

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE SISTEMAS DE INFORMAÇÃO – BACHARELADO

APLICATIVO WEB PARA AUXILIAR NA DOCUMENTAÇÃO
E PROCURA DE PROGRAMAS SIMILARES EM COBOL

FERNANDO BUGMANN

BLUMENAU
2012

2012/2-13

FERNANDO BUGMANN

APLICATIVO WEB PARA AUXILIAR NA DOCUMENTAÇÃO

E PROCURA DE PROGRAMAS SIMILARES EM COBOL

Trabalho de Conclusão de Curso submetido à Universidade Regional de Blumenau para a obtenção dos créditos na disciplina Trabalho de Conclusão de Curso II do curso de Sistemas de Informação— Bacharelado.

Prof. Cláudio Ratke, Mestre - Orientador.

BLUMENAU
2012

2012/2-13

APLICATIVO WEB PARA AUXILIAR NA DOCUMENTAÇÃO E PROCURA DE PROGRAMAS SIMILARES EM COBOL

Por

FERNANDO BUGMANN

Trabalho aprovado para obtenção dos créditos na disciplina de Trabalho de Conclusão de Curso II, pela banca examinadora formada por:

Presidente: _____
Prof. Cláudio Ratke, Mestre – Orientador, FURB

Membro: _____
Prof. Mauro Marcelo Mattos, Doutor – FURB

Membro: _____
Prof. Roberto Heinzle, Doutor – FURB

Blumenau, 03 de dezembro de 2012.

Dedico este trabalho a meus pais, familiares e amigos, especialmente aqueles que me ajudaram diretamente na realização deste.

AGRADECIMENTOS

A Deus, pelo seu imenso amor e graça.

À minha família, que sempre me deu forças para lutar.

Aos meus amigos, pelos empurrões e cobranças.

À T-Systems que permitiu me ausentar da empresa quando precisei.

Ao meu orientador, Cláudio Ratke, por seu auxílio e por ter acreditado na conclusão deste trabalho.

Aos professores do Departamento de Sistemas e Computação da Universidade Regional de Blumenau por suas contribuições durante os semestres letivos.

Se você quer ser bem sucedido, precisa ter dedicação total, buscar seu último limite e dar o melhor de si mesmo.

Ayrton Senna

RESUMO

Este trabalho apresenta um aplicativo para ambiente *Web* que auxilia no desenvolvimento de programas COBOL, permitindo fazer a busca de códigos fonte por determinado(s) termo(s) trazendo resultado(s) específico(s) ou semelhante(s). É utilizada a técnica de Mineração de Texto para efetuar a busca, com o objetivo de maximizar os resultados devolvidos pelo aplicativo. Os códigos fonte resultantes da busca podem ser utilizados como base para o desenvolvimento de novos programas. Foi desenvolvido na linguagem C#, utilizando banco de dados SQL Server 2008 para armazenamento das informações. Ele também permite ao usuário manter manuais técnicos e/ou de procedimentos. Como resultado destaca-se o ganho de tempo no desenvolvimento de novos programas, já que o aplicativo retorna códigos fonte cadastrados em sua base de dados que tenham semelhança com o programa a ser desenvolvido. Também possibilita a pesquisa por procedimentos já realizados, por exemplo, quando o programador precisa realizar uma tarefa e ela já foi feita e cadastrada no aplicativo, ele poderá seguir as ações de acordo com o registro, ganhando tempo na execução dela. Os resultados obtidos foram satisfatórios, pois o aplicativo auxilia na tarefa de busca de um código fonte base para o desenvolvimento de novos programas. Ele também auxilia a documentar as tarefas realizadas no dia-a-dia.

Palavras-chave: *Web*. Mineração de texto. COBOL. Código fonte.

ABSTRACT

This work presents an application to a Web environment that assists in developing COBOL programs, allowing you to search for specific source code(s) term(s) bringing the result(s) specific(s) or similar(s). It used the technique for Text Mining to search, aiming to maximize the results returned by the application. The source code search result can be used as the basis for the development of new programs. It was developed in C # using SQL Database Server 2008 for storing information. It also allows the user to maintain technical manuals and/or procedures. As a result highlights the gain of time in developing new programs, since the application source code returns registered in its database that have similarity with the program being developed. It also allows searching by procedures already performed, for example, when the programmer needs to accomplish a task and she has already been made and registered in the application, it may follow the actions according to the record, buying time in implementing it. The results were satisfactory, since the application assists in the task of looking for a source basis for the development of new programs. It also helps to document the tasks performed on a day-to-day.

Key-words: Web. Text mining. COBOL. Source Code.

LISTA DE FIGURAS

| | |
|---|----|
| Figura 1 - Índice de programas exemplo | 20 |
| Figura 2 - Tela de visualização gráfica..... | 21 |
| Figura 3 - Interface da ferramenta | 22 |
| Figura 4 - Descrição dos botões da barra de ferramentas..... | 22 |
| Figura 5 - Diagrama de casos de uso | 26 |
| Figura 6 - Diagrama de atividades do cadastro de programas..... | 27 |
| Figura 7 - Modelo Entidade Relacionamento..... | 28 |
| Figura 8 - Organização dos projetos do aplicativo | 30 |
| Figura 9 - <i>Logon</i> do sistema | 31 |
| Figura 10 - Menu do aplicativo | 32 |
| Figura 11 - Administra Manuais..... | 33 |
| Figura 12 - Dados do Manual | 33 |
| Figura 13 - Administra Módulos | 34 |
| Figura 14 - Dados do Módulo | 35 |
| Figura 15 - Administra Usuários | 35 |
| Figura 16 - Dados para autenticação | 36 |
| Figura 17 - Administra Áreas | 36 |
| Figura 18 - Dados da Área..... | 37 |
| Figura 19 - Administra Sinônimos | 38 |
| Figura 20 - Dados do Sinônimo..... | 38 |
| Figura 21 - Administra Radicais..... | 39 |
| Figura 22 - Dados do Radical..... | 39 |
| Figura 23 - Administra <i>Stopwords</i> | 40 |
| Figura 24 - Dados da <i>Stopword</i> | 41 |
| Figura 25 - Administra Programas | 41 |
| Figura 26 - Dados do Programa..... | 42 |
| Figura 27 - Resultado da busca pelas palavras chaves "imposto de renda" | 44 |
| Figura 28 - Código fonte COBOL do programa CALCIRPF | 45 |
| Figura 29 - Código fonte COBOL do programa CALCTAXA..... | 45 |
| Figura 30 - Resumo das respostas da questão 1 | 46 |
| Figura 31 - Resumo das respostas da questão 2 | 47 |

| | |
|--|----|
| Figura 32 - Resumo das respostas da questão 3 | 47 |
| Figura 33 - Resumo das respostas da questão 4 | 47 |
| Figura 34 - Formulário de avaliação do trabalho desenvolvido | 64 |

LISTA DE QUADROS

| | |
|---|----|
| Quadro 1 - Experimentos em mineração de texto realizados por Wives..... | 17 |
| Quadro 2 - Requisitos funcionais | 24 |
| Quadro 3 - Requisitos não funcionais | 25 |
| Quadro 4 - Função para buscar radical da palavra | 43 |
| Quadro 5 - Select dinâmico montado para buscar os fontes..... | 44 |
| Quadro 6 - Descrição do Caso de Uso “Efetuar Login” | 51 |
| Quadro 7 - Descrição do Caso de Uso “Alterar Senha” | 52 |
| Quadro 8 - Descrição do Caso de Uso “Manter Programas” | 53 |
| Quadro 9 - Descrição do Caso de Uso “Manter Manuais Técnicos” | 54 |
| Quadro 10 - Descrição do Caso de Uso “Buscar Programas” | 55 |
| Quadro 11 - Descrição do Caso de Uso “Manter Acessos ao Sistema” | 56 |
| Quadro 12 - Descrição do Caso de Uso “Manter Sinônimos” | 57 |
| Quadro 13 - Descrição do Caso de Uso “Manter Stopwords” | 58 |
| Quadro 14 - Descrição do Caso de Uso “Manter Radicais” | 59 |
| Quadro 15 - Descrição do Caso de Uso “Manter Áreas” | 60 |
| Quadro 16 - Dicionário de dados da tabela Manual | 61 |
| Quadro 17 - Dicionário de dados da tabela Modulo..... | 61 |
| Quadro 18 - Dicionário de dados da tabela Programa_Modulo | 62 |
| Quadro 19 - Dicionário de dados da tabela Programa..... | 62 |
| Quadro 20 - Dicionário de dados da tabela Usuario..... | 62 |
| Quadro 21 - Dicionário de dados da tabela Area..... | 63 |
| Quadro 22 - Dicionário de dados da tabela Radical | 63 |
| Quadro 23 - Dicionário de dados da tabela Stopword..... | 63 |
| Quadro 24 - Dicionário de dados da tabela Radical | 63 |

LISTA DE SIGLAS

COBOL - *Common Business Oriented Language*

C# – C Sharp (Linguagem de programação da Microsoft baseada em C)

CSS - *Cascading Style Sheets*

DFD - Diagrama de Fluxo de Dados

MER - Modelo Entidade Relacionamento

RF - Requisito Funcional

RNF - Requisito Não Funcional

TI - Tecnologia da Informação

VS – Microsoft Visual Studio (Ambiente de desenvolvimento .NET)

SUMÁRIO

| | |
|---|-----------|
| 1 INTRODUÇÃO..... | 12 |
| 1.1 OBJETIVOS DO TRABALHO | 13 |
| 1.2 ESTRUTURA DO TRABALHO | 14 |
| 2 FUNDAMENTAÇÃO TEÓRICA | 15 |
| 2.1 GESTÃO DO CONHECIMENTO..... | 15 |
| 2.2 MINERAÇÃO DE TEXTO | 16 |
| 2.3 COBOL..... | 18 |
| 2.4 SISTEMA ATUAL | 19 |
| 2.5 TRABALHOS CORRELATOS..... | 20 |
| 3 DESENVOLVIMENTO DO APLICATIVO | 23 |
| 3.1 LEVANTAMENTO DE INFORMAÇÕES | 23 |
| 3.2 ESPECIFICAÇÃO | 24 |
| 3.2.1 Requisitos Funcionais | 24 |
| 3.2.2 Requisitos não Funcionais | 24 |
| 3.2.3 Diagrama de casos de uso | 25 |
| 3.2.4 Fluxo de Atividades | 26 |
| 3.2.5 Modelo Entidade Relacionamento | 28 |
| 3.3 IMPLEMENTAÇÃO | 29 |
| 3.3.1 Técnicas e ferramentas utilizadas..... | 29 |
| 3.3.2 Operacionalidade da implementação | 31 |
| 3.3.3 Funcionalidade da implementação..... | 42 |
| 3.4 RESULTADOS E DISCUSSÃO | 45 |
| 4 CONCLUSÕES..... | 48 |
| 4.1 EXTENSÕES | 48 |
| REFERÊNCIAS | 49 |
| APÊNDICE A – Descrição dos Casos de Uso | 51 |
| APÊNDICE B – Detalhamento do dicionário de dados..... | 61 |
| ANEXO A - Formulário de avaliação do trabalho Desenvolvido | 64 |

1 INTRODUÇÃO

Possivelmente a informação tornou-se o principal instrumento da transição da sociedade industrial para a Sociedade do Conhecimento. Nesta nova sociedade, os profissionais com maior valor são aqueles que possuem a habilidade de comunicá-la e usá-la para o avanço do conhecimento e a capacidade que uma empresa tem em acumular capital intelectual (SABBAG, 2007).

Para que as pessoas tenham a possibilidade de evoluir tecnicamente e/ou profissionalmente, elas precisam adquirir informação suficiente para realizar seu trabalho do dia-a-dia. Para tanto não devem perder tempo com os problemas que outros já resolveram, ou seja, não é coerente investir tempo na resolução dos mesmos problemas. A partir do momento em que há uma boa base de procedimentos, as pessoas têm a oportunidade de crescerem juntas, pois criam e compartilham as informações, dessa forma a resolução de muitos problemas fica registrada e na próxima vez que alguém se deparar com o mesmo, já vai existir a resposta no aplicativo.

A parte mais difícil da gestão do conhecimento é adequar os recursos tecnológicos com os conhecimentos pessoais. Pois a gestão do conhecimento não é apenas a criação de um banco de dados central que armazene a repetição das experiências e informações que é de conhecimento dos trabalhadores ou dos sistemas de informação como um todo (LARA, 2004).

Em uma empresa de serviços onde o bem mais precioso é o conhecimento coletivo sobre os clientes, processos e a concorrência, as informações fazem o papel mais importante no trabalho de cada colaborador. A divisão de atribuições no trabalho é feita para que se alcancem os objetivos da empresa. É crescente a ênfase na "espiral do conhecimento", tendo-se como foco conhecimentos específicos sobre métodos, técnicas e gestão da informação (ROSINI; PALMISANO, 2003).

Visto a importância da informação em uma empresa de serviços, foi construído neste trabalho um aplicativo para auxiliar no compartilhamento da informação. O foco do trabalho desenvolvido está nas equipes que desenvolvem programas COBOL.

Foi desenvolvido um aplicativo específico para buscar fragmentos de programas para auxiliar no desenvolvimento de novos programas. Pois esta demanda ficava por conta do programador, em especial, esta é uma dificuldade maior para os programadores novos.

Atualmente, quando um programador inicia o desenvolvimento de um programa, ele

acaba interagindo com outro programador para saber se este tem um programa com as funcionalidades do mesmo que ele irá desenvolver, para que ele possa aproveitar o código fonte. Assim sendo, os outros programadores perdem tempo, muitas vezes, procurando um programa para o colega aproveitar.

Outro problema é que os manuais técnicos que existem são de propriedade do cliente, não podendo ser alterados ou complementados pelos programadores. Portanto é interessante a proposta de que os programadores possam criar seus manuais para auxiliar na prática da programação.

A idéia de desenvolver um aplicativo de biblioteca de fontes e procedimentos utilizando a técnica de mineração de texto surgiu porque os programas na linguagem COBOL, são muito similares uns com os outros. Por esse motivo, um aplicativo para auxiliar o programador a selecionar um programa para orientar, ou seja, um que tenha funcionalidades semelhantes ao programa que ele irá desenvolver diminui muito o tempo gasto no desenvolvimento, sobrando mais tempo para os testes unitários.

1.1 OBJETIVOS DO TRABALHO

O objetivo geral do trabalho é desenvolver um aplicativo *web* que pesquise na base de fontes em COBOL e recupere fontes que possam ter similaridade com a tarefa atual com base na mineração de texto, para auxiliar o desenvolvimento de programas no ambiente *Mainframe*.

Os objetivos específicos do trabalho proposto são:

- a) criação de uma biblioteca de fontes COBOL;
- b) criação de uma biblioteca de manuais e procedimentos;
- c) utilizar técnicas de similaridade;
- d) implementar técnica de mineração de texto no aplicativo.

