
106             if (idServico == null)
107                 idServico = "1";
108             System.out
109                 .println("Query:");
110
111             String selectInfo = "select idCredencial,
112                 token, dtUltimaColeta from
113                 publicservices.credencial where
114                 idUsuario = '"
115                 + uid + "' AND idServico = '" +
116                 idServico + "'";
117
118             comando.executeQuery(selectInfo);
119             ResultSet rs = comando.getResultSet();

```

```

111         System.out.println("
112             Buscando info
113             do Usuario");
114     };
115     if (rs.next()) {
116         System.out.println("Usuario encontrado!");
117     };
118     String token = rs.getString("token");
119     String queryInsert = "";
120     FacebookClient facebookClient = new
121         DefaultFacebookClient(
122             token, MY_APP_SECRET);
123     System.out
124     .println("Criado objeto de consulta do
125         Face");
126
127     Date dt = rs.getDate("dtUltimaColeta");
128     Calendar d = Calendar.getInstance();
129     d.set(1971, 12, 12, 0, 0, 0);
130     if (dt != null && dt.after(d.getTime()))
131         d.setTime(dt);
132
133     long lDtUltimaColeta = d.getTimeInMillis
134         () / 1000L;
135
136     // Grupo A Salvador Que o Nao Viu
137     String queryStatus = "SELECT created_time
138         , message FROM stream where source_id
139         = "
140         + uid
141         + " and message <> ' ' and
142         created_time > ' '
143         + lDtUltimaColeta + " LIMIT 50";
144
145     List<Message> status = facebookClient
146     .executeFqlQuery(
147         queryStatus, Message.class);
148
149     System.out
150     .println("FQL Query: ");
151     System.out
152     .println(queryStatus);
153
154     int cont = 0;
155
156     queryInsert = "insert into publicsServices
157         .interacao(idCredencial, tipo,
158         conteudo, data) values (?, ?, ?, ?)";
159     PreparedStatement ps = con.
160     prepareStatement(queryInsert);
161
162     for (Message msg : status) {
163         ps.setString(1, rs.getString("
164             idCredencial"));
165         ps.setString(2, "mensagem");
166         ps.setString(3, msg.getMessage());
167         ps.setString(4, new java.text.
168             SimpleDateFormat (
169                 "yyyy-MM-dd HH:mm:ss").format(msg.
170                 getCreatedTime()));
171
172         cont++;
173         ps.addBatch();
174     }
175
176     if (cont > 0) {
177         ps.executeBatch();
178     }
179
180     System.out.println(
181         cont + "
182         interacoes
183         coletadas");
184
185     ps.close();
186 }
187
188 Calendar dtAtual = Calendar.getInstance()
189 ;
190 String queryUpdate = "update
191     publicsServices.credencial set
192     dtUltimaColeta=' "
193     + new java.text.SimpleDateFormat ("
194         yyyy-MM-dd HH:mm:ss")
195     .format(dtAtual.getTime())
196     + "' where idCredencial = ' "
197     + rs.getString("idCredencial") + "'";
198 ;
199
200 System.out
201 .println("Query: ");
202 System.out.print(queryUpdate);
203 comando.executeUpdate(queryUpdate);
204 System.out
205 .println("Atualizada data de coleta");
206
207 }
208
209 } catch (SQLException e) {
210     // TODO Auto-generated catch block
211     System.out
212     .println(e.toString());
213     System.out
214     .println(e.getMessage());
215     e.printStackTrace();
216 }
217
218 }
219
220 }

```

APÊNDICE B CONVERSOR FREQTOARFF

```

1
2 <%@page import="util.Arquivo"%>
3 <%@page import="java.io.BufferedReader"%>
4 <%@page import="java.io.InputStreamReader"%>
5 <%@page import="java.io.FileInputStream"%>
6 <%@ page language="java" contentType="text/html;
7     charset=ISO-8859-1"
8     pageEncoding="ISO-8859-1"%>
9 <%
10 String[] vetorLinha;
11 String arquivo = "../app-root/repo/src/main/
12     webapp/redesPalavras/ORappa.txt_FF.freq";
13 FileInputStream stream = new FileInputStream(
14     arquivo);
15 InputStreamReader reader = new
16     InputStreamReader(stream);
17 BufferedReader br = new BufferedReader(reader);
18 String linha = br.readLine();
19 int cont_linha = 0;
20 String campos = request.getParameter("campos");
21 String tipoArquivo = request.getParameter("tipo
22     ");
23 String classes = request.getParameter("classes
24     ");
25 String relation = request.getParameter("nome");
26 String replace = request.getParameter("replace
27     ");
28 String vClasses[] = null;

```

