O Processo de Desenvolvimento da BDL-NILC *

Juliana Galvani Greghi
Ronaldo Teixeira Martins
Maria das Graças Volpe Nunes
NILC- ICMC/USP-São Carlos

Resumo

Neste trabalho são relatados os passos da construção de uma base de dados lexicais para o português do Brasil, que deverá servir como repositório centralizador de informações. Além da base de dados, é documentado o processo de desenvolvimento das interfaces e ferramentas de acesso aos dados, divididos em quatro módulos: a) interface de consultas via web, b) interface gráfica do thesaurus, c) ferramenta de edição/inserção dos dados e d) ferramenta de geração de listas especializadas.

1 Introdução

O objetivo deste relatório é reportar as premissas que direcionaram a organização da BDL-NILC, base de dados lexicais para a língua portuguesa desenvolvida pelo Núcleo Interinstitucional de Lingüística Computacional (NILC). Além disso, este relatório deve registrar, documentar e discutir os detalhes do processo de desenvolvimento e implementação da BDL, já que tal processo não foi simples e as decisões tomadas devem ficar registradas para o bom gerenciamento da BDL daqui para frente.

As fases de desenvolvimento do projeto são apresentadas na Seção 2 e as características da implementação são apresentadas na Seção 3. As Seções 4, 5, 6 e 7 trazem informações sobre o desenvolvimento da interface de consulta via Web, da ferramenta de geração de listas especializadas, da interface gráfica do TeP e da ferramenta de edição/inserção de dados, respectivamente. Este documento apresenta, ainda, três anexos: I – Descrição das Tabelas da Base de Dados TotalLex, II – Descrição da Implementação da Interface de Consultas via Web, e III – Descrição da Implementação da Ferramenta de Geração de listas Especializadas.

2 A BDL: da Modelagem à Implementação

A construção da base de dados lexicais foi dividida em 3 etapas:

- Migração dos dados do léxico do ReGra
- Migração dos dados do Thesaurus
- Migração dos dados do dicionário UNL

* Trabalho desenvolvido com apoio da Fapesp (#00/03294-5), CNPq(#301365-91.1)

Como a terceira etapa ainda não foi concluída, aqui serão detalhadas somente as duas primeiras etapas de desenvolvimento da BDL.

2.1 Migração dos dados do léxico do ReGra

O ReGra é um revisor gramatical para o português do Brasil e conta com um léxico com mais de 1,5 milhão de palavras, incluindo nomes próprios e abreviações. O ReGra está disponível como parte do produto Redação Língua Portuguesa, da Itautec/Philco S.A., podendo também ser acoplado ao *Microsoft Word* como uma ferramenta independente, e como parte do MSOffice 2000, versão português.

Esta primeira etapa foi dividida em 7 tarefas:

a) Análise dos dados do léxico
b) Elaboração do Modelo Lingüístico
c) Escolha do Modelo de Dados a ser usado
d) Modelagem Computacional segundo o Modelo de Dados escolhido na etapa anterior
e) Escolha do Sistema de Gerenciamento de Banco de Dados
f) Implementação da Base de Dados
g) Inserção dos dados na base

a) Análise dos dados do léxico: O léxico originalmente usado no revisor gramatical ReGra é armazenado em um arquivo tipo texto e os dados referentes a um item lexical são expressos em uma cadeia única, com as informações separadas por caracteres especiais. Cada entrada é constituída de uma palavra ou, no máximo, palavras compostas hifenizadas. É importante ressaltar o papel da forma canônica, presente em toda entrada, que tem a função de ligar toda palavra à forma básica que lhe deu origem. Com isso, possibilita-se recuperar as várias flexões e derivações de uma mesma forma básica. Assim, *menina, meninas, meninos*, por exemplo, estão todas ligadas à forma canônica *menino* e, conseqüentemente, todas ligadas entre si com seus atributos. Todas as entradas desse léxico estarão presentes na BDL.

A seguir, são apresentados alguns exemplos de entradas presentes no léxico.

dado=<ADJ.M.SI.N.[a.com.contra.em.por.]?.?.[dado]4.#V.[][PARTIC.M.SI.]N.[][dar]4.>
mundo=<S.M.SI.N.[]?.?.[mundo]0.>

Cada classe gramatical presente no léxico foi analisada separadamente, verificando qual o conjunto de dados expresso para cada uma e como esses dados foram ordenados na cadeia de informações. Esse estudo foi realizado com a análise do léxico e com a descrição detalhada do léxico apresentada em (NUNES ET AL, 1996).

b) Elaboração do Modelo Lingüístico: A BDL-NILC deriva da reorganização de pelo menos duas outras bases de dados existentes no Núcleo, que cumpriam finalidades muito diversas: o léxico do ReGra - ferramenta de revisão gramatical automática[1]; e o dicionário Português-UNL, que servia a um projeto de tradução automática multilingual baseada em interlíngua[2]. Essas duas bases, por extremamente dependentes das aplicações a que serviam, possuíam estrutura bastante diversificada: no caso do léxico do ReGra, dicionarizaram-se todas as cadeias mínimas de caracteres isoladas por espaços em branco que poderiam pontuar nos textos de língua portuguesa, ainda que não fossem abonadas por grandes dicionários da língua (caso dos neologismos) e ainda que não constituíssem, isoladamente, palavras (caso dos elementos de composição de lexias complexas, como "Sri" e "Lanka", dicionarizados separadamente); no caso do dicionário Português-UNL, foram registradas apenas as cadeias de caracteres recorrentes que pudessem estar associadas a alguma das unidades semânticas da interlíngua adotada, pouco importando a presença ou a ausência de espaços em branco (caso de "há muito tempo", dicionarizado como um único verbete) ou a consistência morfológica da representação (caso de "fi", registrado para permitir a formação das formas do verbo "ficar", dada a alomorfia do radical {fik}, em que o segmento final poderia ser atualizado ora como "c", em "ficou", ora como "qu", em "fiquem"). Essas configurações foram em grande parte determinadas pela estrutura das ferramentas em que eram empregadas: no caso do léxico do ReGra, a representação de unidades morfológicas menores que a palavra autorizaria casos de

[1] O Projeto ReGra vem sendo desenvolvido desde 1996 com auxílio de várias agências de fomento brasileiras (FINEP, FAPESP), em parceria com a iniciativa privada. Seus resultados podem ser observados no conjunto de ferramentas de auxílio à escrita Redação Língua Portuguesa (RLP) e nos aplicativos de revisão ortográfica e gramatical incorporados ao editor de textos Word, da MicroSoft, versão 2000 em diante. Para maiores informações sobre o ReGra, consulte-se (NUNES ET Al., 1996)

[2] Universal Networking Language (UNL) é uma interlíngua eletrônica desenvolvida para comportar a representação de informação de origem multilíngüe. Conforma a base do Projeto UNL, coordenado pelo IAS/UNU e pela UNDL Foundation, que subsidiaram a construção da base lexical correspondente para o português brasileiro. Para maiores informações sobre a representação UNL e o Projeto UNL consulte-se (UCHIDA ET AL, 1999).

derivação, composição e flexão que, embora pertinentes aos processos de criação lexical da língua (como em "casação", "pãos" e "neurocadeira"), não deveriam ser tolerados por uma ferramenta (necessariamente conservadora) de revisão gramatical[3]; no caso do dicionário Português-UNL, o alto custo do *backtracking* no processo de linearização da sentença indicava a necessidade de serem gerados, no primeiro momento, apenas os morfemas lexicais, para que os morfemas gramaticais (como os sufixos flexionais) pudessem lhes ser posteriormente justapostos.

A diferenciação entre os dois modelos implicava, porém, não apenas a duplicação de informações morfossintáticas, mas igualmente a duplicação de esforços na manutenção das bases lexicais. Não se revela razoável que alterações do conjunto de informações dos verbetes tenham que ser realizadas duas vezes, principalmente se considerado o altíssimo custo dessas intervenções, quase sempre feitas manualmente por pessoal especializado. Cedo se instalou, portanto, não apenas o desejo, mas principalmente a necessidade de fazer as duas bases de dados convergirem, sem que isso implicasse a perda das especificidades inerentes a cada uma das finalidades a que se prestavam. O desafio se revelava tanto mais insuperável porque a simples junção das entradas compiladas em uma e outra base de dados lexicais afetaria o desempenho de ambas as ferramentas, na medida em que instalaria contradições (sob a forma de novos casos de homografia) para as quais não poderiam ser previstas estratégias de desambigüização.

Como tentativa de fusão das bases decidiu-se por um modelo híbrido, caracterizado pela convivência de duas diferentes estruturas de dados e cujo ponto de interseção seriam os verbetes do dicionário. Haveria uma estrutura do tipo rede, que serviria à representação das relações entre os verbetes e sua referência no mundo e na cultura, requeridas pelo Projeto UNL; e haveria uma estrutura do tipo árvore, em que estariam representadas as relações entre os verbetes e a língua, requeridas por ambos os projetos (UNL e ReGra). O verbete, ao ocupar simultaneamente a função de nó da estrutura gnosiológica e a posição de raiz da estrutura lingüística, exerceria a condição de unidade-ponte entre os dois conjuntos de informação. Assegurar-se-ia, desta forma, a plasticidade imprescindível à codificação de todas as informações necessárias. A Figura 1 ilustra, em linhas gerais, a arquitetura prevista para a nova base de dados:

[3] Para um maior detalhamento da estrutura do léxico que serve ao ReGra consulte-se (NUNES ET AL, *op cit.*)

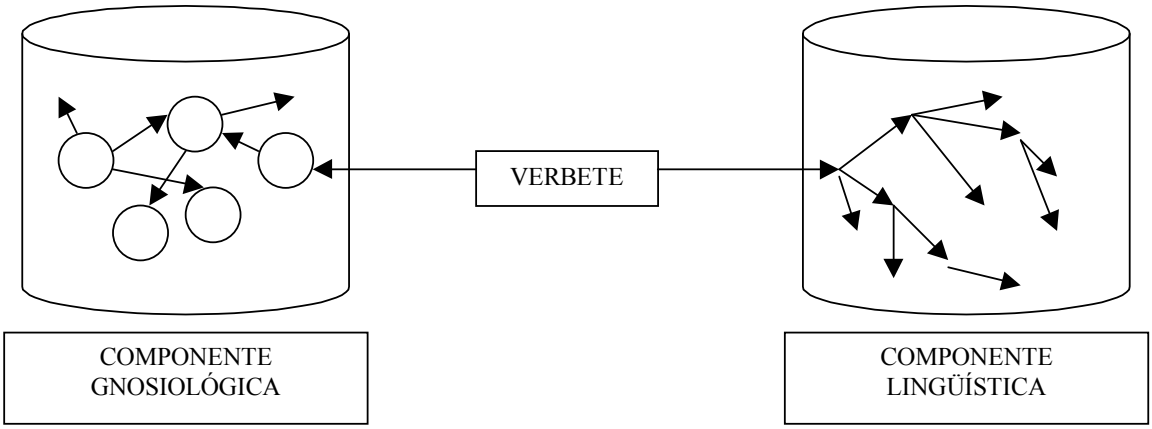

Figura 1 - Macroestrutura da BDL-NILC

A componente gnosiológica

A componente gnosiológica referida pelo modelo compreende um conjunto de nós, hipernós e arcos que corresponderiam à estrutura de conhecimentos (do mundo) recortada e definida pela cultura. Os verbetes estariam relacionados a este repositório de informações na medida em que apontariam para subcomponentes desta estrutura, que constituiriam, em última instância, seus significados. Trata-se, em última instância, daquilo que na tradição da teoria lingüística vem sido referido como "designatum", "sentido" ou "intensão"[4].

Torna-se particularmente importante salientar que, no modelo proposto, o sentido de um verbete não corresponde a um conceito isolado, mas a estruturas conceituais complexas[5], indicadas pela natureza reticulada da componente gnosiológica. O verbete não está associado apenas a um nó isolado, mas também a um conjunto de arcos que chegam ao nó referido ou partem dele em direção a outros nós, de forma a estabelecer uma ativação em cadeia de sentidos co-relacionados, que possam ser utilizados no processo de desambigüização ou de validação dos enunciados lingüísticos[6].

[4] Os conceitos de designatum (em oposição a denotatum), de sentido (em oposição a referência) e de intensão (em oposição a extensão) vêm sendo utilizados principalmente a partir do clássico Sobre o sentido e a referência, de Gotlob Frege .

[5] Reproduz-se, neste sentido, a distinção estabelecida em (JACKENDOFF, 1983), entre "linguistic expressions" e "conceptual structures".

[6] A estrutura da componente gnosiológica recupera, em linhas gerais, a idéia de acesso lexical prevista pelo modelo de Cohort (MARSLEN-WILSON & WELSH 1978); (MARSLEN-WILSON & TYLER, 1980), embora este último tenha sido utilizado como um modelo de ativação interativa para o reconhecimento de palavras.

Do ponto de vista de sua estrutura interna, a componente gnosiológica compreenderia dois conjuntos de dados: um conjunto de primitivos conceptuais (os nós) e um conjunto de relações entre esses primitivos conceptuais (os arcos). Os primitivos conceptuais corresponderiam a categorias axiomáticas definidas pela cultura. Embora pressuponham uma "teoria do mundo" muitas vezes ainda por ser explicitada, conformariam definições de natureza antes espontânea, intuitiva, pré-teórica[7]. No modelo adotado, seriam representadas como entidades indivisíveis, inanalisáveis em conjuntos de traços e irredutíveis a uma instância prototípica ou a várias instâncias exemplares[8]. Para efeito de citação, são rotuladas por palavras da língua inglesa.

São dois os tipos previstos de relações que se estabelecem entre os primitivos conceptuais: as relações ontológicas e as relações psicológicas. As relações ontológicas corresponderiam a arcos que definiriam uma de quatro relações lógicas disponíveis:

a) sinonímia (equ): para a identidade de conceitos (face/cara/rosto)
b) antonímia (ant): para a oposição de conceitos por
    b1) polarização (verdadeiro/falso);
    b2) por gradação (fervente/quente/morno/frio/gelado);
    b3) por inversão (pai/filho); e
    b4) por exclusão (sábado/domingo);
c) hiponímia (icl): para a inclusão de conceitos (carro/veículo/objeto); e
d) partonímia (pof): para a partição de conceitos (flor/jardim).

Essas inter-relações, estabelecidas para todo o conjunto de verbetes, terminariam por caracterizar uma ontologia navegável, que instrumentalizaria vários dos recursos previstos para desambigüização e para o aconselhamento lexical, e serviria à operação do *thesaurus*, recurso adicional atualmente incluído pela ferramenta de revisão gramatical.

As relações psicológicas materializariam as situações de co-ocorrência dos conceitos, a representar um conjunto cristalizado de informações a respeito do mundo. Caracterizariam,

[7] Cf. (MEDIN & ORTONY, 1989); (MURPHY & MEDIN, 1985).
[8] Diferentemente do que postulam, portanto, os modelos de categorização clássicos, que apostam na possibilidade de serem definidos conjuntos necessários e suficientes de traços diferenciadores (KATZ & FODOR, 1963); o modelo proposto por (ROSCH 1973, 1975), que se apóia na premissa de que algumas instâncias (os protótipos) de uma categoria são mais representativas do que outras, e que passariam a conduzir, por isso, o processo de

desta forma, uma base de conhecimento de mundo, novamente navegável, constituída a partir das 38 relações binárias definidas pela interlíngua UNL (como agente, objeto, instrumento, beneficiário, etc.)[9]. Esses novos arcos, a par de permitirem o refinamento dos recursos de desambigüização lexical, instruiriam um conjunto de procedimentos de validação e revisão semântica, não apenas do revisor gramatical, mas das ferramentas de codificação do português para a UNL e da UNL para o português.

A Figura 2 traz uma amostra de estrutura gnosiológica prevista para a base de dados lexical.

