

UNIVERSIDADE FEDERAL DE SANTA CATARINA

**Desenvolvimento de um Dicionário Eletrônico de
Apoio à Produção de Textos em Língua
Estrangeira**

Herbert Mattei de Borba
Tiago Müller Gil Cardoso

UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CURSO DE CIÊNCIAS DA COMPUTAÇÃO

Desenvolvimento de um Dicionário Eletrônico de Apoio a Produção de Textos em Língua Estrangeira

Autores:

Herbert Mattei de Borba
Tiago Müller Gil Cardoso

Orientador:

Philippe Humblé

Co-Orientador:

José Eduardo De Lucca

Banca Examinadora:

Walter Carlos Costa

Palavras-Chave:

Dicionário de Produção, Philippe Humblé, XML, XQuery, XQEngine, Java, JAXB

Florianópolis, 20 de Fevereiro de 2003

SUMÁRIO

1. INTRODUÇÃO	1
1.1. APRESENTAÇÃO	1
1.2. OBJETIVOS.....	1
1.2.1. <i>Objetivo Geral</i>	1
1.2.2. <i>Objetivos Específicos</i>	1
1.3. METODOLOGIA.....	2
2. UM DICIONÁRIO ELETRÔNICO DE PRODUÇÃO DE TEXTOS EM LÍNGUA ESTRANGEIRA.....	3
2.1. PARA QUE UM DICIONÁRIO DE PRODUÇÃO DE TEXTOS?.....	3
2.2. O QUE É O DICIONÁRIO DE PRODUÇÃO DE TEXTOS EM LÍNGUA ESTRANGEIRA?.....	4
2.3. A ORGANIZAÇÃO DO DICIONÁRIO ELETRÔNICO DE PRODUÇÃO	4
3. CONCEBENDO O DICIONÁRIO ELETRÔNICO DE PRODUÇÃO DE TEXTOS EM LÍNGUA ESTRANGEIRA.....	6
3.1. O ESTADO DA ARTE	6
3.2. DEFININDO OS REQUISITOS DO SISTEMA.....	7
3.2.1. <i>Requisitos Funcionais</i>	7
3.2.2. <i>Atributos do sistema</i>	7
3.3. ESCOLHENDO AS TECNOLOGIAS DE SUPORTE AO DICIONÁRIO.....	8
4. IMPLEMENTANDO O DICIONÁRIO ELETRÔNICO DE PRODUÇÃO.....	9
4.1. MODELANDO O SISTEMA	9
4.2. DETALHES DA IMPLEMENTAÇÃO	10
5. CONCLUSÃO	14
6. REFERÊNCIAS BIBLIOGRÁFICAS.....	15
ANEXO I - ARTIGO RELACIONADO AO PROJETO.....	17

ANEXO II - O DICIONÁRIO BILÍNGUE PORTUGUÊS-ESPAHOL DA UNIVERSIDADE FEDERAL DE SANTA CATARINA	23
ANEXO III - CÓDIGO-FONTE DESENVOLVIDO NO PROJETO	32
XML SCHEMAS E DTDS	33
<i>DiccionarioProducaoSchema.xsd</i>	33
<i>DiccionarioResultadoLemaXSD.xsd</i>	34
<i>DicProdResultadoExemploXSD.xsd</i>	35
<i>DicProdResultadoExemploXSL.xsl</i>	36
<i>DicProdResultadoLemaXSL.xsl</i>	36
<i>DicProdResultadoExemploDTD.dtd</i>	37
<i>DicProdResultadoLemaDTD.dtd</i>	38
<i>DicProdResultados.css</i>	38
PACKAGE DICPROD	39
<i>DiccionarioXMLPesquisa.java</i>	39
<i>diccionarioXMLResultadosGerente.java</i>	44
<i>diccionarioXMLGerente.java</i>	48
<i>diccionarioConfig.java</i>	49
JSP	51
<i>index.jsp</i>	51
<i>DicProd.css</i>	52
<i>DicProdResultados.jsp</i>	53
JAVABEANS	53
<i>Config.java</i>	53
<i>Pesquisa.java</i>	54
IMPLEMENTAÇÃO EM JAVA SWING	56
<i>DicProdGUI.java</i>	56
<i>DicProdGUIConfig.java</i>	61

RESUMO

Este projeto apresenta a definição e implementação de um dicionário eletrônico de apoio à produção de textos em língua estrangeira, concebido inicialmente em um projeto do Departamento de Línguas e Literatura Estrangeiras da UFSC. Este documento demonstra o desenvolvimento de um esquema de armazenamento, pesquisa e gerenciamento concebido com base nas tecnologias Java e XML. Foram criadas duas versões do sistema: a primeira pode ser instalada a partir de um CD-ROM (Java Swing) e a segunda possibilita o acesso remoto, através de um servidor web (JSP, JavaBeans).

ABSTRACT

This project presents the definition and implementation of a foreign language text production electronic dictionary, early created in a project of the Foreign Language and Literature Department from UFSC. This document illustrates the development of a scheme for storage, searching and management based on Java and XML technologies. There were create two versions for the system: the first one may be installed from a CD-ROM (Java Swing) and the second allows the remote access through a web server (JSP, JavaBeans).

1. INTRODUÇÃO

1.1. APRESENTAÇÃO

O desenvolvimento do dicionário eletrônico de apoio à produção de textos em língua estrangeira, é apresentado como projeto de conclusão no curso de Ciências da Computação da Universidade Federal de Santa Catarina. A oportunidade para este trabalho surgiu da interação entre professores do Departamento de Línguas e Literatura Estrangeiras (LLE) e do Departamento de Informática e Estatística (INE) da UFSC, dando continuidade ao projeto do *Dicionário Bilingüe Português-Espanhol da Universidade Federal de Santa Catarina*, coordenado pelo Prof. Philippe Humblé.

1.2. OBJETIVOS

1.2.1. Objetivo Geral

O objetivo deste trabalho é definir os requisitos, especificar, implementar e implantar um sistema de apoio à produção de textos em língua estrangeira. A consulta ao dicionário pode ser realizada remotamente através de navegadores web ou localmente através de uma distribuição em CD-ROM. O dicionário deverá permitir a adição de novos idiomas distribuídos separadamente.

1.2.2. Objetivos Específicos

- Definir uma representação para as informações lingüísticas utilizando a tecnologia XML, resguardando as características de associabilidade (os exemplos selecionados do corpus da linguagem fonte devem estar associados aos respectivos exemplos traduzidos para a linguagem destino) e extensibilidade (novos idiomas, novos lemas);

- Implementar um esquema de armazenamento das informações lingüísticas de forma a permitir pesquisas eficientes;
- Implementar as interfaces de acesso às informações como um servidor web e como um aplicativo local;

1.3. METODOLOGIA

- Definição de requisitos do sistema;
- Definição de um modelo conceitual e arquitetural do sistema;
- Definição das interfaces de interação com o usuário;
- Definição das tecnologias que serão utilizadas para implementar o sistema (linguagens, ferramentas, servidor web, SGBD...);
- Definição de um formato XML para o dicionário;
- Pesquisa de maneiras de preparar documentos XML para permitir pesquisas eficientes;
- Conversão dos dados do dicionário para o formato XML;
- Implementação de um mecanismo de busca eficiente para o dicionário;
- Implementação das interfaces com o usuário e acoplamento ao mecanismo de busca;

2. UM DICIONÁRIO ELETRÔNICO DE PRODUÇÃO DE TEXTOS EM LÍNGUA ESTRANGEIRA

2.1. PARA QUE UM DICIONÁRIO DE PRODUÇÃO DE TEXTOS?

Em *O Dicionário Bilingüe Português-Espanhol da Universidade Federal de Santa Catarina* (HUMBLÉ, Philippe. ANEXO II), encontramos argumentos de que a produção de textos em língua estrangeira torna-se complicada quando o autor não dispõe de um dicionário que contemple informações além da mera tradução de vocábulos.

Humblé afirma também que, utilizando um dicionário bilíngüe tradicional, pode-se encontrar uma palavra em um idioma estrangeiro com o mesmo significado de uma palavra pertencente à língua nativa do autor, mas isto tende a deixar o texto com anomalias como sentenças mal formadas ou até mesmo erradas. É necessário então encontrar, dentre várias alternativas, aquela que melhor se adapte ao significado pretendido, além de conhecer sua correta utilização ao nível sintático. O autor também precisa levar em consideração as possíveis variações na tradução das sentenças, capazes de tornar a frase pouco harmoniosa, ou mesmo sem sentido, quando lida por uma pessoa nativa no idioma.

Portanto, os dicionários bilíngües tradicionais são adequados para conhecer o significado de um termo escrito em língua estrangeira, porém, estes mesmos dicionários mostram-se ineficazes no momento de produzir textos. Foi pensando neste problema que os estudiosos da lingüística propuseram este tipo de dicionário, concebido especialmente para o auxílio à produção de textos em língua estrangeira.

2.2. O QUE É O DICIONÁRIO DE PRODUÇÃO DE TEXTOS EM LÍNGUA ESTRANGEIRA?

No dicionário de produção definido pelo Prof. Humblé não ocorre a tradicional correspondência entre palavras e seus respectivos significados como em dicionários convencionais, nem tampouco palavras e suas possíveis traduções como seria esperado encontrar em dicionários bilíngües.

Em um dicionário de produção encontram-se frases prontas, ou seja, exemplos de utilização de vocábulos em diferentes contextos, selecionados criteriosamente por estudiosos do idioma para ilustrar o uso natural dos vocábulos. O usuário pode visualizar exemplos de utilização de uma palavra ao localizar as diversas ocorrências desta palavra no dicionário.

2.3. A ORGANIZAÇÃO DO DICIONÁRIO ELETRÔNICO DE PRODUÇÃO

O dicionário eletrônico de produção está organizado hierarquicamente em volumes, lemas e exemplos. Cada volume contém lemas referentes a determinado idioma-fonte (normalmente o idioma nativo do usuário) e idioma-destino. Cada lema contém uma coleção de exemplos que ilustram o uso de determinado vocábulo. Este vocábulo é considerado o núcleo do lema.

Um exemplo é obrigatoriamente seguido de uma tradução no idioma-destino, mas a tradução do núcleo é opcional. A figura a seguir ilustra a estrutura hierárquica do dicionário.

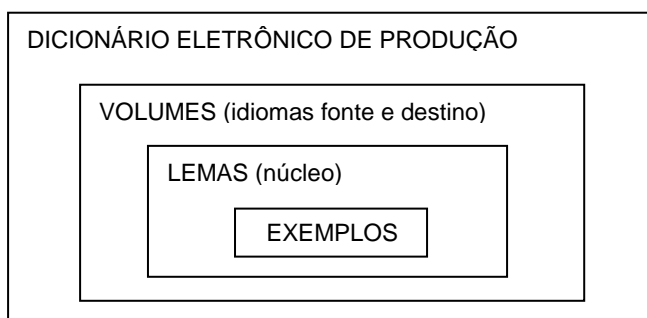


Figura 1 - Organização do dicionário de produção multilíngue

3. CONCEBENDO O DICIONÁRIO ELETRÔNICO DE PRODUÇÃO DE TEXTOS EM LÍNGUA ESTRANGEIRA

3.1. O ESTADO DA ARTE

Atualmente, não se encontra nenhum dicionário de produção de textos em língua estrangeira que possa ser destacado como exemplo de estado da arte. Todavia pode-se encontrar dicionários de produção de textos especializados, voltados a uma área específica do conhecimento (cinema, teatro, engenharia, economia, etc) ou dicionários de produção monolíngües [LON 2002]. A maior parte destes dicionários encontram-se disponíveis somente impressos em papel.

Implementações comerciais de dicionários eletrônicos baseados em tecnologia proprietária estão disponíveis, mas em consequência das restrições de acesso ao código-fonte desses produtos, nada pode ser reaproveitado além de idéias sobre a organização da interface com o usuário ou funcionalidades.

Por outro lado, pessoas vêm trabalhando, através da Internet, na definição de uma tecnologia padrão de suporte a dicionários eletrônicos multilíngües. Um exemplo desse esforço é o caso do projeto Papillon [PAP 2000], que infelizmente, é encontrado apenas em caráter experimental e não possui documentação ou ferramentas suficientes para uma implementação de baixo custo.

Um protótipo do dicionário de produção multilíngüe foi construído por Humblé com o auxílio da ferramenta *AskSam* (ver ANEXO II). Este protótipo oferece uma visão das funcionalidades básicas do sistema. No entanto esta implementação depende de um software proprietário, que não pode ser redistribuído, e apresenta uma

interface com o usuário demasiadamente complexa por tratar-se de uma ferramenta de indexação e pesquisa em textos de propósito geral.

3.2. DEFININDO OS REQUISITOS DO SISTEMA

Durante as reuniões iniciais com os orientadores definimos o escopo do dicionário. A tarefa era definir um modelo e criar um mecanismo de pesquisa para dicionários de produção de textos. O software deveria ser desenvolvido de forma que pudesse ser distribuída para instalação stand-alone (chamada de Versão CD pois essa seria a mídia para distribuição) e que também pudesse ser instalada em um servidor web para ser acessada remotamente, através de um navegador web (Versão Web).

3.2.1. Requisitos Funcionais

R1. Consultas aos lemas

R1.1. Consultas pelos núcleos dos lemas;

R1.1.1. Apresentação da listagem do lema junto com seus respectivos exemplos;

R1.2. Consultas por termos nos exemplos dos idiomas fonte e destino;

R1.2.1. Apresentação da listagem dos lemas junto com seus respectivos exemplos;

R1.2.2. Apresentação apenas dos exemplos que contenham o(s) termo(s) procurado(s);

R1.3. Apresentação de opções para habilitação dos requisitos R1.1.1, R1.2.1 e R1.2.2;

3.2.2. Atributos do sistema

A1. Deve possuir uma interface com o usuário adequada para utilização de todas as funcionalidades atribuídas ao sistema e descritas neste documento;

A2. Deve realizar as consultas eficientemente, valendo-se do esquema de indexação de documentos;

A3. Deve permitir a extensão do sistema para a inclusão de novos volumes;

A4. As tecnologias utilizadas para a implementação não podem possuir restrições quanto à livre distribuição do sistema e do código-fonte.

3.3. ESCOLHENDO AS TECNOLOGIAS DE SUPORTE AO DICIONÁRIO

Juntamente com o Prof. De Lucca, decidimos que a linguagem de programação escolhida para a implementação do sistema seria Java, porque além de suas características já bem divulgadas e a tendência de transformar-se em padrão de mercado, possibilitou a construção de um único núcleo que pôde ser utilizado tanto para a Versão CD como para a Versão Web.

Foi definido que o dicionário estaria em formato XML [W3C 2000a], padrão que vem sendo adotado como solução definitiva para o armazenamento e intercâmbio de informações entre plataformas, sistemas operacionais, protocolos e aplicações.

4. IMPLEMENTANDO O DICIONÁRIO ELETRÔNICO DE PRODUÇÃO

4.1. MODELANDO O SISTEMA

O protótipo do dicionário Português-Espanhol produzido por Philippe Humblé serviu de modelo para a implementação. Abaixo apresentamos um trecho deste dicionário, para posterior detalhamento.

```

@@
lp[ abacate
th[ aguacate, palta.

ep[ Eu adoro abacate amassadinho com açúcar e limão.
ee[ Me encanta el aguacate / la palta machacada con azúcar y limón.

ep[ O abacate é uma fruta rica em óleo e vitamina A.
ee[ El aguacate / La palta es una fruta rica en grasas y vitamina A.