```

23 String saida = "";
24 int iClasses = 0;
25 Arquivo arq = new Arquivo();
26 if (tipoArquivo == null) {
27     tipoArquivo = "";
28 } else if (tipoArquivo.equals("wk")) {
29
30     if (classes == null) {
31         classes = "spam,ham";
32     }
33     if (relation == null) {
34         relation = "spam_text";
35     }
36     vClasses = classes.split(",");
37
38     out.println("@relation " + relation + "\n\n<br><br>" +
39
40     "@attribute spamclass {" + classes + "}" + "\n<br>"
41     + "@attribute text String" + "\n\n<br><br>" +
42
43     "@data" + "\n<br>");
44
45     saida = "@relation " + relation + "\n\n" +
46
47     "@attribute spamclass {" + classes + "}" + "\n"
48     + "@attribute text String" + "\n\n" +
49
50     "@data" + "\n";
51
52 }
53 if (replace == null) {
54     replace = "";
55 }
56
57 while (linha != null) {
58     cont_linha++;
59     if (cont_linha > 4) {
60
61         vetorLinha = linha.split(" ");
62         if (replace.equals("s")) {
63             vetorLinha[0] = vetorLinha[0].replace("-",
64             , " ");
65         } else if (!replace.equals("")) {
66             vetorLinha[0] = vetorLinha[0].replace("-",
67             , replace);
68         }
69         if (!tipoArquivo.equals("wk")) {
70
71             if (campos == null || campos.equals(""))
72             {
73                 out.println(vetorLinha[0] + " " +
74                 vetorLinha[3]
75                 + "<br>");
76                 saida += vetorLinha[0] + " " +
77                 vetorLinha[3] + "\n";
78             } else {
79                 out.println(vetorLinha[0]);
80
81                 if (campos.indexOf("2") > -1) {
82                     out.println(" " + vetorLinha[1]);
83                 }
84                 if (campos.indexOf("3") > -1) {
85                     out.println(" " + vetorLinha[2]);
86                 }
87                 if (campos.indexOf("4") > -1) {
88                     out.println(" " + vetorLinha[3]);
89                 }
90                 if (campos.indexOf("5") > -1) {
91                     out.println(" " + vetorLinha[4]);
92                 }
93             }
94         }
95     }
96 }

```

```

88         if (campos.indexOf("6") > -1) {
89             out.println(" " + vetorLinha[5]);
90         }
91     }
92 } else {
93     iClasses = iClasses % vClasses.length;
94     out.println(vClasses[iClasses] + "," + ""
95     +
96     + vetorLinha[0] + """);
97     saida += vClasses[iClasses] + "," + "" +
98     + vetorLinha[0]
99     + ""\n";
100     iClasses++;
101 }
102 out.println("<br>");
103
104 }
105
106 linha = br.readLine();
107
108 }
109
110 arq.content = saida;
111 arq.tipo = tipoArquivo;
112 arq.nome = relation;
113 if (tipoArquivo.equals("wk"))
114     if (arq.salvar() == 0)
115         out.println("Foi salvo com sucesso o
116         arquivo " + relation + ".arff');
117 %>

```

Código 4. Conversor de Arquivo .freq para .arff

APÊNDICE C NOTIFICADOR

```

1 <%@ page import="java.net.*, java.io.*" %>
2 <%
3 String MY_ID="100001341139185",
4 TOKEN_APP="291802777521539|33
5 JcX5KnVUk9cMVeImujIPd1XaA", //App_Id +
6 App_Secret
7 targetURL="https://graph.facebook.com/" + MY_ID + "
8 /notifications",
9 mensagem="Uma%20nova%20sugest%C3%A3o%20est%C3%A1
10 %20esperando%20por%20v%C3%AA!",
11 urlDestino="index.html",
12 urlParameters="access_token="+TOKEN_APP+"&href="+
13 urlDestino+"&template="+mensagem;
14 URL url;
15 HttpURLConnection connection = null;
16 try {
17     url = new URL(targetURL); //Criar Conexao
18     connection = (HttpURLConnection)url.
19     openConnection();
20     connection.setRequestMethod("POST");
21     connection.setRequestProperty("Content-Type", "
22     application/x-www-form-urlencoded");
23     connection.setRequestProperty("Content-Length",
24     "" + Integer.toString(urlParameters.
25     getBytes().length));
26     connection.setRequestProperty("Content-Language
27     ", "en-US");
28     connection.setUseCaches (false);
29     connection.setDoInput (true);
30     connection.setDoOutput (true);
31     //Enviar requisicao
32     DataOutputStream wr = new DataOutputStream (
33     connection.getOutputStream ());
34     wr.writeBytes (urlParameters);
35     wr.flush ();

```

```

25  wr.close ();
26  //Obter resposta
27  InputStream is = connection.getInputStream();
28  BufferedReader rd = new BufferedReader(new
    InputStreamReader(is));
29  String line;
30  StringBuffer responseBf = new StringBuffer();
31  while((line = rd.readLine()) != null) {
32      responseBf.append(line);
33      responseBf.append('\r');
34  }
35  rd.close();
36  out.print( responseBf.toString());
37
38  } catch (Exception e)
39  {
40      e.printStackTrace();
41
42  } finally {
43      if(connection != null) {
44          connection.disconnect();
45      }
46  }
47  %>

```

Código 5. Código para gerar notificação utilizando Graph API

APÊNDICE D CLASSES.ARFF