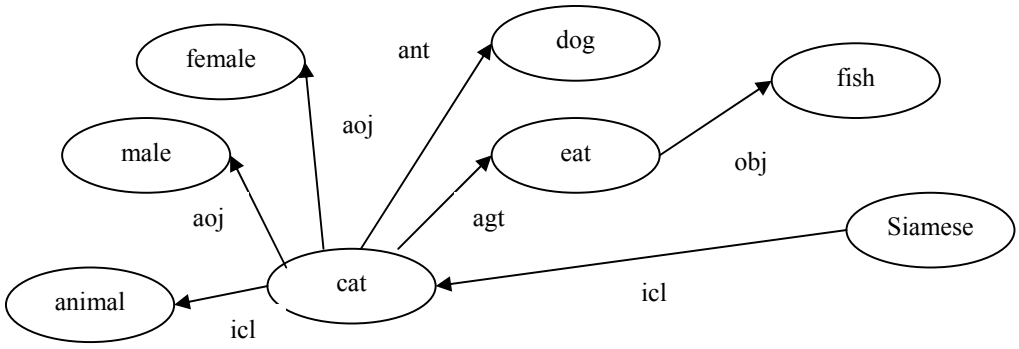

Figura 2 - Amostra de parte de estrutura conceptual prevista para o conceito representado por "cat"[10]

A componente lingüística

A estrutura da componente lingüística, diferentemente do que ocorre na estrutura gnosiológica, tem por objetivo representar o conjunto de relações que se estabelece entre o verbete e os demais verbetes que compõem o sistema lingüístico. Para permitir essa associação, optou-se por representar o verbete em cada um dos níveis de análise lingüística, acompanhando a idéia de que a língua é um sistema multiestratificado, comportando diferentes níveis de descrição, cada um dos quais elegendo problemas e utilizando modelos teóricos

categorização; e o modelo dos exemplares (MEDIN & SCHAFFER, 1978), que postula que a categorização é empreendida a partir de várias instâncias (exemplares), em vez de apenas uma.

[9] Para a lista exaustiva das relações semânticas previstas pela especificação UNL consulte-se (UCHIDA ET AL, 1999.

[10] Na figura apresentada, os arcos com a etiqueta "icl" correspondem a relações ontológicas de hiponímia, e o com a etiqueta "ant", a relações de antonímia; os arcos com as etiquetas "agt" (agente), "obj" (objeto) e "aoj" (atributo) correspondem a relações psicológicas definidas pela representação UNL.

diferentes[11]. Foram propostos 5 diferentes níveis de descrição lingüística, cada um dos quais envolvendo categorias que lhes seriam específicas:

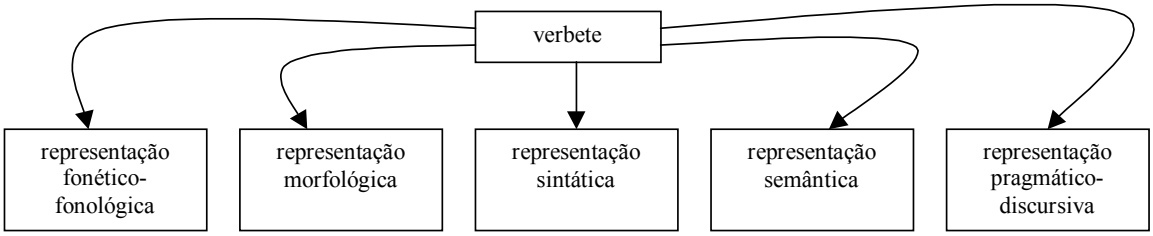

Figura 3 - Níveis de representação lingüística do verbete

Representação fonético-fonológica

A representação fonético-fonológica compreenderia, em princípio, apenas a oposição [± tônico], válida para a diferenciação dos pronomes pessoais oblíquos. Prevê-se, em um futuro próximo, que este nível possa também contemplar a transcrição fonética da palavra, com a indicação da sílaba tônica, para que as estratégias de aconselhamento ortográfico possam ser aprimoradas. No entanto, a base, neste primeiro momento, tem por objetivo apenas a fusão dos léxicos já existentes, sem incorporação de traços que não estejam sendo atualmente utilizados pelos aplicativos a que serve. A informação da tonicidade dos pronomes é pertinente para a análise sintática automática na medida em que implica diferentes distribuições sintáticas: o pronome tônico, por exemplo, mas não o átono, deve vir precedido de preposição.

Representação morfológica

A representação morfológica do verbete compreenderia a classificação do verbete em uma de quatro categorias: morfemas, lexias simples, lexias compostas ou lexias complexas. Os morfemas corresponderiam às unidades mínimas de significação da língua. Seriam as raízes e os afixos (prefixos ou sufixos), e não constituiriam, isoladamente, itens lexicais da língua. As lexias simples envolveriam combinações de raízes e afixos previstas pelos dicionários da língua ou consagradas pelo uso. Seriam cadeias de caracteres isoladas por espaços em branco. As lexias compostas envolveriam combinações de mais de uma raiz, e seriam classificadas segundo

[11] Para a pertinência da concepção de diferentes níveis de análise lingüística, consulte-se (BENVENISTE, 1966.)

a forma da composição (por justaposição ou por aglutinação). As lexias complexas corresponderiam a expressões fixas da língua, ou seja, a cadeias de caracteres que incluem espaços em branco. A Figura 4 apresenta a estrutura do nível morfológico de representação. A separação entre os morfemas e os diferentes tipos de lexia tem também motivação sintática, por envolverem informação necessária à análise e à geração dos enunciados da língua portuguesa[12].

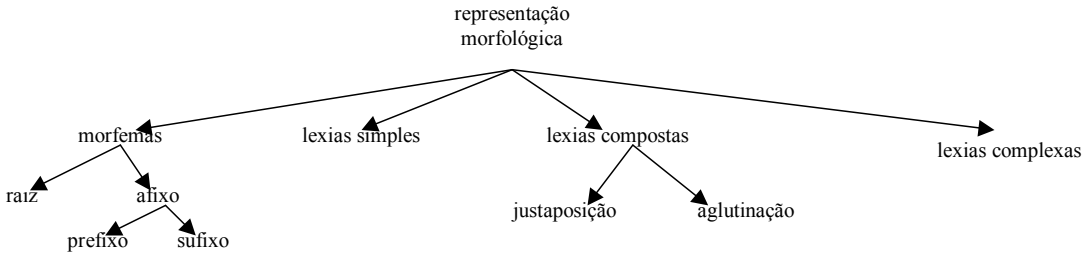

Figura 4 - Estrutura da representação morfológica do verbete

Para efeito de implementação, este modelo está provisoriamente simplificado para o seguinte, em função da ausência de informação sobre a complexidade das lexias nos dicionários de que os verbetes vêm sido obtidos.

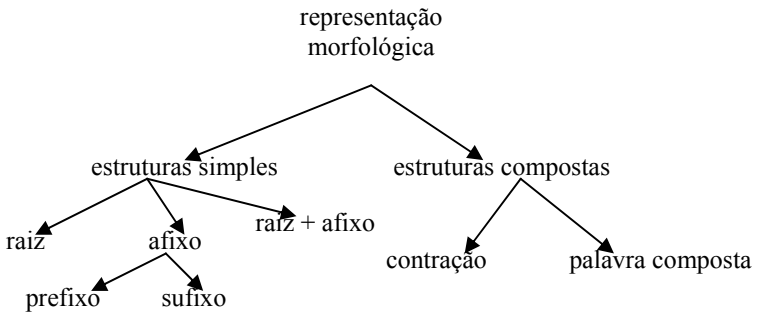

Figura 5 - Representação simplificada da estrutura morfológica do verbete

Representação sintática

[12] O corretor ortográfico não pode aceitar, por exemplo, a ocorrência de raízes sem os correspondentes afixos, ou a de afixos que não estejam ligados a raízes. As palavras compostas por justaposição (como os adjetivos e substantivos compostos), mas não as compostas por aglutinação (caso das contrações, por exemplo) seguem regras específicas de flexão e derivação.

Do ponto de vista de sua representação sintática, os verbetes seriam classificados segundo 1) seu comportamento gramatical e 2) seu comportamento sintático. Por comportamento gramatical, deve-se entender, aqui, a capacidade de um determinado verbete assumir um conjunto específico de flexões que o caracterizam como pertencente a uma determinada classe gramatical da língua portuguesa. Por comportamento sintático, presume-se o conjunto de relações de dependência que o verbete assume na sentença.

Seriam quatro as possibilidades de classificação morfossintática do verbete, derivadas da combinação dos primitivos sintáticos nome [N] e verbo [V][13]:

a) [+N,-V], que caracterizaria os verbetes cujo comportamento sintático coincidiria com o preenchimento da posição de núcleo do sintagma nominal: os substantivos propriamente ditos, os nomes próprios, as abreviaturas, as siglas, os pronomes pessoais do caso reto e do caso oblíquo, os pronomes de tratamento, alguns pronomes demonstrativos (como "isto"), alguns pronomes indefinidos (como "alguém), alguns pronomes interrogativos (como "quem"), alguns pronomes relativos (como "que"), os numerais coletivos (como "década") e os numerais multiplicativos (como "dobro"), nomeadamente;

b) [-N,+V], que caracterizaria os verbetes cujo comportamento sintático coincidiria com o preenchimento da posição de núcleo do sintagma verbal, ou seja, os verbos propriamente ditos;

c) [+N,+V], que caracterizaria os verbetes cujo comportamento sintático coincidiria com o preenchimento da posição de núcleo do sintagma modificador do sintagma nominal: os adjetivos, os pronomes possessivos, alguns pronomes demonstrativos (como "este"), alguns pronomes indefinidos (como "algum"), alguns pronomes interrogativos (como "qual"), alguns pronomes relativos (como "cujo"), os numerais cardinais (como "dois"), os numerais ordinais (como "primeiro"), os numerais fracionários (como "terço") e os artigos;

[13] Para a utilização de [nome] e [verbo] como primitivos gramaticais consulte-se (CHOMSKY, 1970) e (CHOMSKY & LASNIK, 1977).

d) [-N,-V], que caracterizaria os verbetes cujo comportamento sintático coincidiria com o preenchimento da posição de núcleo do sintagma modificador do sintagma verbal ou de núcleo do sintagma modificador de outros sintagmas modificadores: os advérbios, as preposições, as conjunções, as interjeições.

Para cada um dos ramos derivados dessa quadripartição, estariam associadas categorias morfossintáticas específicas, acompanhando a estrutura da língua portuguesa[14]. O Grupo [+N,-V] traria, desta forma, informação relativa:

a) ao gênero gramatical do verbete, indicado pela combinação dos primitivos [masculino] e [feminino], para a formação do masculino (*sapato*) [+masculino,-feminino], do feminino (*saia*) [-masculino,+feminino], do comum-de-dois ou uniforme (*pianista*) [+masculino,+feminino], do neutro, invariável ou não representado (*isso, bonit*) [-masculino,-feminino];

b) ao número gramatical do verbete, indicado pela combinação dos primitivos [singular] e [plural], para a formação do singular (*livro*) [+singular,-plural], do plural (*livros*) [-singular,+plural], do número uniforme (*lápis*) [+singular,+plural], do invariável ou não representado (*três, bonit*) [-singular,-plural];

c) à classe gramatical do verbete, representada por uma entre as seguintes possibilidades:

c1) substantivo comum (*mesa, cadeira, livro, saia, sapato*)
c2) substantivo próprio (*João, Rio de Janeiro*)
c3) sigla (ABNT, OAB)
c4) abreviatura (*dr., prof., p.*)
c5) pronome
c5a) demonstrativo (*isto*)
c5b) interrogativo (*quem*)
c5c) tratamento (*Vossa Senhoria*)
c5d) relativo (*o qual*)

[14] Na definição das classes e subclasses morfossintáticas, optou-se pela representação das categorias definidas pela Nomenclatura Gramatical Brasileira, reproduzidas em praticamente todas as gramáticas normativas da língua portuguesa, como (CUNHA & CINTRA, 1985), (BECHARA, 1976) e (ROCHA LIMA,1972).

c5e) indefinido (*alguém*)

c5f) pessoal (*eu, me, mim, comigo*)

c6) numeral (*dobro*)

c7) verbo (*fazer*)

A Figura 6 apresenta a estrutura do Grupo [+N,-V]

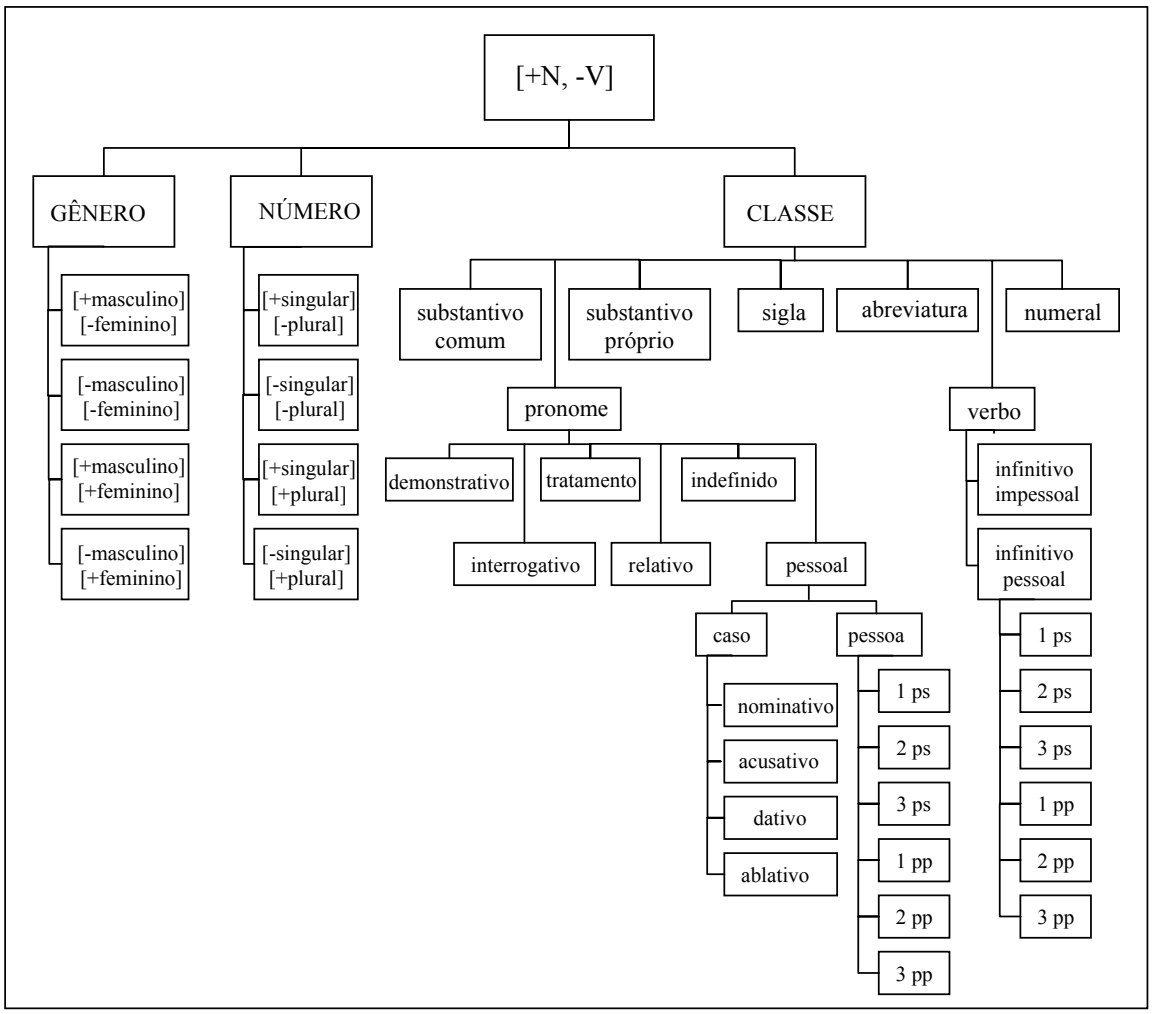

Figura 6 – Estrutura do Grupo [+n,-V]

O grupo [-N,+V] representaria as categorias pertinentes às formas verbais:

a) tempo: subdividido em tempo da referência e tempo do evento, acompanhando sugestão de (REICHENBACH, 1947);

b) modo: indicativo, subjuntivo e imperativo;

c) aspecto: perfeito e imperfeito.

d) pessoa: variando da primeira à terceira pessoa, do singular e do plural;

e) tipo de verbo: auxiliar (*estar*, em *Ela está fazendo isso*), de ligação (*estar*, em *Ela está doente*) ou nocional (*fazer*, em *Ela fez isso*)

A Figura 7 reproduz a estrutura do Grupo [-N,+V].