1.2 ESTRUTURA DO TRABALHO

No primeiro capítulo tem-se a introdução ao tema principal deste trabalho com a apresentação da justificativa e dos objetivos.

No segundo capítulo apresenta-se a fundamentação teórica pesquisada sobre Gestão do Conhecimento, Mineração de Texto, COBOL, bem como os trabalhos correlatos.

O terceiro capítulo apresenta o desenvolvimento do aplicativo, com as técnicas e ferramentas utilizadas, alguns diagramas para melhor compreendê-lo, a operacionalidade dele e resultados e discussões.

No quarto capítulo têm-se as conclusões deste trabalho bem como se apresentam sugestões para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo aborda assuntos a serem apresentados nas seções a seguir, tais como a gestão do conhecimento, mineração de texto, COBOL, sistema atual além de trabalhos correlatos.

2.1 GESTÃO DO CONHECIMENTO

Atualmente é importante que as pessoas busquem mais conhecimento para desempenhar suas tarefas do dia-a-dia, principalmente no segmento de TI, onde as informações mudam constantemente. Simonetti (1996 apud ROSINI; PALMISANO 2003, p. 104) falam a respeito do aumento e transformação no conhecimento disponível:

Para Simonetti (1996), no século XX, foi vivenciado um processo contínuo de brutal aumento e de transformação no conhecimento disponível. Atualmente, diz-se que o conhecimento dobra a cada cinco anos. Isso leva a uma obsolescência do conhecimento adquirido, dentro da própria existência, e a uma necessidade de permanente atualização desses conhecimentos. Na prática, o aumento do conhecimento leva à necessidade das pessoas serem o que podemos chamar de “multi-especialistas atualizados”, isto é, dominar profundamente mais de uma área de conhecimento e manter atualizados esses conhecimentos.

Segundo Turban, Wetherbe e Mclean (2002, p. 329), a gestão do conhecimento é baseada em conceitos de aprendizado organizacional e memória organizacional. Quando os membros de uma empresa são colaborativos dentro do grupo, trocam experiências, ensinam e aprendam, o conhecimento é transformado e passado de pessoa a pessoa.

A memória organizacional é uma forma de guardar, representar e compartilhar o conhecimento da empresa. As pessoas, quando se veem diante de problemas a serem solucionados, recorrem a essa memória. A inteligência humana utiliza a memória da empresa e acrescenta-lhe valor, desta forma é criado um novo conhecimento. O sistema de gestão do conhecimento é capaz de absorver esse novo conhecimento e torná-lo disponível em sua forma maximizada (TURBAN; WETHERBE; MCLEAN, 2002).

O aprendizado organizacional é o desenvolvimento de um novo conhecimento e compreensão que pode influenciar o comportamento da empresa. A tecnologia da informação

tem um papel vital no aprendizado organizacional visto que as empresas estão se tornando cada vez mais de natureza virtual (TURBAN; WETHERBE; MCLEAN, 2002).

De acordo com Turban, Wetherbe e Mclean (2002, p. 331) um sistema de gestão do conhecimento, para funcionar, deve seguir 6 passos de um ciclo. O ciclo funciona da seguinte maneira:

- a) criar conhecimento – cria-se conhecimento quando as pessoas descobrem novas formas de fazer as coisas;
- b) capturar conhecimento – identificar se o conhecimento é relevante e uma forma de representá-lo;
- c) depurar o conhecimento – organizar o conhecimento de tal forma que possa ser facilmente acessado;
- d) armazenar conhecimento – armazenar conhecimento de forma a possibilitar o acesso a todos que busquem o conhecimento;
- e) administrar conhecimento – para não se tornar obsoleto, o conhecimento precisa ser constantemente revisado para certificar-se que ainda seja relevante;
- f) difundir conhecimento – o conhecimento deve estar acessível a todos que precisem dele.

Conforme Davenport (2003, p. 193) o sucesso da gestão do conhecimento depende de um conjunto de habilidades humanas, técnicas e econômicas. Esses atributos devem ser vistos não apenas na empresa de modo geral, mas também em projetos unitários. Costuma ser difícil ver todos estes aspectos em uma única equipe, mas não é impossível.

2.2 MINERAÇÃO DE TEXTO

Mineração de Texto, conforme Tufféry (2011, p. 627), é o conjunto de métodos e técnicas utilizadas para o processamento eletrônico de grandes volumes de dados em linguagem natural de texto, os quais estão armazenados em sistemas computacionais, para fins de extração e estruturação de conteúdos e temas. Esse processo visa auxiliar a descoberta de dados escondidos nos textos, bem como a tomada de decisão automática. Pode-se dizer esquematicamente que Mineração de Texto é a junção de Lexicometria com Mineração de dados.

Para Rezende (2005, p. 338), Mineração de Texto é um conjunto de técnicas e

processos que está sendo utilizada atualmente em diversas áreas para a descoberta de conhecimento inovador. O Quadro 1 apresenta um passo-a-passo que Wives (2004, p. 79) utilizou para implementação de mineração de texto na descoberta de conhecimento em texto.

| Experimento | Objetivo |
|---|---|
| (1) Normal | Verificar o resultado de um processo de agrupamento tradicional, no qual todas as palavras são utilizadas sem qualquer espécie de pré-processamento, a fim de verificar o ganho real na utilização dos conceitos. |
| (2) <i>Stopwords</i> | Verificar o resultado do processo de agrupamento com o pré-processamento mais comum e simples que corresponde à exclusão de palavras irrelevantes devido à sua categoria linguística (artigos, preposições, etc). |
| (3) <i>Stemmer</i> | Verificar a influencia de um <i>stemmer</i> (reduzidor de palavras ao seu radical) no resultado do agrupamento. |
| (4) <i>Stopwords</i> + <i>stemmer</i> | Avaliar a combinação dos dois métodos anteriores. |
| (5) Conceitos | Avaliar a influencia dos conceitos no processo de agrupamento. |
| (6) <i>Stopwords</i> + conceitos | Identificar se a remoção de <i>stopwords</i> facilita a identificação de conceitos e se melhora o resultado do agrupamento. |
| (7) <i>Stopwords</i> + <i>stemmer</i> + Conceitos | Identificar se a remoção de <i>stopwords</i> e o <i>stemmer</i> facilitam a identificação de conceitos e se melhoram o resultado do agrupamento. |
| (8) Refinamento dos conceitos pelo <i>WordWeb</i> | Identificar se a adição manual de sinônimos (similar, <i>synonyms</i>) e subtipos (<i>sub-type</i>), com base no dicionário <i>WordWeb</i> , melhora a qualidade dos conceitos. OBS: também se exclui <i>stopwords</i> e se realiza <i>stemmer</i> . |
| (9) Refinamento por subdivisão de conceitos | Verificar se conceitos mais específicos (menos abrangentes) descrevem melhor os documentos e obtêm melhores resultados. |

Fonte: adaptado de Wives (2004, p. 79).

Quadro 1 - Experimentos em mineração de texto realizados por Wives

Dos passos descritos no Quadro 1, foi utilizado na solução do problema proposto neste

trabalho, *Stopwords*, *stemmer* e conceitos, os quais serão detalhados a seguir.

Conforme Tufféry (2011), *Stopwords* são palavras que não agregam nenhum valor na análise de um texto: por exemplo, artigos e preposições. Como estas palavras não têm importância no texto, elas são eliminadas da busca, sendo possível o ganho de tempo no processamento.

Para Miner *at al.* (2012, p. 47) *Stemmer* é o processo de normalização de palavras relacionadas em apenas uma forma. Tipicamente este processo inclui a identificação e retirada de prefixos, sufixos e pluralizações inapropriadas. Como exemplo, podem-se citar as palavras andar, anda, andou, andador a assim por diante, as quais um típico algoritmo de *stemming* transformaria em "and".

Os conceitos que Wives (2004) utilizou consistem em identificar palavras sinônimas para buscar no texto, utilizando um exemplo em particular deste trabalho, as palavras "tabela" e "arquivo" representam o mesmo conceito, o qual em um código fonte seriam tipos de base de dados.

2.3 COBOL

COBOL é a abreviação da palavra inglesa *Common Business Oriented Language*, traduzindo para o português significa Linguagem Comum Orientada aos Negócios. Tem como objetivo permitir o desenvolvimento de aplicações comerciais para pessoas que não tem um profundo conhecimento de computadores. Por esse motivo que a linguagem COBOL utiliza de palavras normais da língua inglesa. A estrutura é parecida com um texto, dividido em parágrafos e frases. Depois que é desenvolvido o programa, conhecido também como programa fonte, é feita a tradução para a linguagem do computador, que o converte em um programa objeto. O compilador é o responsável por esta tarefa (CAMPOS, 2010).

Um compilador é um programa que converte informações da linguagem em informações binárias que o sistema do computador entende e representa. Em agosto de 1961, foi editada a primeira versão dessa linguagem, permitida para exercer algumas diretrizes como: Orientação comercial, Independência (portabilidade) de máquina e Modular (CAMPOS, 2010).

A história do COBOL teve início em 1959, onde em uma reunião no Pentágono em Washington - EUA, usuários de computadores na indústria privada e governamental,

fabricantes de computadores e outros interessados, decidiram elaborar uma fonte apropriada para a linguagem de dados comerciais. O encontro teve como objetivo discutir a necessidade de criar uma linguagem específica para serviços dentro de dados comerciais. Teve como resultado a criação da comissão denominada de *Conference On Data System Languages* (CODASYL). No ano de 1960, a gramática foi apresentada, chamada de Relatório COBOL-60, que continha um grande número de equívocos, mas logo no ano seguinte foram corrigidos. Surgiu então o COBOL-61, que trouxe uma versão mais ampla do COBOL, exposto em um Relatório provisório “COBOL-61 – Versão Estendida”, apresentada em 1962 (DORNELLES, 2012).

O COBOL apresenta várias características positivas: é correspondida por arquivos direcionados como armazenamento e restauração de informações; suporta bem o idioma inglês e; suas instruções são de fácil entendimento (DORNELLES, 2012).

O COBOL ainda é uma linguagem influente no mundo dos negócios. Estima-se que existem mais de 220 bilhões de linhas de COBOL, que equivalem a cerca de 80 por cento do código usado ativamente no mundo. Oitenta por cento das transações de ponto de vendas são realizadas através desta linguagem, e estima-se que existem mais de um milhão de programadores COBOL no mundo. A cada dia ocorrem 200 vezes mais transações COBOL do que buscas no Google (CRAIG, 2011).

Há sempre novas necessidades, por isso o COBOL é constantemente revisado e atualizado. Vale lembrar que COBOL não é uma linguagem universal para todos os tipos de processamento (DORNELLES, 2012).

2.4 SISTEMA ATUAL

Atualmente são utilizados para cadastrar os programas desenvolvidos no ambiente *Mainframe*, documentos texto no formato do Word. Por exemplo, alguns programadores que fazem sua própria documentação e armazenam em seu computador, não ficando disponível para os demais membros da equipe.

No início do desenvolvimento do programa, geralmente após a leitura da especificação, o programador busca um programa similar ao que ele irá desenvolver para poder reproduzir ou usar o código fonte como referência. Caso não encontre ele indaga os colegas sobre um programa similar ao que ele irá desenvolver.

Os manuais técnicos ou manuais de módulos e sub-rotinas servem para descrever os parâmetros utilizados na chamada dos mesmos. Eles são de propriedade do cliente, não podem ser alterados e/ou complementados pelos programadores. Há uma opção de acessar esses manuais dentro do próprio *Mainframe*, mas a interface de acesso não facilita a busca e a visualização dos mesmos, sendo assim, não é muito eficaz essa função. A Figura 1 demonstra um exemplo de um documento Word com o nome do programa e suas funcionalidades, utilizado para consultas:

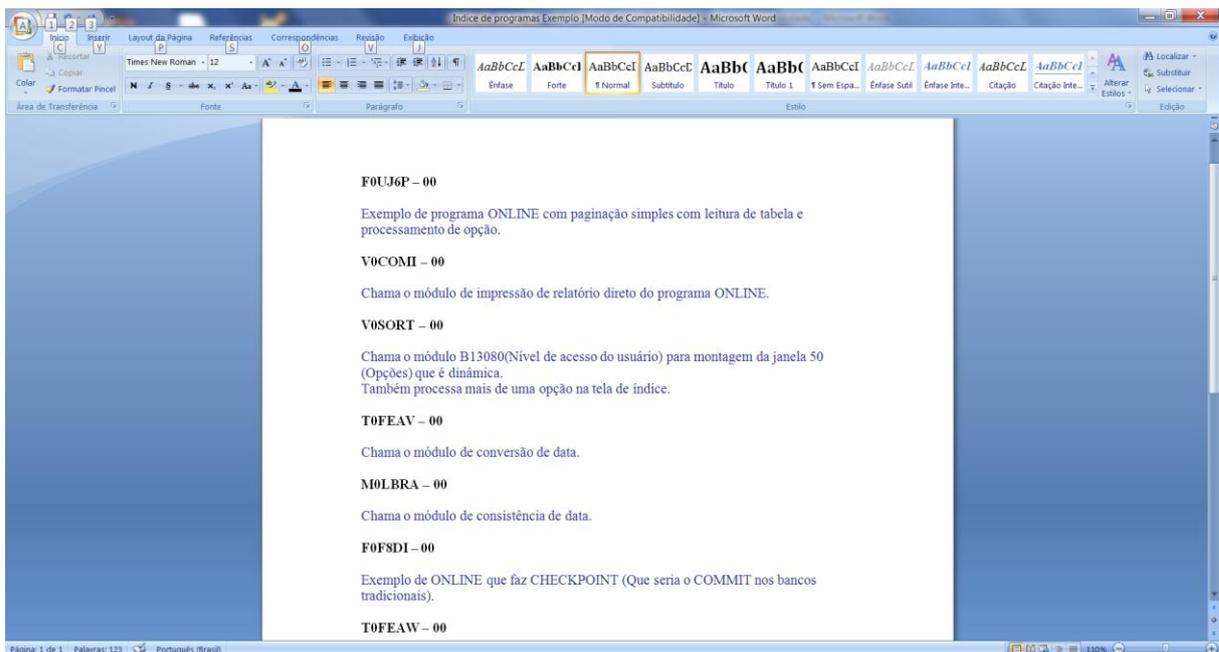


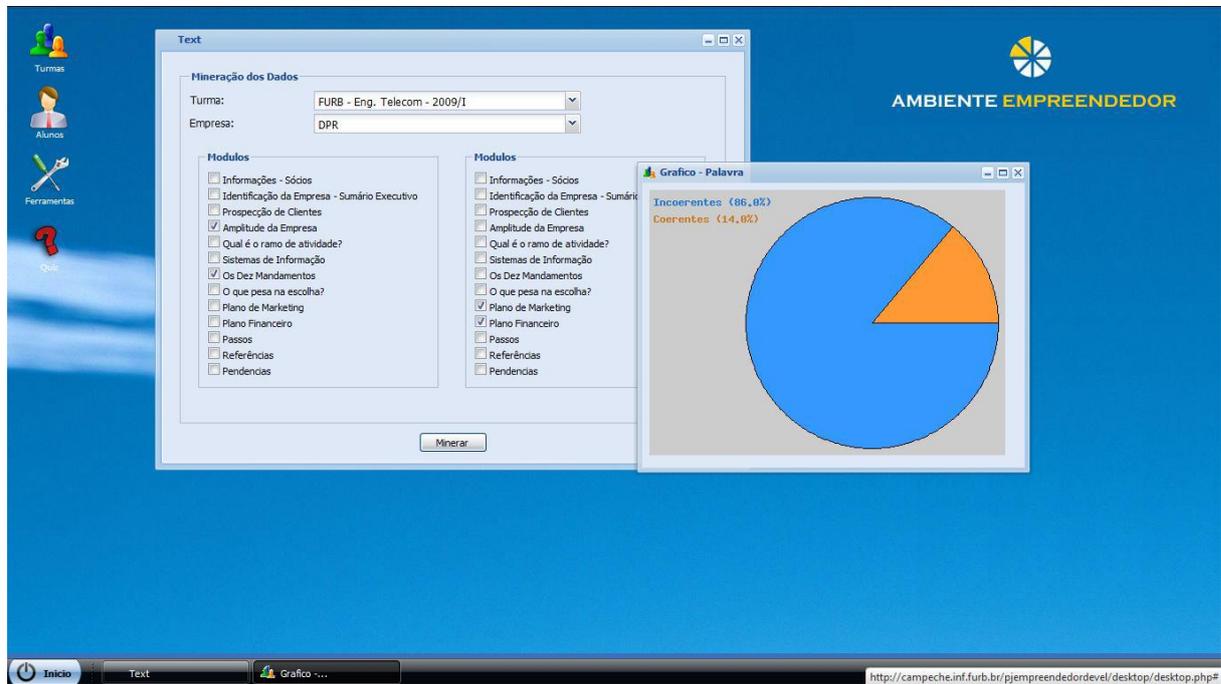
Figura 1 - Índice de programas exemplo

2.5 TRABALHOS CORRELATOS

Podem-se citar como trabalhos correlatos os trabalhos de Gaulke (2011), Justino (1999) e Ramos (1998), todos eles para conclusão do curso de Ciência da Computação na Universidade Regional de Blumenau.

Gaulke (2011) implementou a técnica de mineração de texto no Ambiente do Empreendedor. O objetivo foi comparar textos entre os módulos para que fossem verificadas as inconsistências nos planos de negócio criados no Ambiente do Empreendedor. O trabalho tinha o intuito de auxiliar o gestor na análise do plano de negócio. Foram desenvolvidas as funções de: pesquisa por assunto, pesquisa por empresa já desenvolvida, visualização gráfica

das porcentagens das comparações efetuadas entre os módulos conforme Figura 2 e a visualização de um relatório com todos os assuntos já desenvolvidos.



Fonte: Gaulke (2011).

Figura 2 - Tela de visualização gráfica

Justino (1999) desenvolveu uma “ferramenta de apoio ao processo de reutilização de especificação estruturada”. O trabalho teve por objetivo criar um processo prático para facilitar a reutilização de especificações estruturadas, permitindo ao usuário reutilizar diagramas de fluxo de dados, modelos entidade-relacionamento e dicionários de dados. Na Figura 3 é apresentada a interface da ferramenta.

O ambiente é estruturado através de um menu principal com as opções de Arquivo, Editar, Componentes e Ajuda. Abaixo do menu principal está localizada uma barra de ferramentas para acesso aos comandos mais comuns. Ao lado esquerdo está a barra de ferramentas de desenho, que serve para a construção de DFD e MER. A Figura 4 tem o objetivo de mostrar a descrição dos botões da barra de ferramentas.



Fonte: Justino (1999).

Figura 3 - Interface da ferramenta



Fonte: Justino (1999).

Figura 4 - Descrição dos botões da barra de ferramentas

Ramos (1998) desenvolveu uma “ferramenta para gerenciamento de componentes reutilizáveis em ACCESS” abrangendo o estudo da reusabilidade e técnicas de classificação e gerenciamento de componentes reutilizáveis.

3 DESENVOLVIMENTO DO APLICATIVO

Neste capítulo são apresentadas as características do aplicativo, por meio de fluxogramas, especificação de requisitos funcionais e não funcionais, diagrama de casos de uso e do Modelo de Entidade Relacional (MER) que definem o ciclo de desenvolvimento do aplicativo. Também são mostradas as técnicas e ferramentas utilizadas no processo de implementação, a operacionalidade do aplicativo e os resultados obtidos.

3.1 LEVANTAMENTO DE INFORMAÇÕES

Foi desenvolvido um aplicativo *web* de biblioteca de fontes e procedimentos utilizando a técnica de mineração de texto para auxiliar o desenvolvimento de programas COBOL. É um aplicativo que facilite para o programador encontrar programas exemplo antes de começar o desenvolvimento de um novo programa.

O aplicativo permite o cadastro de programas e suas funcionalidades/módulos. Dessa forma pode ser feita uma busca pelos programas com determinadas funcionalidades/módulos de acordo com a necessidade do novo programa a ser desenvolvido. Isso permite ao programador, de acordo com o resultado da busca, localizar o código fonte buscado e reproduzi-lo para a biblioteca de fontes COBOL dele. Essa busca também pode ser feita por sinônimos. Por exemplo, o programador pesquisa “Calculo de imposto”, mas não há resultado exato para a palavra “Calculo de imposto”, então o aplicativo retorna um programa com uma funcionalidade similar através da busca por sinônimos, por exemplo, "Calculo de taxa". Isso é possível graças à técnica de mineração de texto.

Além das funcionalidades citadas anteriormente, o aplicativo possibilita o armazenamento de manuais técnicos, para que estes sejam de fácil acesso aos programadores. Também é possível a criação, alteração, exclusão e pesquisa de manuais, permitindo a atualização dos mesmos. Deve-se destacar que a alteração de manuais não será feita nos manuais que são de propriedade do cliente, e sim nos próprios manuais criados pelos programadores com base no conhecimento adquirido na prática da programação.

3.2 ESPECIFICAÇÃO

A seguir são apresentados os requisitos funcionais (RF), os requisitos não funcionais (RNF), o diagrama de casos de uso, fluxo de atividades e o modelo de entidade relacionamento (MER).

3.2.1 Requisitos Funcionais

O Quadro 2 apresenta os requisitos funcionais previstos para o sistema e sua rastreabilidade, ou seja, vinculação com o(s) caso(s) de uso associado(s).

| Requisitos Funcionais | Caso de Uso |
|--|--------------------|
| RF01: O sistema deverá permitir que o usuário efetue <i>login</i> no sistema. | UC01 |
| RF02: O sistema deverá permitir que o usuário altere a senha do <i>login</i> . | UC02 |
| RF03: O sistema deverá permitir o usuário manter programas e suas funcionalidades. | UC03 |
| RF04: O sistema deverá permitir o usuário manter manuais técnicos. | UC04 |
| RF05: O sistema deverá permitir o usuário fazer a busca de programas e suas funcionalidades. | UC05 |
| RF06: O sistema deverá permitir o administrador manter acessos ao sistema. | UC06 |
| RF07: O sistema deverá permitir o usuário manter sinônimos | UC07 |
| RF08: O sistema deverá permitir o usuário manter <i>stopwords</i> | UC08 |
| RF09: O sistema deverá permitir o usuário manter radicais | UC09 |
| RF10: O sistema deverá permitir o usuário manter áreas | UC10 |

Quadro 2 - Requisitos funcionais

3.2.2 Requisitos não Funcionais

O Quadro 3 lista os requisitos não funcionais previstos para o sistema.

| Requisitos Não Funcionais |
|---|
| RNF01: O sistema deverá utilizar banco de dados SQL Server 2008. |
| RNF02: O sistema será implementado em C# e ASP.NET. |
| RNF03: O sistema será construído para que diferentes equipes possam utilizá-lo. |
| RNF04: O sistema será desenvolvido para <i>web</i> . |
| RNF05: O sistema utilizará a técnica de mineração de texto. |

Quadro 3 - Requisitos não funcionais

3.2.3 Diagrama de casos de uso

Esta subseção apresenta o diagrama de casos de uso na Figura 5 do aplicativo desenvolvido, sendo que o detalhamento dos casos de uso está descrito a partir do Apêndice A.

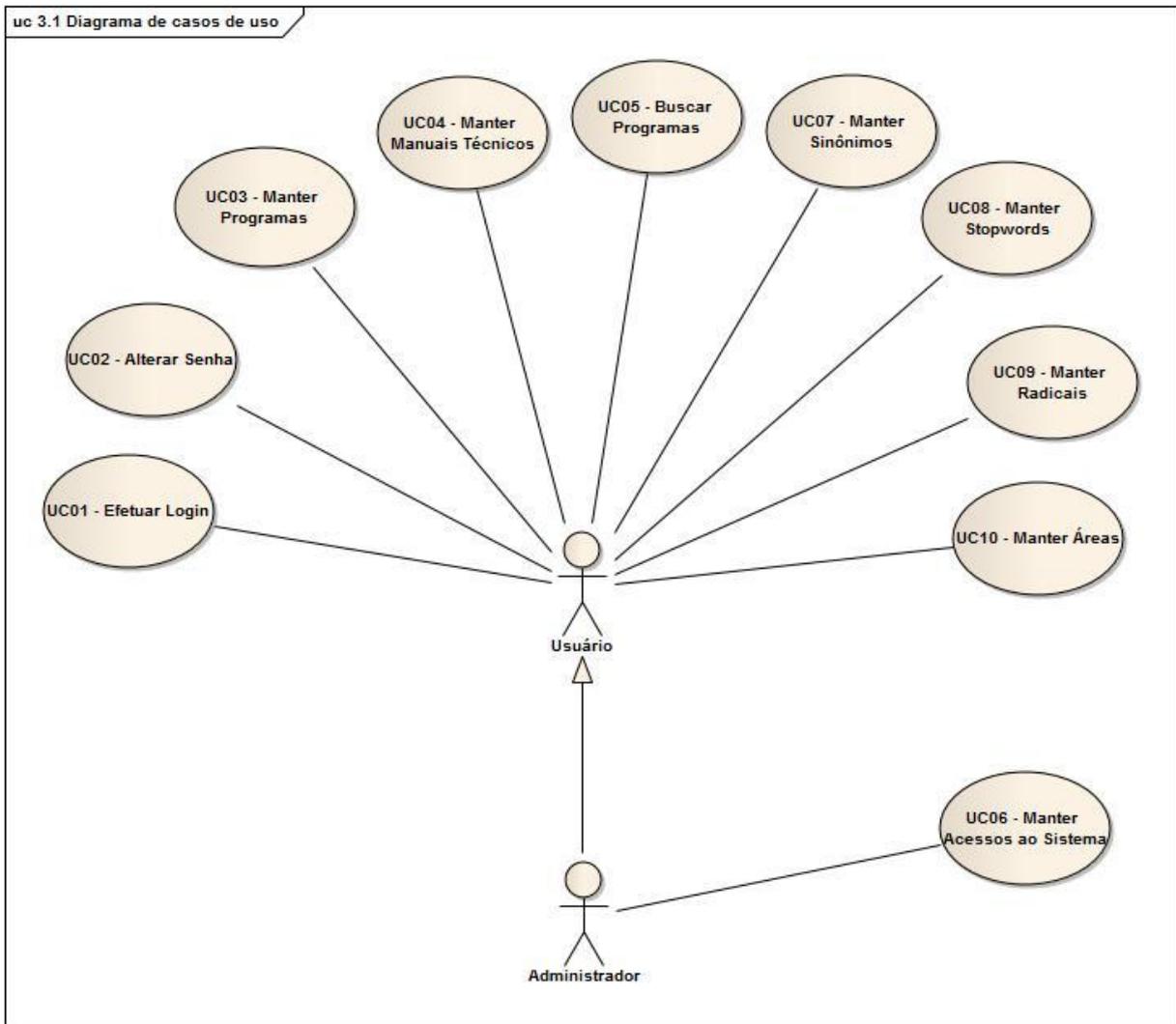


Figura 5 - Diagrama de casos de uso

3.2.4 Fluxo de Atividades

Esta subseção apresenta o fluxo de atividades do aplicativo. A Figura 6 mostra o fluxo de atividades para o cadastramento de programas, que é associado ao caso de uso UC03 - Manter Programas, considerado a base para a funcionalidade principal do programa, a de busca de códigos fonte.

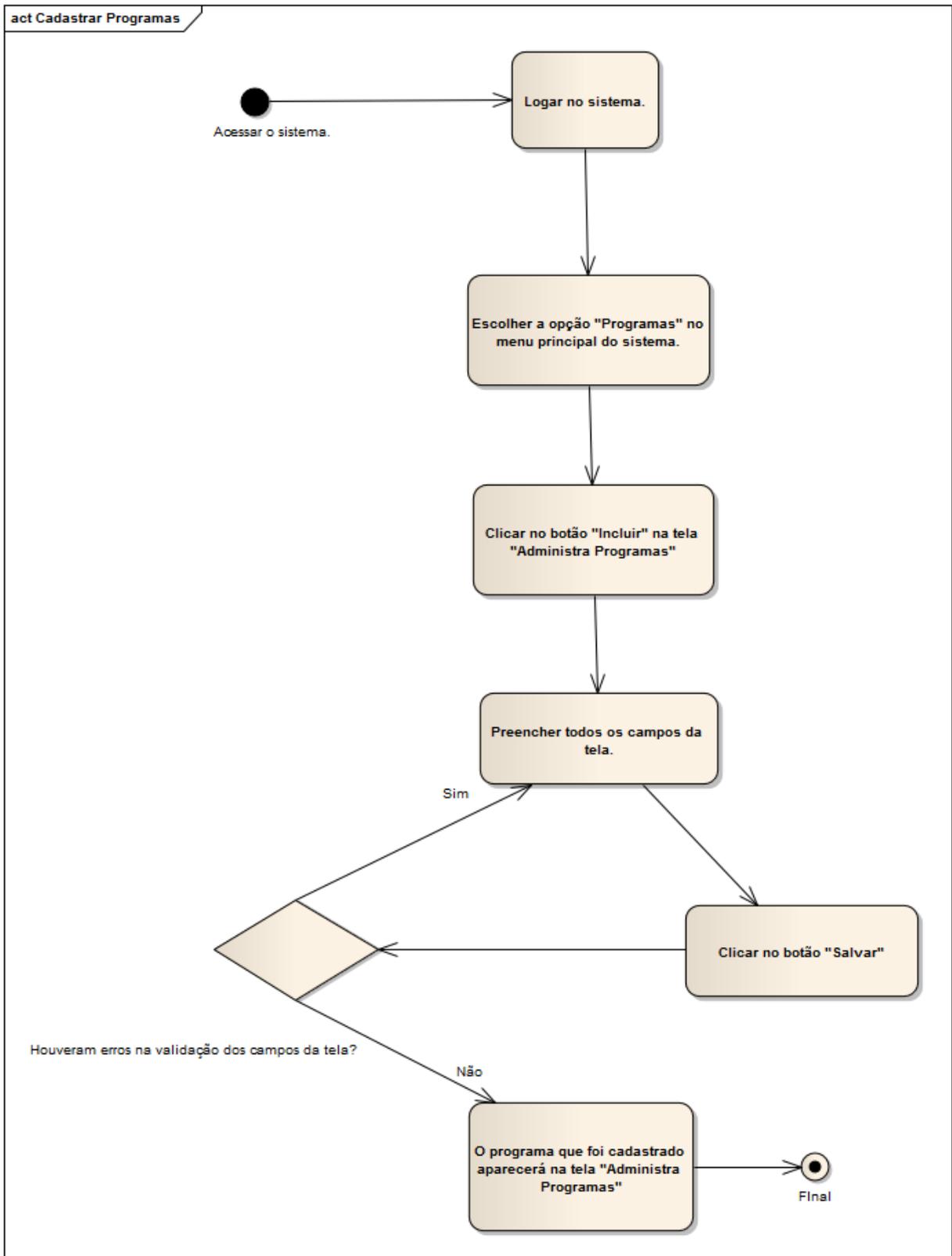


Figura 6 - Diagrama de atividades do cadastro de programas

3.2.5 Modelo Entidade Relacionamento

A Figura 7 apresenta o modelo entidade relacionamento que possui as tabelas utilizadas no aplicativo.