```

Listagem 1 – Exemplo de um lema definido pelo dicionário original

Os símbolos “@@” indicam o início de um **lema**, que é o conjunto de informações composto por um **núcleo**, sua respectiva **tradução**, e um ou vários **exemplos** com variantes de utilização em frases do termo do núcleo nas duas linguagens.

O núcleo é definido na linha seguinte ao “@@”, indicado pelo símbolo “lp[” seguido pelo termo do núcleo propriamente dito (no caso, “abacate”).

Logo após, “th[” indica o início da tradução, ou traduções possíveis para o núcleo. Um exemplo no idioma fonte está indicado por “ep[” (exemplo em Português), sempre seguido por sua tradução indicada por “ee[” (exemplo em

Espanhol). A listagem de exemplos termina onde começa o próximo “@@”, ou seja, o próximo lema.

4.2. DETALHES DA IMPLEMENTAÇÃO

Após a definição, pesquisa e aprendizado das tecnologias que seriam utilizadas, foi possível iniciar a implementação efetiva do dicionário. O primeiro passo dado foi o processo de definição de uma nova estrutura, em XML, que comportasse os dados do dicionário original (Listagem 1). Para tanto foi criado um **Schema XML** [W3C 2000b] (ver Listagem 2) que nada mais é que um conjunto de regras que deve ser seguido pelos documentos XML, ou seja, por qualquer dicionário que irá rodar no sistema. Segundo esse Schema XML, o exemplo da Listagem é representado da seguinte forma:

```
<lema>
  <nucleo>abacate</nucleo>
  <traducao>aguacate, palta.</traducao>
  <exemplo>
    <fonte>Eu adoro abacate amassadinho com açúcar e limão.</fonte>
    <destino>Me encanta el aguacate / la palta machacada con azúcar y
limón.</destino>
  </exemplo>
  <exemplo>
    <fonte>O abacate é uma fruta rica em óleo e vitamina A.</fonte>
    <destino>El aguacate / La palta es una fruta rica en grasas y vitamina
A.</destino>
</lema>
```

Listagem 2 - O lema com núcleo "abacate", em XML

O Schema também define regras para a especificação de um nome de referência para o dicionário, além dos códigos do padrão RFC 1766 que identificam os idiomas fonte e destino que estão sendo utilizados, em um elemento raiz chamado volume. Um exemplo para o dicionário Português-Espanhol é apresentado na Listagem 3, que mostra também como o Schema XML é invocado:

```
<volume xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="file:DicionarioProducaoSchema.xsd"
nome="Dicionário TCC" idioma-fonte="pt-BR" idioma-destino="es">
```

Listagem 3 – Exemplo do elemento raiz <volume>, que “envolve” todos os elementos <lema> do dicionário

Após a definição de como os dados seriam armazenados, iniciou-se a produção das classes Java do núcleo do dicionário, o pacote `dicprod`. Para a implementação da classe de pesquisa (ver Listagem 4), foi utilizada uma biblioteca para busca em documentos XML chamada **XQEngine** [XQE 2003], distribuído livremente e de código fonte aberto (existem mecanismos semelhantes e mais desenvolvidos, porém de código fechado e disponíveis apenas comercialmente). Antes de poder realizar a primeira pesquisa, a XQEngine precisa indexar o documento XML. A indexação serve para aumentar a performance das pesquisas, e é necessária apenas uma vez a cada instalação do software, pois os índices gerados dependem do sistema de arquivo que está sendo utilizado (Window, UNIX, etc.) e são salvos em disco para posterior utilização.

A XQEngine utiliza a **XQuery** [W3C 2000c] , linguagem proposta e desenvolvida (ainda em desenvolvimento) pelo World Wide Web Consortium (<http://www.w3.org>) para realizar consultas em estruturas XML. O trecho do código que realiza buscas pelos exemplos do idioma fonte é apresentado na Listagem 4, para ilustração:


```

/**
 * Faz uma busca genérica
 */
private void busca( String xQuery ) {
    _engine.setResultListener( this );
    try {
        _engine.setQuery( xQuery );
    }
    catch( InvalidQueryException e ) {
        System.out.println( "Pesquisa invalida: " + e.getMessage() );
        return;
    }
    catch ( com.fatdog.xmlEngine.exceptions.TypeException e ) {
        System.out.println( e.toString( ) );
    }
}

/**
 * Busca pelo conteúdo da tag <fonte>, trazendo apenas os exemplos que
 * encontrar
 */
public void busqueFonteExemplos( String expr ) {
    this._exemplos = true;
    this._busca = expr;
    this.busca( "//lema/exemplo[fonte &= '"+ expr + "'" ] );
}

/**
 * Result listener da XQEngine
 * Os resultados da busca são mandados para este método
 */
public void results( String results ) {
    _resultados = results;
    _resultadosXML = montaResultadosXML( );
}

```

Listagem 4 – Métodos da classe `dicprod.dicionarioXMLPesquisa` que invoca a busca por exemplos do idioma fonte, mostrando como a XQuery é utilizada

Quando o usuário clica o botão Procurar em uma das interfaces (CD e Web), o termo digitado na caixa de pesquisa é passado como parâmetro para o método `busqueFonteExemplos`, que o utiliza para montar a expressão XQuery passada ao método `busca`. O método de busca invoca o objeto da classe `XQEngine`, que retornará os resultados utilizando o mecanismo de *result listeners*, enviando-os para o método `results`, que chama `montaResultadosXML` para organizar a resposta que será capturada pela interface.

Os resultados também são estruturas XML, definidas igualmente segundo um Schema XML. O exemplo dos resultados para uma busca por `ObjClassePesq.busqueFonteExemplos("limão com açúcar")`, que geraria a Xquery `“//lema/exemplo[fonte &= 'limão com açúcar']”`, é mostrado na Listagem 5.

```
<?xml version="1.0" encoding="ISO-8859-15"?>
<!-- Gerado pelo motor de busca do dicprod (dicionarioXMLPesquisa.java) -->
<!-- Thu Feb 06 13:01:10 BRST 2003 -->
<?xml-stylesheet type="text/xsl" href="file:xml\DicProdResultadoExemploXSL.xsl" ?>
<exemplos
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="file:xml\DicProdResultadoExemploXSD.xsd"
  pesquisa="açúcar
  com limão" idioma-fonte="pt-BR" idioma-destino="es">
<exemplo>
  <fonte>Eu adoro abacate amassadinho com açúcar e limão.</fonte>
  <destino>Me encanta el aguacate / la palta machacada con azúcar y limón. </destino>
</exemplo>
</exemplos>
```

Listagem 5 – Exemplo de como é gerada a estrutura resultado para a busca “Açúcar com limão”

O gerente de resultados utiliza as classes compiladas pelo JAXB [SUN 2003a, JXB 2003] a partir de um Schema XML específico para os resultados (ver Código Fonte – `DicProdResultadoExemploXSD.xsd`), para convertê-los para o formato HTML, reconhecendo os termos procurados através de um mecanismo de expressões regulares para enfatizá-los no código HTML gerado para resposta. A aplicação em questão então mostra o código HTML, gerando uma saída como a mostrada na Figura 2.

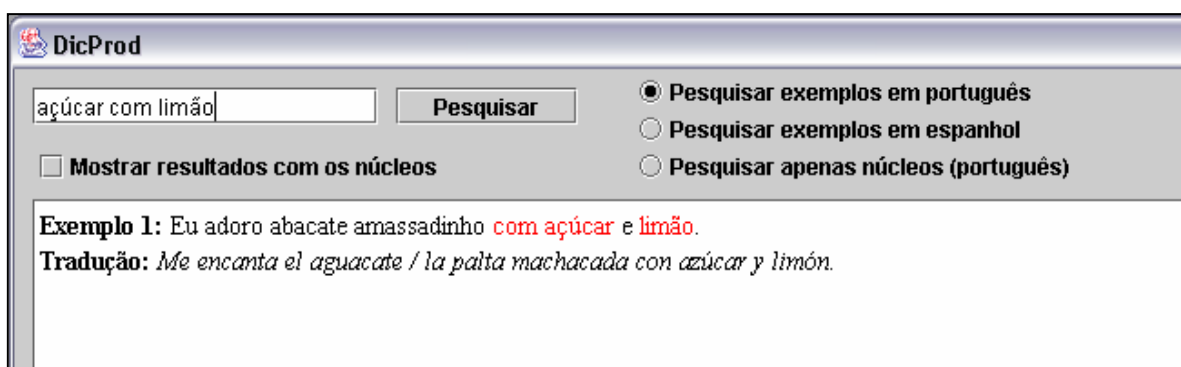


Figura 2 – Saída para a interface da Versão CD (Java Swing), que utiliza um container chamado `JEditorPane` para mostrar o código HTML gerado com o auxílio da JAXB para o resultado

5. CONCLUSÃO

A tecnologia XML possui uma arquitetura que possibilita a rápida implementação de diferentes tipos de sistemas de informação. Apesar da especificação da linguagem de pesquisa XQuery encontrar-se em fase de desenvolvimento, foi possível desenvolver um protótipo funcional para o sistema proposto utilizando poucos recursos.

A ferramenta XQEngine fornece uma solução de código aberto para o problema da indexação e pesquisa em textos estruturados em documentos XML. No entanto, este mecanismo de busca necessita de mais tempo até que todas as funcionalidades necessárias estejam realmente operacionais, como a possibilidade de utilizar caracteres coringas (expressões regulares) para aumentar a capacidade das pesquisas.

Para uma possível versão comercial do dicionário é essencial resolver ainda questões como:

- permitir a utilização de caracteres coringas nas pesquisas;
- limitar o número de exemplos retornados nas pesquisas
- criar uma interface amigável de busca avançada
- converter todo o conteúdo do dicionário para o formato XML

No entanto, a falta de um produto similar a este dicionário assinala como o segmento ainda é pouco explorado, ao mesmo tempo em que destaca a importância deste trabalho como referencial para futuros sistemas semelhantes.

6. REFERÊNCIAS BIBLIOGRÁFICAS

[W3C 2000a] W3C. *Extensible Markup Language (XML) 1.0*. 2. ed. 2000. Disponível eletronicamente em < <http://www.w3.org/TR/2000/REC-xml-20001006> >, acessado em 27/08/2002.

[SUN 2002a]. Sun Microsystems, Inc. *java.sun.com: The Source for Java Technology*. 2002. Disponível eletronicamente em < <http://java.sun.com> >, acessado em 27/08/2002.

[SUN 2002b]. Sun Microsystems, Inc. *JFC/Swing GUI Components*. 2002. Disponível eletronicamente em < <http://java.sun.com/products/jfc/#components> >, acessado em 28/08/2002.

[SUN 2002c]. Sun Microsystems, Inc. *THE AWT IN 1.0 AND 1.1*. 2002. Disponível eletronicamente em < <http://java.sun.com/products/jdk/awt/> >, acessado em 28/08/2002.

[SUN 2002d]. Sun Microsystems, Inc. *Java Servlet Technology*. 2002. Disponível eletronicamente em < <http://java.sun.com/products/servlet/> >, acessado em 28/08/2002.

[SUN 2002e]. Sun Microsystems, Inc. *JavaBeans*. 2002. Disponível eletronicamente em < <http://java.sun.com/products/javabeans/> >, acessado em 28/08/2002.

[W3C 1999]. W3C. *HTML 4.01 Specification: W3C Recommendation 24 December 1999*. Disponível eletronicamente em < <http://www.w3.org/TR/1999/REC-html401-19991224/> >, acessado em 28/08/2002.

[LON 2002]. Longman Dictionaries. *Longman Essential Activator*. 2002. Disponível eletronicamente em < http://www.longman-elt.com/dictionaries/which_dict/essact.html >, acessado em 28/08/2002.

[PAP 2003]. Papillon Project. *Papillon Multilingual Dictionary Web Server*. 2003. Disponível eletronicamente em < <http://bushido.imag.fr/papillon/ConsultInformations.po?lang=eng> >, acessado em 3/01/2003.

[W3C 2000b]. W3C. *W3C XML Schema*. 2000. Disponível eletronicamente em < <http://www.w3.org/XML/Schema> >, acessado em 6/01/2003.

[XQE 2003]. KATZ, Howard: Fatdog Software. *XQEngine (XML Query Engine)*. 2003. Disponível eletronicamente em < <http://www.fatdog.com/> >, acessado em 10/01/2003.

[W3C 2000c]. W3C. *XML Query*. 2000. Disponível eletronicamente em < <http://www.w3.org/XML/Query> >, acessado em 10/01/2003.

[JXB 2003]. The JAXB API, Kohsuke Kawaguchi. 2003. Disponível eletronicamente em < <http://www.xml.com/pub/a/2003/01/08/jaxb-api.html> >, acessado em 12/01/2003.

[SUN 2003a]. Sun Microsystems, Inc. *Java Architecture for XML Binding*. 2003. Disponível eletronicamente em < <http://java.sun.com/xml/jaxb/> >, acessado em 12/01/2003.

[SUN 2003b]. Sun Microsystems, Inc. *JavaServer Pages*. 2003. Disponível eletronicamente em < <http://java.sun.com/products/jsp/> >, acessado em 12/01/2003.

Anexos

Anexo I - Artigo Relacionado ao Projeto

Universidade Federal de Santa Catarina
Curso de Ciências da Computação

Projeto de Conclusão de Curso

**Desenvolvimento de um Dicionário Eletrônico de Apoio à
Produção de Textos em Língua Estrangeira**

Herbert Mattei de Borba

herb@inf.ufsc.br

Tiago Müller Gil Cardoso

tmuller@inf.ufsc.br

Orientador:

Prof. Dr. Philippe Humblé

humbled@cce.ufsc.br

Co-Orientador:

Prof. Dr. José Eduardo De Lucca

delucca@inf.ufsc.br

Banca Examinadora:

Prof. Dr. Walter Carlos Costa

wcosta@cce.ufsc.br

Resumo: Este artigo apresenta o desenvolvimento de um dicionário eletrônico de auxílio à produção de textos em língua estrangeira, de um esquema de armazenamento, pesquisa e gerenciamento baseado nas tecnologias Java e XML. Foram criadas duas interfaces para o sistema: a primeira pode ser instalada a partir de um CD-ROM (Java Swing) e a segunda possibilita o acesso remoto através de um servidor web (JSP, JavaBeans).

Abstract: *This paper presents the development of a foreign language text production electronic dictionary, of a scheme for storage, searching and management based on Java and XML technologies. There were created two interfaces to the system: the first may be installed from a CD-ROM (Java Swing) and the second allows remote access through a web server (JSP, JavaBeans).*

Palavras-Chave: Dicionário de Produção, XML, XQuery, XQEngine, Java, JAXB.

1. Introdução

O desenvolvimento do dicionário eletrônico de apoio à produção de textos em língua estrangeira é apresentado como projeto de conclusão do curso de Ciências da Computação da Universidade Federal de Santa Catarina e dá continuidade ao projeto do *Dicionário Bilingüe Português-Espanhol da Universidade Federal de Santa Catarina*, coordenado pelo Prof. Philippe Humblé do Departamento de Línguas e Literatura Estrangeiras.

2. O Desenvolvimento do Dicionário Eletrônico de Produção de Textos em Língua Estrangeira

2.1. Objetivo

O objetivo do projeto foi desenvolver um sistema de apoio à produção de textos em língua estrangeira como definido pelos orientadores do projeto. A consulta ao sistema pode ser realizada remotamente através de navegadores web ou localmente através de uma distribuição em CD-ROM.