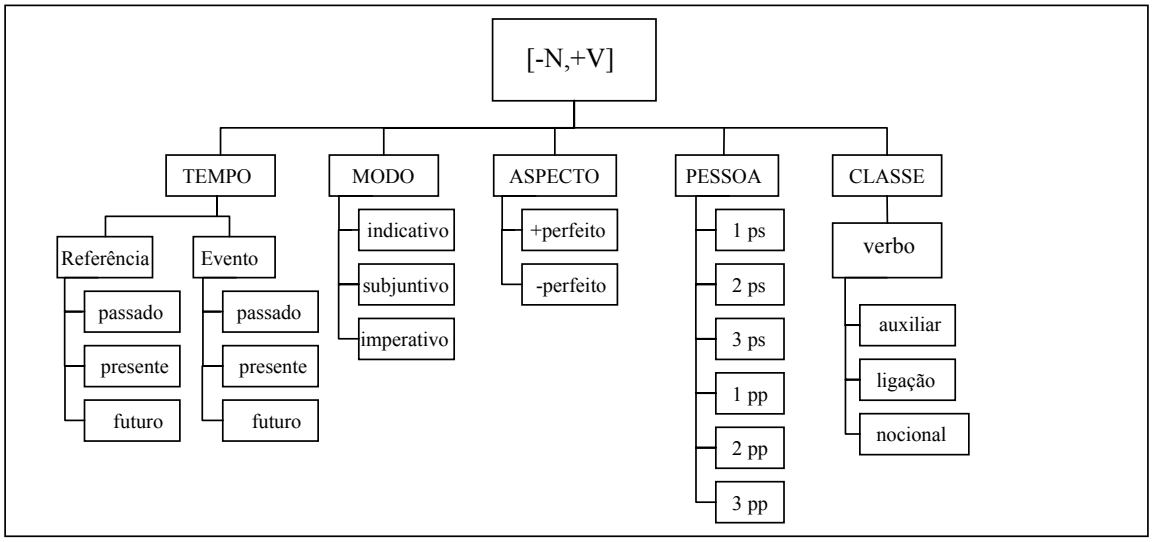

Figura 7 – Estrutura do grupo [-N,+V]

O grupo [+N,+V] reproduziria as informações de gênero e número já referidas no grupo [+N,-V], às quais acrescentaria a subclassificação prevista para as classes gramaticais, conforme indicado na figura 8.

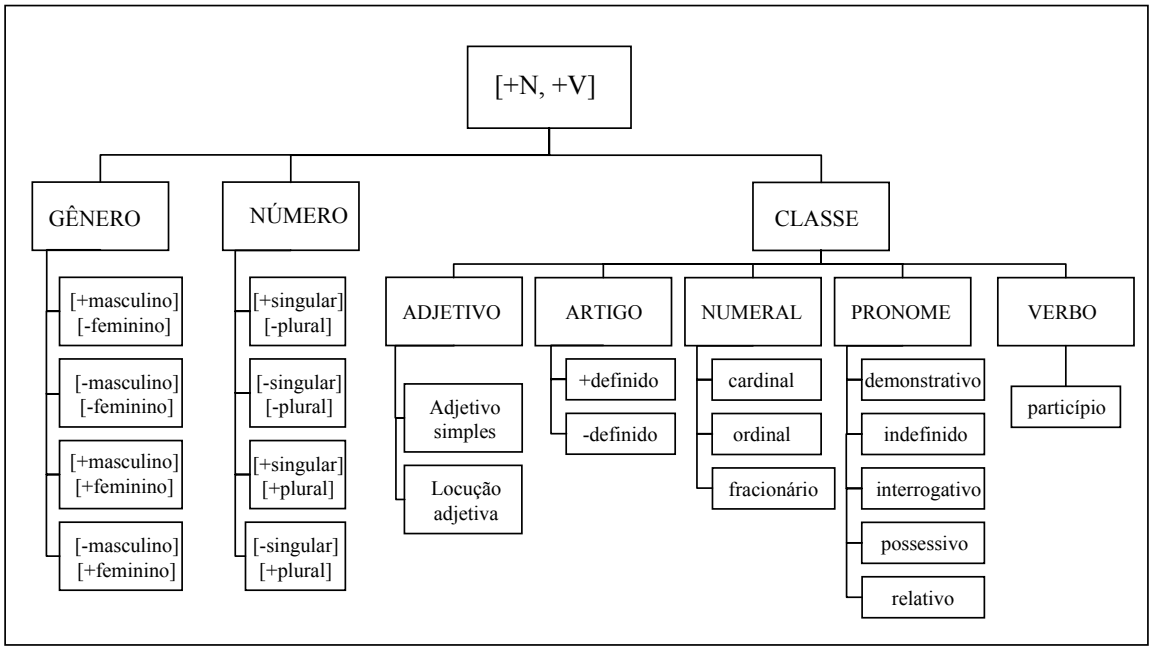

Figura 8 – Estrutura do grupo [+N,+V]

O grupo [-N,-V] representaria, por fim, as informações pertinentes às preposições, advérbios, interjeições, conjunções e as formas do gerúndio dos verbos, como indicado na figura 9.

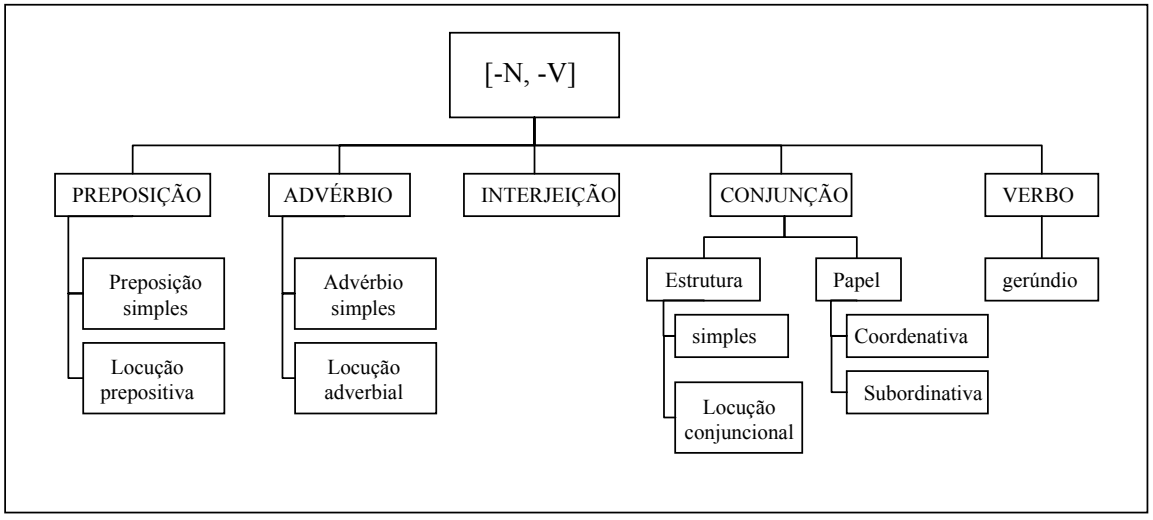

Figura 9 – Estrutura do grupo [-N,-V]

Em relação ao comportamento sintático, os verbetes representariam a) sua estrutura argumental e b) suas relações de regência. A estrutura argumental do verbete indicaria o número de argumentos por ele selecionado: verbetes impessoais (como os verbos impessoais e os substantivos não deverbais) não selecionariam nenhum argumento (0 argumento); verbetes intransitivos (com os verbos intransitivos, os adjetivos que não admitem complemento nominal, e substantivos deverbais) selecionariam um argumento; verbetes transitivos (como os verbos transitivos diretos ou indiretos, adjetivos e advérbios que requerem complemento nominal, e preposições) selecionariam dois argumentos; verbos bitransitivos (como os verbos transitivos diretos e indiretos e algumas preposições, como "entre") selecionariam três argumentos. A regência dos verbetes poderia ser b1) direta (se os argumentos selecionados não vêm precedidos por preposição), b2) indireta (se pelo menos um argumento selecionado vem precedido por preposição) e b3) pronominal (no caso dos verbos que selecionam obrigatoriamente como complemento o pronome pessoal oblíquo átono). Esses dois conjuntos de informação instrumentalizariam a análise sintática automática.

A figura 10 ilustra o nível sintático de representação dos verbetes.

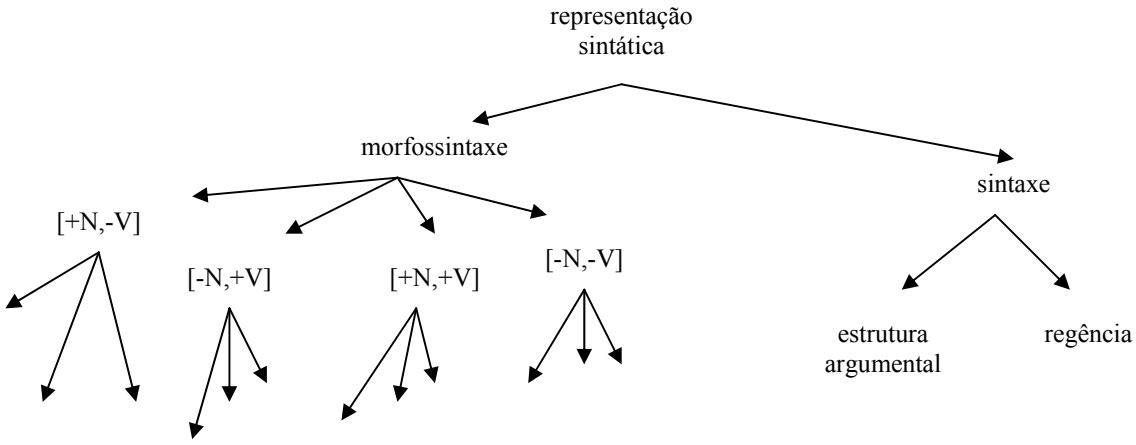

Figura 10 - Estrutura da representação sintática do verbete

Representação semântica e pragmático-discursiva

Os níveis de representação semântico e pragmático-discursivo do verbete estão ainda por serem desenvolvidos, a partir dos resultados obtidos pelo projeto TraSem (RINO ET AL, 2001), para a definição dos traços a serem incorporados às entradas lexicais.

Além dos dados do léxico do ReGra e do Dicionário Português-UNL, a BDL também armazena dados do projeto Thesaurus Eletrônico para o Português do Brasil (TeP). Esses dados são conjuntos de sinônimos e antônimos, presentes na componente gnosiológica da estrutura proposta. A Figura 11 mostra uma representação total dessa estrutura.

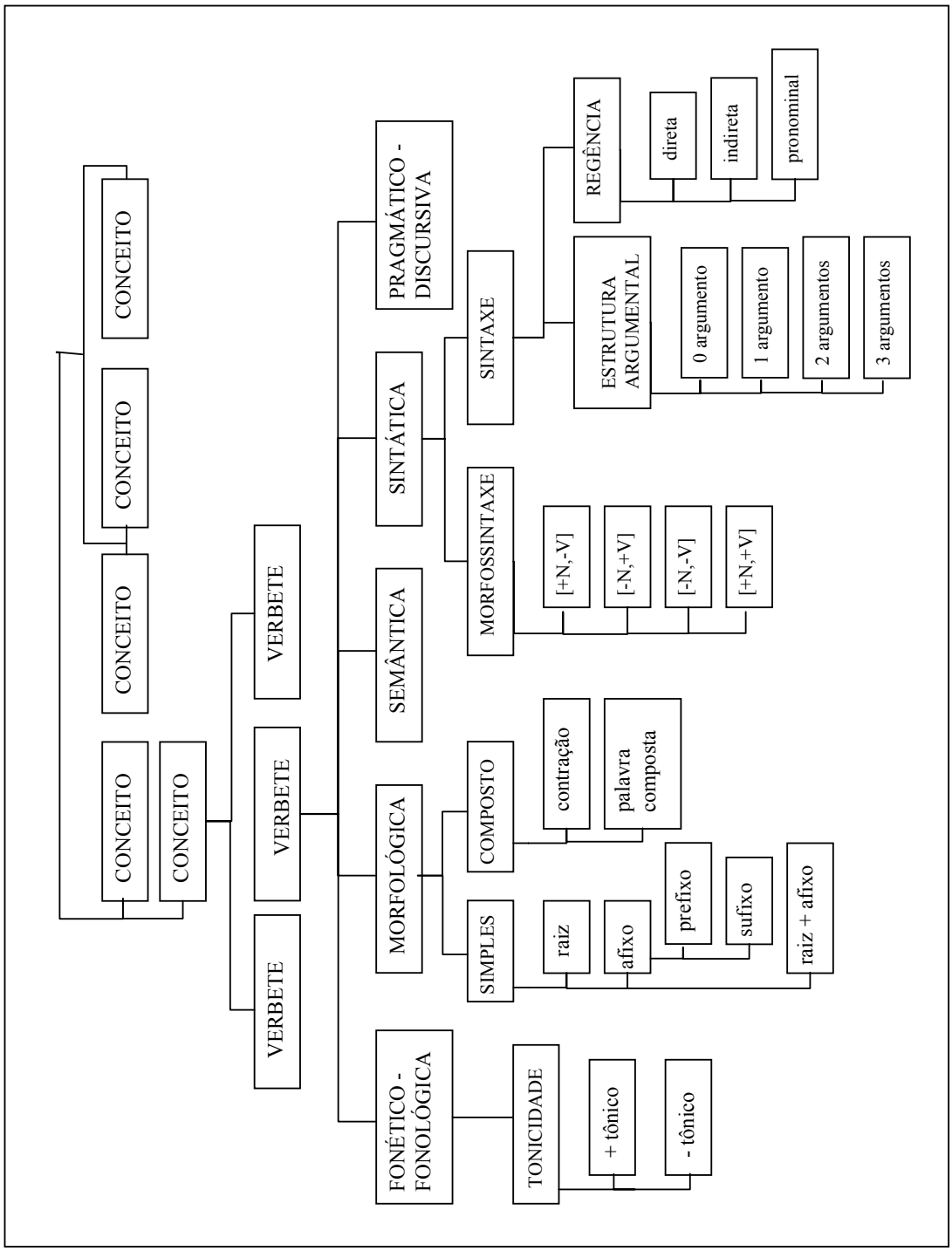

Figura 11 - Estrutura Proposta

c) Escolha do Modelo de Dados a ser usado: O armazenamento de dados lexicais desperta várias discussões a respeito da forma como tais dados devem ser armazenados. Enquanto alguns trabalhos optam por armazenar os dados em arquivos tipo texto especiais, com marcações apropriadas, por exemplo, as diretrizes do grupo *TEI Consortium* (*Text Encoding Initiative Consortium*) (http://www.tei-c.org/);(JANSZ, 1998), outras iniciativas apontam para a utilização de Sistemas de Gerenciamento de Bancos de Dados (ZAJAC, 1998); (IDE ET AL., 1993).

O armazenamento dos dados em arquivos especiais torna-os mais portáveis, já que os usuários potenciais não precisam ter acesso ao SGBD em questão. Tais arquivos devem ser processados por ferramentas específicas e a atualização dos dados deve ser minimizada, já que a cada alteração o arquivo deve ser novamente processado. Entretanto, quando se pensa em um volume de dados considerável como a BDL, que deverá servir como um ambiente centralizador, seguro e de fácil manipulação, as vantagens inerentes à utilização de um SGBD (consultas rápidas e elaboradas e maior segurança e consistência dos dados) parecem se sobrepor a qualquer argumento apresentado anteriormente.

Dessa forma, os fatos que levaram à conclusão de que a BDL deveria ser armazenada em um sistema próprio para manipulação de bancos de dados podem ser assim resumidos:

- necessidade de organizar as informações disponíveis nos vários aplicativos desenvolvidos pelo NILC, padronizando a representação das informações, tentando evitar a inconsistência dos dados e aumentando a possibilidade de reutilização destes;
- maior segurança dos dados, já que os SGBDs têm dispositivos próprios para evitar danos acidentais aos dados armazenados e para restringir a forma de acesso que cada usuário pode ter, por exemplo, concedendo permissão de alteração dos dados somente a pessoas autorizadas.

Depois de definir que a BDL seria armazenada em um SGBD, passou-se à escolha do Modelo de Dados a ser usado. Os dois modelos viáveis para o desenvolvimento de uma base desse tipo são o Modelo Relacional e o Modelo baseado em *features* (IDE ET AL., *op.cit*). Neste trabalho optou-se pelo Modelo Relacional e essa escolha pode ser justificada pelos seguintes fatos:

- a utilização do outro modelo, implementado em um SGBD-OO, exigiria um esforço inicial muito grande, já que os dados do léxico deveriam ser completamente remodelados para serem utilizados em tal modelo

- a perda de desempenho devido à junção de tabelas não é exatamente um problema, já que o número de aplicações existentes, que utilizam sistemas relacionais, é muito grande e, por isso, os algoritmos de junção de tabelas presentes nos SGBDs são bastante otimizados.