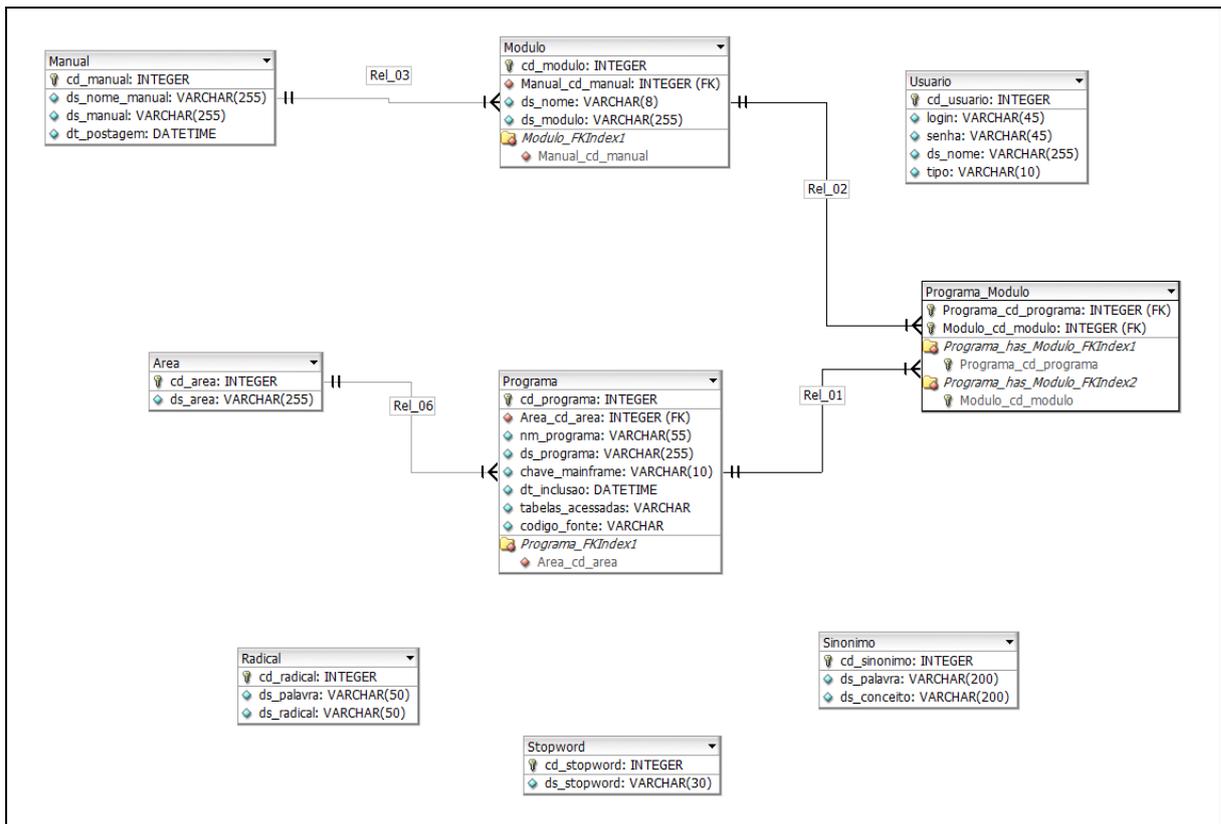


Figura 7 - Modelo Entidade Relacionamento

Foram definidas nove tabelas no banco de dados para utilizar no aplicativo desenvolvido. No Apêndice B consta o dicionário de dados.

A tabela "Manual" é utilizada para manter os manuais de procedimentos que os programadores escrevem com base na prática da programação. Também pode armazenar manuais de módulos utilizados nos programas COBOL.

A tabela "Modulo" serve para manter os módulos que são utilizados nos programas COBOL. Estes módulos são sub-rotinas padrão do sistema, por exemplo, uma função para cálculo de data.

A tabela "Usuario" é utilizada para manter os usuários que utilizam o sistema. Ela armazena as informações de *login* e senha para acesso ao sistema.

A tabela "Area" serve para manter as áreas ou departamentos da empresa. Ela tem a função de indicar a área de origem do programa.

A tabela "Programa" é a mais importante do aplicativo, pois ela mantém as informações do programa, bem como o código fonte do mesmo para que o programador possa efetuar as buscas pelo programa que deseja. A mineração de texto é feita com base nos campos "codigo_fonte" e "ds_programa" desta tabela.

A tabela "Programa_Modulo" é resultado de uma ligação n para n entre as tabelas "Programa" e "Modulo". Ela armazena a informação de quais módulos são utilizados em determinado programa. Ela é usada na busca de programas por módulos.

A tabela "Radical" mantém os radicais das palavras. Esses radicais são utilizados na busca por palavras chave, que é feita através de mineração de texto.

A tabela "Stopword" mantém artigos, preposições e outros, os quais não são utilizados na busca através da mineração de texto.

A tabela "Sinonimo" mantém os sinônimos das palavras. Esses sinônimos são utilizados para pesquisar termos similares na mineração de texto.

3.3 IMPLEMENTAÇÃO

A seguir são mostradas as técnicas e ferramentas utilizadas e a operacionalidade da implementação.

3.3.1 Técnicas e ferramentas utilizadas

No desenvolvimento do aplicativo foram utilizadas as linguagens de programação C# e Asp.Net, e o ambiente de desenvolvimento Microsoft Visual Studio 2008 com o .NET framework 3.5. Para a conexão com o banco de dados foi utilizada a biblioteca do Microsoft SQL Server 2008, que também criou o próprio banco de dados.

Os dados das tabelas radical, sinônimo e *stopword* são provenientes da base de dados da monografia realizada por Gaulke (2011). Apenas foram feitas mudanças na tabela de sinônimos para que atendesse as necessidades deste trabalho.

O aplicativo foi desenvolvido em três camadas: apresentação, negócio e dados.

Também foi dividido em dois projetos no Visual Studio (VS). O projeto BaseProgram é exclusivo da camada de apresentação, onde estão as telas organizadas em pastas específicas, imagens, *Cascading Style Sheets* (CSS), e o arquivo de configuração WebConfig. O projeto Library contém as camadas de negócio e dados. Cada tela possui sua própria classe de negócio e de dados, que estão organizadas em pastas específicas, assim como os enumeradores utilizados. A classe "Mensagem" contém todas as mensagens utilizadas no sistema e a classe "Seguranca" é responsável por criptografar a senha do usuário. Pode-se ver a organização dos projetos na Figura 8.

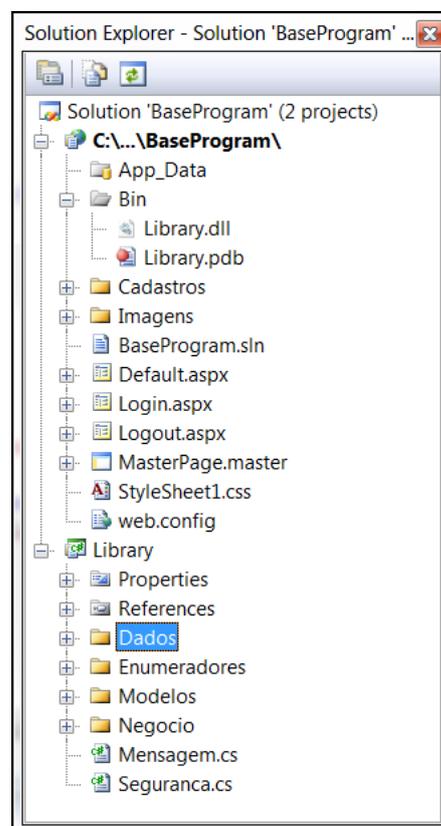


Figura 8 - Organização dos projetos do aplicativo

Para realizar a busca de programas por palavras chave foi utilizada a técnica de mineração de texto. Esta técnica consiste em primeiro retirar as *stopwords* do argumento de pesquisa digitado pelo usuário. As *stopwords* são os artigos, preposições e outros, que não tem importância na busca. Em seguida são recuperados os sinônimos das palavras digitadas pelo usuário com o intuito de enriquecer a busca, uma vez que são pesquisados os fontes que tem as palavras chaves ou as palavras sinônimas. Por fim é recuperado o radical das palavras chave e sinônimas, dessa forma a pesquisa fica mais abrangente, pois são pesquisadas palavras diferentes que possuem o mesmo radical. Por questões de desempenho, a mineração

de texto é executada através de funções e procedimentos armazenados no próprio banco de dados SQL Server.

3.3.2 Operacionalidade da implementação

Nesta subseção são apresentadas as telas do aplicativo e suas funcionalidades. Para utilizá-lo é necessário efetuar *logon* (Figura 9). Caso o usuário ou senha sejam inválidos uma tela de erro será exibida informando que usuário ou senha são inválidos.

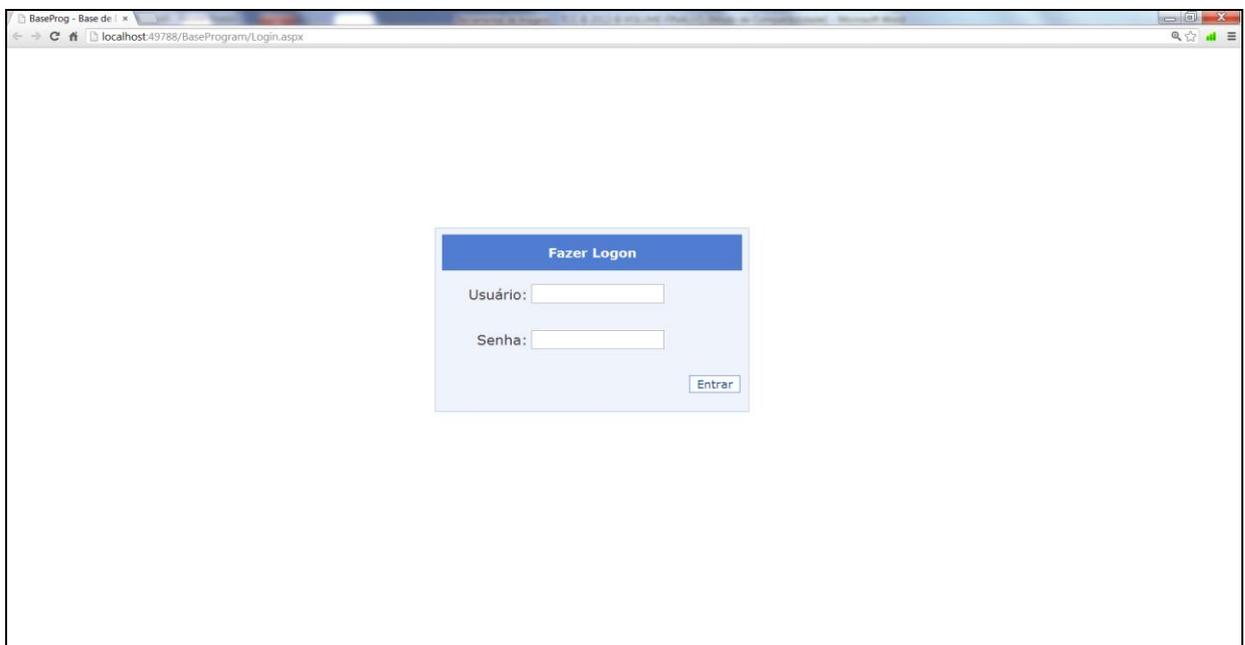


Figura 9 - *Logon* do sistema

Na Figura 10 observa-se a tela de menu principal onde é possível direcionar o processamento para as telas de cadastros específicas.



Figura 10 - Menu do aplicativo

Na Figura 11 é exibida a tela de administração de manuais, onde é possível visualizar, editar e excluir os que já estão cadastrados e incluir novos cadastros. Estas informações servem para ajudar no processo de programação, uma vez que são descritos para que os módulos sirvam e também podem ser cadastrados procedimentos. Esta funcionalidade contempla o objetivo de criação de uma base de manuais e procedimentos.