2.2. Requisitos

Em conjunto com os orientadores do projeto foi definido que:

- as informações do dicionário disponíveis no protótipo criado pelo Prof. Humblé seriam convertidas para um formato XML;
- deveria ser utilizado um esquema de indexação para permitir pesquisas eficientes no dicionário;
- as pesquisas no dicionário deveriam ser realizadas tanto por uma aplicação local quanto através de navegadores web.

2.3. Descrição do Sistema

Cada **volume** do dicionário de produção é organizado em **lemas** contendo um vocábulo como **núcleo** e **exemplos** de uso desse vocábulo com as respectivas **traduções** para o idioma destino. Para a definição do formato XML [1] dos dados do dicionário foi criado um *Schema XML* [2], baseado na estrutura encontrada no protótipo de Humblé. A listagem a seguir ilustra um lema já convertido para o formato XML.

```
<lema>
  <nucleo>abacate</nucleo>
  <traducao>aguacate, palta.</traducao>
  <exemplo>
    <fonte>Eu adoro abacate amassadinho com açúcar e limão.</fonte>
    <destino>Me encanta el aguacate / la palta machacada con azúcar y
limón.</destino>
  </exemplo>
  <exemplo>
    <fonte>O abacate é uma fruta rica em óleo e vitamina A.</fonte>
    <destino>El aguacate / La palta es una fruta rica en grasas y vitamina
A.</destino>
</lema>
```

Após a definição de como os dados seriam armazenados, iniciou-se a produção das classes *Java* [3] do núcleo do dicionário, o pacote `dicprod`. Para a implementação da classe de pesquisa, foi utilizada uma biblioteca para indexação e busca em documentos XML chamada *XQEngine* [4], distribuída livremente e de código-fonte aberto. A *XQEngine* utiliza a *XQuery* [5], linguagem proposta (ainda em desenvolvimento) pelo W3C para realizar consultas em estruturas como documentos XML. Os resultados das consultas também são estruturas XML, definidas segundo um outro Schema XML (XSD). A classe gerente de resultados utiliza as classes geradas pelo JAXB [6] a partir deste XSD para converter os resultados das consultas em código HTML [7], ao mesmo tempo em que reconhece os termos procurados utilizando expressões regulares e enfatiza-os no código HTML resposta. Finalmente, esta resposta pode ser apresentada na aplicação com interface *Swing* [8] ou enviada ao navegador web pela interface criada com *JSP* [9] e *JavaBeans* [10].

3. Conclusão

A tecnologia XML possibilita rápidas implementações de diferentes tipos de sistemas de informação. Apesar da especificação da linguagem de pesquisa XQuery encontrar-se em fase de desenvolvimento, foi possível desenvolver um protótipo funcional para o sistema proposto utilizando poucos recursos. A ferramenta XQEngine fornece uma solução de código aberto para o problema da indexação e pesquisa em textos estruturados em documentos XML, no entanto, este mecanismo de busca necessita de mais tempo até que todas as funcionalidades necessárias estejam operacionais, como a possibilidade de utilizar caracteres coringas (expressões regulares) para aumentar a capacidade das pesquisas.

O estado atingido pelo trabalho não foi suficiente para um produto comercialmente viável, porém, a falta de um similar a este dicionário assinala o quanto este segmento é pouco explorado, ao mesmo tempo em que destaca a importância deste trabalho como um referencial para sistemas semelhantes futuros.

4. Referências Bibliográficas

[1]. W3C. *Extensible Markup Language (XML) 1.0*. 2. ed. 2000. Disponível eletronicamente em < <http://www.w3.org/TR/2000/REC-xml-20001006> >, acessado em 27/08/2002.

[2]. W3C. *W3C XML Schema*. 2000. Disponível eletronicamente em < <http://www.w3.org/XML/Schema> >, acessado em 6/01/2003.

[3]. Sun Microsystems, Inc. *java.sun.com: The Source for Java Technology*. 2002. Disponível eletronicamente em < <http://java.sun.com> >, acessado em 27/08/2002.

[4]. KATZ, Howard: Fatdog Software. *XQEngine (XML Query Engine)*. 2003. Disponível eletronicamente em < <http://www.fatdog.com/> >, acessado em 10/01/2003.

[5]. W3C. *XML Query*. 2000. Disponível eletronicamente em < <http://www.w3.org/XML/Query> >, acessado em 10/01/2003.

[6]. Sun Microsystems, Inc. *Java Architecture for XML Binding*. 2003. Disponível eletronicamente em < <http://java.sun.com/xml/jaxb/> >, acessado em 12/01/2003.

[7]. W3C. *HTML 4.01 Specification*: W3C Recommendation 24 December 1999. Disponível eletronicamente em < <http://www.w3.org/TR/1999/REC-html401-19991224/> >, acessado em 28/08/2002.

[8]. Sun Microsystems, Inc. *JFC/Swing GUI Components*. 2002. Disponível eletronicamente em < <http://java.sun.com/products/jfc/#components> >, acessado em 28/08/2002.

[9]. Sun Microsystems, Inc. *JavaServer Pages*. 2003. Disponível eletronicamente em < <http://java.sun.com/products/jsp/> >, acessado em 12/01/2003.

[10]. Sun Microsystems, Inc. *JavaBeans*. 2002. Disponível eletronicamente em < <http://java.sun.com/products/javabeans/> >, acessado em 28/08/2002.

**Anexo II - O Dicionário Bilingüe Português-
Espanhol da Universidade Federal de Santa
Catarina**

**O DICIONÁRIO BILINGÜE PORTUGUÊS-ESPAÑHOL DA UNIVERSIDADE
FEDERAL DE SANTA CATARINA**

Philippe Humblé (humblé@cce.ufsc.br)
(Universidade Federal de Santa Catarina)

Resumo

Esta comunicação descreve um *Dicionário Bilingüe Português-Espanhol* que está sendo compilado na *Universidade Federal de Santa Catarina* e que tem as seguintes características:

1. Ele se dirige à *produção* do espanhol e não à *compreensão*;
2. Ele se dirige a um público determinado, os alunos brasileiros;
3. Ele foi concebido desde o início como um dicionário eletrônico.

Estas características são ilustradas nesta comunicação.

Abstract

This communication describes a *Bilingual Portuguese-Spanish Dictionary* which is being compiled at the *Universidade Federal de Santa Catarina* and has the following characteristics:

1. It was designed specifically to help with *encoding* and not with *decoding*;
2. It has a specific audience: Brazilian university students;
3. It was designed from the start on as an electronic dictionary.

These characteristics are explained and illustrated in this communication.

Um dicionário de *produção*

Na hora de traduzir uma palavra desconhecida de uma língua estrangeira para o português, geralmente os alunos brasileiros têm poucos problemas. Apesar de os dicionários bilingües existentes no mercado não satisfazerem sob todos os aspectos, haverá ao menos um que conterà uma determinada palavra espanhola desconhecida que encontrarmos num texto. Em muitas ocasiões este determinado dicionário oferecerá várias possibilidades e escolheremos a alternativa que melhor combina com o contexto do que estamos lendo. Mesmo no caso de não encontrarmos entre essas alternativas a palavra exata, o número de alternativas bastará para entendermos o que o autor quis comunicar. No caso de uma palavra desconhecida não ser encontrada, o aluno brasileiro não encontrará dificuldades em entender a definição dada por um dos muitos dicionários monolingües existentes por causa da sabida ‘proximidade dos códigos’.

Na hora de *produzir* o idioma estrangeiro, no entanto, surgem os problemas. Se a palavra desconhecida em espanhol se encontrar no dicionário o usuário ainda terá que escolher o termo certo dentre várias alternativas; ainda mais, ele terá que descobrir como este termo é usado a nível sintático.

O primeiro dilema: a escolha do termo certo

O que se vê geralmente ao usar um dicionário bilingüe português-espanhol é que na maioria das vezes a palavra procurada se encontra efetivamente no dicionário. No entanto, os equivalentes que se dão para a tradução são geralmente mais do que um. A que restrições está submetida a escolha? Acho que pelo menos a três.

Primeiro, a palavra tem que ter o mesmo significado básico do que a palavra que temos em mente em português. Isso significa que, no caso de palavras polissêmicas, a alternativa deve corresponder ao contexto em que se usa esse determinado significado em português. Por exemplo, a palavra ‘grade’ será traduzida por ‘reja’ no contexto de uma prisão, por ‘horário’ no caso de uma ‘grade curricular’, ‘parrilla’ no caso de um churrasco, e assim por diante.

Mas não é suficiente ter a certeza de estar usando a alternativa certa em termos de sentido. Temos que usar a palavra também no *registro* certo. ‘Tipo’ é uma tradução certa para ‘pessoa’ dependendo do caso. Nem sempre distinções de registro feitas numa língua são feitas também na outra e às vezes uma determinada pode ter traduções que são diferentes em termos de registro.

Finalmente, as palavras geralmente se encontram em companhia de determinadas outras palavras para formar ‘colocações’. Dizemos ‘exortar a população a manter a calma’ e não freqüentemente ‘exortar a

população a conservar a calma’ ou ‘exortar a população a manter a serenidade’, mesmo que qualquer pessoa entendesse o que queremos dizer com estas últimas alternativas. Isso não é só verdade no caso de fórmulas, mas também no caso da maioria das orações. Por exemplo, nas linhas que acabo de escrever usei sucessões de palavras como ‘em algumas ocasiões’ ‘ter a certeza’, ‘em determinados casos’ e outros conjuntos de palavras. É importante respeitar as colocações porque elas fazem com que soemos ‘natural’ e não só ‘gramatical’. Os dicionários, em parte por causa do suporte material de papel, que por um lado limita a quantidade de texto que se põe num livro e por outro oferece alternativas limitadas de classificação dos dados, têm dado pouca atenção às colocações. As colocações não são importantes na hora de *entender*, mas elas são fundamentais na hora de *produzir*. Escolher a palavra certa para o contexto certo não é uma opção, é uma exigência.

O segundo dilema: o uso do termo

Toda palavra está submetida a restrições de ordem sintática. Em português se diz ‘traduzir para’ quando em espanhol se diz ‘traducir a’. Também neste aspecto os dicionários bilingües tanto quanto monolingües têm se mostrado deficientes. Os monolingües, porque supõem com razão que seu público alvo não tem problemas de tipo sintático. Nenhum falante de espanhol duvidará entre dizer ‘traducir para el griego’ ou ‘al griego’. Os dicionários bilingües até agora não deram muita atenção para esse tipo de informação porque sua inclusão implicaria numa pesquisa demorada, talvez custosa, dos problemas que o público alvo tem ao *produzir* a língua estrangeira. Os dicionários hoje, em sua grande maioria, foram pensados para ajudar o usuário a *entender* a língua estrangeira, não a *produzir*.

A escolha do público

Foi durante e depois de uma análise de como um aprendiz de língua estrangeira resolve seus problemas de busca produtiva num dicionário¹ que resolvemos fazer um dicionário que desse uma solução aos problemas de *produção*. Em muitos casos isso significou fazer exatamente o contrário do que fazem os dicionários tradicionais.

¹ Os resultados desta pesquisa estão no meu livro *Dictionaries and Language Learners*. Haag und Herchen, Frankfurt am Main, 2001, ISBN 3-89846-110-6

Primeiro decidimos definir o nosso público. Por razões comerciais os editores de dicionários tradicionais não definem seu público. Um dicionário Português-Espanhol tradicional tem que atrair tanto brasileiros quanto portugueses, espanhóis, colombianos, cabo-verdianos e assim por diante. Não há problema em não escolher nenhum público em particular quando se trata de fornecer ao usuário um dicionário para *entender* uma língua estrangeira, mas para *produzir* esta língua o usuário precisa de mais informações e estas, por sua vez, devem ser mais precisas, como foi demonstrado anteriormente. O público de nosso dicionário foi definido como o *aluno universitário brasileiro*. Como veremos mais adiante, isto não implica que outras pessoas não possam tirar proveito deste dicionário, mas a intenção primeira foi de ir ao encontro dos problemas enfrentados por este grupo de pessoas.

A escolha deste público definiu a escolha dos verbetes. Percebemos ao longo da nossa prática docente que os alunos tinham dificuldades para se expressar naturalmente ao falar o espanhol. A língua escrita é a que tem precedência nos dicionários, mas o que primeiramente interessa aos alunos é poder se expressar verbalmente e naturalmente. Isto implica em palavras e usos que não são representados nos dicionários tradicionais e que decidimos incluir.

Se a primeira característica do nosso dicionário foi de escolher um público específico, a segunda característica foi de concebê-lo exclusivamente como um dicionário de *produção*. Isso implica que ele tem uma 'direção de uso'. Ao contrário de outros dicionários tradicionais, ele foi pensado para ser usado na direção Português-Espanhol, e não ao contrário. Isso quer dizer que, mesmo se falantes de espanhol encontrarem aqui respostas a suas perguntas, o dicionário não foi feito para que esses usuários resolvam suas dúvidas com relação ao português. Na hora de selecionar os exemplos para os verbetes, o que foi tomado em conta foram os usos e os significados da palavra em português e não em espanhol. O fato do dicionário ter uma finalidade bem definida, a de ajudar para a *produção* da língua, fez com que cortássemos tudo o que julgávamos não ser necessário para este processo. Assim chegamos à conclusão que um dicionário que constasse só de exemplos seria o mais adequado. Terminamos com cerca de 25.000 exemplos. Estas últimas considerações, junto com a escolha do público, determinou a escolha dos verbetes.

A escolha dos verbetes

É costume entre as editoras de dicionários fazer do número de verbetes a principal propaganda. Um dicionário com menos de 30.000 verbetes é considerado 'micro'. Quando pensamos nos verbetes do nosso

dicionário, levamos em conta que o usuário precisa mais do que nada de palavras comuns. Estas são também as palavras que mais significados têm e que mais se envolvem em colocações. Por exemplo, no *Aurélio* ‘pegar’ tem 45 acepções, ‘tirar’ 48, ‘pôr’ 67 e assim por diante. Decidimos limitar o número de verbetes do dicionário a 5.000 e tratá-los da maneira mais completa possível. Mesmo assim e pelo fato do dicionário ser eletrônico, o nosso dicionário dá exemplos de um número muito maior de palavras porque cada exemplo é exemplo para cada palavra que compõe este exemplo. Num dicionário eletrônico o exemplo ‘A casa foi terminada o ano passado’ é exemplo para as palavras ‘casa’, ‘terminar’, ‘ano’ e ‘passado’. Desta maneira poderia se dizer que o número de verbetes do dicionário equivale ao número de palavras diferentes usadas no dicionário. Como há 25.000 exemplos isso nos dá uma cifra muito mais alta do que 5000.

A escolha dos exemplos

Tradicionalmente os lexicógrafos costumam inventar os exemplos com os quais ilustram os verbetes. Mas recentemente a tendência virou no sentido de selecionar exemplos autênticos, porque mesmo um falante nativo não consegue inventar exemplos que reflitam fielmente o uso de uma palavra. Como o nosso interesse era basicamente ser um dicionário para aprendizes brasileiros ficamos muito felizes de poder contar com o banco de dados valiosíssimo do projeto VARSUL do qual a UFSC faz parte. Como o material que constitui o corpus VARSUL é autêntico, pegamos um máximo de exemplos deste corpus. Não seguimos, no entanto, a ortodoxia e mudamos os exemplos toda vez que sentimos que era necessário. Com efeito, as pessoas quando falam erram, se repetem e corrigem, e não havia necessidade para que o dicionário fosse tão autêntico.