```

@relation teste
@attribute          spamclass          ANA-
TEL,COELBA,EMBASA,Policia @attribute text String
@data
ANATEL, 'tim continuum'
ANATEL, 'tim sinal'
ANATEL, 'continuum sinal'
ANATEL, 'celular estar'
ANATEL, 'celular serviço'
ANATEL, 'estar serviço'
ANATEL, 'celular início'
ANATEL, 'estar início'
ANATEL, 'serviço início'
ANATEL, 'celular manhã'
ANATEL, 'estar manhã'
ANATEL, 'serviço manhã'
ANATEL, 'início manhã'
ANATEL, 'celular internet'
ANATEL, 'internet claro'
ANATEL, 'internet bosta'
ANATEL, 'claro bosta'
ANATEL, 'não servir'
ANATEL, 'não what'
ANATEL, 'servir what'
ANATEL, 'não app'
ANATEL, 'servir app'
ANATEL, 'what app'
ANATEL, 'ligação vivo'
ANATEL, 'ligação cair'
ANATEL, 'vivo cair'
ANATEL, 'ligação hora'
ANATEL, 'vivo hora'
ANATEL, 'cair hora'

```

```

ANATEL, 'ligação nunca'
ANATEL, 'ligação completo'
ANATEL, 'nunca completo'
ANATEL, 'ligação precisar'
ANATEL, 'nunca precisar'
ANATEL, 'completo precisar'
ANATEL, 'celular ter'
ANATEL, 'internet ter'
ANATEL, 'estar gvt'
ANATEL, 'estar carroça'
ANATEL, 'gvt carroça'
ANATEL, 'não velox'
ANATEL, 'servir velox'
ANATEL, 'não socorro'
ANATEL, 'servir socorro'
ANATEL, 'velox socorro'
COELBA, 'estar energia'
COELBA, 'energia aqui'
COELBA, 'blackout aqui'
COELBA, 'estar rua'
COELBA, 'blackout rua'
COELBA, 'aqui rua'
COELBA, 'estar morar'
COELBA, 'rua morar'
COELBA, 'estar breu'
COELBA, 'rua breu'
COELBA, 'morar breu'
COELBA, 'estar total'
COELBA, 'rua total'
COELBA, 'morar total'
COELBA, 'breu total'
COELBA, 'fazer saber'
COELBA, 'fazer ter'
COELBA, 'saber ter'
COELBA, 'fazer gato'
COELBA, 'saber gato'
COELBA, 'ter gato'
COELBA, 'aqui casa'
COELBA, 'fazer casa'
COELBA, 'saber casa'
COELBA, 'ter casa'
COELBA, 'gato casa'
COELBA, 'energia ser'
COELBA, 'aqui ser'
COELBA, 'energia só'
COELBA, 'aqui só'
COELBA, 'ser só'
COELBA, 'energia faltar'
COELBA, 'aqui faltar'
COELBA, 'casa faltar'
COELBA, 'ser faltar'
COELBA, 'só faltar'
COELBA, 'total escuridão'
COELBA, 'estar poste'
COELBA, 'estar avenida'
COELBA, 'poste avenida'

```

COELBA, 'estar apagar'
COELBA, 'poste apagar'
COELBA, 'avenida apagar'
COELBA, 'aqui luz'
COELBA, 'casa luz'
COELBA, 'faltar luz'
COELBA, 'aqui véspera'
COELBA, 'casa véspera'
COELBA, 'faltar véspera'
COELBA, 'luz véspera'
COELBA, 'aqui prova'
COELBA, 'casa prova'
COELBA, 'faltar prova'
COELBA, 'luz prova'
COELBA, 'véspera prova'
COELBA, 'aqui matemático'
COELBA, 'casa matemático'
COELBA, 'faltar matemático'
COELBA, 'luz matemático'
COELBA, 'véspera matemático'
COELBA, 'prova matemático'
EMBASA, 'estar dia'
EMBASA, 'estar água'
EMBASA, 'dia água'
EMBASA, 'embasar cobrar'
EMBASA, 'embasar valor'
EMBASA, 'cobrar valor'
EMBASA, 'embasar exorbitante'
EMBASA, 'cobrar exorbitante'
EMBASA, 'valor exorbitante'
EMBASA, 'embasar conta'
EMBASA, 'cobrar conta'
EMBASA, 'valor conta'
EMBASA, 'exorbitante conta'
EMBASA, 'embasar não'
EMBASA, 'cobrar não'
EMBASA, 'valor não'
EMBASA, 'exorbitante não'
EMBASA, 'conta não'
EMBASA, 'embasar reclamar'
EMBASA, 'cobrar reclamar'
EMBASA, 'valor reclamar'
EMBASA, 'exorbitante reclamar'
EMBASA, 'conta reclamar'
EMBASA, 'não reclamar'
EMBASA, 'embasar ficar'
EMBASA, 'cobrar ficar'
EMBASA, 'valor ficar'
EMBASA, 'exorbitante ficar'
EMBASA, 'conta ficar'
EMBASA, 'não ficar'
EMBASA, 'reclamar ficar'
EMBASA, 'água haver'
EMBASA, 'água gostar'
EMBASA, 'haver gostar'
EMBASA, 'água ter'

EMBASA, 'haver ter'
EMBASA, 'gostar ter'
EMBASA, 'água tomar'
EMBASA, 'haver tomar'
EMBASA, 'gostar tomar'
EMBASA, 'ter tomar'
EMBASA, 'água banho'
EMBASA, 'haver banho'
EMBASA, 'gostar banho'
EMBASA, 'ter banho'
EMBASA, 'tomar banho'
EMBASA, 'água sugestão'
EMBASA, 'haver sugestão'
EMBASA, 'ter sugestão'
EMBASA, 'água verificar'
EMBASA, 'haver verificar'
EMBASA, 'ter verificar'
EMBASA, 'sugestão verificar'
EMBASA, 'água vazamento'
EMBASA, 'haver vazamento'
EMBASA, 'ter vazamento'
EMBASA, 'sugestão vazamento'
EMBASA, 'verificar vazamento'
EMBASA, 'água vaso'
EMBASA, 'haver vaso'
EMBASA, 'ter vaso'
EMBASA, 'sugestão vaso'
EMBASA, 'verificar vaso'
EMBASA, 'vazamento vaso'
EMBASA, 'água sanitário'
EMBASA, 'haver sanitário'
EMBASA, 'ter sanitário'
EMBASA, 'sugestão sanitário'
EMBASA, 'verificar sanitário'
EMBASA, 'vazamento sanitário'
EMBASA, 'vaso sanitário'
EMBASA, 'água falto'
EMBASA, 'água bairro'
EMBASA, 'falto bairro'
Polícia, 'acabar ser'
Polícia, 'acabar roubar'
Polícia, 'ser roubar'
Polícia, 'acabar passarela'
Polícia, 'ser passarela'
Polícia, 'roubar passarela'
Polícia, 'acabar extra'