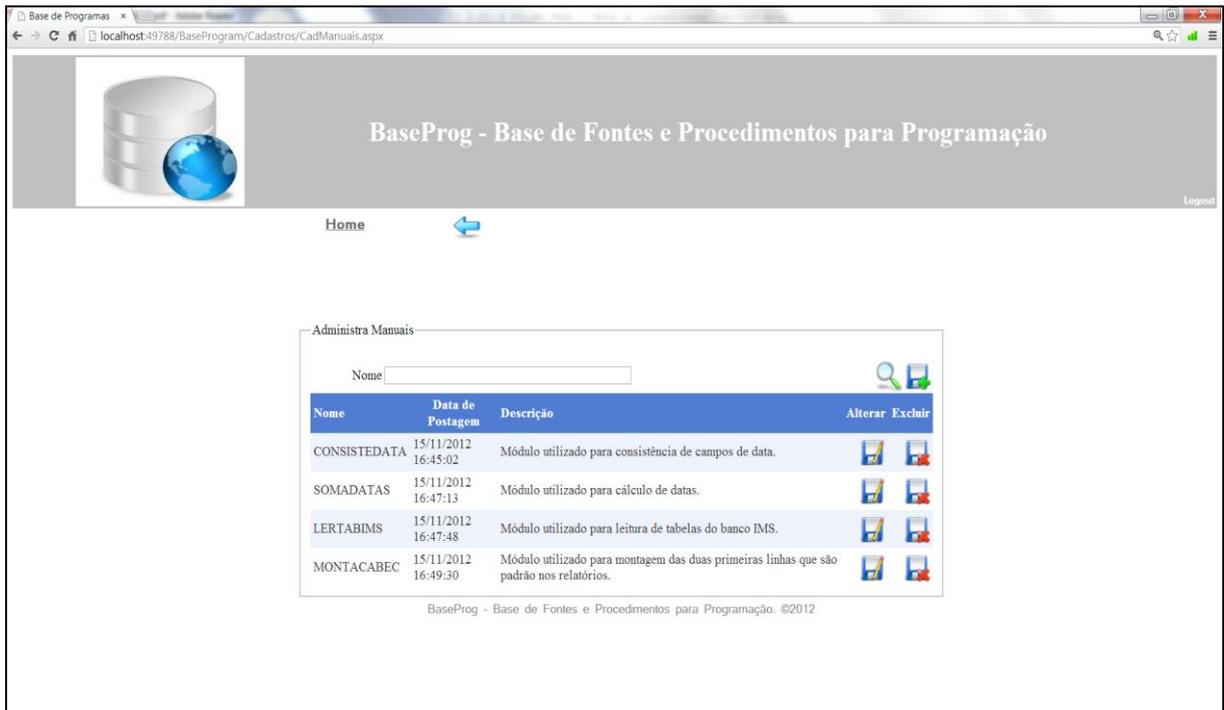


Figura 11 - Administra Manuais

A Figura 12 mostra a tela de dados dos manuais onde devem ser informados os campos nome e descrição do manual para inclusão ou alteração de manuais.



Figura 12 - Dados do Manual

Na Figura 13 é exibida a tela de administração de módulos, onde é possível visualizar, editar e excluir os que já estão cadastrados e incluir novos cadastros. Esta tela mostra os

módulos ou sub-rotinas que são utilizados nos programas.



Figura 13 - Administra Módulos

A Figura 14 mostra a tela de dados dos módulos onde devem ser informados os campos nome, descrição e manual do módulo para inclusão ou alteração de módulos. Esta tela também possui um *link* para a inclusão de manuais para o caso de ter que incluir o manual do módulo antes da inclusão do mesmo.



Figura 14 - Dados do Módulo

Na Figura 15 é exibida a tela de administração de usuários, onde é possível visualizar, editar e excluir os que já estão cadastrados e incluir novos cadastros. Esta tela permite ao administrador do aplicativo manter os acessos ao mesmo.



Figura 15 - Administra Usuários

A Figura 16 mostra a tela de dados dos usuários onde devem ser informados os campos

nome, *login*, senha, confirmação de senha e tipo para inclusão ou alteração de usuários.

The screenshot shows a web browser window with the URL `localhost:49788/BaseProgram/Cadastrros/CadastroUsuario.aspx`. The page header includes a logo of a database cylinder and a globe, and the title "BaseProg - Base de Fontes e Procedimentos para Programação". A "Home" link with a blue arrow is visible. The main content area is titled "Dados para autenticação" and contains a form with the following fields: "Nome do Usuário", "Login", "Senha", "Confirme a senha", and "Tipo". There are also icons for file upload and a red prohibition sign. The footer text reads "BaseProg - Base de Fontes e Procedimentos para Programação. ©2012".

Figura 16 - Dados para autenticação

Na Figura 17 é exibida a tela de administração de áreas, onde é possível visualizar, editar e excluir as que já estão cadastradas e incluir novos cadastros. Estas áreas servem para identificar de que área o programa faz parte.

The screenshot shows a web browser window with the URL `localhost:49788/BaseProgram/Cadastrros/CadArea.aspx`. The page header is identical to Figure 16. The main content area is titled "Administra Áreas" and features a search box labeled "Nome". Below the search box is a table with the following data:

| Nome da Área | Alterar | Excluir |
|------------------|---------|---------|
| Administração | | |
| Vendas | | |
| Produção | | |
| Contabilidade | | |
| Recursos Humanos | | |

The footer text reads "BaseProg - Base de Fontes e Procedimentos para Programação. ©2012".

Figura 17 - Administra Áreas

A Figura 18 mostra a tela de dados das áreas onde deve ser informado o campo nome para inclusão ou alteração de áreas.



The screenshot displays a web browser window with the address bar showing 'localhost:49788/BaseProgram/Cadastros/CadArea.aspx'. The page title is 'BaseProg - Base de Fontes e Procedimentos para Programação'. The main content area features a form titled 'Dados da Área' with a single text input field labeled 'Nome da Área'. A 'Home' link with a left-pointing arrow is located above the form. At the bottom of the form, there are icons for a document and a red prohibition sign. The footer of the page reads 'BaseProg - Base de Fontes e Procedimentos para Programação. ©2012'.

Figura 18 - Dados da Área

Na Figura 19 é exibida a tela de administração de sinônimos, onde é possível visualizar, editar e excluir os que já estão cadastrados e incluir novos cadastros. No processo de mineração de texto a busca de códigos fonte quando é feita por palavras chave é complementada por esses sinônimos. Por exemplo, quando se busca pela palavra chave "Imposto", na tabela de sinônimos "Imposto" tem um sinônimo cadastrado como "Taxa", dessa forma o aplicativo retorna os códigos fonte que tenham a palavra "Imposto" e também os que tenham a palavra "Taxa".



Figura 19 - Administra Sinônimos

A Figura 20 mostra a tela de dados dos sinônimos onde devem ser informados os campos palavra e sinônimo para inclusão ou alteração de sinônimos.

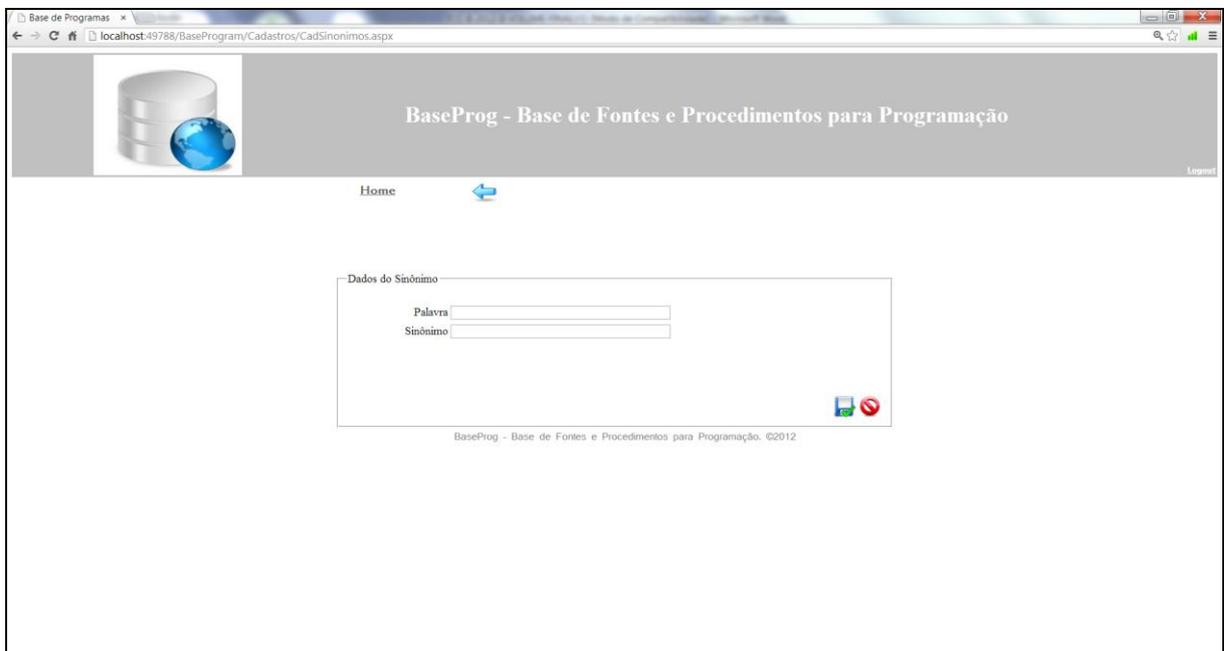


Figura 20 - Dados do Sinônimo

Na Figura 21 é exibida a tela de administração de radicais, onde é possível visualizar, editar e excluir os que já estão cadastrados e incluir novos cadastros. No processo de mineração de texto quando a busca por programas é feita por palavras chave, o aplicativo

compara o que foi digitado pelo usuário na tabela de radicais e busca os programas fonte com o radical da palavra digitada, caso ela esteja cadastrada na tabela de radicais.

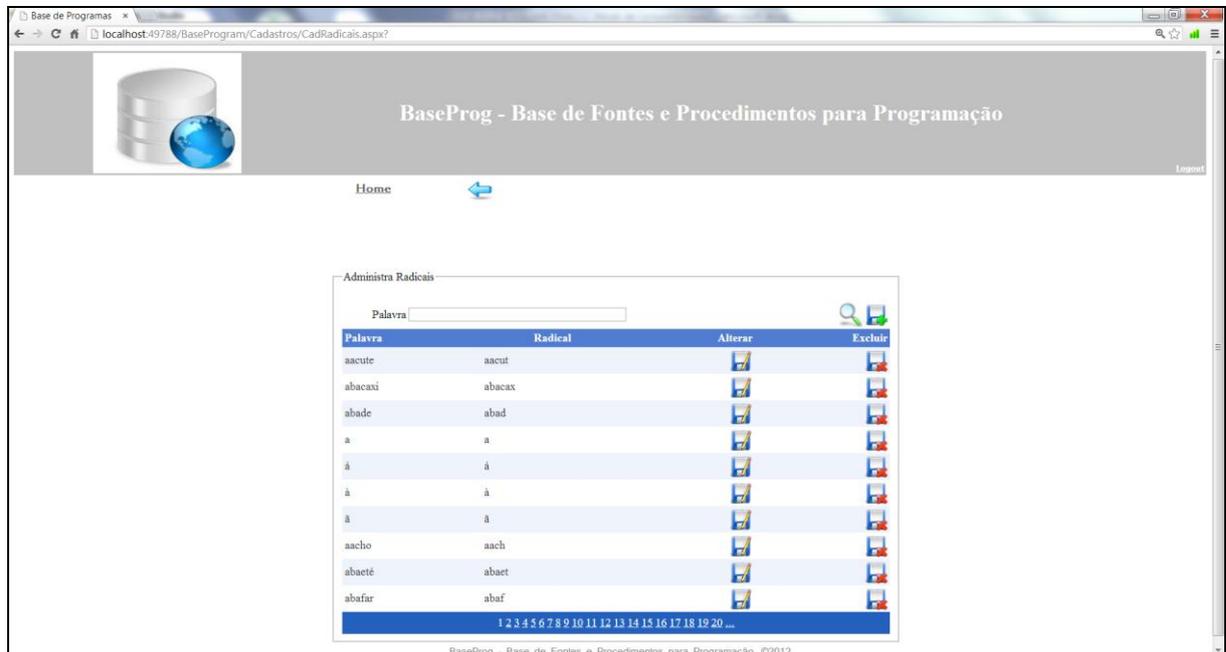


Figura 21 - Administra Radicais

A Figura 22 mostra a tela de dados dos radicais onde devem ser informados os campos palavra e radical para inclusão ou alteração de radicais.



Figura 22 - Dados do Radical

Na Figura 23 é exibida a tela de administração de *stopwords*, onde é possível visualizar, editar e excluir as que já estão cadastradas e incluir novos cadastros. No processo de mineração de texto as *stopwords* são as palavras que não tem importância na busca, ou seja, quando o usuário fizer a pesquisa por palavras chave, as palavras que ele digitar que estiverem cadastradas na tabela de *stopword* não serão procuradas nos códigos fonte. Essas palavras são os artigos, preposições.

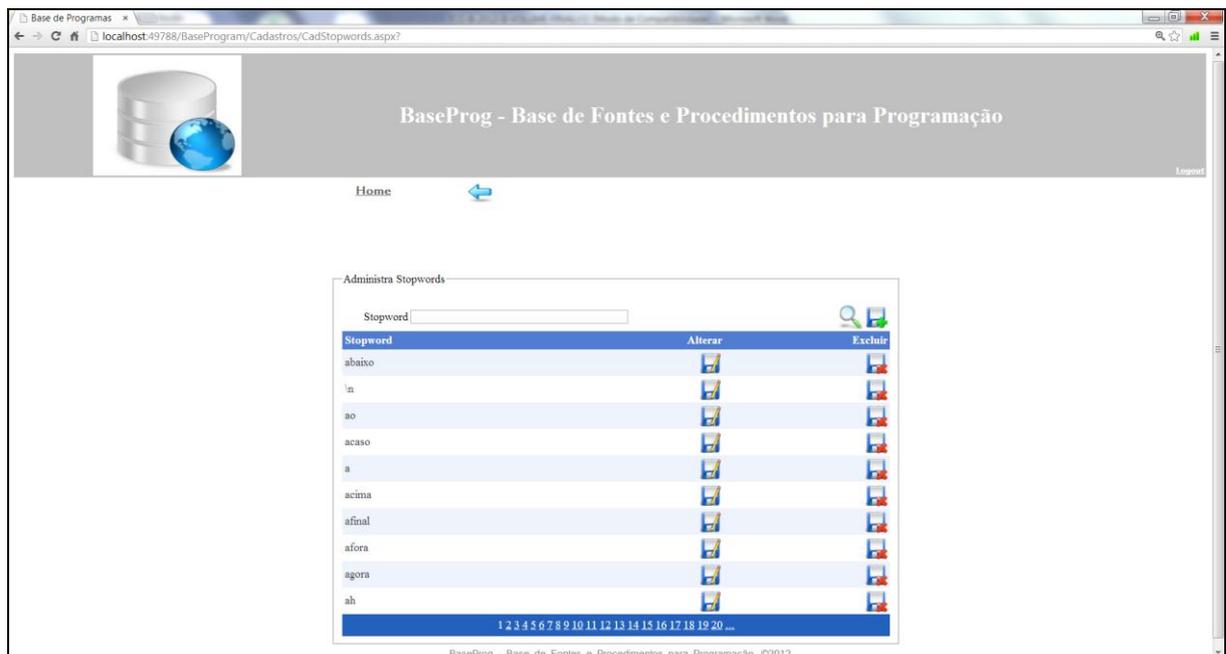


Figura 23 - Administra *Stopwords*

A Figura 24 mostra a tela de dados das *stopwords* onde deve ser informado o campo *stopword* para inclusão ou alteração de *stopwords*.



Figura 24 - Dados da *Stopword*

Na Figura 25 é exibida a tela de administração de Programas, onde é possível visualizar, editar e excluir os que já estão cadastrados e incluir novos cadastros. Nesta tela é possível pesquisar os códigos fonte pelo nome do programa, por algum módulo que ele acesse por alguma tabela que é lida ou ainda por palavras chave, onde é feito o processo de mineração de texto.



