

# Consultas em Bancos de Dados Utilizando Linguagem Natural

**Renato Rocha Silva, Sérgio Muinhos Barroso Lima**

renatorrs2004@yahoo.com.br; smblima@grabnery.edu.com

*Faculdade Metodista Granbery*

*Rua Batista de Oliveira, 1145 – CEP 36010-530 – Juiz de Fora/MG*

**Abstract.** The Natural Language (LN) is an alternative way for database queries, used for computer naive users, that probably have difficulties to use query languages such SQL or similar. This work presents natural language processing (PLN) techniques, its applications, advantages, disadvantages and a syntactic parser implementation.

Key words: natural language, queries, database, context-free grammars, SQL

**Resumo.** A Linguagem Natural (LN) é uma alternativa para consultas em bancos de dados, principalmente para usuário leigos em computação, que provavelmente têm dificuldades em utilizar linguagens de consulta como o SQL ou similares. Este trabalho apresenta, então, as técnicas para o processamento de linguagem natural (PLN) mostrando suas principais aplicações, vantagens, desvantagens e a implementação de um analisador sintático exemplo.

Palavras Chave: linguagem natural, consultas, bancos de dados, gramáticas livres de contexto, SQL.

## INTRODUÇÃO

O Processamento de Linguagem Natural é uma área da Inteligência Artificial (IA) utilizada para a compreensão e produção de linguagem natural, como por exemplo o português e o inglês.

As facilidades que a linguagem natural pode disponibilizar para consultas em bancos de dados (BD) são inúmeras: o usuário não precisa saber detalhes sobre a estrutura do (BD), nem o conhecimento de uma linguagem para consultas em bancos de dados (SQL) para

realizar suas consultas, ou seja, numa elaboração convencional de uma consulta a um BD, o usuário deve ter conhecimento sobre as tabelas, campos, seus relacionamentos e, além disso, ter que utilizar uma linguagem para consulta com regras de sintaxe rígidas, artificiais e de semântica limitada. Além disso, através da aplicação de técnicas de PLN é possível o entendimento de consultas com erros (termos digitados erroneamente) e incompletas, nesses casos o PLN busca por palavras próximas (em forma e/ou significado) e pelo contexto da conversação.

Com a LN, o usuário deve conhecer somente do que se trata o BD, ou seja, ter conhecimentos acerca do domínio do BD. Por exemplo, em um banco de dados acadêmico, o usuário poderia formular a seguinte questão:

“Quais são os nomes dos alunos matriculados e os cursos que estudam”?

Em SQL: “select aluno.nome, curso.nome from aluno, curso  
where aluno.idcurso = curso.idcurso”

Nota-se que, para a formulação da consulta em LN, não é necessário que o usuário saiba detalhes sobre a estrutura do BD (meta-informação); já em SQL essas informações são cruciais, inclusive a notação exata dos termos no BD (campos e tabelas) além da construção da junção relacional (*join*) e da sintaxe rígida da linguagem SQL.

O PLN também possui algumas desvantagens, pois como o usuário pode criar suas próprias consultas e não terá um padrão para seguir, poderão ocorrer resultados que não irão satisfazer a expectativa do usuário e erros na passagem da LN para uma linguagem de consulta, portanto, o usuário pode criar uma resistência ao uso da aplicação ou, até mesmo, não acreditar nas respostas geradas pela mesma.

O objetivo deste trabalho é apresentar técnicas de PLN para consultas em BD, com a implementação, em PROLOG, de um analisador sintático. Para exemplificar algumas dessas técnicas, e implementar o analisador sintático, será utilizado como exemplo o BD acadêmico da Faculdade Metodista Granbery (FMG) com informações sobre alunos, currículos, cursos e estágios.

Este artigo está organizado da seguinte maneira: após esta introdução, são apresentados conceitos sobre o PLN: histórico, principais aplicações e técnicas; a seguir apresenta-se as gramáticas livres de contexto (CFG); a implementação de um analisador sintático implementado em PROLOG e, por fim, as conclusões do trabalho.