- a perda de informação sobre a hierarquia das informações pode ser solucionada através do uso de campos especiais, que simulam a hierarquia das informações, armazenando dados adicionais. Esses dados, por sua vez, podem ser extraídos do próprio conjunto de informações.

d) Modelagem Computacional segundo o Modelo de Dados escolhido: como citado acima, o Modelo de Dados escolhido foi o Relacional. Para melhor visualização da estrutura da base foi elaborado um Diagrama de Dados Entidade-Relacionamento, um modelo de dados conceitual de alto nível muito utilizado para o projeto conceitual de bases de dados. Esse diagrama é apresentado na Figura 12 e foi elaborado com base no modelo lingüístico apresentado anteriormente.

Esse modelo representa uma possível interpretação do relacionamento existente entre as palavras e respectivas informações lingüísticas. Neste modelo, há uma entidade denominada PALAVRA que possui o atributo *Lexema*. Este atributo representa o item lexical propriamente dito. PALAVRA está ligada a quatro relacionamentos: **tem**, com a entidade CLASSIFICAÇÃO, **é sinônimo de** e **é antônimo de**, com ela mesma e, por último, **expressa**, com a entidade CONCEITO. O relacionamento **tem** indica que cada palavra tem pelo menos uma classificação em uma classe gramatical qualquer. Os relacionamentos **é sinônimo de** e **é antônimo de** indicam quais palavras são sinônimas/antônimas umas das outras. E o relacionamento **expressa** indica que cada palavra expressa um conceito, representado em uma ontologia da língua portuguesa. Essa ontologia, entretanto, ainda não foi criada e, por isso, tal relacionamento é representado por um conjunto de linhas tracejadas. Neste ponto o leitor pode se perguntar "Por que não colocar o atributo *Lexema* como atributo da entidade classificação?" A resposta é simples: esse modelo está em contínua atualização e deve ser incrementado com novas relações semânticas e pragmático-discursivas entre os itens lexicais.

Dessa forma, é provável que novas entidades venham se a relacionar com PALAVRA, sendo importante manter as entradas independentes de qualquer classificação até que estes novos dados sejam implementados.

A entidade CLASSIFICAÇÃO apresenta os atributos *Codigo*, que identifica de forma única cada classificação, e *Canônica*, que armazena a forma canônica do item lexical em questão. Ela é superclasse das especializacões GRUPO1, GRUPO2 e GRUPO3. GRUPO1, por sua vez, é superclasse de PRONOME_PESSOAL, VERBO, SINTAXE1 e MORFOLOGIA1. Os atributos de GRUPO1 são: *Classe, Tipo, Gênero e Número* e armazenam informações sobre a classe classe gramatical, subclassificação na classe gramatical, gênero e número respectivamente. Essa subclassificação pode ser mais bem entendida com o seguinte exemplo:

> o item lexical *aonde* é classificado como pronome indefinido e pronome relativo. A informação armazenada pelo atributo *Classe* é pronome, e as informações armazenadas pelo atributo *Tipo* são indefinido e relativo.

A especialização PRONOME_PESSOAL armazena informações pertencentes exclusivamente aos pronomes pessoais: caso, pessoa e tonicidade. A especialização VERBO armazena informações pertencentes exclusivamente aos itens lexicais classificados como verbo infinitivo pessoal. SINTAXE1 armazena informações a respeito da regência e preposições regidas por um verbete com os atributos *Regencia, Estr_Arg* e *ListPrep*. MORFOLOGIA1 é representada pelo conjunto de atributos *Tipo, Componentes* e *Atributos*.

GRUPO2 pode ser descrita pelo seguinte conjunto de atributos: *TRef,* o tempo de referência do evento, *TEv*, o tempo de ocorrência do evento, *Modo,* a que modo de conjugação pertence aquele item lexical e *Pessoa*, que indica a pessoa a que o item se refere. GRUPO2 é também uma superclasse em relação a SINTAXE2, que tem como atributos *Regência, Estr_Arg e ListPrep*, que armazenam a regência, a estrutura argumental do verbete e uma lista de preposições regidas por tal verbete.

GRUPO3 é representada pelo conjunto de atributos *Classe*, que indica a qual classe gramatical determinado item lexical pertence, e *Tipo*, que indica subclassificação de tal item dentro de uma classe gramatical, e é superclasse de duas especializações: CONJUNÇÃO e MORFOLOGIA3. A primeira, como o próprio nome indica, armazena informações pertencentes somente àqueles itens classificados como conjunções. Ela é representada pelo atributo *Papel*, que indica qual o papel representado por determinado item (se coordenativo ou

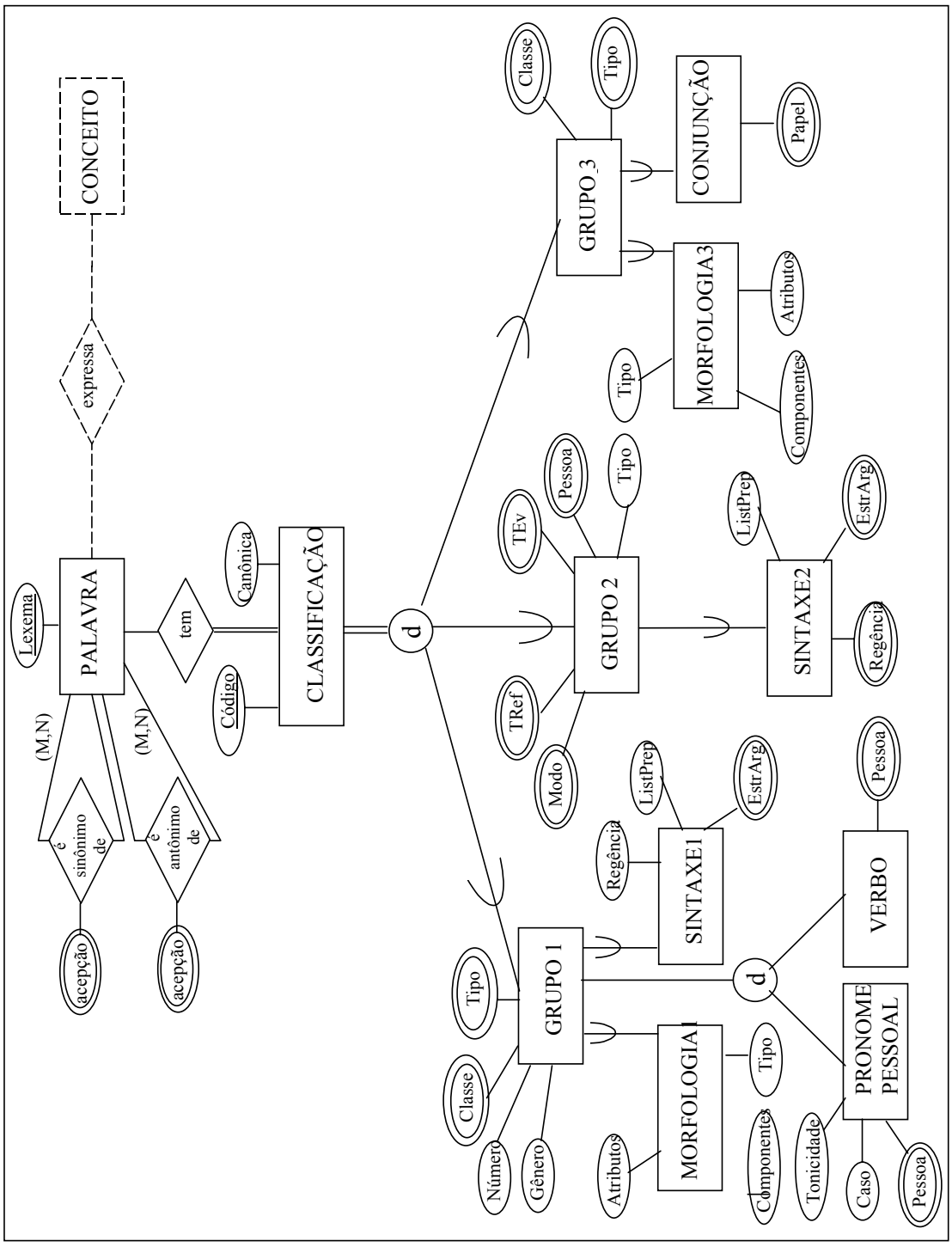

Figura 12 - Diagrama Entidade-Relacionamento

subordinativo). A subclasse MORFOLOGIA3 é representada pelos atributos *Tipo*, que indica se a entrada é um item lexical simples ou composto, *Atributos*, indicando se o item é uma raiz, um

prefixo, etc., e *Componentes*, indicando, no caso de itens compostos, quais são os itens lexicais que o compõe.

As subclasses MORFOLOGIA herdam os atributos ora da superclasse GRUPO1, ora da superclasse GRUPO3. Dessa forma, são duas especializações distintas, modeladas como subclasses MORFOLOGIA1 e MORFOLOGIA3. O mesmo ocorre com as subclasses SINTAXE1 e SINTAXE2. Ora os atributos são herdados de GRUPO1, ora de GRUPO2.

É importante ressaltar que só foram modeladas as estruturas que já possuem dados para serem inseridos na base. Os módulos Pragmática-Discursiva e Semântica, como citado anteriormente, ainda não foram modeladas por não haver, ainda, dados que possam ser utilizadas para preenchê-los.

e) Escolha do Sistema de Gerenciamento de Banco de Dados: a escolha do SGBD deve levar em consideração as consultas e operações de gerenciamento que serão realizadas com maior freqüência, as características e variáveis do SGBD que podem ser personalizadas de acordo com a aplicação, e quais as conseqüências resultantes dessas alterações para o sistema como um todo.

No caso da BDL, o SGBD utilizado é o Microsoft SQL Server, 6.5. Ele tem demonstrado ser um SGBD robusto, com várias operações de gerenciamento que previnem erros, falhas ou perdas involuntárias de dados. Todas as informações necessárias para a manipulação do sistema podem ser encontradas nos manuais que acompanham o software ou estão disponíveis on-line.

f) Implementação da Base de Dados: as tabelas foram implementadas de acordo com o mapeamento feito com base no Diagrama apresentado na Figura 12. Além do mapeamento, foi realizado um breve levantamento sobre algumas variáveis do sistema que poderiam ser alteradas para o melhor desempenho da aplicação. Os detalhes sobre o mapeamento e tal levantamento são apresentados na Seção 3.

g) Inserção dos dados na base: para que a transferência dos dados do léxico para a BDL pudesse ser realizada, foi necessário o desenvolvimento de uma ferramenta, daqui a diante referida como IMDL (Interface de Migração dos Dados do Léxico), que fizesse a conversão e

formatação dos dados existentes para o novo formato exigido pela modelagem realizada. Vale lembrar que, originalmente, os dados do léxico são armazenados em arquivos tipo txt.

A ferramenta IMDL[15] lê o arquivo de entradas do léxico e analisa, linha a linha, todas as informações disponíveis sobre cada item lexical. Conforme as informações vão sendo recuperadas, elas são inseridas em arquivos tipo txt específicos, que representam cada tabela implementada no SGBD. As informações são separadas pelo caracter \ e já são escritas no novo formato exigido. Depois que os arquivos são gerados, eles devem ser abertos no Word e salvos como documentos de texto do DOS. Essa conversão garante que todos os sinais gráficos usados serão conservados na cópia dos dados para as tabelas da base de dados. Com todas as entradas analisadas e devidamente separadas, esses arquivos são copiados para as respectivas tabelas através de um comando oferecido pelo SQL Server, o bcp (*bulk copy*), descrito em detalhes na Seção 2.3.

2.2 A Migração do Thesaurus

O projeto do *Thesaurus* Eletrônico para Português do Brasil (TeP) tem por objetivo produzir um *thesaurus* eletrônico, cuja finalidade é oferecer ao "usuário comum" da língua portuguesa um repertório de sinônimos e antônimos para a palavra que ele (o usuário) deseja, por razões de estilo e/ou precisão, substituir no processo de produção de seu texto escrito. Cada conjunto de sinônimos e antônimos associado a uma determinada acepção de uma forma lexical é construído com base nas relações semânticas - sinonímia e antonímia - que se estabelecem entre essas formas, não entre os conceitos por elas atualizados. Por essa razão, essas relações semânticas são denominadas "relações de sentido" (DIAS-DA-SILVA ET AL, 2000). A base de informações do thesaurus já conta com mais de 30 conjuntos de sinônimos e antônimos entre verbos, substantivos e adjetivos e advérbios.

A base de informações do Thesaurus, da mesma forma que o léxico, é armazenada em um arquivo tipo txt, com as informações separadas por caracteres especiais. Para que esses dados pudessem ser inseridos na BDL foi desenvolvida uma ferramenta semelhante à IMDL, que formata os dados do Thesaurus e insere os dados formatados em um novo arquivo tipo txt.

[15] A IMDL foi implementada usando a linguagem Delphi, versão 4.0 e encontra-se armazenada na máquina usada para o desenvolvimento deste projeto, na pasta c:\BDL\ferramenta.

Essa ferramenta será, a partir de agora, referenciada como IMDT[16] (Interface de Migração de Dados do Thesaurus).

Depois de processados, os arquivos gerados pela IMDT devem ser abertos no Word e salvos como documentos de texto do DOS. Isso garante que todos os sinais gráficos presentes nos dados convertidos serão preservados.

A transferência desses arquivos também é feita através do comando bcp, apresentado em detalhes na seção seguinte.

2.3 A migração pelo comando bcp (linha de comando)

Um dos grandes problemas encontrados no início da fase de inserção dos dados na BDL foi o tempo gasto para inserir um determinado conjunto de dados. O método de indexação usado pelo SQL Server é o B-Tree (LANGSAM ET. AL, 1996), (MICROSOFT, 1995), que usa páginas de índices para recuperar os dados. Há momentos em que, quando uma página de índices é completa, deve haver uma reorganização da árvore e, dependendo da profundidade alcançada pela árvore, essa reorganização pode levar um tempo considerável. No caso da BDL, quando o número de entradas inseridas na base alcançou valores acima de 1milhão de entradas, a inserção de um novo registro passou a tomar um tempo muito alto, o que fez com que a inserção dos dados, feita da forma tradicional, fosse substituída pela cópia dos mesmos através do comando bcp.

Para que esse comando possa ser utilizado, é necessário que uma das propriedades da base de dados esteja habilitada (*Select Into/Bulk Copy* - para verificar essa propriedade, basta clicar duas vezes sobre a base de dados desejada e será apresentada uma janela com as características gerais da base).

O comando bcp[17] permite que os dados sejam copiados em uma tabela sem que qualquer operação de indexação seja realizada, permitindo assim, que uma grande massa de dados seja inserida nas tabelas da base em pouco tempo. Para que esse comando possa ser utilizado de forma correta, todos os índices (chave-primária, chave-estrangeira e qualquer índice secundário) devem ser "desligados", ou seja, não devem ser definidos até que todos os dados

[16] A IMDT foi implementada com a linguagem Delphi, versão 5.0 e encontra-se residente na máquina usada para o desenvolvimento do projeto, na pasta c:\BDL\Thesaurus\ferramenta. Os arquivos originais, usados pela IMDT, estão armazenados na pasta c:\BDL\Thesaurus\Originais e os arquivos convertidos são armazenados em c:\BDL\Thesaurus.

[17] O comando bcp deve ser executado pelo prompt do DOS e a linha de comando utilizada é: c:\>bcp NomedaBase..NomedaTabela in c:\NomedoArquivo.txt -SNomedoServidor -Uusuario -PSenhadoUsuario

tenham sido transferidos (Esse procedimento não segue as recomendações da teoria de bancos de dados, mas é muito utilizado, na prática, por viabilizar a rápida inserção de dados em uma tabela).

Durante a execução, o sistema pedirá que usuário confirme o tipo do dado a ser copiado e o tamanho do campo. O usuário deve apenas pressionar a tecla "enter" para confirmar as informações que o sistema apresenta. A última confirmação pedida diz respeito ao caractere delimitador do campo. Todos os campos, exceto o último, têm como delimitador o caractere \. Dessa forma, o usuário deve, quando inquirido sobre tal caractere (*field terminator*), digitar a seqüência \\. O SQL Server pede que, antes de qualquer campo delimitador, seja digitado o caractere \. Para o último campo, o usuário deve digitar a seqüência \n, que indica que o delimitador do último campo é a marca de parágrafo. Mais detalhes podem ser obtidos no manual *Administrator's Companion* que acompanha o SQL Server.