Polícia, 'ser extra'
Polícia, 'roubar extra'
Polícia, 'passarela extra'
Polícia, 'levar celular'
Polícia, 'levar carteira'
Polícia, 'celular carteira'
Polícia, 'ser estar'
Polícia, 'levar estar'
Polícia, 'celular estar'
Polícia, 'carteira estar'

Polícia, 'levar bem'
Polícia, 'celular bem'
Polícia, 'carteira bem'
Polícia, 'estar bem'
Polícia, 'levar dono'
Polícia, 'celular dono'
Polícia, 'celular perder'
Polícia, 'celular tempo'
Polícia, 'estar tempo'
Polícia, 'celular indeterminar'
Polícia, 'estar indeterminar'
Polícia, 'tempo indeterminar'
Polícia, 'acabar celta'
Polícia, 'ser celta'
Polícia, 'roubar celta'
Polícia, 'acabar placa'
Polícia, 'ser placa'
Polícia, 'roubar placa'
Polícia, 'celta placa'
Polícia, 'acabar jqy'
Polícia, 'ser jqy'
Polícia, 'roubar jqy'
Polícia, 'celta jqy'
Polícia, 'placa jqy'
Polícia, 'acabar lagoa'
Polícia, 'ser lagoa'
Polícia, 'roubar lagoa'
Polícia, 'celta lagoa'
Polícia, 'placa lagoa'
Polícia, 'jqy lagoa'
Polícia, 'acabar abaeté'
Polícia, 'ser abaeté'
Polícia, 'roubar abaeté'
Polícia, 'celta abaeté'
Polícia, 'placa abaeté'
Polícia, 'jqy abaeté'
Polícia, 'lagoa abaeté'
Polícia, 'celular triste'
Polícia, 'celular instante'
Polícia, 'triste instante'
Polícia, 'celular deixar'
Polícia, 'triste deixar'
Polícia, 'instante deixar'
Polícia, 'celular pio'
Polícia, 'triste pio'
Polícia, 'instante pio'
Polícia, 'deixar pio'
Polícia, 'celular banheiro'
Polícia, 'triste banheiro'
Polícia, 'instante banheiro'
Polícia, 'deixar banheiro'
Polícia, 'pio banheiro'
Polícia, 'celular shopping'
Polícia, 'triste shopping'
Polícia, 'instante shopping'
Polícia, 'deixar shopping'

Polícia, 'pio shopping'
Polícia, 'banheiro shopping'
Polícia, 'celular iguatemi'
Polícia, 'triste iguatemi'
Polícia, 'instante iguatemi'
Polícia, 'deixar iguatemi'
Polícia, 'pio iguatemi'
Polícia, 'banheiro iguatemi'
Polícia, 'shopping iguatemi'
Polícia, 'celular sumir'
Polícia, 'triste sumir'
Polícia, 'instante sumir'
Polícia, 'deixar sumir'
Polícia, 'pio sumir'
Polícia, 'banheiro sumir'
Polícia, 'shopping sumir'
Polícia, 'iguatemi sumir'
Polícia, 'ser violência'
Polícia, 'estar violência'
Polícia, 'ser minuto'
Polícia, 'estar minuto'
Polícia, 'violência minuto'
Polícia, 'ser engarrafamento'
Polícia, 'estar engarrafamento'
Polícia, 'violência engarrafamento'
Polícia, 'minuto engarrafamento'
Polícia, 'ser suficiente'
Polícia, 'estar suficiente'
Polícia, 'violência suficiente'
Polícia, 'minuto suficiente'
Polícia, 'engarrafamento suficiente'
Polícia, 'ser assaltar'
Polícia, 'estar assaltar'
Polícia, 'violência assaltar'
Polícia, 'minuto assaltar'
Polícia, 'engarrafamento assaltar'
Polícia, 'suficiente assaltar'

APÊNDICE E APIS DO FACEBOOK

Em 30 de Abril de 2014, foi lançada a versão 2.0 da plataforma de desenvolvimento do Facebook. Apesar disso, ainda será dado suporte à versão anterior, referida como 1.0, a mesma utilizada durante os estudos realizados nesse projeto. No entanto, só será permitida a utilização da antiga versão pelas aplicações criadas antes da data de lançamento da nova e ainda assim até o dia 30/04/2015, ficando a cargo do desenvolvedor decidir qual usar. Portanto, deve-se atentar a isso caso os exemplos citados ou códigos desenvolvidos sejam utilizados para continuar ou utilizar como base o projeto. Para as aplicações antigas migrarem para a plataforma 2.0, devem modificar o caminho o qual utilizavam para fazer as consultas, seguindo a estrutura <https://graph.facebook.com/v2.0/objeto> [37].

Graph API, [40] É a forma primária de obter e enviar dados da rede social do Facebook, uma API baixo nível que utiliza

HTTP¹⁰ e pode ser usada para fazer consultas, postar novas histórias, enviar fotos e uma variedade de outras tarefas que uma aplicação pode necessitar realizar. Caracterizada por ser flexível, já que qualquer linguagem que possua uma biblioteca HTTP, como *cURL*, *urllib* ou um navegador WEB podem fazer uso dela. Apesar da facilidade de utilização, possui um requisito obrigatório na maioria de suas requisições, é necessário a utilização de *tokens* de acesso - chaves criptografadas - que podem ser gerados através da implementação de uma autenticação (*login*) na rede. Um dos recursos da chamada da API seria realizar uma requisição do tipo GET¹¹ através de uma URL com a seguinte estrutura *graph.facebook.com/objeto*, onde objeto poderia ser, por exemplo, o identificador da página da Coca-Cola, apresentando como resultado uma *string*¹² em JSON¹³, seguindo a estrutura do Código 6.