Figura 25 - Administra Programas

A Figura 26 mostra a tela de dados dos programas onde devem ser informados os campos área, nome, descrição, chave *Mainframe*. Opcionalmente podem ser selecionados os módulos que o programa acessa, informadas as tabelas que ele lê e também pode ser feito upload do código fonte do programa no formato .txt (documento de texto) para inclusão ou alteração de programas. Essa funcionalidade contempla o objetivo de criação de uma base de fontes COBOL.

BaseProg - Base de Fontes e Procedimentos para Programação

Home

Dados do Programa

Área [Selecione]

Nome

Descrição

Chave Mainframe

| Seleção | Nome do Módulo | Descrição |
|--------------------------|----------------|------------------------|
| <input type="checkbox"/> | CONSISTEDATA | Consistência de datas |
| <input type="checkbox"/> | SOMADATAS | Calcula datas |
| <input type="checkbox"/> | LERTABIMS | Leitura de tabelas IMS |
| <input type="checkbox"/> | MONTACABEC | Monta cabeçalho padrão |

Tabelas acessadas

Programa [Escolher arquivo](#) Nenhum arquivo selecionado

BaseProg - Base de Fontes e Procedimentos para Programação. ©2012

Figura 26 - Dados do Programa

3.3.3 Funcionalidade da implementação

Nesta subseção é demonstrado um caso de teste onde o usuário busca pelas palavras chaves "Imposto de Renda". Quando é clicado no botão pesquisar, o aplicativo efetua o processo de mineração de texto conforme os passos abaixo.

O primeiro passo é retirar as *stopwords* obtendo o termo "Imposto Renda" para pesquisa. O segundo passo é procurar os sinônimos para cada palavra chave na tabela de sinônimos, obtendo as seguintes palavras sinônimas para pesquisa: dinheiro, justiça, taxa, tributo, dinheiro, economia, tecido, vestuário. O terceiro passo é buscar o radical das palavras chave e sinônimas obtendo os seguintes radicais: impost, dinheir, justic, tax, tributo, rend, dinheir, econom, tec, vestuári. Para esta tarefa existe a função *buscaRadical* cadastrada no banco de dados conforme o Quadro 4.

```

USE [Baseprog]
GO
/***** Object:  UserDefinedFunction [dbo].[buscaRadical]      Script Date:
11/20/2012 18:53:37 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER FUNCTION [dbo].[buscaRadical](@frase VARCHAR(MAX))
RETURNS VARCHAR(MAX)
BEGIN
    DECLARE @palavra VARCHAR(50)
    DECLARE @radical VARCHAR(50)
    DECLARE @fraseTrab VARCHAR(MAX)

    SET @fraseTrab = ' '+@frase+' '
    DECLARE radic CURSOR FOR SELECT ' '+ds_palavra+' ',ds_radical FROM
        radical ORDER BY ds_palavra
    OPEN radic
    FETCH NEXT FROM radic
    INTO @palavra,@radical
    WHILE @@FETCH_STATUS = 0
    BEGIN
        SELECT @fraseTrab = replace(@fraseTrab, @palavra, ' '+@radical+' ')
        FETCH NEXT FROM radic
        INTO @palavra,@radical
    END
    CLOSE radic
    DEALLOCATE radic
    SELECT @fraseTrab = SUBSTRING(@fraseTrab, 2, len(@fraseTrab) - 1)
    RETURN @fraseTrab
END

```

Quadro 4 - Função para buscar radical da palavra

Por fim é montado um *select* dinâmico para selecionar os fontes de acordo com os radicais das palavras chaves e sinônimas conforme o Quadro 5.

```

SELECT * FROM programa
WHERE (1=1) AND ( codigo_fonte LIKE '%impost%' OR ds_programa LIKE '%impost%'
OR codigo_fonte LIKE '%dinheir%' OR ds_programa LIKE '%dinheir%'
OR codigo_fonte LIKE '%justic%' OR ds_programa LIKE '%justic%'
OR codigo_fonte LIKE '%tax%' OR ds_programa LIKE '%tax%'
OR codigo_fonte LIKE '%tributo%' OR ds_programa LIKE '%tributo%' )

AND ( codigo_fonte LIKE '%rend%' OR ds_programa LIKE '%rend%'
OR codigo_fonte LIKE '%dinheir%' OR ds_programa LIKE '%dinheir%'
OR codigo_fonte LIKE '%econom%' OR ds_programa LIKE '%econom%'

```

```
OR codigo_fonte LIKE '%tec%' OR ds_programa LIKE '%tec%'
OR codigo_fonte LIKE '%vestuári%' OR ds_programa LIKE '%vestuári%' )
```

Quadro 5 - Select dinâmico montado para buscar os fontes

A seguir na Figura 27 é mostrado o resultado da busca que são os programas CALCIRPF e CALCTAXA, pois o programa CALCIRPF possui as palavras "IMPOSTO" e "RENDA" em seu código fonte e o CALCTAXA possui as palavras "TAXA" e "DINHEIRO" conforme evidenciado nas Figuras 28 e 29. Com essa funcionalidade são atingidos os objetivos de utilizar técnicas de similaridade e implementar a técnica de mineração de texto no aplicativo.



The screenshot shows a web browser window with the URL `localhost:49788/BaseProgram/Cadastros/CadProgramas.aspx`. The page title is "BaseProg - Base de Fontes e Procedimentos para Programação". Below the title is a navigation bar with a "Home" link and a "Logout" button. The main content area is titled "Administra Programas" and contains a search form with fields for "Programa", "Módulo", "Tabela", and "Palavras Chave". The "Palavras Chave" field contains the text "imposto de renda". Below the search form is a table with the following data:

| Programa | Chave Mainframe | Data Inclusão | Tabelas Acessadas | Alterar | Excluir |
|----------|-----------------|---------------------|-------------------|---------|---------|
| CALCIRPF | CHAV01 | 15/11/2012 19:33:06 | | | |
| CALCTAXA | CHAV10 | 15/11/2012 19:43:42 | YUK | | |

At the bottom of the page, there is a copyright notice: "BaseProg - Base de Fontes e Procedimentos para Programação. ©2012".

Figura 27 - Resultado da busca pelas palavras chaves "imposto de renda"

```

CALCIRPF - Bloco de notas
Arquivo Editar Formatar Exibir Ajuda
011800
011900 WRITE REG-TEXTO FROM DETALHE-2
012000
012100 WRITE REG-TEXTO FROM LINHA-TRACOS
012200
012300 ELSE
012400
012500 ADD 1 TO WK-DESPREZADOS
012600
012700 END-IF.
012800
012900*-----*
013000
013100 CABECALHO.
013200
013300 ADD 1 TO WS-PAG.
013400 WRITE REG-TEXTO FROM CABECALHO-1.
013500
013600 WRITE REG-TEXTO FROM CABECALHO-2.
013700
013800 WRITE REG-TEXTO FROM LINHA-ESPACOS.
013900
014000 WRITE REG-TEXTO FROM CABECALHO-3.
014100
014200*-----*
014300
014400 INICIALIZACAO.
014500 OPEN INPUT RECEITA.
014600
014700 OPEN OUTPUT RELATI.
014800
014900*-----*
015000
015100 FINALIZACAO.
015200 DISPLAY 'LIDOS.....:' WK-LIDOS.
015300 DISPLAY 'CONSIDERADOS:' WK-CONSIDERADOS.
015400 DISPLAY 'DESPREZADOS:' WK-DESPREZADOS.
015500
015600 CLOSE RECEITA
015700 CLOSE RELATI.
015800
015900 STOP RUN.
016000*-----*
016100 CALCULAIRPF.
016200 MOVE IRPF TO WK-IRPF.
016300 MOVE ALIQUOTA TO WK-ALIQUOTA.
016500
016600
016900 /OBJECT 'RECURSOS HUMANOS'
016900 'CALCULA IMPOSTO RENDA'
016900 USING WK-IRPF
016900 WK-ALIQUOTA
017000 WK-VALOR
017100 WITHOUT TEST/.
017200
017300*-----*

```

Figura 28 - Código fonte COBOL do programa CALCIRPF

```

CALCTAXA - Bloco de notas
Arquivo Editar Formatar Exibir Ajuda
051100
051200 COMPUTE WK-PREMED = (WK-TOTEST / WK-QTDPECA).
051300 MOVE WK-PREMED TO WK-MEDIO.
051400 MOVE CAB-STATUS-3 TO BRE-VALOR.
051500 WRITE FD-BRELVAL.
051600 ADD 1 TO WK-CONTLNHA-V.
051700 INITIALIZE BRE-VALOR.
051800
051900 MOVE ALL '-' TO BRE-VALOR.
052000 WRITE FD-BRELVAL.
052100 ADD 1 TO WK-CONTLNHA-V.
052200 INITIALIZE BRE-VALOR.
052300*-----*
052400 TERMINO.
052500 CLOSE BCADPE.
052600 PERFORM CALCULA-GERAL.
052700 CLOSE BRELVA.
052800 STOP RUN.
052900*-----*
053000 CALCULATAXAIRPF.
053100
053110* VALOR EXPRESSO EM DINHEIRO
053200 MOVE VALOR TO WK-VALOR.
053300
053400
053500 /OBJECT 'RECURSOS HUMANOS'
053600 'CALCULA TAXA IRPF'
053700 USING WK-VALOR
053800 WK-TAXA
053900 WITHOUT TEST/.
054000
054100*-----*
054000 CALCULA-GERAL.
054100 MOVE CAB-STATUS-TITULO TO BRE-VALOR.
054200 WRITE FD-BRELVAL.
054300 INITIALIZE BRE-VALOR.
054400
054500 MOVE ALL '-' TO BRE-VALOR.
054600 WRITE FD-BRELVAL.
054700 INITIALIZE BRE-VALOR.
054800
054900 MOVE WK-PECAS-GER TO WK-PECAS-G
055000 MOVE CAB-STATUS-G1 TO BRE-VALOR.
055100 WRITE FD-BRELVAL.
055200 INITIALIZE BRE-VALOR.
055300
055400 MOVE WK-TOT-GERAL TO WK-TOTAL-G.
055500 MOVE CAB-STATUS-G2 TO BRE-VALOR.
055600 WRITE FD-BRELVAL.
055700 INITIALIZE BRE-VALOR.
055800
055900 COMPUTE WK-CUSTOM-GER = (WK-TOT-GERAL / WK-PECAS-GER).
056000 MOVE WK-CUSTOM-GER TO WK-MEDIO-G.
056100 MOVE CAB-STATUS-G3 TO BRE-VALOR.
056200 WRITE FD-BRELVAL.
056300 INITIALIZE BRE-VALOR.

```

Figura 29 - Código fonte COBOL do programa CALCTAXA

3.4 RESULTADOS E DISCUSSÃO

Foi desenvolvido um aplicativo *web*, que possibilita a documentação e a busca de

programas fonte similares utilizando mineração de texto e o cadastro de manuais ou procedimentos. O objetivo do aplicativo foi alcançado e todos os requisitos propostos foram desenvolvidos.

Em relação aos trabalhos correlatos, embora o foco dos demais trabalhos serem diferentes, o aplicativo desenvolvido neste trabalho tem a característica de ser desenvolvido em ambiente *web* o que facilita a interação entre os usuários.

O maior desafio deste trabalho foi em relação ao pouco conhecimento das linguagens de programação C# e ASP.NET. Muito esforço foi empregado na procura de métodos para programar as funcionalidades do programa. Com isso, este trabalho agregou conhecimento técnico que poderá ser utilizado futuramente.

Foram realizados testes no início do desenvolvimento, a cada alteração e após o fim do desenvolvimento do sistema na perspectiva de alcançar um bom nível de qualidade no aplicativo. O aplicativo facilita o trabalho do usuário para cadastrar as tabelas e para efetuar as buscas, pois as telas são padronizadas, e os dados são apresentados da mesma forma em todo o aplicativo.

Com o intuito de avaliar o resultado final deste trabalho, foi elaborado um formulário através da ferramenta Google Docs. Seis pessoas que trabalham na empresa T-Systems avaliaram o trabalho, onde 83% delas possuem de 1 a 5 anos de experiência na área de TI. Os resultados mostram claramente que este projeto proporciona benefícios na sua utilização. Detalhes dos resultados obtidos podem ser vistos nas Figuras 30, 31, 32 e 33.

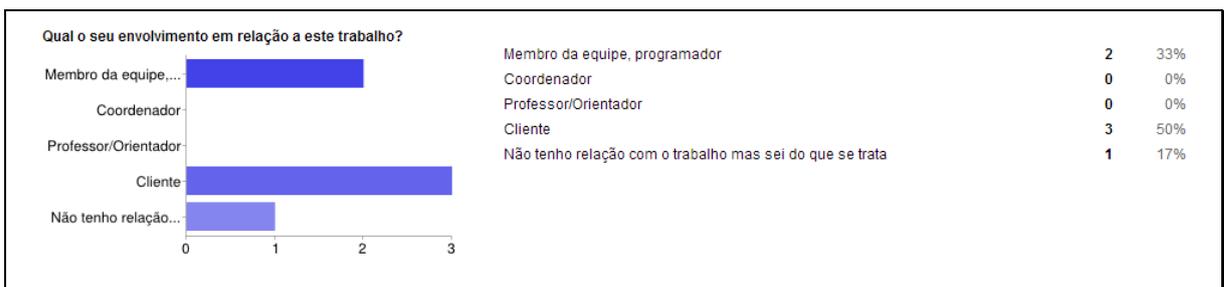


Figura 30 - Resumo das respostas da questão 1

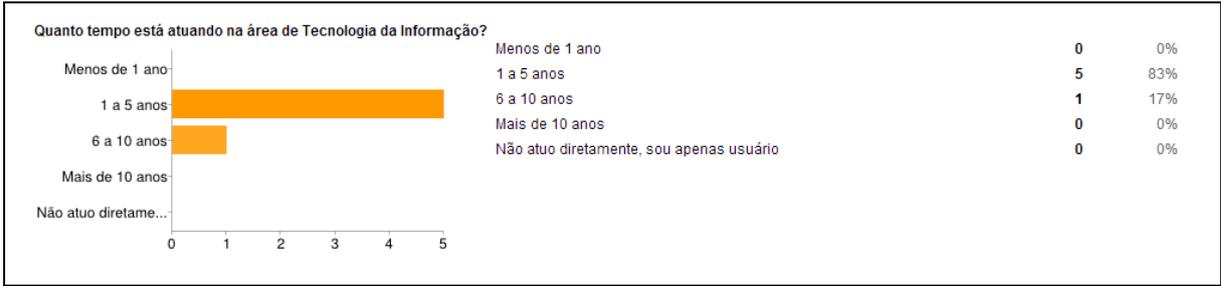


Figura 31 - Resumo das respostas da questão 2

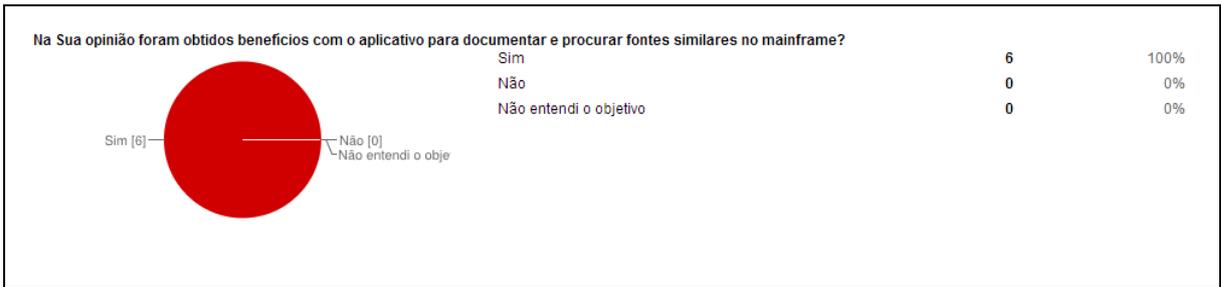


Figura 32 - Resumo das respostas da questão 3

Escreva em poucas palavras como você pode contribuir com o sucesso da implantação e continuidade da utilização deste projeto?