Depois que todos os dados já foram copiados para as respectivas tabelas, os índices devem ser definidos. O sistema pode demorar vários minutos até que consiga acabar a indexação dos dados da base.

3 A Construção da Base de Dados no SQL Server

O primeiro passo realizado para a implementação da BDL foi o mapeamento das tabelas. Este, por sua vez, seguiu todas diretrizes de normalização e mapeamento de tabelas apresentado em (ELMASRI & NAVATHE, 1999), e será apresentado a seguir.

CLASSIFICAÇÃO
(Codigo , Canonica)

CONJUNCAO
(Codigo , Papel)

E_SINONIMO
(Lexema1, Lexema2, Acepcao)

E_ANTONIMO
(Lexema1, Lexema2, Acepcao)

GRUPO1
(Codigo , Gênero, Número)

GRUPO1A
(Codigo , Classe , Tipo)

GRUPO2
(Codigo , Tipo)

GRUPO2A
(Codigo, Tref, Tev, Modo, Pessoa)

GRUPO3
(Codigo, Classe, Tipo)

MORFOLOGIA1
(Codigo , Componentes, Tipo, Atributos)

MORFOLOGIA3
(Codigo , Componentes, Tipo, Atributos)

PALAVRA
(Lexema)

PRONOME_PESSOAL
(Codigo, Pessoa , Tonicidade)

SINTAXE1
(Codigo , Regencia, ListPrep)

SINTAXE1A
(Codigo , EstrArg)

SINTAXE2
(Codigo , ListPrep)

SINTAXE2A
(Codigo , Regencia)

SINTAXE2B
(Codigo , EstrArg)

VERBO
(Codigo , Pessoa)

O próximo passo realizado foi calcular o tamanho provável de cada tabela e, dessa forma, estimar um tamanho máximo para a base de dados. Esse cálculo foi realizado da seguinte forma:

- Definiu-se o tamanho máximo de um registro da tabela.

  Para cada registro da tabela há um *overhead* de 2 bytes. Dessa forma, deve-se calcular o tamanho do registro através da seguinte soma:

  Tamanho do registro = soma dos bytes ocupados por cada campo + 2 bytes

  A esse valor foram acrescidas duas unidades. Esse acréscimo é feito para que o *overhead* usado pelo SQL Server possa ser considerado.

- Calcula-se o número de páginas de dados usadas.

  O SQL Server usa páginas de recuperação de dados de tamanho 2K. Cada página usa 32 bytes de *overhead*. Dessa forma, para calcular o número de páginas de dados usadas, deve-se realizar o seguinte cálculo:

  Número de registros por página = 2016 / Tamanho do registro

  $$\text{Número de páginas de dados} = \frac{\text{Número máximo de registros da tabela}}{\text{Número de registros por página}}$$

- Calcula-se o tamanho dos registros de índice.

Para cada registro, há um *overhead* de 5 bytes. Deve-se efetuar o seguinte cálculo:

Tamanho do registro indexado = Soma dos bytes dos campos de índice + 5 bytes

- Calcula-se o número de páginas indexadas.

Número de registros indexados por página = 2016/ Tamanho do registro indexado

$$\text{Número de páginas de índice de nível 0} = \frac{\text{Número de páginas de dados}}{\text{Número de registros indexados por página}}$$

$$\text{Número de páginas de nível 1} = \frac{\text{Número de páginas de índice de nível 0}}{\text{Número de registros indexados por página}}$$

$$\text{Número de páginas de nível 2} = \frac{\text{Número de páginas de nível 1}}{\text{Número de registros indexados por página}}$$

Os cálculos devem ser efetuados até que o número de páginas do nível *n* seja 1.

Finalmente, deve-se realizar a soma do total de páginas de dados necessárias com o total de páginas de todos os níveis e o resultado será o número de páginas, de tamanho 2K, necessárias para a tabela analisada. Mais detalhes e exemplos podem ser obtidos em (MICROSOFT, *op cit.*)

Esse processo foi repetido para todas as tabelas e, dessa forma, pôde-se estimar um valor máximo da base de dados: 396Mb.

Para que a base de dados pudesse ser criada, foi definido um *DatabaseDevice*. Este é o arquivo .dat sobre o qual a base de dados é criada e os dados são armazenados. O tamanho estipulado para o *DatabaseDevice* deve ser o tamanho máximo previsto para a base de dados. No caso da BDL o valor usado para definição do *device* foi de 600Mb, um valor acima do estimado pelos cálculos realizados. Essa definição de tamanho foi uma conseqüência direta do fato de que novos dados deveriam ser inseridos futuramente e a base de dados teria seu tamanho

alterado. Para evitar que futuramente a base de dados tivesse que ser estendida e um novo *device* tivesse que ser criado, optou-se por definir uma base de dados de tamanho superior ao estimado e, quando todos os dados já tiverem sido inseridos na base, efetuar uma diminuição do tamanho da base, se necessário.

O próximo passo foi a criação de um *DatabaseDevice* utilizado para armazenar o Log de todas as transações efetuadas na base de dados. O tamanho desse arquivo de Log dever ser definido com um valor entre 10 e 25% do tamanho total da base de dados. No caso da BDL o arquivo de Log foi definido com 25% do tamanho do arquivo de dados – 150 Mb.

Devem ser criados mais dois *DatabaseDevices*, que serão usados para estender os segmentos de sistema e padrão (*system segment* e *default segment*), com tamanhos de 40 e 20 Mb, respectivamente. A determinação do tamanho desses segmentos foi empírica. Conforme a BDL ia sendo usada e o tamanho das consultas aumentava, o sistema falhava e indicava que tais segmentos estavam cheios, sendo necessário aumentá-los para que as consultas especificadas pudessem ser realizadas. Como não há qualquer tipo de recomendação a respeito do tamanho destes segmentos, eles foram sendo aumentados aos poucos, até que se atingisse um tamanho que permitisse a realização de tais operações sem que ocorressem falhas no sistema e o tamanho atingido foi 40 Mb.

Em seguida, passou à criação da BDL: na definição da nova base de dados, o usuário deve especificar como *DataDevice* o arquivo criado para armazenar os dados da BDL, e como *LogDevice* o arquivo criado para armazenar o Log das transações.

A base de dados foi criada e os arquivos de dados preparados. Todas as tabelas foram definidas de acordo com o mapeamento realizado, mas os índices (chave primária, chave estrangeira, índice secundário) não foram definidos até que todos os dados tivessem sido inseridos na base. Mais detalhes sobre a migração dos dados e variáveis utilizadas pode ser obtido no Anexo I (Descrição das Tabelas da Base de Dados TotalLex.).

Depois de analisar cuidadosamente um conjunto de variáveis do SGBD e as conseqüências de suas alterações para o sistema como um todo, algumas foram alteradas de acordo com a aplicação. Somente aquelas variáveis que não iriam causar perdas ao sistema foram alteradas. As variáveis que tiveram seus valores alterados foram:

Variável	Valor default	Valor atual
Memory	4096	8192
User connections	20	30
Remote Conn Timeout	10	10000

4 A Interface de Consulta via Web

A BDL, além de repositório central de dados, também deve servir como fonte de consulta para usuários que queiram obter informações detalhadas a respeito dos itens lexicais da língua portuguesa. Para tornar tal funcionalidade viável, foi necessário desenvolver uma interface que permitisse a qualquer usuário, leigo ou especialista, consultar as diversas informações disponíveis na base de dados.

Uma das idéias que acompanharam o processo de criação dessa interface foi que ela deveria ser de fácil acesso ao maior número de usuários possível. E a maneira encontrada para tal objetivo ser alcançado foi construir a interface como uma página de dados da Internet.

Antes que essa interface fosse criada, foi feito um breve levantamento junto ao usuário potencial do sistema, que são pesquisadores e estudantes da área de computação, lingüística e lingüística computacional, sobre as informações que deveriam estar disponíveis e a forma como essa consulta poderia ser realizada. A partir desse levantamento, um protótipo inicial[18] está sendo construído e deve ser avaliado pelo usuário em uma etapa posterior (Figuras 13a e 13b).

Além de permitir que o usuário faça consultas a respeito de itens lexicais do português, essa interface oferecerá ao usuário a oportunidade de conhecer um pouco mais sobre a gramática da língua portuguesa através da " Minigramática Online do Português do Brasil ", um aplicativo desenvolvido pelo NILC, disponível em http://nilc.icmc.sc.usp.br/minigramatica/sejabem.vindo..htm. Esse acesso será feito de forma direta, ou seja, a informação sobre a classe gramatical do item lexical terá um link que levará o usuário diretamente às informações sobre aquela classe específica e, a partir daí, o usuário poderá interagir com a minigramática livremente. Outras funcionalidades que deverão, no futuro, estar disponíveis ao usuário são a possibilidade de visualização gráfica das relações de sinonímia/antonímia, apresentado em detalhes na Seção 6, e traduções para a língua inglesa

dos itens consultados. As traduções só estarão disponíveis após a inserção dos dados do dicionário UNL na base.

Figura 13a – Interface de Consulta vai Web

[18] Esse protótipo está sendo desenvolvido em *Active Server Pages* (ASP), com o auxílio da ferramenta *Microsoft Visual Interdev*, 6.0. Maiores detalhes sobre a implementação podem ser obtidos no Anexo II (Descrição da Implementação da Interface de Consultas via Web).

Figura 13b – Resultado da consulta realizada

Todas as interfaces desenvolvidas serão avaliadas junto ao usuário através de métodos específicos para avaliação de interfaces. São eles:

- *Think Aloud:* essa técnica nos ajudará a conhecer a maneira como o usuário utiliza o sistema. Uma determinada tarefa será realizada pelo usuário, que deverá descrever (em voz alta) todos os passos utilizados para tal realização, bem como outras dúvidas ou comentários sobre o sistema. Essa declaração do usuário deverá ser gravada para que se possa ter todas as informações registradas (DIX ET AL., 1998)

- Heurística: essa técnica avaliará o sistema de acordo com um subconjunto selecionado entre as dez heurísticas propostas por Nielsen e Molich (1990) (MOLICH & NIELSEN 1990)[19].

As heurísticas a serem utilizadas neste projeto são:

- visibilidade do status do sistema: o sistema deve sempre manter o usuário informado sobre o que está fazendo, em um tempo de resposta razoável.

[19] Mais informações a respeito das dez heurísticas definidas por Nielsen podem ser encontradas em http://www.useit.com/papers/heuristic/heuristic_list.html

- controle e liberdade do usuário: as escolhas equivocadas do usuário levam o sistema a um estado indesejado. O sistema deve sempre oferecer, de forma clara e aparente, funções que permitam ao usuário desfazer/refazer uma determinada operação.
- consistência e padronização: o sistema não deve usar palavras, ícones ou ações diferentes que tenham o mesmo significado. Ele deve seguir os padrões já estabelecidos.
- prevenção de erros: o sistema deve, antes de ter mensagens de erros de boa qualidade, tentar evitar que os erros ocorram. Isso pode ser conseguido com uma modelagem cuidadosa, que previna e informe o usuário sobre as ações realizadas.
- equivalência entre o sistema e o mundo real: o sistema deve comunicar-se com vocabulário familiar ao usuário, seguir as convenções do mundo real, apresentando as informações em uma ordem lógica e natural.
- projeto minimalista e estético: os diálogos apresentados pelo sistema não devem conter informações irrelevantes ou raramente necessárias. Cada informação extra compete com as informações relevantes e diminui a visibilidade relativa.
- ajuda e documentação: embora seja melhor que o sistema possa ser utilizado sem ajuda ou documentação, é necessário que tais informações estejam disponíveis. Essas informações devem ser de fácil entendimento e recuperação, listar os passos que podem ou devem ser seguidos pelo usuário para realizar determinada tarefa, e não devem ser muito extensas.

Esse conjunto de heurísticas visa avaliar completamente as interfaces, verificando, por exemplo, se a interface fornece ajuda ao usuário sempre que necessário, e se essa ajuda é relevante, se existe documentação sobre o sistema, e se essa documentação está disponível, além de outros pontos relevantes para a avaliação.

5 A ferramenta de Geração de Listas

A BDL foi projetada para servir como fonte de consulta e também como fonte de recursos para o desenvolvimento de novos aplicativos para PLN. Dessa forma, foi projetada e construída uma interface que permite ao usuário gerar listas de informações de acordo com sua necessidade.

A interface permite que o usuário faça escolhas sobre o tipo de lista que deseja gerar, a(s) classe(s) gramatical(is), o gênero, o número e o intervalo a que os itens lexicais da lista devem pertencer. Os tipos de listas possíveis são:

a) lista de palavras da classe gramatical X, ou seja, todas as palavras classificadas em X; independentemente de pertencer à outra classe gramatical.

   Exemplo: caso o usuário deseje gerar uma lista de palavras pertencentes à classe "Substantivo", uma das palavras que iriam figurar nesta lista seria "casa", apesar de também estar classificada como flexão do verbo casar ;

b) lista de palavras que pertencem SOMENTE à classe gramatical Y, ou seja, todas as palavras que só estão classificadas em Y. Neste caso, a palavra "casa" não estaria presente na lista, pois está classificada também como verbo. Figurariam nesta lista palavras como abacate, caixa, ou prelo, que só podem ser classificadas como substantivo

c) lista de palavras que possam estar classificadas AO MESMO TEMPO nas classes gramaticais X, Y e Z, ou seja, palavras que pertencem à todas as classes especificadas.

   Exemplo: se as classes gramaticais X, Y e Z fossem as classes Adjetivo, Substantivo e Verbo, esta lista seria composta por palavras que, necessariamente, pertencem às classes em questão, como é o caso de "abandonado", "machucado" e "partido".

A seqüência de figuras 14a, 14b, 14c, 14d mostra um exemplo de interação em que deverá ser gerada uma lista com itens classificados como substantivos, do gênero feminino, sem qualquer restrição a respeito do número, e que estejam compreendidos no intervalo entre as letras h e p.