Em alguns casos fomos obrigados a inventar exemplos, porque de certas acepções de palavras importantes não constava nenhum exemplo no corpus. Foi algumas vezes o caso de acepções que consideramos básicas. É um fato sabido¹ que é difícil achar um exemplo ‘típico’ do uso de uma palavra.

Um dicionário eletrônico

O fato de termos partido de zero permitiu que não fôssemos obrigados a romper com velhos hábitos. Foram analisadas as necessidades do público alvo, suas características e de acordo com esses dados foi analisado quais teriam que ser as características do dicionário. Optou-se muito rapidamente por um dicionário eletrônico

¹ Veja Sinclair *Corpus, concordance, collocation*, na Introdução e em todo o capítulo 8.

pela sua facilidade de uso. Do ponto de vista da redação, o fato de termos nos livrado do suporte em papel permitiu que o nosso dicionário fosse adquirindo características diferentes. A principal delas decorre do fato que o computador libera o lexicógrafo da ditadura do alfabeto. Num dicionário impresso, as palavras têm que ser classificadas individualmente e ilustradas individualmente. ‘De graça’ não pode estar classificado ao mesmo tempo com ‘graça’, ‘de’ e muito menos com ‘até’, ‘nem’, ‘trabalhar’, ‘distribuir’, etc., sendo estas últimas palavras as que mais freqüentemente aparecem com ‘de graça’. No caso de um dicionário eletrônico, ‘de graça’ vai aparecer onde quer que esteja no dicionário. Isso terá a vantagem de mostrar a tradução da palavra não só no caso, deveras hipotético, em que ela é usada individualmente, mas também quando ela é usada em diferentes contextos e com a vantagem de que esses contextos também aparecem traduzidos.

Um dicionário feito por professores

Por último, o que distingue este dicionário da grande maioria é o fato dele ter nascido da prática de professores de língua, que são mais conscientes do que os lexicógrafos de quais são as necessidades do usuário. Esta é uma tendência que parece estar despontando mundialmente. Posso aqui me remeter ao trabalho de Jean Binon e Serge Verlinde da Universidade Católica de Louvain, igualmente professores, e que redigiram um dicionário de francês econômico desenhado com as necessidades do público alvo em mente.

O uso do dicionário

Como funciona o dicionário? Até o presente momento não há ainda software próprio para o dicionário, mas como este terá basicamente as características do programa AskSam usado para a compilação do dicionário, uma demonstração com este software dá uma boa impressão do que será o resultado final. Suponhamos que procuremos a palavra ‘atrapalhar’. Uma procura pela palavra ‘atrapalh*’ nos dará os seguintes resultados (*lp* significa ‘lema em português’; *ep* ‘exemplo em português’; *ee* ‘exemplo em espanhol’).

1. *ep* [*A falta de recursos poderá atrapalhar o andamento dos debates sobre os projetos do Governo.*
ee [*La falta de recursos puede dificultar el desarrollo de los debates sobre los proyectos del gobierno.*
 (No verbete ‘andamento’)
2. ***lp*** [**atrapalhar**
ep [*Nós não atrapalhamos a senhora muito, não?*
ee [*¿No la incomodamos demasiado a Usted?*
ep [*O Exército foi que me atrapalhou.*
ee [*Lo que desbarató mis planes fue el ejército.*

<p><i>ep[O senhor acha que o progresso atrapalhou um pouco a vida das pessoas aqui?</i> <i>ee[¿Le parece que el progreso estropeó un poco la vida de esta gente?</i> <i>ep[Ele se atrapalhou todo na hora de responder as perguntas para o policial.</i> <i>ee[(Él) se confundió / no daba pie con bola al momento de responder las preguntas del policía. (No próprio verbete ‘atrapalhar’)</i></p> <p>3. <i>ep[Jogaram no lixo aquele bagulho que atrapalhava a entrada da frente.</i> <i>ee[Echaron a la basura aquel cacharro / trique (Méx.) que estorbaba la entrada . (No verbete ‘bagulho’.)</i></p> <p>4. <i>ep[Os faróis altos dos carros me atrapalham a visão.</i> <i>ee[Las luces altas de los autos me dificultan la visión. (No verbete ‘farol’.)</i></p> <p>5. <i>ep[O brasileiro classe A já está habitado a levar a família para almoçar fora aos domingos e a crise não atrapalhou tanto o movimento como era esperado.</i> <i>ee[El brasileiro de la clase alta está acostumbrado a llevar a la familia a comer afuera los domingos, la crisis no afecto tanto el movimiento como era esperado. (No verbete ‘habitado’.)</i></p> <p>6. <i>ep[O senhor acha que o progresso atrapalhou um pouco a vida das pessoas aqui?</i> <i>ee[¿A Usted le parece que el progreso perturbó un poco la vida de las personas? (No verbete ‘progresso’.)</i></p> <p>7. <i>ep[Outra coisa que eu acho assim que deve atrapalhar um casamento é a rotina.</i> <i>ee[Otra cosa que yo creo que necesariamente perjudica el casamiento es la rutina. (No verbete ‘rotina’.)</i></p> <p>8. <i>ep[Ele é bem atrapalhado.</i> <i>ee[Él es muy desordenado / confuso. (No verbete ‘ser’.)</i></p>

Um verbo como ‘atrapalhar’ há 11 exemplos espalhados pelo dicionário rendendo 11 traduções diferentes dependendo do contexto. Secundariamente, o dicionário oferece não só a tradução de *atrapalhar* mas também de todas as palavras neste contexto com as quais *atrapalhar* forma colocações. O exemplo mais eloqüente disso é ‘atrapalhar o andamento dos debates’. Outro exemplo, num caso como ‘média’, há exemplos do uso desta palavra incluídos nos verbetes dedicados a *abaixo*, *acima*, *alumínio*, *atingir*, *baixo*, *botar*, *campeonato*, *consumo*, *história*, *idade*, *litro*, *morar*, *média*, *paralelo*, *ponto*, *treinar*.

Conclusão

Nesta comunicação apresentamos o dicionário bilingüe português-espanhol que estamos desenvolvendo na Universidade Federal de Santa Catarina. Este dicionário tem como principais características o fato de ter a finalidade expressa de ajudar na *produção* do espanhol, de se dirigir especificamente a um público brasileiro e, finalmente, de ser eletrônico. A publicação do dicionário está prevista para o começo do ano 2003.

Bibliografia

1. Binon J. and Verlinde S 1994. *The Dictionnaire contextuel du français économique. A production oriented dictionary of business French.* Martin W., Meys W., Moerland M., Plas ten E., Sterkenburg van, Vossen (Eds.) 1994. *Euralex '94 Proceedings. Papers Submitted to the 6th Euralex International Congress on Lexicography in Amsterdam.* The Netherlands, 523-529.
2. Ooi, V, 1999, *Corpus Linguistics and Lexicography*, University of Edinburgh Press
3. Sinclair J. M. (Ed.) 1987. *Looking Up*, Collins, London and Glasgow.
4. Sinclair J. M. 1988. Naturalness in Language. McCarthy, M. (Ed.) 1988. *Naturalness in Language.* English Language Research Journal Vol. 2. University of Birmingham.
5. Sinclair J. M. 1991. *Corpus, Concordance, Collocation*, Oxford University Press, Oxford.

Anexo III - Código-Fonte Desenvolvido no Projeto

XML SCHEMAS E DTDs

DicionarioProducaoSchema.xsd

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Mecanismo utilizado para validação: http://www.w3.org/2001/03/webdata/xsv -->

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <!-- TIPOS SIMPLES -->
  <xs:simpleType name="tipoIdioma">
    <xs:annotation>
      <xs:documentation xml:lang="pt-BR">Define os códigos que serão utilizados
para identificar os idiomas fonte e destino do volume.</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:enumeration value="pt-BR"/>
      <xs:enumeration value="es"/>
      <!-- deve ser incrementado com todos os códigos dos países (segundo RFC 1766)
... -->
    </xs:restriction>
  </xs:simpleType>

  <!-- ELEMENTOS -->
  <xs:element name="volume">
    <xs:complexType>
      <xs:sequence minOccurs="1" maxOccurs="unbounded">
        <xs:element ref="lema" />
      </xs:sequence>
      <xs:attribute name="nome" type="xs:string"/>
      <xs:attribute name="idioma-fonte" type="tipoIdioma"/>
      <xs:attribute name="idioma-destino" type="tipoIdioma"/>
      <!-- FALTANDO: complementar com atributos como Autor, Data, ... -->
    </xs:complexType>
  </xs:element>

  <xs:element name="lema">
    <xs:annotation>
      <xs:documentation xml:lang="pt-BR">O elemento <lema> tem como "chave
primária" o elemento <nucleo>, que dessa forma deve ser único para cada
volume.</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="nucleo" type="xs:ID" />
        <xs:element name="traducao" type="xs:string" minOccurs="1" maxOccurs="1" />
        <xs:element ref="exemplo" minOccurs="1" maxOccurs="unbounded" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="exemplo">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="fonte" type="xs:string" />
        <xs:element name="destino" type="xs:string" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>

```

DicionarioProducaoDocType.dtd

```
<?xml version='1.0' encoding='ISO-8859-1'?>

<!ELEMENT lema (exemplo|traducao|nucleo)*>

<!ELEMENT exemplo (fonte|destino)*>

<!ELEMENT fonte (#PCDATA)>

<!ELEMENT destino (#PCDATA)>

<!ELEMENT traducao (#PCDATA)>

<!ELEMENT nucleo (#PCDATA)>

<!ELEMENT volume (lema)*>
<!ATTLIST volume
  nome CDATA #IMPLIED
  xmlns:xsi CDATA #IMPLIED
  idioma-destino CDATA #IMPLIED
  idioma-fonte CDATA #IMPLIED
  xsi:noNamespaceSchemaLocation CDATA #IMPLIED
  >
```

DicionarioResultadoLemaXSD.xsd

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Mecanismo utilizado para validação: http://www.w3.org/2001/03/webdata/xsv -->

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <!-- TIPOS SIMPLES -->
  <xs:simpleType name="tipoIdioma">
    <xs:annotation>
      <xs:documentation xml:lang="pt-BR">Define os códigos que serão utilizados
para identificar os idiomas fonte e destino do volume.</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:enumeration value="pt-BR"/>
      <xs:enumeration value="es"/>
      <!-- deve ser incrementado com todos os códigos dos países (segundo RFC 1766)
... -->
    </xs:restriction>
  </xs:simpleType>

  <!-- ELEMENTOS -->
  <xs:element name="lemas">
    <xs:complexType>
      <xs:sequence minOccurs="1" maxOccurs="unbounded">
        <xs:element ref="lema" />
      </xs:sequence>
      <xs:attribute name="pesquisa" type="xs:string"/>
      <xs:attribute name="idioma-fonte" type="tipoIdioma"/>
      <xs:attribute name="idioma-destino" type="tipoIdioma"/>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```

    <!-- FALTANDO: complementar com atributos como Autor, Data, ... -->
  </xs:complexType>
</xs:element>

<xs:element name="lema">
  <xs:annotation>
    <xs:documentation xml:lang="pt-BR">O elemento <lt;lema> tem como "chave
primária" o elemento <lt;nucleo>, que dessa forma deve ser único para cada
volume.</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="nucleo" type="xs:ID" />
      <xs:element name="traducao" type="xs:string" minOccurs="1" maxOccurs="1" />
      <xs:element ref="exemplo" minOccurs="1" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="exemplo">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="fonte" type="xs:string" />
      <xs:element name="destino" type="xs:string" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>

```

DicProdResultadoExemploXSD.xsd

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Mecanismo utilizado para validação: http://www.w3.org/2001/03/webdata/xsv -->

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <!-- TIPOS SIMPLES -->
  <xs:simpleType name="tipoIdioma">
    <xs:annotation>
      <xs:documentation xml:lang="pt-BR">Define os códigos que serão utilizados
para identificar os idiomas fonte e destino do volume.</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:enumeration value="pt-BR"/>
      <xs:enumeration value="es"/>
      <!-- deve ser incrementado com todos os códigos dos países (segundo RFC 1766)
... -->
    </xs:restriction>
  </xs:simpleType>

  <!-- ELEMENTOS -->
  <xs:element name="exemplos">
    <xs:complexType>
      <xs:sequence minOccurs="1" maxOccurs="unbounded">
        <xs:element ref="exemplo" />
      </xs:sequence>
      <xs:attribute name="pesquisa" type="xs:string"/>
      <xs:attribute name="idioma-fonte" type="tipoIdioma"/>
    </xs:complexType>
  </xs:element>

```



```

        <xs:attribute name="idioma-destino" type="tipoIdioma"/>
    </xs:complexType>
</xs:element>

<xs:element name="exemplo">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="fonte" type="xs:string" />
            <xs:element name="destino" type="xs:string" />
        </xs:sequence>
    </xs:complexType>
</xs:element>

</xs:schema>

```

DicProdResultadoExemploXSL.xsl

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:fo="http://www.w3.org/1999/XSL/Format">
    <xsl:template match="/">
        <html>
            <head>
                <link rel="stylesheet" href="xml/DicProdResultados.css" type="text/css" />
            </head>
            <body>
                <table border="0" width="500">
                    <xsl:for-each select="exemplos/exemplo" >
                        <tr>
                            <td class="fonte"><xsl:number count="exemplo" format="1. " /></td>
                            <td class="fonte"><xsl:value-of select="fonte" /></td>
                        </tr>
                        <tr>
                            <td></td><td class="destino"><xsl:value-of select="destino" /></td>
                        </tr>
                    </xsl:for-each>
                    <tr>
                        <td colspan="2"><hr/></td>
                    </tr>
                </table>
            </body>
        </html>
    </xsl:template>
</xsl:stylesheet>

```

DicProdResultadoLemaXSL.xsl

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:fo="http://www.w3.org/1999/XSL/Format">
    <xsl:template match="/">
        <html>

```

```

    <head>
      <link rel="stylesheet" href="xml/DicProdResultados.css" type="text/css"
/>
    </head>
    <body>
      <table border="0" width="500">

        <xsl:for-each select="lemas/lema" >
          <tr>
            <td colspan="2" class="nucleo">Núcleo: <xsl:value-of select="nucleo"
/></td>
          </tr>
          <tr>
            <td colspan="2" class="traducao"><xsl:value-of select="traducao"
/></td>
          </tr>

          <xsl:for-each select="exemplo" >
            <tr>
              <td class="fonte"><xsl:number count="exemplo" value="position()"
format="1. " /></td>
              <td class="fonte"><xsl:value-of select="fonte" /></td>
            </tr>
            <tr>
              <td></td><td class="destino"><xsl:value-of select="destino" /></td>
            </tr>
          </xsl:for-each>
          <tr>
            <td colspan="2"><hr/></td>
          </tr>
        </xsl:for-each>

      </table>
    </body>
  </html>

</xsl:template>

</xsl:stylesheet>

```

DicProdResultadoExemploDTD.dtd

```

<?xml version='1.0' encoding='UTF-8'?>

<!--
  Typical usage:

  <?xml version="1.0"?>

  <!DOCTYPE exemplos SYSTEM "DicProdResultados_exemplos.dtd">

  <exemplos>
    ...
  </exemplos>
-->

<!ELEMENT destino (#PCDATA)>

<!ELEMENT exemplo (destino|fonte)*>

<!ELEMENT exemplos (exemplo)*>

```

```

<!ATTLIST exemplos
  idioma-destino CDATA #IMPLIED
  xsi:noNamespaceSchemaLocation CDATA #IMPLIED
  pesquisa CDATA #IMPLIED
  xmlns:xsi CDATA #IMPLIED
  idioma-fonte CDATA #IMPLIED
>