```

1 {
2   "id": "40796308305",
3   "about": "The Coke Page.",
4   "can_post": false,
5   "category": "Food/beverages",
6   "checkins": 21688,
7   "description": "The Coke Company",
8   "founded": "1886",
9   "has_added_app": false,
10  "is_community_page": false,
11  "is_published": true,
12  "likes": 83030953,
13   "name": "Coca-Cola",
14  "talking_about_count": 755437,
15  "username": "Coca-Cola",
16  "website": "http://www.coke.com",
17  "were_here_count": 0
18 }

```

Código 6. Exemplo de código JSON recebido como resposta a uma requisição da Graph API.

1) *Time-based Pagination* ou *Paginação Temporal*: A Graph API possui um artifício que permite filtrar o conteúdo trazidos em suas consultas, é o *Time-based Pagination* ou *Paginação Temporal*. Para realizar a paginação dos resultados, são utilizados dois parâmetros: *since* - determina o início do intervalo de tempo no formato Unix timestamp - ou *until* - delimita o fim de um intervalo de tempo no formato Unix timestamp. No Código 7 é exibida uma resposta em JSON a uma requisição utilizando paginação e os dois recursos.

```

1 {
2   "data": [ ... Endpoint data is here ],
3   "paging": {
4     "previous": "https://graph.facebook.com/me/feed?limit=25&since=1364849754",
5     "next": "https://graph.facebook.com/me/feed?limit=25&until=1364587774"
6   }
7 }
8 }

```

Código 7. Exemplo de código JSON recebido como resposta a uma requisição da Graph API

¹⁰Hypertext Transfer Protocol

¹¹GET é um tipo de Requisição utilizando o protocolo HTTP que obtém qualquer informações de um formulário identificado por um endereço. Fonte: <http://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html>

¹²<http://www.techterms.com/definition/string>

¹³JavaScript Object Notation

FQL API, Facebook Query Language, ou em português, Linguagem de consulta do Facebook. Caracterizada por ter uma sintaxe baseada na linguagem SQL, realiza consultas aos dados exibidos pela Graph API, além de possuir recursos avançados não disponíveis na outra interface de programação. Apesar de apresentar um diferencial, será descontinuada pelo Facebook nas próximas versões da plataforma de desenvolvimento da empresa. Em razão disso, não será abordada com mais detalhes.

2) *SDKs do Facebook*: Em razão da utilização das APIs do Facebook necessitarem alguns conhecimentos específicos e ainda necessitar que alguns procedimentos sejam realizados para sua utilização, como o gerenciamento da chave de acesso aos dados, foram desenvolvidas bibliotecas com essa e outras funcionalidades já implementadas, os SDKs.

Javascript SDK, [38] é mantido pelo Facebook para desenvolvimento utilizando a linguagem Javascript, o SDK provê um conjunto rico de funcionalidades feitas para executar no computador cliente podendo, adicionar plugins sociais¹⁴, realizar chamadas de API e implementar *login* para se autenticar na rede.

Dentre esses recursos, os mais interessantes para serem usados no projeto são o *login*, por ser a ferramenta recomendada para implementar a autenticação na rede e, as chamadas à Graph API, por reduzir os passos necessários para coletar os dados. Na seção Autenticação no Facebook será abordada com mais detalhes o fluxo de utilização do SDK. No código 8 é exibido a utilização do SDK, onde, inicialmente há uma chamada à função *FB.login* - responsável por obter acesso à rede, cuidando inclusive do tratamento da utilização da chave de acesso. Como parâmetro é passada a chamada à função, *FB.api* - faz chamadas à Graph API e no exemplo está fazendo uma postagem na página do usuário. Há ainda outro parâmetro, *scope*, uma requisição de permissão para a aplicação publicar em nome do usuário.