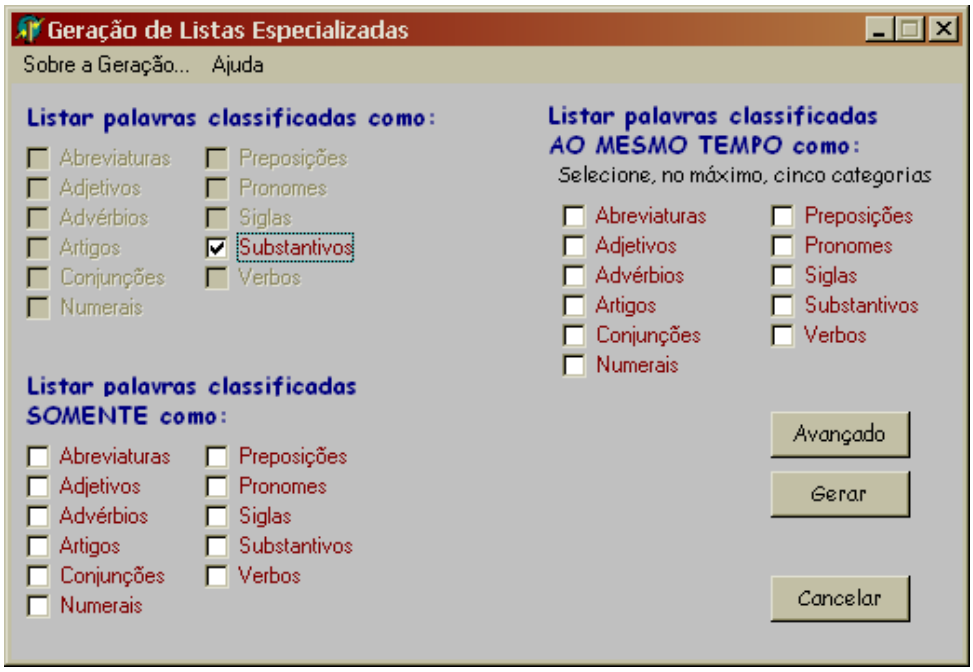

Figura 14a – Seleção do tipo de lista a ser gerada

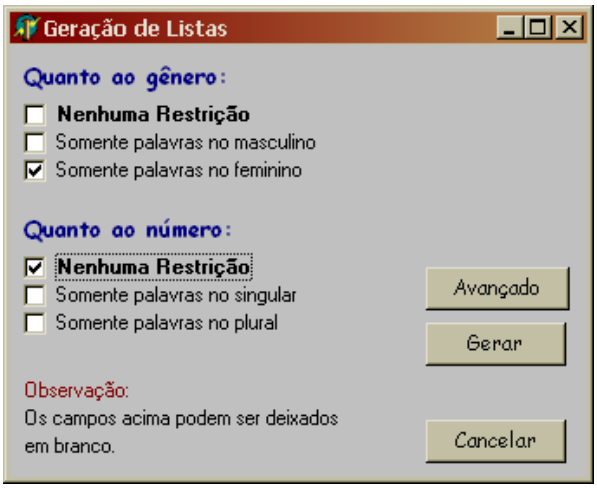

Figura 14b – Conjunto de Restrições Possíveis

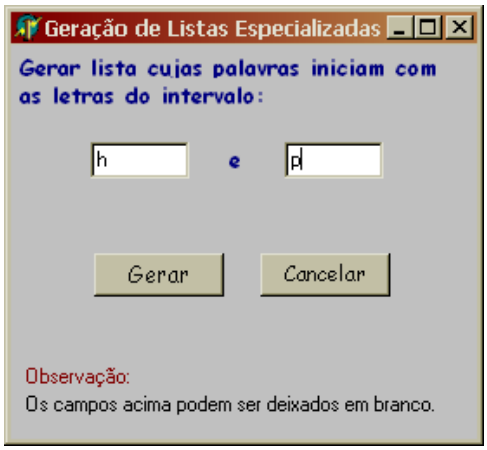

Figura 14c – Intervalo de Pertinência

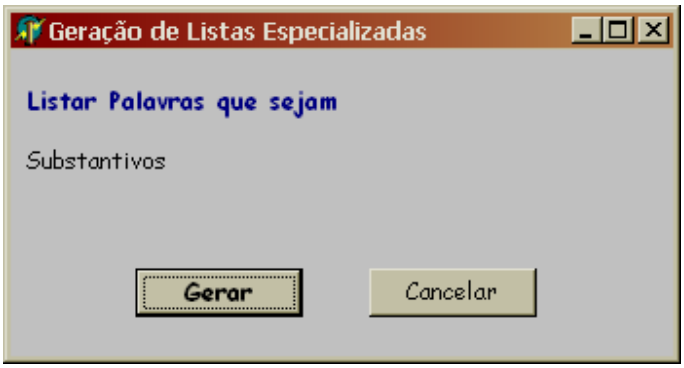

Figura 14d – Tela Final

Essas listas são arquivos tipo txt e as informações são expressas no mesmo formato usado para representar os dados do léxico. Detalhes sobre a implementação podem ser encontrados no Anexo III (Descrição da Implementação da Ferramenta de Geração de Listas Especializadas).

6 A Interface Gráfica do Thesaurus

Uma das questões levantadas durante o desenvolvimento da BDL foi: "Qual a forma de apresentação mais conveniente para se apresentar os dados do thesaurus ao usuário?" Depois de se analisar algumas possibilidades, percebeu-se que uma interface gráfica, que permitisse a identificação dos relacionamentos existentes entre os itens lexicais, e a possibilidade de uma interação direta do usuário seriam características desejáveis. Uma outra decisão tomada foi que essa interface também deveria ser disponibilizada na Web.

Foi realizado um breve levantamento a respeito das ferramentas que poderiam ser utilizadas para o desenvolvimento de tal interface e a que foi considerada mais adequada foi o *ThinkMap Studio* 5.0 (http://www.thinkmap.com). Essa ferramenta pode ser facilmente conectada a uma base de dados relacional, permitindo que seus dados sejam apresentados ao usuário em uma interface gráfica, com visualização dos dados, bem como o relacionamento entre eles, em 3D. Existe uma interface gráfica para os itens lexicais da língua inglesa, desenvolvida em um projeto de parceria entre a *Plumb Design* e a Universidade de Princeton para a visualização dos dados disponíveis na *WordNet* (MILLER, 1993)[20].

Essa interface serviu de fonte de inspiração e motivou o desenvolvimento de uma interface semelhante para a língua portuguesa do Brasil usando os dados disponíveis no *Thesaurus* do Português. Já foi desenvolvido um protótipo inicial da interface de visualização dos dados sobre sinonímia do TeP, inicialmente denominada de TeP3D. Essa interface foi desenvolvida com uma versão para testes da ferramenta *ThinkMap Studio* 5.0, com validade de 30 dias e, por esse motivo, esse protótipo não poderá, ainda, ser acessado pela interface de consulta aos dados da BDL. As Figuras 15a, 15b e 15c mostram quais foram os resultados obtidos na construção desse protótipo inicial do TeP3D.

A Figura 15a mostra a palavra "apoquentado" no centro das arestas. Esta é o alvo da consulta realizada, e as demais palavras, que aparecem nas extremidades das arestas são

[20] Mais informações podem ser obtidas em http://www.thinkmap.com/projects/plumbthesaurus

sinônimos dessa palavra, em suas várias acepções. Caso o usuário deseje visualizar os sinônimos de qualquer uma das palavras sinônimas de "apoquentado", basta clicar sobre a palavra desejada e a mesma passará a ser o centro do grafo e seus sinônimos serão apresentados nas extremidades das arestas. Da mesma forma, a Figura 15b mostra a palavra "amoroso" como centro da consulta. Outra forma de identificar a palavra que está sendo consultada, é observar a terceira caixa de texto, da esquerda para direita, apresentada na base da interface. Essa caixa permite que o usuário escolha a nova palavra a ser visualizada, mesmo que esta não seja sinônimo da palavra atual, ou verifique qual palavra está sendo analisada no momento. A Figura 15c mostra uma outra possibilidade de visualização dos dados.

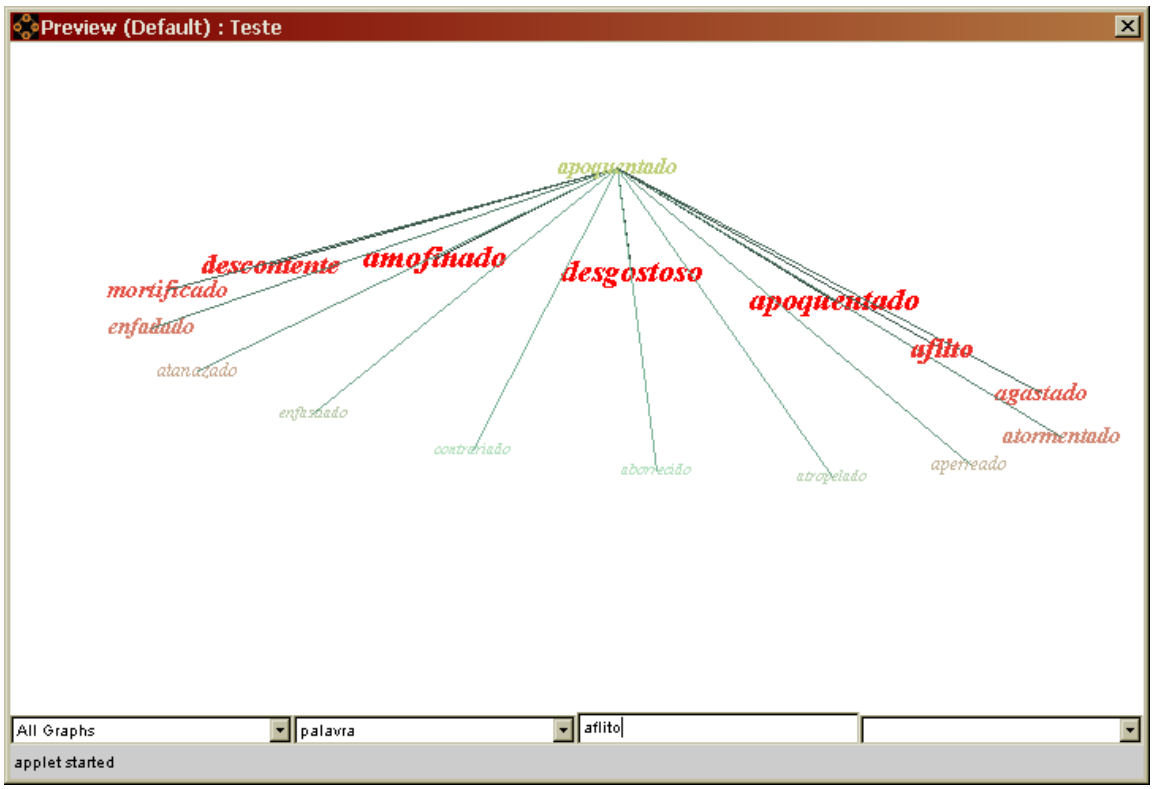

Figura 15a – TeP3D

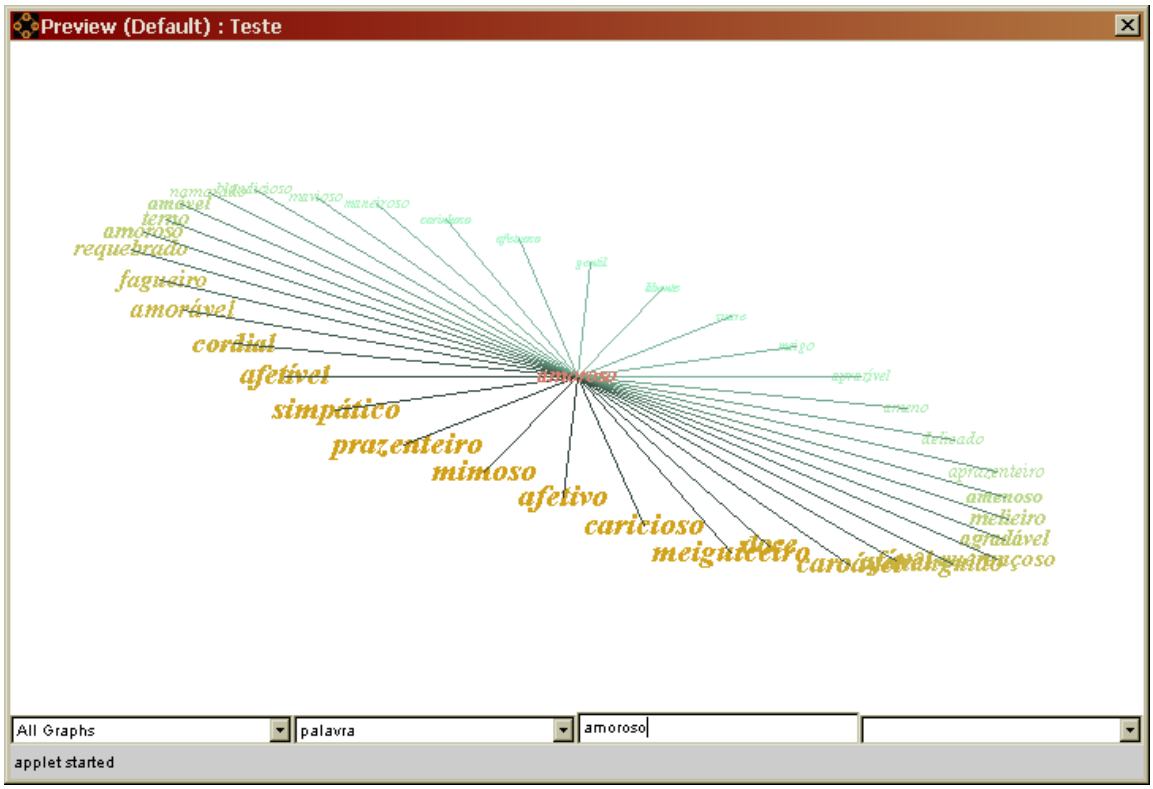

Figura 15b – TeP3D

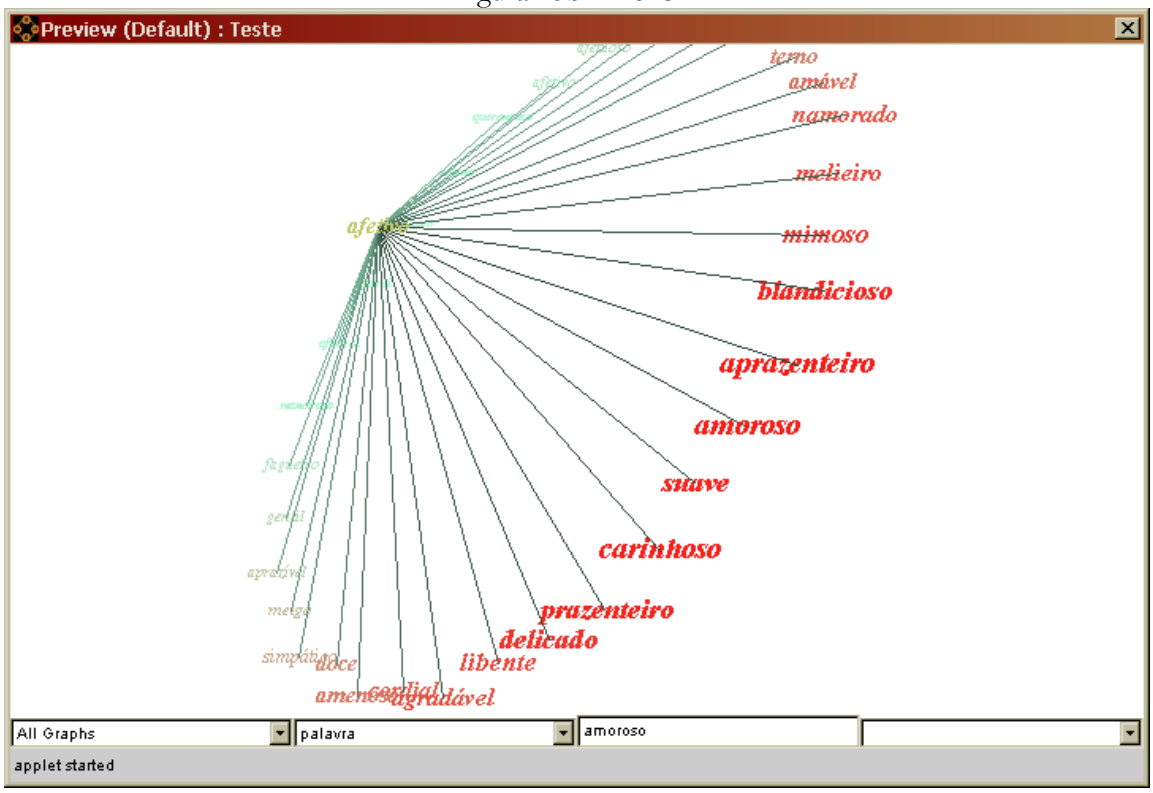

Figura 15c – TeP3D

Uma outra possibilidade é a utilização das ferramentas *DreamWeaver Ultradev* 4.0 e *Flash* 5.0, da *Macromedia*. Essas ferramentas, quando utilizadas em conjunto, permitem a conexão e recuperação de informações armazenadas em bases de dados e a manipulação dessas informações de forma a tornar sua apresentação ao usuário mais agradável. A conexão e recuperação dos dados são feitas com a ferramenta *DreamWeaver Ultradev* e a animação e manipulação dos dados é feita com *Flash*, ferramenta essa que tem sido bastante utilizada para o desenvolvimento de *sites* animados e interativos.

7 A Ferramenta de Edição

Uma última preocupação a respeito do acesso aos dados da BDL é desenvolver uma interface que permita a fácil alteração dos dados armazenados e, também, a inserção de novos dados, sem que seja necessário que o usuário conheça a estrutura da BDL ou a forma como as informações devem ser distribuídas nas tabelas da base de dados.

Está em fase de desenvolvimento, em Delphi 5.0, uma interface de edição/inserção dos dados, que ficará disponível somente a pessoas autorizadas, para que haja maior controle e segurança dos dados que serão alterados ou inseridos na BDL. As figuras 16a, 16b, 16c e 16d ilustram a forma como o usuário irá interagir com a ferramenta. É importante lembrar que esta ferramenta está em desenvolvimento e, por isso, as funcionalidades não estão completamente implementadas.