Posso ajudar em melhorias e oferecer a solução aos meus colegas da área de TI. Sempre ir cadastrando novos programas com seus respectivos objetos/códigos utilizados. Dessa forma a ferramenta se tornará de grande utilidade para a criação de novos programas muito mais rápido, também ajudando novos programadores em suas tarefas. Utilizando e dando feedback ao projeto. Utilizando ele e a medida que houver necessidade de uma melhoria ou uma nova função, avisar ao desenvolvedor do sistema, para haver sempre uma melhoria constante na ferramenta. Estarei trabalhando com o aplicativo diariamente, cadastra ...

Figura 33 - Resumo das respostas da questão 4

4 CONCLUSÕES

Diante dos desafios propostos neste trabalho, pode-se dizer que foram alcançados os seus objetivos. O aplicativo permite que sejam encontrados fontes através da mineração de texto, contribuindo com o programador na tarefa de encontrar um fonte similar ao que ele irá desenvolver. O aplicativo também possibilita o controle de manuais e procedimentos, os quais ficam disponíveis para todos os usuários para que utilizem na resolução de tarefas já resolvidas por outros, ganhando tempo na realização desta.

O aplicativo é *web*, o acesso é restrito conforme a permissão do tipo de usuário, pois somente os administradores podem manter o acesso ao sistema. Todas as outras funcionalidades são permitidas para todos os tipos de usuários.

A linguagem C# mostrou-se adequada para as necessidades do aplicativo, pois trabalha de forma estável com o banco de dados SQL Server, o que torna segura a programação do sistema e futuras alterações.

O investimento em um aplicativo com esta finalidade justifica-se, uma vez que facilita o trabalho do programador que poderá encontrar um programa fonte similar ao que ele irá desenvolver, fazendo com que ele ganhe tempo no desenvolvimento, conseqüentemente sobrando mais tempo para a atividade de testes do programa. Com a implantação do sistema que deve acontecer em um futuro próximo, a tendência é aumentar o nível de qualidade dos programas entregues ao cliente.

4.1 EXTENSÕES

Como sugestões para trabalhos futuros, na busca por palavras chave poderia ser colocado um grau de similaridade na tabela de sinônimos para que o usuário pudesse buscar por grau de similaridade, por exemplo: Imposto -> taxa > 0,6.

Outro caso que poderia ser implementado futuramente é a busca de programas por área, pois o cadastro de programas tem essa informação, que poderia ser levada em conta na pesquisa dos fontes, restringindo ainda mais a pesquisa.

REFERÊNCIAS

CAMPOS, Carlos. **COBOL, o que é ?**. [S.l.], 2010. Disponível em: <<http://carloscamposinfo.com/cjec/?p=144>>. Acesso em: 06 out. 2012.

CRAIG, Bruce. **COBOL: still doing the business**. [S.l.], 2011. Disponível em: <<http://www.governmentnews.com.au/2011/11/04/article/COBOL-still-doing-the-business/WPIMFXRADW.html>>. Acesso em: 14 nov. 2012.

DAVENPORT, Thomas H.; PRUSAK, Laurence. **Conhecimento Empresarial: como as organizações gerenciam o seu capital intelectual**. Tradução de Lenke Peres – Rio de Janeiro: Elsevier, 2003. – 14ª reimpressão

DORNELLES Carlos Alberto. **A história do COBOL**. Brasília: 2012. Disponível em: <http://www.cadcobol.com/histor_1.htm>. Acesso em: 06 out. 2012.

GAULKE, Roger Erivan. **Implementação da técnica *Text Mining* aplicado no ambiente do empreendedor**. 2011. 37 f. Trabalho de Conclusão de Curso (Bacharel em Ciência da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

JUSTINO, Gilvan. **Ferramenta de apoio ao processo de reutilização de especificação estruturada**. 1999. 82 f. Trabalho de Conclusão de Curso (Bacharel em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

LARA, Consuelo Rocha Dutra de. **A atual gestão do conhecimento : a importância de avaliar e identificar o capital intelectual nas organizações**. São Paulo: Nobel, 2004.

MINER, Gary et al. **Practical Text Mining : and statistical analysis for non-structured text data applications**. Waltham, MA 02451, USA: Elsevier Inc., 2012.

RAMOS, Débora Cristina Leira. **Ferramenta para gerenciamento de componentes reutilizáveis em ACCESS**. 1998. 111 f. Trabalho de Conclusão de Curso (Bacharel em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

REZENDE, S. O. **Sistemas Inteligentes: Fundamentos e Aplicações**. São Paulo, Manole, 2005.

ROSINI, Alessandro Marco; PALMISANO, Angelo. **Administração de sistemas de informação e a gestão do conhecimento**. São Paulo: Pioneira Thomson Learning, 2003.

SABBAG, Paulo Yazigi. **Espirais do conhecimento** : ativando indivíduos, grupos e organizações. São Paulo: Saraiva, 2007.

TUFFÉRY, Stéphane. **Data mining and statistics for decision making**. The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, United Kingdom: John Wiley & Sons Ltd., 2011.

TURBAN, Efraim; WETHERBE, James C.; MCLEAN, Ephraim. **Tecnologia da informação para gestão** : transformando os negócios na economia digital. São Paulo: Atmed Editora S.A., 2002.

WIVES, Leandro K. **Utilizando conceitos como descritores de textos para o processo de identificação de conglomerados (clustering) de documentos**. 2004. 136 f. Tese (Curso de Pós-Graduação em Computação) - Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre.

APÊNDICE A – Descrição dos Casos de Uso

Este Apêndice apresenta a descrição dos casos de uso conforme previstos no diagrama apresentado na subseção 3.2.3.

O Quadro 6 apresenta o detalhamento do Caso de Uso “Efetuar *Login*”.

UC01 - Efetuar *Login*

Ator: Usuário

Objetivo: Ao acessar a aplicação o usuário será direcionado a tela de *login* para fornecer os dados de acesso à aplicação.

Pré-condições: Estar cadastrado na base de dados.

Pós-condições: O usuário estará conectado no aplicativo.

Cenário Principal:

1. O usuário acessa a aplicação;
2. A aplicação é direcionada para a tela de *login*;
3. O usuário informa os dados;
4. A aplicação faz a validação dos dados;
5. A aplicação redireciona para a tela inicial.

Cenário Alternativo:

Se no item 4 o usuário informa *login* ou senha inválidos:

- 4.1 É enviada mensagem informativa para a tela;

Quadro 6 - Descrição do Caso de Uso “Efetuar Login”

O Quadro 7 apresenta o detalhamento do Caso de Uso “Alterar Senha”.

UC02 - Alterar Senha

Ator: Usuário

Objetivo: O Usuário acessa a tela de alteração de senha, informa a nova senha que ele irá utilizar e repete a nova senha para garantir a operação.

Pré-condições: Usuário deve fazer *login* no sistema.

Pós-condições: O usuário alterou a senha de acesso.

Cenário Principal:

1. Sistema apresenta a tela alterar senha;
2. Usuário informa nova senha e confirma a nova senha;
3. Usuário escolhe a opção “alterar”;
4. Sistema verifica *login* e senha atual;
5. Sistema verifica nova senha e confirmação da nova senha;
6. Sistema apresenta mensagem “Sua senha foi alterada”.

Cenário Alternativo:

Se no item 4 o *login* ou senha estiverem inválidos:

- 4.1 É enviada mensagem informativa para a tela;

Cenário Alternativo:

Se no item 5 a nova senha for diferente da confirmação da nova senha ou nova senha igual senha anterior:

- 5.1 É enviada mensagem informativa para a tela;

Quadro 7 - Descrição do Caso de Uso “Alterar Senha”

O Quadro 8 apresenta o detalhamento do Caso de Uso “Manter Programas”.

UC03 - Manter Programas

Ator: Usuário

Objetivo: O Usuário acessa a tela de cadastro de programas para informar os dados do programa. Serão cadastrados os seguintes dados: código do programa, área do programa, nome, descrição, chave no *Mainframe*, data de inclusão, módulos acessados, tabelas acessadas e código fonte.

Pré-condições: Usuário deve fazer *login* no sistema.

Pós-condições: Usuário alterou, excluiu ou incluiu um programa.

Cenário Principal:

1. Sistema informa os programas cadastrados;
2. Usuário opta por alterar, excluir ou incluir um programa;
3. Sistema verifica o preenchimento dos campos obrigatórios na alteração ou inclusão.

Cenário Alternativo:

Se no item 3 faltar preencher algum campo obrigatório:

- 3.1 É enviada mensagem informativa para a tela;

Cenário Alteração:

1. Sistema mostra os programas cadastrados;
2. Usuário seleciona um programa para alteração;
3. Sistema mostra os dados do programa para alteração;
4. Usuário realiza as alterações necessárias;
5. Usuário seleciona a opção alterar;
6. Sistema altera o registro e apresenta a mensagem “Registro Alterado com Sucesso”.

Cenário Inclusão:

1. Sistema mostra a tela de cadastro de programas;
2. Usuário informa todos os campos obrigatórios;
3. Usuário seleciona a opção incluir;
4. Sistema inclui o registro e exibe a mensagem “Registro incluído com sucesso”.

Cenário Exclusão:

1. Sistema mostra os programas cadastrados;
2. Usuário seleciona um programa para exclusão;
3. Usuário seleciona a opção excluir;
4. Sistema pede confirmação da exclusão;
5. Sistema exclui o registro e exibe a mensagem “Registro excluído com sucesso”.

Quadro 8 - Descrição do Caso de Uso “Manter Programas”

O Quadro 9 apresenta o detalhamento do Caso de Uso “Manter Manuais Técnicos”.

UC04 - Manter Manuais Técnicos

Ator: Usuário

Objetivo: O Usuário acessa a tela de cadastro de manuais técnicos para informar os dados do mesmo. Serão cadastrados os seguintes dados: código do manual, nome do manual, descrição e data de postagem.

Pré-condições: Usuário deve fazer *login* no sistema.

Pós-condições: Usuário alterou, excluiu ou incluiu um manual.

Cenário Principal:

1. Sistema informa os manuais técnicos cadastrados;
2. Usuário opta por alterar, excluir ou cadastrar um manual técnico;
3. Sistema verifica o preenchimento dos campos obrigatórios na alteração ou exclusão.

Cenário Alternativo:

Se no item 3 for verificada alguma inconsistência no preenchimento dos campos:

- 3.1 É enviada mensagem informativa para a tela;

Cenário Alteração:

1. Sistema mostra os manuais cadastrados;
2. Usuário seleciona um manual para alteração;
3. Sistema mostra os dados do manual para alteração;
4. Usuário realiza as alterações necessárias;
5. Usuário seleciona a opção alterar;
6. Sistema altera o registro e apresenta a mensagem “Registro Alterado com Sucesso”.

Cenário Inclusão:

1. Sistema mostra a tela de cadastro de manuais;
2. Usuário informa todos os campos obrigatórios;
3. Usuário seleciona a opção incluir;
4. Sistema inclui o registro e exibe a mensagem “Registro incluído com sucesso!”.

Cenário Exclusão:

1. Sistema mostra os manuais cadastrados;
2. Usuário seleciona um manual para exclusão;
3. Usuário seleciona a opção excluir;
4. Sistema exclui o registro e exibe a mensagem “Registro excluído com sucesso”.

Quadro 9 - Descrição do Caso de Uso “Manter Manuais Técnicos”

O Quadro 10 apresenta o detalhamento do Caso de Uso “Buscar Programas”.

UC05 - Buscar Programas

Ator: Usuário

Objetivo: O usuário acessa a tela de programas onde ele pode informar os seguintes campos para pesquisa: programa, módulos acionados, tabelas ou palavras chave.

Pré-condições: Usuário deve fazer *login* no sistema.

Pós-condições: O sistema retorna o nome do programa e a chave onde o mesmo se encontra no *Mainframe*.

Cenário Principal:

1. Sistema apresenta a tela de busca detalhada de programas;
2. Usuário informa programa, módulos acionados, tabelas ou palavras chave;
3. O usuário seleciona a opção pesquisar;
4. O sistema apresenta o programa e a chave onde o mesmo está cadastrado no *Mainframe*.

Cenário Alternativo:

Se no item 4 não houver resultado para a pesquisa:

- 4.1 É enviada mensagem informativa para a tela;

Quadro 10 - Descrição do Caso de Uso “Buscar Programas”

O Quadro 11 apresenta o detalhamento do Caso de Uso “Manter Acessos ao Sistema”.

UC06 - Manter Acessos ao Sistema

Ator: Administrador

Objetivo: O Administrador seleciona a opção cadastro de usuários, onde ele informará os campos usuário, senha e confirmação de senha

Pré-condições: Administrador deve fazer *login* no sistema.

Pós-condições: Administrador incluiu ou excluiu um usuário no sistema.

Cenário Principal:

1. Administrador opta por incluir ou excluir um usuário;
2. Administrador preenche os campos usuário, senha e confirmação de senha;
3. Sistema valida os campos digitados;

Cenário Alternativo:

Se no item 3 for verificada alguma inconsistência no preenchimento dos campos:

- 3.1 É enviada mensagem informativa para a tela;

Cenário Inclusão:

1. Sistema mostra a tela de cadastro de usuários;
2. Administrador informa todos os campos obrigatórios;
3. Administrador seleciona a opção incluir;
4. Sistema inclui o registro e exibe a mensagem “Registro incluído com sucesso”.

Cenário Exclusão:

1. Sistema mostra os usuários cadastrados;
2. Administrador seleciona um usuário para exclusão;
3. Administrador seleciona a opção excluir;
4. Sistema exclui o registro e exibe a mensagem “Registro excluído com sucesso”.