```

1 FB.login(
2   function() {
3     FB.api(
4       '/me/feed',
5       'post',
6       {message: 'Hello, world!'}
7     );
8   },
9   {scope: 'publish_actions'}
10 );

```

Código 8. Obtenção de token do usuário e postagem na linha do tempo utilizando o Javascript SDK. [43]

PHP SDK Apesar de não ter sido utilizada na construção do projeto, foi citada devido a ter suporte oficial pela rede. Possui qualidades como acesso nativo a Graph API e integração com o *login* do Facebook. Geralmente usado para desenvolver aplicações canvas, sites ou adicionando funcionalidades do lado servidor [39] - papel desempenhado na solução proposta por pelo RestFB, comentado na seção E-3 - para uma aplicação que já utiliza o SDK Javascript. No código 9, é exibida a utilização da API, uma requisição de dados do próprio usuário.

¹⁴<https://developers.facebook.com/docs/plugins/>

```

1 $request = new FacebookRequest (
2     $session,
3     'GET',
4     '/me'
5 );
6 $response = $request->execute();
7 $graphObject = $response->getGraphObject();

```

Código 9. Obtenção de dados do usuário com o SDK PHP. [43]

3) *SDKs de terceiros*: Existem diversos SDKs desenvolvidos para acessar os dados da rede, no entanto só será abordado o utilizado no projeto. Ele utiliza a linguagem Java, escolhida por ser uma linguagem bastante popular e que possui ferramentas feitas nessa linguagem que serão necessárias para a construção, como o WEKA, aplicação para mineração de dados.

Conforme pode ser visto na página de desenvolvedores do Facebook, a linguagem Java não possui suporte oficial. Apesar de não ser possível requisitar auxílio da própria empresa, existem APIs de terceiros que suprem as necessidades dos desenvolvedores de outras linguagens, com os recursos necessários e ajuda através de documentação e fóruns de desenvolvimento. O Facebook possuía uma API para a linguagem Java, que conforme [42], serviu como base para a criação da Facebook-java-api e assim como o cliente original, foi implementada em Java. No entanto, ao navegar no site do projeto¹⁵ verifica-se que a mesma também foi descontinuada e ainda em sua página, há incentivos para a utilização da API RestFB¹⁶.

4) *Autenticação no Facebook*: A navegação tradicional na rede do Facebook requer previamente a realização de autenticação do usuário. Somente após determinar a identidade de quem está no site, permite acesso ao seu e aos demais perfis de acordo as permissões possuídas na conta. Semelhante ao processo ocorrido com os usuários comuns, também faz-se necessário passar os dados das credenciais antes de obter acesso aos dados via API.

O fluxo de *login* no Facebook pode ser realizado de diversas formas, seja através da biblioteca Javascript - recomendado pelo site - ou, através de outra estratégia adequada ao projeto. Na página de desenvolvedores da rede, são listados os passos necessários para integrar o *login* do Facebook a um site. Os mesmos são descritos nas subseções abaixo.

- Primeiramente é verificado se a pessoa está logada na aplicação, ou seja, permitiu acesso a seus dados pela mesma e, se a pessoa está conectada ao Facebook.
- Solicitar *login* e permissão de acesso aos dados. Caso não esteja logado, é aberta uma janela na tela para solicitar que o usuário faça a autenticação e dê permissão de acesso ao conjunto de dados que a aplicação fará uso.
- No intuito de confirmar se a pessoa que respondeu à solicitação de *login* é a mesma que estava utilizando a aplicação no momento que a rotina foi chamada, é feita a verificação da identidade do autor da resposta. Como

o Javascript SDK já faz essa verificação, não é necessário implementá-la a menos que seja usado um SDK não oficial para autenticação.

- No fim do processo de *login* uma cadeia de caracteres é gerada, o *token* de acesso. Ele é como um passaporte que é transmitido a cada chamada da API como prova de que o autor foi uma pessoa específica que através de determinada aplicação realizou a ação. O *token* gerado pelo Javascript SDK fica automaticamente armazenado no navegador e devido a curta validade que normalmente possui, é também renovado pela biblioteca de forma transparente. Em alguns casos pode ser necessário gerar o *token* do lado servidor, nesse caso é necessário criar os *tokens* de vida longa, ou *long lived token*. Para facilitar a compreensão dos passos, as demais informações a respeito de *tokens* de acesso serão comentadas na subseção Tokens de acesso.