<!ELEMENT fonte (#PCDATA)>

```

DicProdResultadoLemaDTD.dtd

```

<?xml version='1.0' encoding='UTF-8'?>

<!--
  Typical usage:

  <?xml version="1.0"?>

  <!DOCTYPE lemas SYSTEM "dsdfsd.dtd">

  <lemas>
  ...
  </lemas>
-->
<!ELEMENT lemas (lema)*>
<!ATTLIST lemas
  idioma-destino CDATA #IMPLIED
  idioma-fonte CDATA #IMPLIED
  pesquisa CDATA #IMPLIED
>

<!ELEMENT lema (nucleo|traducao|exemplo)*>
<!ELEMENT nucleo (#PCDATA)>
<!ELEMENT traducao (#PCDATA)>
<!ELEMENT exemplo (destino|fonte)*>
<!ELEMENT destino (#PCDATA)>
<!ELEMENT fonte (#PCDATA)>

```

DicProdResultados.css

```

.nucleo {
  font-family: Verdana;
  font-size: 10pt;
  font-weight: bold;
}

.traducao {
  font-family: Verdana;
  font-size: 10pt;
  text-decoration: none;
  font-style: italic;
}

```

```
.fonte{
  font-family: Verdana;
  font-size: 10pt;
}

.destino {
  font-family: Verdana;
  font-size: 10pt;
  font-style: italic;
}
```

PACKAGE DICPROD

DicionarioXMLPesquisa.java

```
package dicprod;

/*
 * dicionarioXMLPesquisa.java
 *
 * Created on 13 de Janeiro de 2003, 21:29
 */

import java.io.*;
import java.util.Calendar;
import javax.xml.parsers.*;
import org.xml.sax.XMLReader;
import java.nio.charset.*;

import com.fatdog.xmlEngine.XQEngine;
import com.fatdog.xmlEngine.exceptions.*;
import com.fatdog.xmlEngine.query.ResultListener;

public class dicionarioXMLPesquisa implements ResultListener {

  private dicionarioConfig CONFIG;

  public dicionarioXMLPesquisa( boolean rel, boolean carregaGerente,
dicionarioConfig cfg ) {
    this._relativo = rel;
    inicieXMLPesquisa( true, cfg );
  }
  public dicionarioXMLPesquisa( ) {
    inicieXMLPesquisa( true, null );
  }
  /**
   * Constructor padrão
   */
  public dicionarioXMLPesquisa( boolean carregaGerente, dicionarioConfig cfg ) {
```

```

        this._relativo = false;
        inicieXMLPesquisa( carregaGerente, cfg );
    }

    public void setConfig( dicionarioConfig cfg ) {
        this.CONFIG=cfg;
    }

    public void setRelativo( boolean rel ) {
        this._relativo=rel;
    }

    void inicieXMLVars( ) {
        File xmlp;

        String strpath;
        if ( !this._relativo ) {
            // para utilizar path absoluto, de acordo com o sistema operacional
            this._XMLPATH = "file:" + new File( CONFIG.getVarCfg( "dirbase" ) + "xml/"
).getPath() + File.separatorChar;
        }
        else {
            this._XMLPATH = "xml/";
        }
    }

    void inicieXMLPesquisa( boolean carregaGerente, dicionarioConfig cfg ) {
        setConfig( cfg );
        inicieXMLVars( );
        _engine = new XQEngine();
        if ( carregaGerente ) {
            _gerente = new dicionarioXMLGerente( CONFIG );
        }
        try {
            installXercesXMLReader( );
            _engine.setDoIndexNumbers( false );
            _engine.WRITE_XML_HEADER = false;
            _engine.setDoFullText( true );
            _engine.setMinIndexableWordLength( 1 );
            _engine.setStopList( CONFIG.getVarCfg( "xqe.stoplist" ) );
        }
        catch( java.io.IOException ioe ) {
            System.out.println( ioe ); return;
        }
    }

    /**
     * Instala o reader XML que o XQEngine utilizará
     */
    private void installSunXMLReader( ) {
        SAXParserFactory spf = SAXParserFactory.newInstance();
        try {
            SAXParser parser = spf.newSAXParser();
            XMLReader reader = parser.getXMLReader();
            _engine.setXMLReader( reader );
        }
        catch( Exception e ) {
            System.out.println("(1) XMLReader não pode ser instalado!");
            System.out.println( e );
        }
    }

    /**
     * Instala o reader XML que o XQEngine utilizará
     */
    private void installXercesXMLReader() {
        try {

```

```

String parserName = "org.apache.xerces.parsers.SAXParser";
XMLReader parser = (XMLReader)Class.forName(parserName).newInstance();
_engine.setXMLReader( parser );
// if ( _engine.DEBUG_RESOLVE_ENTITIES == true )
// parser.setEntityResolver( _engine );
}
catch( Exception e ) {
    System.out.println("(2) XMLReader não pode ser instalado!");
    System.out.println( e );
}
}

/**
 * Cria um indice sem persistência
 */
public void crieIndice( ) {
    try {
        _engine.setDocument( CONFIG.getVarCfg("dic.dicionario") );
    }
    catch ( FileNotFoundException e ) {
        System.out.println( e );
    }
    catch ( CantParseDocumentException ce ) {
        System.out.println( ce );
    }
    catch ( MissingOrInvalidSaxParserException me ) {
        System.out.println( me );
    }
    catch( java.io.IOException ioe ) {
        System.out.println( ioe ); return;
    }
}

/**
 * Cria um índice persistente, que será salvo para que a inicialização posterior
 * seja mais rápida.
 */
public void crieIndicePersistente( ) {
    try {
        crieIndice( );
        _engine.setIndexFileName( CONFIG.getVarCfg("xqe.indexpathname") );
        _engine.saveIndex( );
    }
    catch ( FileNotFoundException e ) {
        System.out.println( e );
    }
    catch( java.io.IOException ioe ) {
        System.out.println( ioe ); return;
    }
}

/**
 * Carrega um índice persistente, salvo em disco
 */
public void carregueIndicePersistente( ) {
    carregueIndicePersistente( CONFIG.getVarCfg( "xqe.indexpathname" ) );
}
public void carregueIndicePersistente( String nomeBaseIndices ) {
    try {
        _engine.setIndexFileName( nomeBaseIndices );
        _engine.restoreIndex( );
    }
    catch( java.io.IOException ioe ) {

```

```

        System.out.println( "Indice nao encontrado! \nPrimeira vez que está rodando?
\nCriando indices..." );
        crieIndicePersistente( );
    }
}

/**
 * Faz uma busca genérica
 */
private void busca( String xQuery ) {
    _engine.setResultListener( this );
    try {
        _engine.setQuery( xQuery );
    }
    catch( InvalidQueryException e ) {
        System.out.println( "Pesquisa invalida: " + e.getMessage() );
        return;
    }
    catch ( com.fatdog.xmlEngine.exceptions.TypeException e ) {
        System.out.println( e.toString( ) );
    }
}

/**
 * Busca pelo conteúdo da tag <nucleo>
 */
public void busqueNucleo( String expr ) {
    this._exemplos = false;
    this._busca = expr;
    this.busca( "//lema[nucleo &= '"+ expr +"']" );
}

/**
 * Busca pelo conteúdo da tag <fonte>
 */
public void busqueFonte( String expr ) {
    this._exemplos = false;
    this._busca = expr;
    this.busca( "//lema[exemplo[fonte &= '"+ expr +"']]" );
}

/**
 * Busca pelo conteúdo da tag <destino>
 */
public void busqueDestino( String expr ) {
    this._exemplos = false;
    this._busca = expr;
    this.busca( "//lema[exemplo[destino &= '"+ expr +"']]" );
}

/**
 * Busca pelo conteúdo da tag <fonte>, trazendo apenas os exemplos que
 * encontrar
 */
public void busqueFonteExemplos( String expr ) {
    this._exemplos = true;
    this._busca = expr;
    this.busca( "//lema/exemplo[fonte &= '"+ expr +"']" );
}

/**
 * Busca pelo conteúdo da tag <destino>, trazendo apenas os exemplos que
 * encontrar
 */
public void busqueDestinoExemplos( String expr ) {
    this._exemplos = true;

```

```

    this._busca = expr;
    this.busca( "//lema/exemplo[destino &= '"+ expr + "'" ]" );
}

/**
 * Result listener da XQEngine
 * Os resultados da busca são mandados para este método
 */
public void results( String results ) {
    _resultados = results;
    _resultadosXML = montaResultadosXML( );
}

public String getResultados( ) {
    return _resultados;
}

public String getResultadosXML( ) {
    return _resultadosXML;
}

String montaResultadosXML( ) {
    Calendar agora = Calendar.getInstance();
    String res;
    String dirbase = CONFIG.getVarCfg( "dirbase" );

    String idmfonte;
    String idmdestino;
    if ( _gerente != null ) {
        idmfonte = _gerente.getVolumeIdiomaFonte();
        idmdestino = _gerente.getVolumeIdiomaDestino();
    }
    else {
        idmfonte = "DESCONHECIDO";
        idmdestino = "DESCONHECIDO";
    }

    String dtdstr;
    String xsdstr;
    String xslstr;
    if ( this._exemplos ) {
        dtdstr = "<!-- DTD retirado por incompatibilidade com o Internet Explorer 5.0
!DOCTYPE exemplos SYSTEM \" + _XMLPATH + \"DicProdResultadoExemploDTD.dtd\"-->";
        xsdstr = "xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\"
xsi:noNamespaceSchemaLocation=\" + _XMLPATH + \"DicProdResultadoExemploXSD.xsd\"";
        xslstr = "<?xml-stylesheet type=\"text/xsl\" href=\" + _XMLPATH +
\"DicProdResultadoExemploXSL.xsl\" ?>";

        res = "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n<!-- Gerado pelo motor de
busca do dicprod (dicionarioXMLPesquisa.java) -->\n<!-- " + agora.getTime() + " -
->\n" + dtdstr + "\n" + xslstr + "\n<exemplos " + xsdstr + " pesquisa=\" + _busca
+ "\" idioma-fonte=\" + idmfonte + "\" idioma-destino=\" + idmdestino + "\">\n" +
getResultados( ) + "</exemplos>";
    } else {
        dtdstr = "<!-- DTD retirado por incompatibilidade com o Internet Explorer 5.0
!DOCTYPE exemplos SYSTEM \" + _XMLPATH + \"DicProdResultadoLemaDTD.dtd\"-->";
        xsdstr = "xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\"
xsi:noNamespaceSchemaLocation=\" + _XMLPATH + \"DicProdResultadoLemaXSD.xsd\"";
        xslstr = "<?xml-stylesheet type=\"text/xsl\" href=\" + _XMLPATH +
\"DicProdResultadoLemaXSL.xsl\" ?>";

        res = "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n<!-- Gerado pelo motor de
busca do dicprod (dicionarioXMLPesquisa.java) -->\n<!-- " + agora.getTime() + " -
->\n" + dtdstr + "\n" + xslstr + "\n<lemas " + xsdstr + " pesquisa=\" + _busca
+ "\" idioma-fonte=\" + idmfonte + "\" idioma-destino=\" + idmdestino + "\">\n" +
getResultados( ) + "</lemas>";
    }
}

```



```

    }
    return res;
}

public void salveResultados( ) {
    // String nome = "DICPROD_" + agora.get( agora.MILLISECOND ) + "_" + agora.get(
    agora.SECOND ) + "_" + agora.get( agora.HOUR_OF_DAY ) + "_" + agora.get(
    agora.MINUTE ) + "_" + agora.get( agora.DAY_OF_MONTH ) + "_" + agora.get(
    agora.MONTH ) + "_" + agora.get( agora.YEAR ) + "_" + Math.abs( new
    java.util.Random().nextLong());
    String nome = "DicProdResultados.xml";
    File arq = new java.io.File( CONFIG.getVarCfg( "dirbase" ) + nome );
    try {
        FileOutputStream fos = new FileOutputStream( arq );
        OutputStreamWriter osw = new OutputStreamWriter( fos, _charset );
        osw.write( montaResultadosXML( ) );
        osw.close();
    }
    catch (IOException e) {
        System.out.println( e.toString() );
    }
}

public diccionarioConfig getConfig( ) {
    return CONFIG;
}

// VARIÁVEIS PRIVADAS
private diccionarioXMLGerente _gerente;
private String _busca;
private XQEngine _engine;
private String _resultados;
private String _resultadosXML;
private boolean _exemplos = false;
private String _XMLPATH;
private boolean _relativo;
private static Charset _charset = Charset.forName("ISO-8859-15");
}

```

diccionarioXMLResultadosGerente.java

```

package dicprod;

import java.io.*;
import java.util.*;
import java.util.regex.*;

// Bibliotecas JAXB para gerenciamento do XML
import javax.xml.bind.JAXBContext;
import javax.xml.bind.JAXBException;
import javax.xml.bind.Unmarshaller;

// importa as classes geradas pelo JAXB
import dicprod.jaxbResultadoExemplo.*;
import dicprod.jaxbResultadoLema.*;

public class diccionarioXMLResultadosGerente {

```

```

protected diccionarioConfig CONFIG;

/**
 * Cria um objeto gerenteDiccionarioXML que lê de um arquivo
 *
 *
 */
public diccionarioXMLResultadosGerente( String modo, diccionarioConfig cfg ) {
    try {
        CONFIG = cfg;
        String dirbase = CONFIG.getVarCfg( "dirbase" );
        this._modo = modo;
        JAXBContext jc;
        Unmarshaller u;
        if ( _modo.compareTo("exemplos") == 0 ) {
            jc = JAXBContext.newInstance( "dicprod.jaxbResultadoExemplo" );
            u = jc.createUnmarshaller();
            _exemplos = (Exemplos)u.unmarshal( new FileInputStream( dirbase +
"DicProdResultados.xml" ) );
        }
        else {
            jc = JAXBContext.newInstance( "dicprod.jaxbResultadoLema" );
            u = jc.createUnmarshaller();
            _lemas = (Lemas)u.unmarshal( new FileInputStream( dirbase +
"DicProdResultados.xml" ) );
        }
    } catch( JAXBException je ) {
        je.printStackTrace();
    } catch( IOException ioe ) {
        ioe.printStackTrace();
    }
}

/**
 * Cria um gerente que lê de um StringBuffer
 */
public diccionarioXMLResultadosGerente( String modo, String XMLResults ) {
    try {
        this._modo = modo;
        JAXBContext jc;
        Unmarshaller u;
        if ( _modo.compareTo("exemplos") == 0 ) {
            StringBuffer xmlStr = new StringBuffer( XMLResults );
            jc = JAXBContext.newInstance( "dicprod.jaxbResultadoExemplo" );
            u = jc.createUnmarshaller();
            _exemplos = (Exemplos)u.unmarshal( new
javax.xml.transform.stream.StreamSource( new StringReader( xmlStr.toString() ) ) );
        }
        else {
            StringBuffer xmlStr = new StringBuffer( XMLResults );
            jc = JAXBContext.newInstance( "dicprod.jaxbResultadoLema" );
            u = jc.createUnmarshaller();
            _lemas = (Lemas)u.unmarshal( new javax.xml.transform.stream.StreamSource(
new StringReader( xmlStr.toString() ) ) );
        }
    } catch( JAXBException je ) {
        je.printStackTrace();
    }
}

// GETTERS

public String getPesquisa( ) {
    if ( _modo.compareTo("exemplos") == 0 ) {
        return _exemplos.getPesquisa( );
    }
}

```