Quadro 11 - Descrição do Caso de Uso “Manter Acessos ao Sistema”

O Quadro 12 apresenta o detalhamento do Caso de Uso “Manter Sinônimos”.

UC07 - Manter Sinônimos

Ator: Usuário

Objetivo: O Usuário acessa a tela de cadastro de sinônimos para informar os dados do mesmo. Serão cadastrados os seguintes dados: código do sinônimo, descrição da palavra e descrição do conceito.

Pré-condições: Usuário deve fazer *login* no sistema.

Pós-condições: Usuário alterou, excluiu ou incluiu um sinônimo.

Cenário Principal:

1. Sistema informa os sinônimos cadastrados;
2. Usuário opta por alterar, excluir ou cadastrar um sinônimo;
3. Sistema verifica o preenchimento dos campos obrigatórios na inclusão ou alteração.

Cenário Alternativo:

Se no item 3 for verificada alguma inconsistência no preenchimento dos campos:

- 3.1 É enviada mensagem informativa para a tela;

Cenário Alteração:

1. Sistema mostra os sinônimos cadastrados;
2. Usuário seleciona um sinônimo para alteração;
3. Sistema mostra os dados do sinônimo para alteração;
4. Usuário realiza as alterações necessárias;
5. Usuário seleciona a opção alterar;
6. Sistema altera o registro e apresenta a mensagem “Registro Alterado com Sucesso”.

Cenário Inclusão:

1. Sistema mostra a tela de cadastro de sinônimos;
2. Usuário informa todos os campos obrigatórios;
3. Usuário seleciona a opção incluir;
4. Sistema inclui o registro e exibe a mensagem “Registro incluído com sucesso!”.

Cenário Exclusão:

1. Sistema mostra os sinônimos cadastrados;
2. Usuário seleciona um sinônimo para exclusão;
3. Usuário seleciona a opção excluir;
4. Sistema exclui o registro e exibe a mensagem “Registro excluído com sucesso”.

Quadro 12 - Descrição do Caso de Uso “Manter Sinônimos”

O Quadro 13 apresenta o detalhamento do Caso de Uso “Manter *Stopwords*”.

UC08 - Manter *Stopwords*

Ator: Usuário

Objetivo: O Usuário acessa a tela de cadastro de *stopwords* para informar os dados do mesmo. Serão cadastrados os seguintes dados: código da *stopword*, descrição da *stopword*.

Pré-condições: Usuário deve fazer *login* no sistema.

Pós-condições: Usuário alterou, excluiu ou incluiu uma *stopword*.

Cenário Principal:

1. Sistema informa as *stopwords* cadastradas;
2. Usuário opta por alterar, excluir ou cadastrar uma *stopword*;
3. Sistema verifica o preenchimento dos campos obrigatórios na inclusão ou alteração.

Cenário Alternativo:

Se no item 3 for verificada alguma inconsistência no preenchimento dos campos:

- 3.1 É enviada mensagem informativa para a tela;

Cenário Alteração:

1. Sistema mostra as *stopwords* cadastradas;
2. Usuário seleciona uma *stopword* para alteração;
3. Sistema mostra os dados da *stopword* para alteração;
4. Usuário realiza as alterações necessárias;
5. Usuário seleciona a opção alterar;
6. Sistema altera o registro e apresenta a mensagem “Alteração Efetuada com Sucesso”.

Cenário Inclusão:

1. Sistema mostra a tela de cadastro de *stopwords*;
2. Usuário informa todos os campos obrigatórios;
3. Usuário seleciona a opção incluir;
4. Sistema inclui o registro e exibe a mensagem “Registro incluído com sucesso!”.

Cenário Exclusão:

1. Sistema mostra as *stopwords* cadastradas;
2. Usuário seleciona uma *stopword* para exclusão;
3. Usuário seleciona a opção excluir;
4. Sistema exclui o registro e exibe a mensagem “Registro excluído com sucesso”.

Quadro 13 - Descrição do Caso de Uso “Manter *Stopwords*”

O Quadro 14 apresenta o detalhamento do Caso de Uso “Manter Radicais”.

UC09 - Manter Radicais

Ator: Usuário

Objetivo: O Usuário acessa a tela de cadastro de radicais para informar os dados do mesmo. Serão cadastrados os seguintes dados: código do radical, descrição da palavra e descrição do radical.

Pré-condições: Usuário deve fazer *login* no sistema.

Pós-condições: Usuário alterou, excluiu ou incluiu um radical.

Cenário Principal:

1. Sistema informa os radicais cadastrados;
2. Usuário opta por alterar, excluir ou cadastrar um radical;
3. Sistema verifica o preenchimento dos campos obrigatórios na inclusão ou alteração.

Cenário Alternativo:

Se no item 3 for verificada alguma inconsistência no preenchimento dos campos:

- 3.1 É enviada mensagem informativa para a tela;

Cenário Alteração:

1. Sistema mostra os radicais cadastrados;
2. Usuário seleciona um radical para alteração;
3. Sistema mostra os dados do radical para alteração;
4. Usuário realiza as alterações necessárias;
5. Usuário seleciona a opção alterar;
6. Sistema altera o registro e apresenta a mensagem “Alteração Efetuada com Sucesso”.

Cenário Inclusão:

1. Sistema mostra a tela de cadastro de radicais;
2. Usuário informa todos os campos obrigatórios;
3. Usuário seleciona a opção incluir;
4. Sistema inclui o registro e exibe a mensagem “Registro incluído com sucesso!”.

Cenário Exclusão:

1. Sistema mostra os radicais cadastrados;
2. Usuário seleciona um radical para exclusão;
3. Usuário seleciona a opção excluir;
4. Sistema exclui o registro e exibe a mensagem “Registro excluído com sucesso”.

Quadro 14 - Descrição do Caso de Uso “Manter Radicais”

O Quadro 15 apresenta o detalhamento do Caso de Uso “Manter Áreas”.

UC10 - Manter Áreas

Ator: Usuário

Objetivo: O Usuário acessa a tela de cadastro de áreas para informar os dados da mesma. Serão cadastrados os seguintes dados: código da área, descrição da área.

Pré-condições: Usuário deve fazer *login* no sistema.

Pós-condições: Usuário alterou, excluiu ou incluiu uma área.

Cenário Principal:

1. Sistema informa as áreas cadastradas;
2. Usuário opta por alterar, excluir ou cadastrar uma área;
3. Sistema verifica o preenchimento dos campos obrigatórios na inclusão ou alteração.

Cenário Alternativo:

Se no item 3 for verificada alguma inconsistência no preenchimento dos campos:

- 3.1 É enviada mensagem informativa para a tela;

Cenário Alteração:

1. Sistema mostra as áreas cadastradas;
2. Usuário seleciona uma área para alteração;
3. Sistema mostra os dados da área para alteração;
4. Usuário realiza as alterações necessárias;
5. Usuário seleciona a opção alterar;
6. Sistema altera o registro e apresenta a mensagem “Alteração Efetuada com Sucesso”.

Cenário Inclusão:

1. Sistema mostra a tela de cadastro de áreas;
2. Usuário informa todos os campos obrigatórios;
3. Usuário seleciona a opção incluir;
4. Sistema inclui o registro e exibe a mensagem “Registro incluído com sucesso!”.

Cenário Exclusão:

1. Sistema mostra as áreas cadastradas;
2. Usuário seleciona uma área para exclusão;
3. Usuário seleciona a opção excluir;
4. Sistema exclui o registro e exibe a mensagem “Registro excluído com sucesso”.

Quadro 15 - Descrição do Caso de Uso “Manter Áreas”

APÊNDICE B – Detalhamento do dicionário de dados

Este Apêndice apresenta o dicionário de dados das tabelas do aplicativo, e tem o objetivo de descrever as tabelas e seus respectivos campos. Os campos do Tipo "INTEGER" representam valores numéricos. O tipo "VARCHAR" representa uma sequência de letras ou palavras e o tipo "DATETIME" armazena exclusivamente data e hora.

O Quadro 16 apresenta o dicionário de dados da tabela "Manual".

| Manual - Armazena dados dos manuais e procedimentos. | | | | |
|---|-------------|----------------|--------------------|----------------------------------|
| Campo | Tipo | Tamanho | Obrigatório | Descrição |
| cd_manual | INTEGER | - | SIM | Chave primária. Código do manual |
| ds_nome_Manual | VARCHAR | 255 | SIM | Descrição do nome do manual |
| ds_manual | VARCHAR | 255 | SIM | Descrição do objetivo do manual |
| dt_postagem | DATETIME | - | SIM | Data de postagem do manual |

Quadro 16 - Dicionário de dados da tabela Manual

O Quadro 17 apresenta o dicionário de dados da tabela "Modulo".

| Modulo - Armazena dados dos módulos. | | | | |
|---|-------------|----------------|--------------------|-------------------------------------|
| Campo | Tipo | Tamanho | Obrigatório | Descrição |
| cd_modulo | INTEGER | - | SIM | Chave primária. Código do módulo |
| Manual_cd_manual | INTEGER | - | SIM | Chave estrangeira. Código do manual |
| ds_nome | VARCHAR | 30 | NÃO | Nome do módulo |
| dt_modulo | VARCHAR | 255 | NÃO | Descrição da finalidade do módulo |

Quadro 17 - Dicionário de dados da tabela Modulo

O Quadro 18 apresenta o dicionário de dados da tabela "Programa_Modulo".

| Programa_Modulo - Armazena dados de módulos que existem nos programas. | | | | |
|---|-------------|----------------|--------------------|------------------------------------|
| Campo | Tipo | Tamanho | Obrigatório | Descrição |
| Programa_cd_programa | INTEGER | - | SIM | Chave primária. Código do programa |
| Modulo_cd_modulo | INTEGER | - | SIM | Chave primária. Código do módulo |

Quadro 18 - Dicionário de dados da tabela Programa_Modulo

O Quadro 19 apresenta o dicionário de dados da tabela "Programa".

| Programa - Armazena dados dos programas. | | | | |
|---|-------------|----------------|--------------------|--|
| Campo | Tipo | Tamanho | Obrigatório | Descrição |
| cd_programa | INTEGER | - | SIM | Chave primária. Código do programa |
| Area_cd_area | INTEGER | - | SIM | Chave estrangeira. Código da área |
| nm_programa | VARCHAR | 55 | SIM | Nome do programa |
| ds_programa | VARCHAR | 255 | SIM | Descrição do programa |
| chave_mainframe | VARCHAR | 10 | SIM | Biblioteca onde o programa se encontra no <i>Mainframe</i> |
| dt_inclusao | DATETIME | - | SIM | Data de inclusão do registro |
| tabelas_acessadas | VARCHAR | MAX | NÃO | Tabelas lidas no programa |
| codigo_fonte | VARCHAR | MAX | NÃO | Código fonte do programa |

Quadro 19 - Dicionário de dados da tabela Programa

O Quadro 20 apresenta o dicionário de dados da tabela "Usuario".

| Usuario - Armazena dados dos usuários do aplicativo. | | | | |
|---|-------------|----------------|--------------------|-----------------------------------|
| Campo | Tipo | Tamanho | Obrigatório | Descrição |
| cd_usuario | INTEGER | - | SIM | Chave primária. Código do usuário |
| login | VARCHAR | 45 | SIM | Login do usuário no aplicativo |
| senha | VARCHAR | 45 | SIM | Senha do usuário no aplicativo |
| ds_nome | VARCHAR | 255 | SIM | Nome do usuário |
| tipo | VARCHAR | 10 | SIM | Tipo de usuário |

Quadro 20 - Dicionário de dados da tabela Usuario

O Quadro 21 apresenta o dicionário de dados da tabela "Area".

| Area - Armazena dados das áreas da empresa. | | | | |
|--|-------------|----------------|--------------------|--------------------------------|
| Campo | Tipo | Tamanho | Obrigatório | Descrição |
| cd_area | INTEGER | - | SIM | Chave primária. Código da área |
| ds_area | VARCHAR | 255 | SIM | Descrição da área |

Quadro 21 - Dicionário de dados da tabela Area

O Quadro 22 apresenta o dicionário de dados da tabela "Radical".

| Radical - Armazena dados dos Radicais das palavras. | | | | |
|--|-------------|----------------|--------------------|-----------------------------------|
| Campo | Tipo | Tamanho | Obrigatório | Descrição |
| cd_radical | INTEGER | - | SIM | Chave primária. Código do radical |
| ds_palavra | VARCHAR | 50 | SIM | Descrição da palavra |
| ds_radical | VARCHAR | 50 | SIM | Descrição do radical da palavra |

Quadro 22 - Dicionário de dados da tabela Radical

O Quadro 23 apresenta o dicionário de dados da tabela "Stopword".

| Stopword - Armazena dados das <i>Stopwords</i> (palavras sem importância na busca). | | | | |
|--|-------------|----------------|--------------------|---|
| Campo | Tipo | Tamanho | Obrigatório | Descrição |
| cd_stopword | INTEGER | - | SIM | Chave primária. Código da <i>stopword</i> |
| ds_stopword | VARCHAR | 30 | SIM | Descrição da <i>stopword</i> |

Quadro 23 - Dicionário de dados da tabela Stopword

O Quadro 24 apresenta o dicionário de dados da tabela "Sinonimo".

| Sinonimo - Armazena dados dos Sinônimos das palavras. | | | | |
|--|-------------|----------------|--------------------|------------------------------------|
| Campo | Tipo | Tamanho | Obrigatório | Descrição |
| cd_sinonimo | INTEGER | - | SIM | Chave primária. Código do sinônimo |
| ds_palavra | VARCHAR | 200 | SIM | Descrição da palavra |
| ds_conceito | VARCHAR | 200 | SIM | Descrição do sinônimo da palavra |

Quadro 24 - Dicionário de dados da tabela Radical

ANEXO A - Formulário de avaliação do trabalho Desenvolvido

Conforme descrito na seção 3.4 foi elaborado um formulário para avaliar o trabalho desenvolvido. A Figura 34 exibe o formulário distribuído.

Aplicativo Web para auxiliar na documentação e procura de programas similares sob a plataforma Mainframe

Formulário para análise dos resultados obtidos com a elaboração do projeto para documentação e procura de fontes similares.
* Required

Qual o seu envolvimento em relação a este trabalho? *

- Membro da equipe, programador
- Coordenador
- Professor/Orientador
- Cliente
- Não tenho relação com o trabalho mas sei do que se trata

Quanto tempo está atuando na área de Tecnologia da Informação?

- Menos de 1 ano
- 1 a 5 anos
- 6 a 10 anos
- Mais de 10 anos
- Não atuo diretamente, sou apenas usuário

Na Sua opinião foram obtidos benefícios com o aplicativo para documentar e procurar fontes similares no mainframe? *

- Sim
- Não
- Não entendi o objetivo

Escreva em poucas palavras como você pode contribuir com o sucesso da implantação e continuidade da utilização deste projeto? *

Figura 34 - Formulário de avaliação do trabalho desenvolvido