- Após ter a comprovação da identidade da pessoa e a permissão dela para acesso aos seus dados, a aplicação está pronta para fazer chamadas às APIs de dados do Facebook. Nesse ponto pode ser interessante realizar as chamadas via servidor, para isso, deve ser passado para o mesmo, o *token* e escolher alguma linguagem que possa ser usada no servidor como, PHP ou Java.
- Por último há um passo opcional que é colocar um botão para o usuário sair da aplicação. Esse botão possui uma característica peculiar, ao ser clicado faz o usuário ser deslogado tanto do Facebook quanto da aplicação.

Descrição da estrutura de acesso através de tokens

Ainda em uma das páginas dos desenvolvedores [43], um token de acesso é uma *string* de estrutura não variável que identifica um usuário, aplicação ou página e pode ser usada pela aplicação para realizar chamadas às APIs. O token é obtido quando alguém conecta a uma aplicação usando o *login* do Facebook, ganhando assim acesso temporário e seguro às APIs da rede. Existem quatro tipos de tokens de acesso no Facebook: User Access Token, App Access Token, Page Access Token e Client Token. Desses, somente os dois primeiros serão necessários para o projeto.

User Access Token ou Token de Acesso do Usuário, é o tipo de token mais utilizado, necessário para qualquer chamada às APIs seja para leitura, modificação ou escrita. São obtidos a partir das janelas de *login* e requerem a permissão de uma pessoa para ser obtido. Embora existam diversas maneiras de obter os tokens de acesso do usuário, há uma estrutura básica que normalmente é seguida. Primeiro, através de uma biblioteca é chamada uma janela de *login* e requisitado ao cliente que efetue o *login* e/ou dê permissão de acesso aos seus dados. Após, o cliente deve se autenticar e dar as permissões solicitadas. Para em seguida, ser retornado um token de acesso. Os passos são demonstrados na Figura 18.

Na página de documentação [43] divide os tokens em dois tipos: *Short-Lived Tokens* e *Long-lived Tokens*. Os *Short-Lived Tokens* tem duração de uma a duas horas, enquanto os *Long-lived Tokens*, possuem aproximadamente 60 dias de validade. Apesar de ser uma informação oficial, é necessário utilizar

¹⁵<https://code.google.com/p/Facebook-java-api/>

¹⁶De acordo a definição do próprio projeto, RestFB é um simples e flexível cliente para as APIs do Facebook, Graph e a antiga REST, escrito em Java. Disponível em <http://restfb.com/>

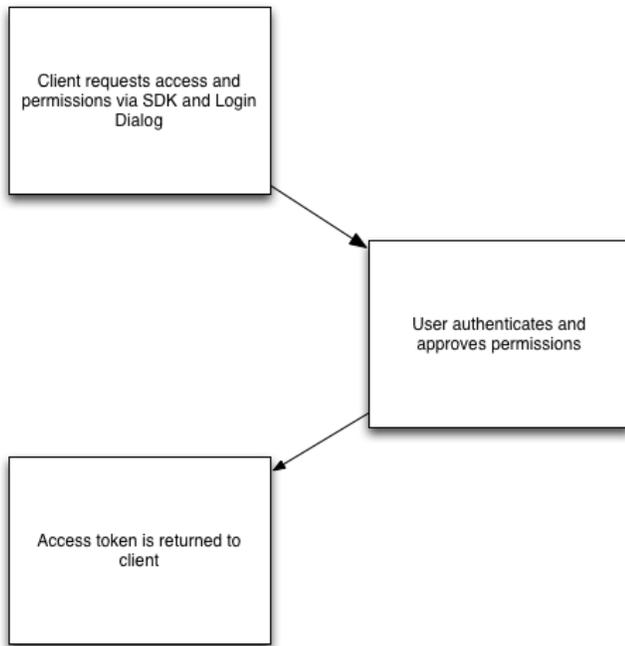


Figura 18. Fluxo básico de obtenção de token do usuário. [43]

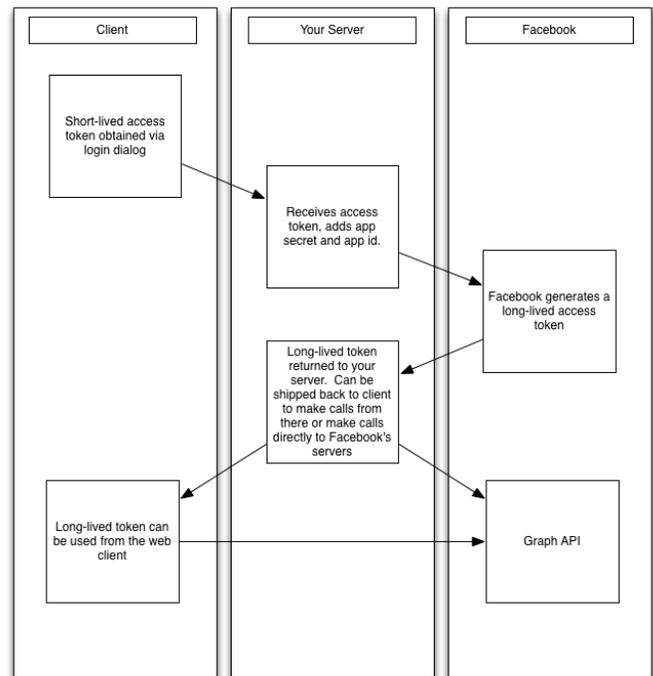


Figura 19. Fluxo de obtenção de token estendido. [43]

técnicas de tratamento de erro para tratar erros de vencimento, pois, na própria página informa a possibilidade de mudança de validade sem aviso prévio. Os tokens obtidos pelas janelas de *login*, são *Short-Lived Tokens*, mas podem ser estendidos para *Long-lived Tokens*. As APIs oficiais do Facebook gerenciam automaticamente o ciclo de vida dos tokens, portanto, quando se utiliza iOS, Android e Javascript SDK não é necessário nenhum procedimento para renovar os tokens, pois já são atualizados antes de expirar. Já em aplicações onde não há o gerenciamento automático do ciclo de vida do token, é aconselhado que sejam gerados os tokens de longa duração. Para isso estender o token, os passos são: do lado cliente é obtido o token de curta duração, em seguida é enviado ao servidor acompanhado do *App Secret* e do *App Id*, esses dados são submetidos ao Facebook e são devolvidos ao servidor em forma de um token de longa validade que pode ser enviado ao lado cliente para que faça requisições por lá ou permanecer no servidor, realizando as requisições diretamente. Na Figura 19 é representado esse fluxo.

Por razões de segurança, a extensão da validade dos tokens deve ser feita sempre do lado servidor devido a chamada utilizar o *App Secret*.

Já os *App Access Token*, ou Tokens de Acesso de Aplicação, são usados para fazer requisições relativas às aplicações. Podem modificar parâmetros da aplicação, criar e gerenciar os usuários de teste ou consultar as estatísticas de acesso da aplicação. Podem também publicar no lugar de uma pessoa que tenha dado a permissão *open_graph*. Assim como acontece com a extensão de token, não deve ser feita a requisição de token de aplicação via cliente, por também utilizar o *App*

Secret.

APÊNDICE F MONTAGEM DE URL PARA GERAR NOTIFICAÇÕES NO FACEBOOK

No código 10 é exibido como a URL para requisitar a notificação é montada, usando no processo informações como: *App_Id*, *App_Secret*, identificação do usuário(*USER_ID*) e a mensagem a ser enviada.