```

        else {
            return _lemas.getPesquisa( );
        }
    }
}
/**
 * Idioma fonte do resultado atual
 */
public String getIdiomaFonte( ) {
    if ( _modo.compareTo("exemplos") == 0 ) {
        return _exemplos.getIdiomaFonte( );
    }
    else {
        return _lemas.getIdiomaFonte( );
    }
}
/**
 * Idioma destino do resultado atual
 */
public String getIdiomaDestino( ) {
    if ( _modo.compareTo("exemplos") == 0 ) {
        return _exemplos.getIdiomaDestino( );
    }
    else {
        return _lemas.getIdiomaDestino( );
    }
}

public java.util.List getListaExemplos( ) {
    return _exemplos.getExemplo( );
}

public String getStrExemplos( ) {
    String strexemplos = "";
    List exemplos = this.getListaExemplos();

    int cntEx = 0;

    for( Iterator iter = exemplos.iterator(); iter.hasNext(); ) {
        dicprod.jaxbResultadoExemplo.ExemploType exemplo =
(dicprod.jaxbResultadoExemplo.ExemploType)iter.next();
        cntEx++;
        strexemplos = strexemplos + "Exemplo " + cntEx + ": " + exemplo.getFonte( )
+
                                                "\nTradução: " + exemplo.getDestino( ) + "\n";
    }
    return strexemplos;
}

/**
 * Função para acender os termos da pesquisa nos resultados
 */
public String doHilite( String texto ) {
    Pattern hipat;
    Matcher m;
    StringBuffer sbret = new StringBuffer( );
    String ret = texto;
    String tok = "";
    String ntext = texto;
    StringTokenizer pes = new StringTokenizer( getPesquisa( ) );

    while ( pes.hasMoreTokens( ) ) {
        sbret = new StringBuffer( );
        tok = pes.nextToken( );
        // Procura pelo termo que deve ser acendido, apenas onde eles aparecem como
uma palavra inteira.

```

```

        hipat = Pattern.compile( "(\\A|[¿;]|\\s)(" + tok + ")(\\s|\\z|[!.,\\?])",
Pattern.CASE_INSENSITIVE );
        m = hipat.matcher( ntext );
        while ( m.find( ) ) {
            m.appendReplacement( sbret, m.group(1) + "<font color=\"red\"> " +
m.group(2) + "</font>" + m.group(3) );
        }
        m.appendTail( sbret );
        ntext = sbret.toString( );
    }
    ret = sbret.toString( );
    return ret;
}

public String getHtmlExemplos( ) {
    String strexemplos = "";
    List exemplos = this.getListaExemplos();

    int cntEx = 0;

    for( Iterator iter = exemplos.iterator(); iter.hasNext(); ) {
        dicprod.jaxbResultadoExemplo.ExemploType exemplo =
(dicprod.jaxbResultadoExemplo.ExemploType)iter.next();
        cntEx++;
        strexemplos = strexemplos + "<b>Exemplo " + cntEx + ": </b>" + doHilite(
exemplo.getFonte( ) ) +
                                "<br><b>Tradução: </b><i>" + doHilite(
exemplo.getDestino( ) ) + "</i><br><br>";
    }
    return strexemplos;
}

public java.util.List getListaLemas( ) {
    return _lemas.getLema( );
}

public String getHtmlLemas( ) {
    String strlemas = "";
    List lemas = this.getListaLemas();

    int cntLema = 0;
    int cntEx;

    for( Iterator iter = lemas.iterator(); iter.hasNext(); ) {
        LemaType lema = (LemaType)iter.next();
        cntLema++;
        strlemas = strlemas + "<b>Núcleo " + cntLema + ": </b>" + lema.getNucleo();
        strlemas = strlemas + "<br><b>Tradução: </b><i>" + lema.getTraducao() +
"</i>";

        List exemplos = lema.getExemplo();

        cntEx = 0;
        strlemas = strlemas + "<blockquote>";
        for( Iterator iter2 = exemplos.iterator(); iter2.hasNext(); ) {
            cntEx++;
            dicprod.jaxbResultadoLema.ExemploType exemplo =
(dicprod.jaxbResultadoLema.ExemploType)iter2.next();
            strlemas = strlemas + "<br><b>Exemplo " + cntLema + "." + cntEx + ":
</b>" + doHilite( exemplo.getFonte( ) ) +
                                "<br><b>Tradução: </b><i>" + doHilite(
exemplo.getDestino( ) ) + "</i><br>";
        }
        strlemas = strlemas + "</blockquote>";
        strlemas = strlemas + "<br>";
    }
}

```

```

    return strlemas;
}

private Exemplos _exemplos;
private Lemas _lemas;
private String _modo;
}

```

dicionarioXMLGerente.java

```

package dicprod;

import java.io.FileInputStream;
import java.io.IOException;

// Bibliotecas JAXB para gerenciamento do XML
import javax.xml.bind.JAXBContext;
import javax.xml.bind.JAXBException;
import javax.xml.bind.Unmarshaller;

// importa as classes geradas pelo JAXB
import dicprod.jaxbDic.*;

public class dicionarioXMLGerente {

    private dicionarioConfig CONFIG;

    /**
     * Cria um objeto gerenteDicionarioXML
     *
     *
     */
    public dicionarioXMLGerente( dicionarioConfig cfg ) {
        try {
            CONFIG = cfg;
            // Instancia um objeto JAXBContext, que utilizará o Schema compilado
            // no pacote pacoteJava
            JAXBContext jc = JAXBContext.newInstance( "dicprod.jaxbDic" );

            Unmarshaller u = jc.createUnmarshaller();

            _volume = (Volume)u.unmarshal( new FileInputStream( CONFIG.getVarCfg(
"dic.dicionario" ) ) );

        } catch( JAXBException je ) {
            je.printStackTrace();
        }
    }
}

```

```

    } catch( IOException ioe ) {
        ioe.printStackTrace();
    }
}

// GETTERS
/**
 * Nome do volume atual
 */
public String getVolumeNome( ) {
    return _volume.getNome( );
}
/**
 * Idioma fonte do volume atual
 */
public String getVolumeIdiomaFonte( ) {
    return _volume.getIdiomaFonte( );
}
/**
 * Idioma destino do volume atual
 */
public String getVolumeIdiomaDestino( ) {
    return _volume.getIdiomaDestino( );
}

private Volume _volume;
}

```

dicionarioConfig.java

```

package dicprod;

import java.util.Properties;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.File;

public class dicionarioConfig {

    // Diretório raiz default. Utilizado para iniciar, aguardando um valor
    // real.
    private String _dirbase_default = "";
    // Diretório raiz passado pela linha de comando java -Ddirbase=xxx
    private String _dirbase; // para a GUI
    // Nome do arquivo
    private String CONF_NOME = "config.properties"; // junta tudo

    public dicionarioConfig( ) {
        try {
            _dirbase = new String( System.getProperty( "dirbase" ) );

```

```

    }
    catch ( NullPointerException e ) {
        _dirbase = _dirbase_default;
    }

    File propsFile = new File( _dirbase + CONF_NOME );
    carregaConfig( propsFile );
}

public dicionarioConfig( String dirbase ) {
    _dirbase = dirbase;
    File propsFile = new File( _dirbase + CONF_NOME );
    carregaConfig( propsFile );
}

/**
 * Utilizado para iniciar o um diretorio posteriormente, caso
 * um config.properties não possa ser encontrado no diretorio
 * padrão
 */
public void setDirBase( String dirbase ) {
    this._dirbase = dirbase;
    File propsFile = new File( _dirbase + CONF_NOME );
    carregaConfig( propsFile );
}

public void carregaConfig( File arqProp ) {
    Properties Defaults = new Properties( );
    Defaults.setProperty( "dirbase", _dirbase );
    _props = new Properties(Defaults);

    System.out.println( "Arquivo de configuracao: " +
arqProp.getAbsolutePath( ) );
    try {
        _props.load(new FileInputStream( arqProp ) );
    } catch (Exception e) {
        System.out.println( e.toString() );
        _tudocerto = false;
    }
}

public void salveConfig( ) {
    File propsFile = new File( _dirbase + CONF_NOME );
    try {
        _props.store( new FileOutputStream( propsFile ), "Arquivo de
configuração do dicprod - (c)2003 herb@inf.ufsc.br" );
    }
    catch ( java.io.FileNotFoundException e ) {
        System.out.println( e.toString() );
    }
    catch ( java.io.IOException e ) {
        System.out.println( e.toString() );
    }
}

public String getDirBase( ) {
    return _dirbase;
}

```

```

public String getVarCfg( String key ) {
    return _props.getProperty( key );
}

public void setVarCfg( String key, String value ) {
    _props.setProperty( key, value );
}

public boolean getStatus( ) {
    return this._tudocerto;
}

private Properties _props;
private boolean _tudocerto = true;
}

```

JSP

index.jsp

```

<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%@page language="java"%>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>DicProd Web</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<link rel="stylesheet" href="DicProd.css" type="text/css" />
</head>

<body bgcolor="#FFFFFF" text="#000000">
<table width="610" border="0" cellspacing="0" cellpadding="0" align="center"
class="pagina_fundo">
<tr>
<td>
<td>
<table width="600" border="0" cellspacing="0" cellpadding="0" align="center">
<tr>
<td class="header_fundo" align="center"><span class="header_titulo"><a
href="DicProdBemvindo.html" target="FrameResultados"
class="header_titulo">Dicionário
de Produções de Textos</a></span><br/>
<span class="header_subtitulo">Versão para a Web</span></td>
</tr>
<tr>
<td align="center">
<form action="DicProdResultados.jsp" method="POST" name="DicHtmlForm"
target="FrameResultados" id="DicHtmlForm">
<td class="corpo_fundo" align="center">
<table width="540" border="0" cellspacing="0" cellpadding="0">
<tr>
<td width="265">
<div align="center">
<input type="text" name="termo" class="form_text" value="" />

```



```

        <input type="submit" name="Submit" value="Procurar"
class="form_button" />
        </div>
    </td>
    <td width="235">
        <input type="radio" name="tipoPesquisa" value="fonte"
checked="checked" />
        <span class="form_caption">Pesquisar exemplos em
Portugu&ecirc;s</span><br />
        <input type="radio" name="tipoPesquisa" value="destino" />
        <span class="form_caption">Pesquisar exemplos em
Espanhol</span>
        <br />
        <input name="tipoPesquisa" type="radio" value="nucleo" />
        <span class="form_caption">Pesquisar apenas nos
n&uacute;cleos</span> </td>
    </tr>
    <tr>
        <td width="265">
            <div align="center">
                <input name="apenasExemplos" type="checkbox" value="false" />
                <span class="form_caption">Mostrar lemas
inteiros</span></div>
            </td>
            <td width="235">&nbsp;</td>
        </tr>
    </table>
</td>
</form>
</tr>
<tr>
    <td class="corpo_fundo" align="center">
        <iframe src="DicProdBemvindo.html" name="FrameResultados" width="600"
height="400" id="FrameResultados"></iframe>
    </td>
</tr>
<tr>
    <td class="header_fundo" align="center">&nbsp;</td>
</tr>
</table>
</td>
</tr>
</table>
</body>
</html>

```

DicProd.css

```

.header_titulo {
    font-family: Verdana, Arial, Helvetica, sans-serif;
    font-size: 16px;
    font-weight: bold;
    color: #333333;
    text-decoration: none;
}
.header_fundo { background-color: #999966}
.header_subtitulo { font-family: Arial, Helvetica, sans-serif; font-size: 10px;
color: #FFFFFF}
.pagina_fundo { background-color: #999966}
.corpo_fundo { background-color: #CCCC99}

```

```
.form_text { font-family: Verdana, Arial, Helvetica, sans-serif; font-size: 10px;
font-weight: bold}
.form_button { font-family: Verdana, Arial, Helvetica, sans-serif; font-size:
10px}
.form_caption { font-family: Verdana, Arial, Helvetica, sans-serif; font-size:
10px; font-weight: bold}
.linkapagado {
    text-decoration: none;
}
.corpotitulo {
    font-family: Arial, Helvetica, sans-serif;
    font-size: 14px;
    font-weight: bold;
}
.corpotexto {
    font-family: Verdana, Arial, Helvetica, sans-serif;
    font-size: 12px;
}
```

DicProdResultados.jsp

```
<%@ page contentType="text/html; charset=iso-8859-1" language="java" errorPage=""
%>
<jsp:useBean class="dicprodjb.Config" id="CONFIG" scope="page"></jsp:useBean>
<jsp:setProperty name="CONFIG" property="dirBase" value="<%=
application.getRealPath( "/" ) %>"/>
<jsp:useBean class="dicprodjb.Pesquisa" id="pesq" scope="page"></jsp:useBean>
<jsp:setProperty name="pesq" property="config" value="<%= CONFIG.getConfig( ) %>"/>
<jsp:setProperty name="pesq" property="*" />
<%= pesq.getResultadosHTML() %>
```

JAVABEANS

Config.java

```
package dicprodjb;

/*
 * DicionarioConfigBean.java
 *
 * Created on 23 de Janeiro de 2003, 04:03
 */

import java.beans.*;
import dicprod.dicionarioConfig;

/**
 *
 * @author Herbert
 */
```

```

public class Config {

    private diccionarioConfig _config;

    public Config( ) {

    }

    public void setDirBase( String dir ) {
        _config = new diccionarioConfig( dir );
    }

    public void setKeyCfg( String key ) {
        this._key = key;
    }

    public void setVarCfg( String value ) {
        if ( this._key != null ) {
            _config.setVarCfg( this._key, value );
        }
        else {
            _config.setVarCfg( "null", value );
        }
    }

    public String getVarCfg( ) {
        return _config.getVarCfg( this._key );
    }

    public boolean getStatus( ) {
        this._status = _config.getStatus();
        return this._status;
    }

    public diccionarioConfig getConfig( ) {
        return this._config;
    }

    private String _key;
    private boolean _status;

}

```

Pesquisa.java

```

package dicprodjb;

/*
 * DicionarioConfigBean.java
 *
 * Created on 23 de Janeiro de 2003, 04:03
 */

import java.beans.*;
import dicprod.*;

/**
 *

```

```

* @author Herbert
*/
public class Pesquisa {

    private String _termo = "";
    private dicionarioConfig _config;
    private dicionarioXMLPesquisa _pesq;
    private String _tipopesquisa = "nucleo";
    private boolean _apenasexemplos = true;
    private String _resultados = "";

    public Pesquisa() {

    }

    public void setConfig( dicionarioConfig cnf ) {
        this._config = cnf;
        _pesq = new dicionarioXMLPesquisa( true, true, this._config );
        _pesq.carregueIndicePersistente( );
        System.out.println( "Carregou pesquisa..." );
    }

    public void setTermo( String termo ) {
        this._termo = termo;
    }

    public String getTermo( ) {
        return this._termo;
    }

    public void setTipoPesquisa( String tipo ) {
        this._tipopesquisa = tipo;
    }

    public String getResultados( ) {
        if ( this._tipopesquisa.compareTo( "nucleo" ) != 0 ) {
            if ( this._apenasexemplos ) {
                if ( this._tipopesquisa.compareTo( "fonte" ) == 0 ) {
                    _pesq.busqueFonteExemplos( this._termo );
                }
                else {
                    _pesq.busqueDestinoExemplos( this._termo );
                }
            }
            else {
                if ( this._tipopesquisa.compareTo( "fonte" ) == 0 ) {
                    _pesq.busqueFonte( this._termo );
                }
                else {
                    _pesq.busqueDestino( this._termo );
                }
            }
        }
        else {
            _pesq.busqueNucleo( this._termo );
        }
        this._resultados = _pesq.getResultadosXML( );
        return this._resultados;
    }

    public String getResultadosHTML( ) {
        String res = getResultados( );
        String htmlres;
        dicionarioXMLResultadosGerente ger;

        if ( this._apenasexemplos ) {

```

```

        ger = new dicionarioXMLResultadosGerente( "exemplos", res );
        htmlres = ger.getHtmlExemplos( );
    }
    else {
        ger = new dicionarioXMLResultadosGerente( "lemas", res );
        htmlres = ger.getHtmlLemas( );
    }
    return "<html><body>" + htmlres + "</body></html>";
}

public void setApenasExemplos( boolean exm ) {
    this._apenasexemplos = exm;
}
}
}

```

IMPLEMENTAÇÃO EM JAVA SWING

DicProdGUI.java