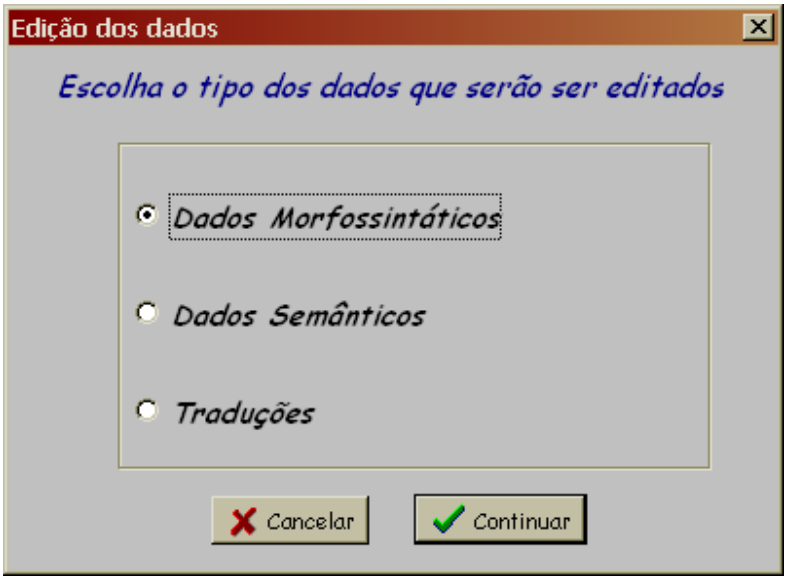

Figura 16a

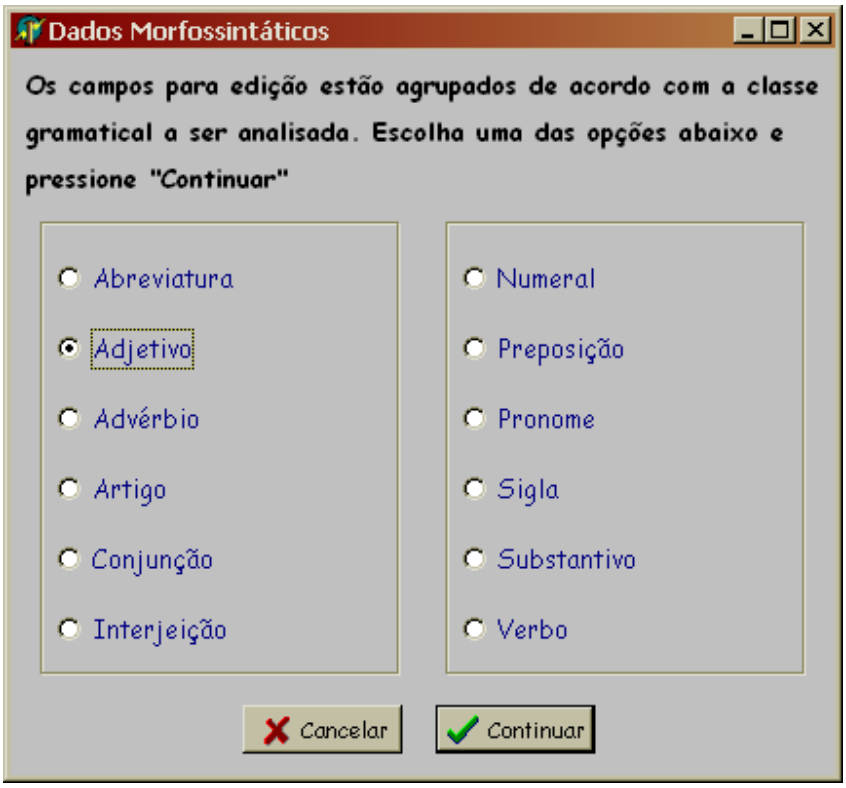

Figura 16b

A Figura 16a mostra a tela inicial apresentada ao usuário. Neste caso, o usuário deve optar pelo tipo de dados a ser atualizado/inserido. Feita a escolha, é apresentada ao usuário a interface correspondente ao tipo de dados escolhido. No caso da Figura 16b, o tipo de dados escolhido foi Morfossintático e a próxima escolha diz respeito à classe gramatical que deverá ser alterada. De acordo com a classe gramatical escolhida, é apresentado um conjunto de atributos que podem ser alterados (Figura 16c). O usuário pode optar por alterar ou não algum campo. Caso o campo seja deixado em branco, aquele dado não terá seu valor alterado na base de dados. Só serão atualizados os valores dos campos preenchidos.

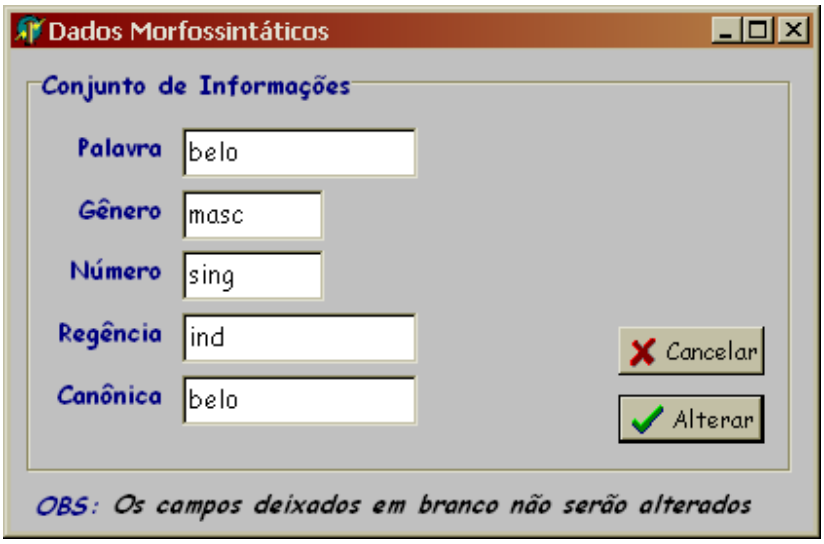

Figura 16c

Figura 16d

Depois de inserir/alterar os dados que deseja, o usuário é avisado pelo sistema se a alteração pôde ser efetuada (Figura 16d).

8 Conclusões

O desenvolvimento de uma base de dados como a BDL tem se mostrado um trabalho bastante complexo. Todo o processo de projeto e desenvolvimento da base foi realizado com muito cuidado e, mesmo assim, ocorreram vários problemas durante o processo.

Um dos grandes problemas encontrados diz respeito ao gerenciamento do SQL Server. Apesar de ser bastante eficiente, em alguns momentos, o software apresenta um comportamento inadequado. Depois de analisar as características dos problemas ocorridos e verificar em que situações tais problemas ocorreram, levantamos a hipótese de que o mau funcionamento se dá devido a incompatibilidades do SQL Server 6.5 com o sistema

operacional Windows 2000. Acreditamos que a versão SQL Server 2000 não apresente os mesmos problemas.

Além dos problemas com gerenciamento do SQL Server, enfrentamos muitos problemas para o desenvolvimento da interface de acesso aos dados via Web. Os problemas enfrentados nesta fase, entretanto, foram sendo superados com o progresso dos trabalhos e com a familiarização com a ferramenta *Microsoft InterDev* 6.0.

O processo de migração dos dados também foi uma tarefa bastante complicada. O volume de dados a ser inserido era muito grande, tornando inviável a inserção dos dados de maneira tradicional. A utilização do comando bcp foi a solução encontrada para tal problema e o conjunto de parâmetros utilizado foi definido de forma empírica.

Todos esses problemas foram bastante positivos, mostrando-nos o quão complicado pode ser o desenvolvimento de uma aplicação que, inicialmente, poderia ser considerada simples. Outra lição reforçada foi a de que a construção de um recurso como este, para o processamento da língua portuguesa, é uma tarefa bastante árdua, que requer um planejamento cuidadoso das tarefas a serem realizadas e o acompanhamento direto de um especialista em lingüística.

Referências Bibliográficas

BECHARA, E. (1976). Moderna gramática portuguesa. São Paulo: Companhia Editora Nacional.

BENVENISTE, E. (1966). Les niveaux de l'analyse linguistique. In Problèmes de linguistique générale. Paris: Gallimard.

CHOMSKY, N. (1970). Remarks on nominalization. In Jacobs, R. A. & Rosenbaum, P. (Eds.) Readings in English Transformational Grammar. Waltham, Mass: Ginn and Company.

CHOMSKY, N. & LASNIK, H. (1977). Filters and control. Linguistic Inquiry, 8:3, 425-504.

CUNHA, C. & CINTRA, L. (1985). Nova gramática do português contemporâneo. Rio de Janeiro: Nova Fronteira.

DIAS-DA-SILVA, B.C. ET AL. (2000) Construção de um Thesaurus para o Português do Brasil In: *Encontro para o Processamento da Língua Portuguesa Escrita e Falada*, *5*. Pp 1-11, Outubro, Atibaia.

DIX, A; FINLAY, J.; ABOWD, G.; BEALE, R. (1998) *Human-Computer Interaction* Second Prentice-Hall Europe, 2 ed., p 427.

ELMASRI, R.; NAVATHE, S.B. (1994) *Fundamentals of Database Systems* The Benjamin/Cummings Publishing Company, California, 2ed.

IDE, N.; LE MAITRE, J.; VÉRONIS, J. (1993) Outiline of a Model for Lexical Databases *Information Processing & Management,* 29(2), pp159-186. Disponível em 09/10/2000 (http://www.up.univ-mrs.fr/~veronis/publis.html)

JACKENDOFF, R. (1983). *Semantics and cognition*. Cambridge, MASS: The MIT Press.

JANSZ,K. (1998) *Intelligent processing, storage and visualisation of dictionary information.* Sydney, Austrália, Tese – Universidade de Sydney.

KATZ, J.; FODOR, J. (1963). *The structure of a semantic theory*. Language, 39, 170-210.

LANGSAM, Y.; AUGENSTEIN, M.J.; TENENBAUM, A.M. (1996) *Data Structures Using C and C++* Prentice Hall, 2 ed.

MARSLEN-WILSON, W. D.; TYLER, L. K. (1980). *The temporal structure of spoken language understanding. Cognition*, 8, 1-71.

MARSLEN-WILSON, W. D.; WELSH, A. (1978). Processing interactions and lexical access during word recognition in continuous speech. *Cognitive Psychology*, 10, 29-63.

MEDIN, D. L.; ORTONY, A. (1989). Psychological essentialism. In. S. Vosniadou and A. Ortony (Eds.) Similarity and analogical reasoning. New York: Cambridge University Press.

MEDIN, D. L.; SCHAFFER, M. M. (1978). Context theory of classification learning. Psychological Review, 85, 207-238.

MILLER, G. A.; BECKWITH, R.; FELLBAUM, C.; GROSS, D.; MILLER, K. (1993). *Introduction to WordNet: An On-line Lexical Database.* Disponível em DATA (ftp://ftp.cogsi.princeton.edu/pub/wornet/5papers.ps)

MOLICH, R., NIELSEN, J. (1990). Improving a human-computer dialogue, *Communications of the ACM* 33, 3 (March), 338-348.

MURPHY, G. L.; MEDIN, D. L. (1985). The role of theories in conceptual coherence. Psychological Review, 92, 289-316.

NIELSEN, J., MOLICH, R. (1990). Heuristic evaluation of user interfaces, *Proc. ACM CHI'90 Conf.* (Seattle, WA, 1-5 April), 249-256.

NIELSEN,J. Avaliação de interfaces através de heurísticas. Disponível em 02/12/2000 em (http://www.useit.com/papers/heuristic/heurist_list.html)

NUNES, M.G.V. ET AL. (1996). *A Construção de um Léxico da Língua Portuguesa do Brasil para Suporte à Correção Automática de Textos*. Rel. Técnico do ICMC, No. 42. ICMC/USP –n São Carlos, Agosto.

RINO, L. H. M.; MARTINS, R. T.; MARCHI, A. R.; KUHN, D. C. S.; PINHEIRO, G. M.; PARDO, T. A. S.; DI FELIPPO, A.; NUNES, M. G. V. (2001) *Projeto TraSem: A investigação teórica sobre o problema da ambigüidade categorial.* Série de Relatórios do NILC. NILC-TR-01-1, Abril, 42p

REICHENBACH, H. (1947). *Elements of symbolic logic*. Berkeley, CA: University of California Press.

ROCHA LIMA, C. H. (1972). *Gramática normativa da língua portuguesa*. Rio de Janeiro: Livraria José Olympio.

ROSCH, E. H. (1973). On the internal sctructure of perceptual and semantic categories. In T. E. Moore (Ed.) Cognitive development and the acquisition of language. New York: Academic Press. pp. 111-144.

ROSCH, E. H. (1975). Cognitive representation of semantic categories. *Journal of Experimental Psychology*: General, 104, 192-133.

UCHIDA, H.; ZHU, MEIYING; DELLA SENTA, T. (1999). Universal Networking Language: a gift for a millenium. Tokyo: United Nations University.

ZAJAC, R; (1998). The Habanera Lexical Knowledge Base Management System – In: *International Conference on Language Resources And Evaluation, 1*. Granada, Spain, 28-30 May.

Anexo I

Descrição das Tabelas da Base de Dados TotalLex

Aqui são apresentados todos os campos usados para a definição das tabelas e também o tipo de dados e o tamanho de cada campo. Essas informações são muito importantes, uma vez que é através delas que o tamanho da base e outras informações importantes para o gerenciamento da BDL são obtidos.

classificacao

cl_Codigo	int
cl_Item	int
cl_Canonica	varchar(50)

conjuncao

cj_Codigo	int
cj_Papel	varchar(6)

e_sinonimo

es_Lexema1	varchar(50)
es_Lexema2	varchar(50)
es_Acepcao	varchar(2)

e_antonimo

ea_Lexema1	varchar(50)
ea_Lexema2	varchar(50)
ea_Acepcao	varchar(2)

grupo1

g1_Codigo	int
g1_Genero	varchar(4)
g1_Numero	varchar(4)

grupo1a

g1a_Codigo	int
g1a_Classe	varchar(6)
g1a_Tipo	varchar(30)

grupo2

g2_Codigo	int
g2_Tipo	varchar(3)

grupo2a

g2a_Codigo	int
g2a_TRef	varchar(3)
g2a_TEv	varchar(3)
g2a_Modo	varchar(5)
g2a_Pessoa	varchar(3)

grupo3

g3_Codigo	int
g3_Classe	varchar(6)
g3_Tipo	varchar(8)

morfologia1

mf_Codigo	int
mf_Tipo	varchar(4)
mf_Atributo	varchar(8)
mf_Componentes	varchar(50)

morfologia3

mf_Codigo	int
mf_Tipo	varchar(4)
mf_Atributo	varchar(8)
mf_Componentes	varchar(50)

palavra

vt_Lexema	varchar(50)

pronome_pessoal

pp_Codigo	int
pp_Caso	varchar(3)
pp_Pessoa	varchar(3)
pp_Tonicidade	varchar(3)

sintaxe1

st_Codigo	int
stb_Regencia	varchar(7)
st_List_Prep	varchar(50)

sintaxe1a

sta_Codigo	int
sta_Estr_Arg	varchar(4)

sintaxe2

st_Codigo	int
st_List_Prep	varchar(50)

sintaxe2a

sta_Codigo	int
sta_Estr_Arg	varchar(4)

sintaxe2b

stb_Codigo	int
stb_Regencia	varchar(7)

verbo

vb_Codigo	int
vb_Pessoa	varchar(3)

Definição das Variáveis utilizadas

Variável	Descrição
_Classe	Classe gramatical a que tal entrada pertence
_Tipo	Classificação dentro de uma classe gramatical (ex: substantivo comum; locução prepositiva; artigo indefinido;)
_Codigo	Número de identificação da classificação de cada entrada
_Genero	Gênero da entrada
_Numero	Número da entrada
_Pessoa	Pessoa a que se refere tal entrada
cj_Papel	Papel exercido pela entrada
cl_Canonica	Forma Canônica de determinada entrada
cl_Item	Identificação da entrada na tabela "verbete"
pp_Tonicidade	Acento das entradas
mf_Atributo	Se a entrada é uma contração, uma raiz, um afixo, etc.
mf_Tipo	Se a entrada é um item simples ou composto
g2a_Modo	A que modo pertence aquele verbo
g2a_TEv	Tempo de Ocorrência do evento
g2a_TRef	Tempo de Referência do Evento
pp_Caso	Se nominativo, acusativo, ablativo ou dativo
sta_Estr_Arg	Número de argumentos exigidos por aquele item
st_List_Prep	Preposições que acompanham determinado item
st_Regencia	Se a regência direta ou indireta
vt_Palavra	Verbete propriamente dito