```

1 TOKEN_APP="291802777181539|33
  JCX5KnKHk9cMVeImujIPd1XaA", //App_Id +
  App_Secret
2 targetURL="https://graph.facebook.com/" + USER_ID
  + "/notifications",
3 mensagem="Novidade%20no%20ar!",
4 urlDestino="index.html",
5 urlParameters="access_token="+TOKEN_APP+"&href="+
  urlDestino+"&template="+mensagem;
6 urlFinal = targetURL + '?' + urlParameters;
  
```

Código 10. Exemplo gerar notificação utilizando Graph API

APÊNDICE G DETALHES DA IMPLEMENTAÇÃO NO FACEBOOK

Segundo [28], o primeiro passo para criar uma aplicação é entrar na página destinada a desenvolvedores e fazer a solicitação. Em seguida, serão solicitadas algumas informações pertinentes à nova aplicação a ser criada. Dentre elas merecem destaque: o nome que a aplicação será conhecida e poderá ser buscada - *Display Name* - e, o identificador textual que aparecerá no endereço da aplicação - *Namespace*. Após criada,

serão fornecidas duas cadeias de caracteres de extrema importância e que devem ser guardadas com segurança devido ao poder que elas dão a quem as possui. São elas o identificador numérico da aplicação - *App ID* - e a outra é uma cadeia alfanumérica exibida somente mediante digitação de senha de acesso à rede do administrador da aplicação - *App Secret*. As duas são utilizadas no processo de desenvolvimento para conseguir acesso à aplicação, suas informações e os dados dos usuários.

As aplicações do Facebook são carregadas dentro da página Canvas, considerada quase uma página em branco dentro do Facebook onde, as aplicações são executadas. O conteúdo da página Canvas é carregado através de uma URL¹⁷ que contenha HTML¹⁸, Javascript¹⁹ e CSS²⁰, formando assim a aplicação. Depois de ter solicitado a criação da aplicação, o próximo passo a ser realizado é definir as informações relativas a essa página. Os principais campos a serem preenchidos são o endereço usado para fazer a requisição da aplicação do Facebook - *Canvas Page* -, o endereço origem do conteúdo a ser exibido dentro do Canvas - *Canvas URL* - e o endereço para carregar o conteúdo criptografado do Canvas - *Secure Canvas URL* [28].

¹⁷Uniform Resource Locator

¹⁸HyperText Markup Language

¹⁹Javascript é uma linguagem de script multi-paradigma, baseada em protótipo que é dinâmica, e suporta estilos de programação orientado a objetos, imperativo e funcional. Fonte: <https://developer.mozilla.org/pt-BR/docs/JavaScript>

²⁰Cascading Style Sheets