```

/*
 * DicProdGUI.java
 *
 * Created on 15 de Janeiro de 2003, 01:40
 */

import dicprod.*;
import javax.swing.JFrame;

public class DicProdGUI extends javax.swing.JFrame {

    private dicionarioConfig CONFIG;
    private dicionarioXMLPesquisa _dicpes;

    /** Creates new form DicProdGUI */
    public DicProdGUI() {
        CONFIG = new dicionarioConfig( );
        System.out.println( CONFIG );
        _dicpes = new dicionarioXMLPesquisa( true, CONFIG );
        _dicpes.carregueIndicePersistente( );
        initComponents();
        this.jTextPaneAlerta.setVisible( false );
        this.jTextFieldBusca.requestFocus();
    }

    /** This method is called from within the constructor to
     * initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is
     * always regenerated by the Form Editor.
     */
    private void initComponents() { //GEN-BEGIN:initComponents

```

```

buttonGroupOpcoes = new javax.swing.ButtonGroup();
jTextPaneAlerta = new javax.swing.JTextPane();
jTextFieldBusca = new javax.swing.JTextField();
jButtonPesquisar = new javax.swing.JButton();
jRadioButtonLema = new javax.swing.JRadioButton();
jRadioButtonFonte = new javax.swing.JRadioButton();
jRadioButtonDestino = new javax.swing.JRadioButton();
jCheckBoxLemas = new javax.swing.JCheckBox();
jScrollPane1 = new javax.swing.JScrollPane();
jEditorPaneResultados = new javax.swing.JEditorPane();
ESPACADOR = new javax.swing.JLabel();

getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

setTitle("DicProd");
setResizable(false);
addWindowListener(new java.awt.event.WindowAdapter() {
    public void windowClosing(java.awt.event.WindowEvent evt) {
        exitForm(evt);
    }
});

jTextPaneAlerta.setBackground(new java.awt.Color(204, 204, 204));
jTextPaneAlerta.setBorder(new javax.swing.border.TitledBorder(null,
"Pesquisando", javax.swing.border.TitledBorder.CENTER,
javax.swing.border.TitledBorder.ABOVE_TOP, new java.awt.Font("Dialog", 1, 12)));
jTextPaneAlerta.setEditable(false);
jTextPaneAlerta.setText(" ");
getContentPane().add(jTextPaneAlerta, new
org.netbeans.lib.awtextra.AbsoluteConstraints(30, 90, 320, 120));

jTextFieldBusca.addFocusListener(new java.awt.event.FocusAdapter() {
    public void focusGained(java.awt.event.FocusEvent evt) {
        jTextFieldBuscaFocusGained(evt);
    }
});

jTextFieldBusca.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        jTextFieldBuscaKeyTyped(evt);
    }
});

getContentPane().add(jTextFieldBusca, new
org.netbeans.lib.awtextra.AbsoluteConstraints(10, 10, 190, -1));

jButtonPesquisar.setText("Pesquisar");
jButtonPesquisar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButtonPesquisarActionPerformed(evt);
    }
});

getContentPane().add(jButtonPesquisar, new
org.netbeans.lib.awtextra.AbsoluteConstraints(210, 10, 100, 20));

jRadioButtonLema.setText("Pesquisar apenas n\u00facleos (portugu\u00eas)");
buttonGroupOpcoes.add(jRadioButtonLema);
jRadioButtonLema.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jRadioButtonLemaActionPerformed(evt);
    }
});

getContentPane().add(jRadioButtonLema, new
org.netbeans.lib.awtextra.AbsoluteConstraints(340, 40, -1, -1));

```

```

jRadioButtonFonte.setSelected(true);
jRadioButtonFonte.setText("Pesquisar exemplos em portugu\u00eas");
buttonGroupOpcoes.add(jRadioButtonFonte);
jRadioButtonFonte.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jRadioButtonFonteActionPerformed(evt);
    }
});

getContentPane().add(jRadioButtonFonte, new
org.netbeans.lib.awtextra.AbsoluteConstraints(340, 0, 270, -1));

jRadioButtonDestino.setText("Pesquisar exemplos em espanhol");
buttonGroupOpcoes.add(jRadioButtonDestino);
jRadioButtonDestino.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jRadioButtonDestinoActionPerformed(evt);
    }
});

getContentPane().add(jRadioButtonDestino, new
org.netbeans.lib.awtextra.AbsoluteConstraints(340, 20, -1, -1));

jCheckBoxLemas.setText("Mostrar resultados com os n\u00f3s");
getContentPane().add(jCheckBoxLemas, new
org.netbeans.lib.awtextra.AbsoluteConstraints(10, 40, -1, -1));

jEditorPaneResultados.setContentType("text/html");
jEditorPaneResultados.setAutoscrolls(false);
jScrollPane.setViewportView(jEditorPaneResultados);

getContentPane().add(jScrollPane, new
org.netbeans.lib.awtextra.AbsoluteConstraints(10, 70, 710, 440));

getContentPane().add(ESPACADOR, new
org.netbeans.lib.awtextra.AbsoluteConstraints(650, 510, 80, 10));

pack();
} //GEN-END: initComponents

private void jTextFieldBuscaFocusGained(java.awt.event.FocusEvent evt) { //GEN-
FIRST:event_jTextFieldBuscaFocusGained
    this.jTextFieldBusca.selectAll();
} //GEN-LAST:event_jTextFieldBuscaFocusGained

private void jButtonConfigurarActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButtonConfigurarActionPerformed
    dicionarioGUIConfig diccfg = new dicionarioGUIConfig( this, true, _dicpes );
    diccfg.show();
} //GEN-LAST:event_jButtonConfigurarActionPerformed

private void jTextFieldBuscaKeyTyped(java.awt.event.KeyEvent evt) { //GEN-
FIRST:event_jTextFieldBuscaKeyTyped
    if ( ( evt.getKeyChar() == '\n' ) || ( evt.getKeyChar() == '\r' ) ) {
        this.jTextFieldBusca.setText( this.jTextFieldBusca.getText().trim() );
        facaBusca( );
    }
} //GEN-LAST:event_jTextFieldBuscaKeyTyped

private void jRadioButtonLemaActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jRadioButtonLemaActionPerformed
    jCheckBoxLemas.setEnabled(false);
} //GEN-LAST:event_jRadioButtonLemaActionPerformed

private void jRadioButtonDestinoActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jRadioButtonDestinoActionPerformed
    jCheckBoxLemas.setEnabled(true);
}

```

```

} //GEN-LAST:event_jRadioButtonDestinoActionPerformed

private void jRadioButtonFonteActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jRadioButtonFonteActionPerformed
    jCheckBoxLemas.setEnabled(true);
} //GEN-LAST:event_jRadioButtonFonteActionPerformed

void mostraResultados( String resultados ) {
    diccionarioXMLResultadosGerente ger;
    String htmlres;

    if ( !this.jRadioButtonLema.isSelected() ) {
        if ( !this.jCheckBoxLemas.isSelected() ) {
            // mostrando exemplos
            ger = new diccionarioXMLResultadosGerente( "exemplos", resultados );
            htmlres = ger.getHtmlExemplos( );
        }
        else {
            // mostrando lemas
            ger = new diccionarioXMLResultadosGerente( "lemas", resultados );
            htmlres = ger.getHtmlLemas( );
        }
    }
    else {
        // mostrando lemas
        ger = new diccionarioXMLResultadosGerente( "lemas", resultados );
        htmlres = ger.getHtmlLemas( );
    }
    this.jEditorPaneResultados.setText( htmlres );
}

void mostraErro( String Erro ) {
    this.jEditorPaneResultados.setText( "<html><body><b>" + Erro +
"</b></html></body>" );
}

void mostraAlerta( String Alerta ) {
    this.jTextPaneAlerta.setText( Alerta );
    this.jTextPaneAlerta.setVisible( true );
    this.jTextPaneAlerta.update( this.jTextPaneAlerta.getGraphics() );
}

void escondeAlerta( ) {
    this.jEditorPaneResultados.requestFocus();
    this.jTextPaneAlerta.setVisible( false );
    this.jTextPaneAlerta.setText( "" );
}

void facaBusca( ) {
    String expr = this.jTextFieldBusca.getText();
    String iniAlerta = "- Procurando \"" + expr + "\"\n";
    String fimAlerta = "\n\nPor favor, aguarde...";

    if ( expr.length() != 0 ) {
        if ( this.jRadioButtonLema.isSelected() ) {
            mostraAlerta( iniAlerta + "- Pesquisando núcleos...\n- Mostrando lemas
inteiros..." + fimAlerta );
            _dicpes.busqueNucleo( expr );
        }
        else if ( this.jRadioButtonFonte.isSelected( ) ) {
            if ( !this.jCheckBoxLemas.isSelected() ) {
                mostraAlerta( iniAlerta + "- Pesquisando no idioma fonte...\n- Mostrando
apenas exemplos..." + fimAlerta );
                _dicpes.busqueFonteExemplos( expr );
            }
            else {

```



```

        mostraAlerta( iniAlerta + "- Pesquisando no idioma fonte...\n-
Mostrando lemas inteiros..." + fimAlerta );
        _dicpes.busqueFonte( expr );
    }
}
else if ( this.jRadioButtonDestino.isSelected( ) ) {
    if ( !this.jCheckBoxLemas.isSelected( ) ) {
        mostraAlerta( iniAlerta + "- Pesquisando no idioma destino...\n-
Mostrando apenas exemplos..." + fimAlerta );
        _dicpes.busqueDestinoExemplos( expr );
    }
    else {
        mostraAlerta( iniAlerta + "- Pesquisando no idioma destino...\n-
Mostrando lemas inteiros..." + fimAlerta );
        _dicpes.busqueDestino( expr );
    }
}

if ( _dicpes.getResultados().compareTo( "-- no hits --" ) != 0 ) {
    // _dicpes.salveResultados();
    mostraResultados( _dicpes.getResultadosXML() );
}
else {
    mostraErro( "<b>Nenhum exemplo encontrado para: </b><i>" + expr +
"</i><br><br><b>Por favor, refine a busca.</b>" );
}
escondeAlerta( );
}
else {
    mostraErro( "Por favor, digite um termo para a pesquisa." );
}
this.jTextFieldBusca.requestFocus();
}

private void jButtonPesquisarActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButtonPesquisarActionPerformed
    facaBusca( );
} //GEN-LAST:event_jButtonPesquisarActionPerformed

/** Exit the Application */
private void exitForm(java.awt.event.WindowEvent evt) { //GEN-FIRST:event_exitForm
    System.exit(0);
} //GEN-LAST:event_exitForm

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    new DicProdGUI().show();
}

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JEditorPane jEditorPaneResultados;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTextPane jTextPaneAlerta;
private javax.swing.JRadioButton jRadioButtonLema;
private javax.swing.JRadioButton jRadioButtonFonte;
private javax.swing.ButtonGroup buttonGroupOpcoes;
private javax.swing.JTextField jTextFieldBusca;
private javax.swing.JLabel ESPACADOR;
private javax.swing.JCheckBox jCheckBoxLemas;
private javax.swing.JRadioButton jRadioButtonDestino;
private javax.swing.JButton jButtonPesquisar;

```

```

// End of variables declaration//GEN-END:variables
}

```

DicProdGUIConfig.java

```

/*
 * dicionarioGUIConfig.java
 *
 * Created on 19 de Janeiro de 2003, 04:41
 */

import javax.swing.JFileChooser;
import javax.swing.JOptionPane;
import dicprod.dicionarioConfig;
import java.io.File;

public class dicionarioGUIConfig extends javax.swing.JDialog {

    private dicionarioConfig CONFIG;
    private dicprod.dicionarioXMLPesquisa _pesq;
    boolean _GUI = false;
    private String _currrdir = "";

    // TRES CONSTRUCTORS DIFERENTES
    public dicionarioGUIConfig ( java.awt.Frame parent, boolean modal ) {
        super (parent, modal);
        this.CONFIG = new dicionarioConfig ( );
        initComponents( );
        iniciaConfig( );
    }
    public dicionarioGUIConfig ( java.awt.Frame parent, boolean modal,
dicionarioConfig cfg ) {
        super (parent, modal);
        this.CONFIG = cfg;
        initComponents( );
        iniciaConfig( );
    }
    public dicionarioGUIConfig (java.awt.Frame parent, boolean modal,
dicprod.dicionarioXMLPesquisa pesq ) {
        super (parent, modal);
        this._pesq = pesq;
        this.CONFIG = _pesq.getConfig( );
        _GUI = true;
        initComponents ( );
        iniciaConfig( );
    }
}

void iniciaConfig( ) {
    if ( !CONFIG.getStatus( ) ) {
        // carrega com valores padrão

```

```

        CONFIG.setVarCfg ( "dic.dicionario", CONFIG.getVarCfg ( "dirbase" ) +
"volume.xml" );
        CONFIG.setVarCfg ( "xqe.stoplist", CONFIG.getVarCfg ( "dirbase" ) +
"stoplist.txt" );
        CONFIG.setVarCfg ( "xqe.indexpathname", CONFIG.getVarCfg ( "dirbase" ) +
"indice" + File.separatorChar + "indice" );
        salveConfiguracaoAtual ( );
    }
    else {
        this.jTextFieldDicDicionario.setText ( CONFIG.getVarCfg ( "dic.dicionario" )
);
        this.jTextFieldXqeStopList.setText ( CONFIG.getVarCfg ( "xqe.stoplist" ) );
        File arqind = new File ( CONFIG.getVarCfg ( "xqe.indexpathname" ) );
        this.jTextFieldDirBaseIndice.setText ( arqind.getParent ( ) );
        // this.jTextFieldNomeBaseIndice.setText ( arqind.getName ( ) );
        if ( configuracaoAtualValida ( ) ) {
            this.jButtonReindexar.setEnabled ( true );
        }
    }
}

boolean configuracaoAtualValida ( ) {
    File dicdic = new File (CONFIG.getVarCfg ( "dic.dicionario" ));
    File stplst = new File (CONFIG.getVarCfg ( "xqe.stoplist" ));
    File idxptn = new File (CONFIG.getVarCfg ( "xqe.indexpathname" ));
    File idxptndir = new File (idxptn.getParent ( ));
    if ( ( !dicdic.exists ( ) || !dicdic.canRead ( ) )
|| ( !stplst.exists ( ) || !stplst.canRead ( ) )
|| ( !idxptndir.exists ( ) || !idxptndir.canWrite ( ) || !idxptndir.isDirectory
( ) )
) {
        return false;
    }
    else {
        return true;
    }
}

void salveNoConfig( ) {
    CONFIG.setVarCfg ( "dic.dicionario", this.jTextFieldDicDicionario.getText ( )
);
    CONFIG.setVarCfg ( "xqe.stoplist", this.jTextFieldXqeStopList.getText ( ) );
    CONFIG.setVarCfg ( "xqe.indexpathname", this.jTextFieldDirBaseIndice.getText (
) + File.separatorChar + "Indice" );
}

void salveConfiguracaoAtual ( ) {
    salveNoConfig();
    if ( configuracaoAtualValida ( ) ) {
        CONFIG.salveConfig ( );
        this.jButtonReindexar.setEnabled ( true );
    }
    else {
        this.jButtonReindexar.setEnabled ( false );
        JOptionPane.showMessageDialog ( this, "Existem erros na configuração!
Verifique se os arquivos podem ser encontrados e que o diretório de índices é
aberto para escrita.", "Erro ao salvar configurações", JOptionPane.WARNING_MESSAGE
);
    }
}