Símbolos utilizados no preenchimento das tabelas

Símbolos Possíveis	
Estrutural Argumental	0arg / 1arg / 2arg / 3arg
Regência	dir / ind / dir-ind
Tipo (Morfologia)	simpl / comp
Atributos (Morfologia)	r(st) / af-pr / af-su / r+af / contr / pal-comp
Tonicidade	ton / ato
Gênero	+m-f \ -m-f / +m+f / -m+f
Número	+s-p / -s-p / +s-p / -s+p
Pessoa	1ps / 2ps / 3ps / 1pp / 2pp / 3pp
Classe (grupo1)	subs / sigl / abrv / num / pron /adj / art / num / verb
Tipo (grupo1)	prop / com / mult /col / alg / dem / trat / indef / int/ pess/ simpl / loc-adj / +def / -def / card / ord / frac / frac-ord /poss / rel / partic
Caso (Pronome Pessoal)	nom / acu / abl / dat
Infinitivo (Verbo)	inf-pess
Tempo de Referência	pas / pre / fut
Tempo de Evento	pas / pre / fut
Modo	indic / subj / imper
Aspecto	+perf / -perf
Classe (grupo3)	prep / adv / interj / conj / verb

Tipo (grupo3)	simpl / loc-prep / loc-adv / gerun
Papel (Conjunção)	coord / subord

Definição dos símbolos utilizados para o preenchimento das tabelas

+def = +definido
-def = -definido

+m-f = +masculino-feminino
-m-f = -masculino-feminino
+m+f=+masculino+feminino
-m+f = -masculino+feminino

+n-v = +nominal-verbal
-n+v = -nominal+verbal
-n-v = -nominal-verbal
+n+v = +nominal+verbal

+perf = +perfeito
-perf = -perfeito

+s-p = +singular-plural
-s-p = -singular-plural
+s+p = +singular+plural
-s+p = -singular+plural

0arg = 0 argumentos
1arg = 1 argumentos
2arg = 2 argumentos
3arg = 3 argumentos

1ps = primeira pessoa do singular
2ps = segunda pessoa do singular
3ps = terceira pessoa do singular
1pp = primeira pessoa do plural
2pp = segunda pessoa do plural
3pp = terceira pessoa do plural

abl = ablativo
abrv = abreviatura
acu = acusativo
adj = adjetivo
adv = advérbio
af-pr = afixo-prefixo
af-su = afixo-sufixo
alg = algarismo
art = artigo
ato = átono
card = cardinal
col = coletivo
com = comum
comp = composto
conj = conjunção
coord = coordenativa
dat = dativo
dem = demonstrativo
dir = direta
frac = fracionário
frac-ord = fracionário e ordinal
fut = futuro
gerun = gerúndio
imper = imperativo
ind = indireta
indef = indefinido
indic = indicativo
inf-imp = infinitivo impessoal
inf-pess = infinitivo pessoal
int = interrogativo
interj = interjeição
loc-adv = locução adverbial
loc-conj = locução conjuncional
loc-prep = locução prepositiva
mult = multiplicativo
nom = nominativo
num = numeral
ord = ordinal
partic = particípio
pas = passado
pess = pessoal
poss = possessivo
pre = presente
prep = preposição
pron = pronome
prop = próprio
r(st) = raiz(stem)
r+af = raiz + afixos
rel = relativo
sigl = sigla
simpl = simples
subj = subjuntivo

subord = subordinativa
subs = substantivo
ton = tônico
trat = tratamento
verb = verbo

Alterações na Nomenclatura utilizada

Os dados utilizados para popular a BDL, inicialmente, foram os dados extraídos do léxico e, dessa forma, algumas siglas utilizadas para preencher os campos da base foram modificadas. Abaixo são apresentadas as siglas que eram usadas no léxico e como elas ficaram na BDL.

Símbolos usados no léxico	**Símbolos usados na BDL**
1S	1PS
2S	2PS
3S	3PS
1P	1PP
2P	2PP
3P	3PP
ABREV.	abrv
ADJ.	adj
ART.	art
BI.	3arg
CAR.	card
CONJ.	conj
COORD.	coord.
DE.	+def
DEM.	dem
F.	-m+f
FRA.	frac
FRA-ORD.	frac-ord
GERUN.	gerun
I.	indef
INDE	-def
INF-PESS	inf-pess
INT.	0arg
INTER.	int
INV.	-m-f
	-s-p
M.	+m-f
MUL.	mult
NOM	prop
ORD.	ord
PARTIC.	partic
PL.	-s+p
POSS.	poss
PREP.	prep
PRES.	pre
PRON.	pron
REL.	rel
S.	subs
SI.	+s-p
SIGL.	sigl

SUBORD.	subord
TD.	1arg
TI.	2arg
TRAT.	trat
V.	verb
2G.	+m+f
2N.	+s+p

Com a subdivisão das áreas de classificação (fonético-fonológico, morfologia, sintaxe, semântica, pragmático-discursivo) dos itens de entrada, os símbolos terminais apresentados abaixo assumem outros significados e representações.

3SP	não é mais utilizado
A/D/SU/N	a característica de grau não é mais analisada
ADIT./ADVE./ALTER./ CONCL./CAUS./COMP./ CONC./COND./CONFOR./ CONS./EXPL./FIN./ INTEG./PROPOR/TEMP	os complementos das conjunções serão representados na classificação semântica
C	contração será marcada na característica morfológica do item
AFIR/CIR-LUG/CIR-TEMP/ CIR-MOD/DUV/INT-CAUS/ INT-LUG/INT-MOD/ INT-LUG/INT-MOD/ INT-TEMP/NEG	as características dos advérbios serão expressas na morfologia de cada item
FUT-PRES./FUT-PRET FUT-SUBJ/PRES-SUBJ/ PRET-IMPERF-SUBJ PRET-IMPERF/PRET-M-Q-P/ PRET-PERF/IMP-AFIRM	as classificações dos verbos quanto a modo e tempo foram divididos em tempo de referência e tempo de ocorrência do evento
OBL-AT/OBL-TO	a caracterização a respeito da tonicidade será colocado em uma subclassificação própria

- Para pronomes que sejam oblíquos: marcar a tonicidade, se átono ou tônico, na tabela fonologia. Essa característica será separada em uma classificação própria.
Ignorar a classificação quanto a grau de qualquer item que apresente essa característica.
- A lista de regências dos itens que apresentam esta característica será colocada em uma tabela específica, a ser determinada.
- A contração das preposições será marcada na tabela morfologia.

- A lista de complementos das conjunções coordenativas e subordinativas será marcada na tabela semântica.
- Os pronomes não são mais classificados com retos e reflexivos
- Qualquer item que apresente a característica pessoa cujo valor seja igual à 3ps deve ter o valor alterado para 3pp ou 3ps (particular de cada caso).
- As formas nominais dos verbos serão preenchidas de acordo com cada tabela específica: os verbos no gerúndio devem ser colocados na tabela grupo3, os verbos no particípio devem ser colocados na tabela grupo1, os verbos no infinitivo pessoal e impessoal devem ser colocados na tabela grupo1. Os demais serão colocados na tabela grupo2.
- A predicação dos verbos será preenchida de acordo com o número de argumentos que o verbo pede: intransitivo - 0argumentos; transitivo direto - 1argumento; transitivo indireto - 2argumentos; bi-transitivo - 3argumentos. Essas informações serão colocadas na tabela sintaxe2a.
- Os tipos dos advérbios serão preenchidos de acordo com suas características semânticas.
- Os prefixos não são mais uma classe separada de palavras. Eles serão preenchidos quanto à morfologia de cada item, na tabela morfologia3.

Anexo II

Descrição da Implementação da Interface de Consultas via Web

A interface de consulta aos dados da BDL foi implementada em *Active Server Pages* (ASP), com auxílio da ferramenta *Microsoft Visual Interdev, 6.0*. Todas as variáveis utilizadas para o desenvolvimento da interface e o algoritmo de realização da interface estão descritos nessa seção. Os arquivos dessa interface estão armazenados na máquina usada para o desenvolvimento do projeto, localizado em E:\inetpub\wwwroot\FinalInterface. Este é o local em que os arquivos são acessados por qualquer usuário através da Internet.

Neste local estão os arquivos "***postform.htm***" e "***postvalues.asp***". O arquivo ***postform.htm*** é a página inicial, em que o usuário faz a escolha sobre a palavra que deve ser consultada e as informações que deseja obter. O arquivo ***postvalues.asp*** é responsável por todo o processamento de recuperação dos dados e apresentação das informações ao usuário.

Uma cópia local desses arquivos pode ser encontrada em c:\AcessoBDL\FinalInterface\FinalInterface_Local.

Descrição das Variáveis

Informações alteradas pelo usuário

strItemProcurado	Palavra digitada pelo usuário no formulário inicial
strInfMorfT	Opção sobre o usuário querer todas as informações morfossintáticas ou não
strInfMorfRC	Opção do usuário quere informações restritas por classe gramatical
strInfMorfAbrv	Opções de Classes
strInfMorfAdj	
strInfMorfAdv	
strInfMorfArt	
strInfMorfCnj	
strInfMorfNmr	
strInfMorfPrp	
strInfMorfPrn	
strInfMorfSgl	
strInfMorfSubs	
strInfMorfVrb	
strInfTrad	Opção de obter traduções do item selecionado

Informações recuperadas pelo sistema

StrCanonica	Armazena a forma canônica do item selecionado
StrClasse	Armazena a classe a que esse item pertence
StrGenero	Armazena o gênero do item selecionado
StrNumero	Armazena o número do item selecionado
StrTestaClasse	Variável usada para testar se um determinado item já foi classificado em uma certa classe gramatical
strTipo	Subclassificação do item dentro de uma classe gramatical
strTipoAux	Subclassificação para pronomes

strPessoa	Armazena informações sobre a pessoa
strPessoaAux	Variável auxiliar para a recuperação da pessoa de verbos infinitivo pessoais
strSQL	Variável usada para criar as consultas necessárias
strRegencia	Armazena informações sobre a regência dos substantivos, adjetivos e verbos infinitivos
strRegenciaVb	Armazena informações sobre a regência dos verbos em outras conjugações
strListPrep	Armazena a lista de preposições regidas pelo item selecionado
strTonicidade	Variável usada para obter a tonicidade dos pronomes pessoais
strModo	Armazena o modo em que determinado verbo é conjugado
strModoAux	Variável usada para montar a string final a respeito das informações de modo, tempo e pessoa de um verbo
strTempo	Armazena a string com o tempo de referência e de evento de um verbo
strPessoaVb	Variável auxiliar para a recuperação da pessoa para verbos do grupo2
strEstrArg	Armazena a estrutura argumental do item selecionado
strEstrArgAux	Variável auxiliar para recuperação da estrutura argumental
strMorfAtrib	Armazena os atributos morfológicos do item selecionado
strMorfComp	Os componentes de um item composto
strTEvAux	Armazena o tempo de ocorrência de um item
strTRefAux	Armazena o tempo de referência de um item
strClass3	Armazena a classe gramatical a qual um item, pertencente ao grupo3, está relacionado
strPapelConj	Armazena informações a respeito da papel desempenhado pelas conjunções
strPapelConjAux	Variável auxiliar para recuperação do papel das conjunções
strConjCompl	Armazena informações complementares sobre as conjunções
strConjTipo	Armazena a subclassificação das conjunções
intCodigo	Código de identificação do item selecionado
intCodigoAux	Verifica se tal código já foi processado
intGrupo	Grupo ao qual tal item está relacionado
c, i, j, k, x, y, l	Variáveis auxiliares, usadas como contadores para o processamento e recuperação das informações

Variáveis usadas para o preenchimento das *labels* de resposta

strTudo

intTodas
intAbrv
intAdj
intAdv
intArt
intConj
intNum
intPrep Variáveis usadas para a apresentação das informações recuperadas
intPron
intSg
intSub
intVb
intIngl
intModo
intTempo

Algoritmo Implementado

Início
Variáveis Contador, codigo

Leia a palavra a ser buscada
Leia que tipo de informação o usuário deseja
Se usuário deseja informações morfossintáticas então
Recupere o(s) código(s) da(s) classificação(ões) da palavra
Contador ← quantidade de códigos recuperados
Codigo ← primeiro codigo recuperado
Enquanto contador ≠ 0 faça
Recupere a classe gramatical para aquele codigo
Recupere as informações para a classe gramatical acima
Se usuário deseja receber informações sobre essa classe então
Escreva as informações recuperadas
Se usuário deseja receber informações de tradução então
Se houver tradução correspondente para a palavra selecionada de acordo com a classe gramatical acima, escreva a informação
Fim Se
Fim Se

contador ← contador – 1
codigo ← proximo codigo
Fim Enquanto
Fim Se
Senão Se usuário deseja receber informações sobre tradução então
Recupere todas as traduções possíveis para a palavra selecionada
Escreva as informações recuperadas

Fim Senão

Fim

Anexo III

Descrição da Implementação da Ferramenta de Geração de Listas Especializadas

A ferramenta de geração de listas foi implementada com a linguagem Delphi, 5.0 e ficará, em um primeiro momento, disponível somente aos integrantes do NILC. A idéia é que o usuário especialista possa se servir de listas personalizadas para o desenvolvimento de novas pesquisas na área. As listas geradas são armazenadas na máquina usada para o desenvolvimento do projeto, no diretório c:\ListasGeradas.

Descrição das Variáveis usadas

sCampo1 sCampo2 sCampo3 sCampo4 sCampo5 sCampo6 sCampo7 sCampo8 sCampo9 sCampo10	Variáveis usadas para armazenar as informações recuperadas antes que sejam tratadas e escritas no arquivo final
sClasse	Armazena o símbolo a ser escrito no arquivo de acordo com a classe selecionada pelo usuário
sGenero	Armazena a informação já tratada sobre o gênero gênero da palavra, que deverá ser escrita no arquivo final
sNumero	Armazena a informação já tratada sobre o número número da palavra, que deverá ser escrito no arquivo final
sInicio	Letra inicial do intervalo em que as palavras devem estar
sFim	Letra final do intervalo
sPessoa	Armazena a pessoa dos pronomes
sTonicidade	Armazena a tonicidade dos pronomes pessoais
sCampoCod	
sTipo	Armazena informação sobre a subclass. da palavra em determinada classe gramatical
sTransit	Armazena a transitividade dos verbos
sCampoTeste sCampoAux	Variáveis usadas no processamento de listas exclusivas
sTempo sTempoAtual sTempoAnterior	Variáveis usadas para proc do tempo de verbos
sEncMeso	Armazena informações sobre ênclise/mesóclise

sRegAtual sRegAnt sRegencia	Variáveis usadas para proc. a regência dos verbos
sPessVb	Armazena a pessoa do verbo inf. sem tratamento
sPessInf	Armazena a pessoa do verbo inf tratada, pronta para ser escrita no arquivo final
CodigoInicial CodigoFinal	Variáveis que marcam um bloco de palavras a serem recuperadas. Usadas para tratamento de verbos
CodMenor CodMenorAuxiliar	Variável usada para controle prioridade entre as várias classificações de uma palavra na geração e listas compostas
count countAux	Variáveis usadas no controle dos diversos laços usados no programa
Lista	Arquivo criado para armazenar os verbetes selecionados
Teste teste1	Booleanos usados para verificar se as informações recuperadas sobre uma palavra já foram todas analisadas
strSel strJoin strCond strWhere strSQL	Variáveis usadas para montar a consulta necessária para geração de listas compostas
strTeste	
sPalavra	Armazena a palavra recuperada
sAbrev sAdj sAdv sArt sConj sNum sPrep sPron sSigl sSub	Variáveis usadas para armazenar a cadeia de informações que devem ser escritas no arquivo texto

sFinal	
strAuxiliar	
sItem	Armazena o código de identificação da palavra em questão
sCodAbrev	
sCodAdj	
sCodAdv	
sCodArt	Variáveis usadas para controle de prioridade
sCodConj	entre as diversas classificações de uma palavra
sCodNum	
sCodPrep	
sCodPron	
sCodSigl	
sCodSub	

Algoritmo

Início

Verifica se há alguma restrição sobre o gênero ou número das palavras a serem recuperadas
Verifica se foi especificado um intervalo no qual as palavras devem estar inseridas
Se a lista a ser gerada é uma lista simples então
início
Verifica qual a classe gramatical a ser considerada
Recupera todas as informações das palavras que satisfazem as condições inicias
Escreve as informações, linha a linha, em um arquivo tipo texto
fim

Se a lista a ser gerada é composta então
início
Selecione todas as palavras que pertencentes à primeira classe gramatical selecionada e que satisfaçam as condições iniciais

Enquanto todas as classes selecionadas não forem consideradas faça
Selecione, do conjunto resultante da seleção anterior, as palavras pertencentes à próxima classe gramatical selecionada

Recupere todas informações do conjunto de palavras resultante da seleção realizada
Escreva as informações, linha a linha, em um arquivo tipo texto
fim

Fim