String procureDiretorio ( String valvelho, String descr ) {
    JFileChooser chooser = new JFileChooser ( );

```

```

chooser.setFileSelectionMode ( JFileChooser.DIRECTORIES_ONLY );

File dir;

dir = new File ( valvelho );
if ( dir.exists ( ) ) {
    chooser.setCurrentDirectory( dir );
}
else {
    chooser.setCurrentDirectory( new File( _currdir ) );
}

int returnVal = chooser.showOpenDialog (this);
_currdir = chooser.getCurrentDirectory( ).getAbsolutePath( );
if ( returnVal == JFileChooser.APPROVE_OPTION ) {
    return chooser.getSelectedFile().getAbsolutePath( );
}
else {
    return valvelho;
}
}

String procureArquivoXML ( String valvelho, String descr ) {
    JFileChooser chooser = new JFileChooser ();
    ExampleFileFilter filter = new ExampleFileFilter ( );
    filter.addExtension ( "xml" );
    filter.setDescription ( descr );
    chooser.setFileFilter ( filter );

    File arq;

    arq = new File ( valvelho );
    if ( arq.exists ( ) ) {
        chooser.setSelectedFile ( arq );
    }
    else {
        chooser.setCurrentDirectory( new File( _currdir ) );
    }

    int returnVal = chooser.showOpenDialog (this);
    if ( returnVal == JFileChooser.APPROVE_OPTION ) {
        _currdir = chooser.getCurrentDirectory( ).getParent( );
        return chooser.getSelectedFile ( ).getAbsolutePath ( );
    }
    else {
        _currdir = chooser.getCurrentDirectory( ).getAbsolutePath( );
        return valvelho;
    }
}

String procureArquivoTxt ( String valvelho, String descr ) {
    JFileChooser chooser = new JFileChooser();
    ExampleFileFilter filter = new ExampleFileFilter( );
    filter.addExtension ( "txt" );
    filter.setDescription ( descr );
    chooser.setFileFilter ( filter );

    File arq;

    arq = new File ( valvelho );
    if ( arq.exists ( ) ) {
        chooser.setSelectedFile ( arq );
    }
    else {
        chooser.setCurrentDirectory( new File( _currdir ) );
    }
}

```

```

int returnVal = chooser.showOpenDialog (this);
if ( returnVal == JFileChooser.APPROVE_OPTION ) {
    _currrdir = chooser.getCurrentDirectory( ).getParent( );
    return chooser.getSelectedFile ( ).getAbsolutePath ( );
}
else {
    _currrdir = chooser.getCurrentDirectory( ).getAbsolutePath( );
    return valvelho;
}
}

/** This method is called from within the constructor to
 * initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is
 * always regenerated by the Form Editor.
 */
private void initComponents() { //GEN-BEGIN:initComponents
    java.awt.GridBagConstraints gridBagConstraints;

    jTextFieldDicDicionario = new javax.swing.JTextField();
    jButtonProcurarDicDicionario = new javax.swing.JButton();
    jTextFieldXqeStopList = new javax.swing.JTextField();
    jButtonProcurarXqeStopList = new javax.swing.JButton();
    jTextFieldDirBaseIndice = new javax.swing.JTextField();
    jButtonProcurarDirBaseIndice = new javax.swing.JButton();
    jLabelPastadoIndice = new javax.swing.JLabel();
    jButtonReindexar = new javax.swing.JButton();
    jLabelXqeStopList = new javax.swing.JLabel();
    jLabelDicDicProd = new javax.swing.JLabel();
    jButtonSalvar = new javax.swing.JButton();
    jButtonFechar = new javax.swing.JButton();

    getContentPane().setLayout(new java.awt.GridBagLayout());

    setTitle("DicProd Vers\u00e3o TCC: Configura\u00e7\u00e3o");
    setResizable(false);
    addWindowListener(new java.awt.event.WindowAdapter() {
        public void windowClosing(java.awt.event.WindowEvent evt) {
            closeDialog(evt);
        }
    });

    jTextFieldDicDicionario.setToolTipText("Arquivo XML definindo o dicion\u00e1rio
atual");
    gridBagConstraints = new java.awt.GridBagConstraints();
    gridBagConstraints.gridx = 0;
    gridBagConstraints.gridy = 1;
    gridBagConstraints.gridwidth = 6;
    gridBagConstraints.fill = java.awt.GridBagConstraints.BOTH;
    gridBagConstraints.ipadx = 376;
    gridBagConstraints.insets = new java.awt.Insets(4, 10, 0, 0);
    getContentPane().add(jTextFieldDicDicionario, gridBagConstraints);

    jButtonProcurarDicDicionario.setText("Procurar...");
    jButtonProcurarDicDicionario.setToolTipText("Procurar arquivo");
    jButtonProcurarDicDicionario.addActionListener(new
java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButtonProcurarDicDicionarioActionPerformed(evt);
        }
    });

    gridBagConstraints = new java.awt.GridBagConstraints();
    gridBagConstraints.gridx = 6;
    gridBagConstraints.gridy = 1;
    gridBagConstraints.gridwidth = 2;

```

```

gridBagConstraints.fill = java.awt.GridBagConstraints.BOTH;
gridBagConstraints.ipady = -6;
gridBagConstraints.insets = new java.awt.Insets(4, 1, 0, 0);
getContentPane().add(jButtonProcurarDicDicionario, gridBagConstraints);

jTextFieldXqeStopList.setToolTipText("Arquivo com a lista de palavras que
ser\u00e3o exclu\u00eddas da pesquisa");
gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 0;
gridBagConstraints.gridy = 4;
gridBagConstraints.gridwidth = 6;
gridBagConstraints.fill = java.awt.GridBagConstraints.BOTH;
gridBagConstraints.ipadx = 376;
gridBagConstraints.insets = new java.awt.Insets(4, 10, 0, 0);
getContentPane().add(jTextFieldXqeStopList, gridBagConstraints);

jButtonProcurarXqeStopList.setText("Procurar...");
jButtonProcurarXqeStopList.setToolTipText("Procurar arquivo");
jButtonProcurarXqeStopList.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButtonProcurarXqeStopListActionPerformed(evt);
    }
});

gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 6;
gridBagConstraints.gridy = 4;
gridBagConstraints.gridwidth = 2;
gridBagConstraints.fill = java.awt.GridBagConstraints.BOTH;
gridBagConstraints.ipady = -6;
gridBagConstraints.insets = new java.awt.Insets(4, 1, 0, 0);
getContentPane().add(jButtonProcurarXqeStopList, gridBagConstraints);

jTextFieldDirBaseIndice.setToolTipText("Pasta onde ser\u00e3o guardados os
arquivos do \u00edndice");
gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 0;
gridBagConstraints.gridy = 7;
gridBagConstraints.gridwidth = 6;
gridBagConstraints.fill = java.awt.GridBagConstraints.BOTH;
gridBagConstraints.ipadx = 376;
gridBagConstraints.insets = new java.awt.Insets(4, 10, 0, 0);
getContentPane().add(jTextFieldDirBaseIndice, gridBagConstraints);

jButtonProcurarDirBaseIndice.setText("Procurar...");
jButtonProcurarDirBaseIndice.setToolTipText("Procurar diret\u00f3rio");
jButtonProcurarDirBaseIndice.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButtonProcurarDirBaseIndiceActionPerformed(evt);
    }
});

gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 6;
gridBagConstraints.gridy = 7;
gridBagConstraints.gridwidth = 2;
gridBagConstraints.fill = java.awt.GridBagConstraints.BOTH;
gridBagConstraints.ipady = -6;
gridBagConstraints.insets = new java.awt.Insets(4, 1, 0, 0);
getContentPane().add(jButtonProcurarDirBaseIndice, gridBagConstraints);

jLabelPastadoIndice.setText("Pasta para o \u00edndice:");
gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 0;
gridBagConstraints.gridy = 6;

```

```

gridBagConstraints.fill = java.awt.GridBagConstraints.BOTH;
gridBagConstraints.insets = new java.awt.Insets(10, 10, 0, 0);
getContentPane().add(jLabelPastadoIndice, gridBagConstraints);

jButtonReindexar.setText("Re-indexar...");
jButtonReindexar.setToolTipText("Remove as refer\u00eancias a um \u00edndice
pre-existente, \ne cria um novo \u00edndice.");
jButtonReindexar.setEnabled(false);
jButtonReindexar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButtonReindexarActionPerformed(evt);
    }
});

gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 0;
gridBagConstraints.gridy = 10;
gridBagConstraints.fill = java.awt.GridBagConstraints.BOTH;
gridBagConstraints.insets = new java.awt.Insets(37, 110, 5, 1);
getContentPane().add(jButtonReindexar, gridBagConstraints);

jLabelXqeStopList.setText("Arquivo de palavras exclu\u00eddas:");
gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 0;
gridBagConstraints.gridy = 3;
gridBagConstraints.gridwidth = 4;
gridBagConstraints.fill = java.awt.GridBagConstraints.BOTH;
gridBagConstraints.insets = new java.awt.Insets(10, 10, 0, 0);
getContentPane().add(jLabelXqeStopList, gridBagConstraints);

jLabelDicDicProd.setText("Arquivo do dicion\u00e1rio:");
gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 0;
gridBagConstraints.gridy = 0;
gridBagConstraints.gridwidth = 2;
gridBagConstraints.fill = java.awt.GridBagConstraints.BOTH;
gridBagConstraints.insets = new java.awt.Insets(20, 10, 0, 0);
getContentPane().add(jLabelDicDicProd, gridBagConstraints);

jButtonSalvar.setText("Salvar...");
jButtonSalvar.setToolTipText("Salva configura\u00e7\u00f5es");
jButtonSalvar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButtonSalvarActionPerformed(evt);
    }
});

gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 5;
gridBagConstraints.gridy = 10;
gridBagConstraints.fill = java.awt.GridBagConstraints.BOTH;
gridBagConstraints.insets = new java.awt.Insets(37, 110, 5, 1);
getContentPane().add(jButtonSalvar, gridBagConstraints);

jButtonFechar.setText("Fechar");
jButtonFechar.setToolTipText("Fecha a janela de configura\u00e7\u00e3o");
jButtonFechar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButtonFecharActionPerformed(evt);
    }
});

gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 7;
gridBagConstraints.gridy = 10;
gridBagConstraints.fill = java.awt.GridBagConstraints.BOTH;
gridBagConstraints.insets = new java.awt.Insets(37, 11, 5, 1);

```

```

        getContentPane().add(jButtonFechar, gridBagConstraints);

        pack();
    } //GEN-END: initComponents

    private void jButtonSalvarActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_jButtonSalvarActionPerformed
        salveConfiguracaoAtual ( );
    } //GEN-LAST:event_jButtonSalvarActionPerformed

    private void jButtonFecharActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_jButtonFecharActionPerformed
        if ( _GUI ) {
            setVisible ( false );
            dispose ( );
        }
        else {
            System.exit ( 0 );
        }
    } //GEN-LAST:event_jButtonFecharActionPerformed

    public void apagueIndiceAtual ( ) {
        salveNoConfig ( );
        if ( configuracaoAtualValida ( ) ) {
            File arq1 = new File(CONFIG.getVarCfg ( "xqe.indexpathname" ) + ".trees");
            File arq2 = new File(CONFIG.getVarCfg ( "xqe.indexpathname" ) + ".xref");
            File arq3 = new File(CONFIG.getVarCfg ( "xqe.indexpathname" ) + ".dict");
            if ( arq1.exists ( ) ) {
                arq1.delete ( );
            }
            if ( arq2.exists ( ) ) {
                arq2.delete ( );
            }
            if ( arq3.exists ( ) ) {
                arq3.delete ( );
            }
        }
    }

    private void jButtonReindexarActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_jButtonReindexarActionPerformed
        // apagueIndiceAtual ( );
        _pesq = new dicprod.dicionarioXMLPesquisa ( false, CONFIG );
        _pesq.crieIndicePersistente ( );
    } //GEN-LAST:event_jButtonReindexarActionPerformed

    private void
    jButtonProcurarDirBaseIndiceActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_jButtonProcurarDirBaseIndiceActionPerformed
        this.jButtonReindexar.setEnabled ( false );
        this.jTextFieldDirBaseIndice.setText ( procureDiretorio (
        this.jTextFieldDirBaseIndice.getText ( ), "Pasta para os arquivos de índice" ) );
    } //GEN-LAST:event_jButtonProcurarDirBaseIndiceActionPerformed

    private void jButtonProcurarXqeStopListActionPerformed(java.awt.event.ActionEvent
    evt) { //GEN-FIRST:event_jButtonProcurarXqeStopListActionPerformed
        this.jButtonReindexar.setEnabled ( false );
        this.jTextFieldXqeStopList.setText ( procureArquivoTxt (
        this.jTextFieldXqeStopList.getText ( ), "Arquivo com termos excluídos" ) );
    } //GEN-LAST:event_jButtonProcurarXqeStopListActionPerformed

    private void
    jButtonProcurarDicDicionarioActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_jButtonProcurarDicDicionarioActionPerformed

```



```

        this.jButtonReindexar.setEnabled ( false );
        this.jTextFieldDicDicionario.setText ( procureArquivoXML (
this.jTextFieldDicDicionario.getText (), "Arquivo de Dicionário de Produção" ) );
    }//GEN-LAST:event_jButtonProcurarDicDicionarioActionPerformed

    /** Closes the dialog */
    private void closeDialog(java.awt.event.WindowEvent evt) {//GEN-
FIRST:event_closeDialog
        if ( _GUI ) {
            setVisible (false);
            dispose ();
        }
        else {
            System.exit (0);
        }
    }//GEN-LAST:event_closeDialog

    /**
     * @param args the command line arguments
     */
    public static void main (String args[]) {
        new dicionarioGUIConfig (new javax.swing.JFrame (), true ).show ();
    }

    // Variables declaration - do not modify//GEN-BEGIN:variables
    private javax.swing.JButton jButtonFechar;
    private javax.swing.JTextField jTextFieldDicDicionario;
    private javax.swing.JButton jButtonProcurarDicDicionario;
    private javax.swing.JButton jButtonProcurarXqeStopList;
    private javax.swing.JButton jButtonProcurarDirBaseIndice;
    private javax.swing.JLabel jLabelXqeStopList;
    private javax.swing.JButton jButtonSalvar;
    private javax.swing.JTextField jTextFieldDirBaseIndice;
    private javax.swing.JTextField jTextFieldXqeStopList;
    private javax.swing.JButton jButtonReindexar;
    private javax.swing.JLabel jLabelPastadoIndice;
    private javax.swing.JLabel jLabelDicDicProd;
    // End of variables declaration//GEN-END:variables
}

```