## PROCESSAMENTO DE LINGUAGEM NATURAL

Os primeiros trabalhos de tratamento da LN com implementações informatizadas surgiram no início da década de 50 (Santos, 2003).

A partir da década de 60, surgiram várias aplicações com compreensão da LN, nesta época, já existiam aplicações capazes de aceitar e de responder a questões em inglês, aplicações que simulavam entrevistas psiquiátricas e várias outras aplicações que auxiliavam nas mais diferentes áreas: álgebra, medicina, relações de parentesco, etc. (Abreu, 2002).

O programa que mais se destacou nesta época foi o Eliza (desenvolvido por Joseph Weizenbaum, em 1966), este programa simulava ser um psiquiatra onde o usuário interagía com a aplicação escrevendo sobre seus problemas, todavia é importante mencionar que o programa não tinha compreensão do tema que estava sendo abordado, ele apenas utilizava truques semânticos (Abreu, 2002).

A aplicação “ed” (<http://www.inbot.com.br/ed/>) é um robô virtual da Petrobrás (figura 1) implementado para conversação de forma similar ao Eliza, porém utilizando modernas técnicas de PLN. As respostas às consultas realizadas (figura 1) contém ainda *hiperlinks* para os assuntos-chave, aumentando as possibilidades de recuperação semântica (CONPET, 2007).

Para que um sistema computacional interprete uma sentença em linguagem natural, é necessário manter informações morfológicas (léxicas), sintáticas (regras gramaticais) e semânticas (significados).

O tópico a seguir ilustra alguns conceitos da Língua Portuguesa essenciais para o entendimento do trabalho.

- **Conceitos da Língua Portuguesa**

A implementação de processadores de LN requer conhecimentos sobre a língua utilizada na comunicação entre o usuário e o sistema computacional, no caso deste trabalho, a Língua Portuguesa. A seguir, alguns conceitos importantes da língua Portuguesa, no âmbito deste trabalho, são apresentados (Cunha, 1982 e Bechara, 1999).



**Figura 1.** Interface da aplicação “ed” – robô virtual da Petrobrás. Fonte CONPET(2007)

**Oração:** é uma frase constituída de sujeito e predicado, ou apenas predicado.

**Frase:** é um conjunto de palavras que formam o sentido completo (sentença).

**Adjunto Adverbial:** denota alguma circunstância do fato expresso pelo verbo, ou intensifica o sentido deste.

**Adjunto Adnominal:** serve para especificar ou delimitar o significado de um substantivo.

**Aposto:** é o termo que se junta a um substantivo, a um pronome, ou a um equivalente destes, a título de explicação.

**Preposição:** são os vocábulos que relacionam dois termos de uma oração, de tal modo que o sentido do primeiro (antecedente) é explicado ou completado pelo sentido do segundo (conseqüente).

**Período:** é um conjunto de orações que formam sentido completo.

**Período Simples:** frase constituída de uma só oração.

**Período Composto:** frase constituída de duas ou mais orações.

**Composição por Subordinação:** são orações sem autonomia gramatical, isto é, as orações que funcionam como essenciais, integrantes ou acessórios de outra oração.

**Composição por Coordenação:** são as orações que têm sentido próprio.

**Sintagma:** seqüência de elementos lingüísticos relacionados entre si.

**Sintagma Nominal:** conjunto de substantivos e seus adjuntos.

**Sintagma Verbal:** conjunto de verbos e seus adjuntos.

**Sintagma Preposicional:** são grupos preposicionais não ligados, independentes da noção de regência.

**Sintagma Adjetival:** formado por adjetivo ou grupos de adjetivos.

**Sujeito:** termo sobre o qual se faz uma declaração.

**Predicado:** tudo aquilo que se diz do sujeito.

**Complemento Verbal:** palavras que integram o sentido do verbo.

**Complemento Nominal:** palavras que completam o sentido de substantivo, adjetivo ou advérbio.

- **Análise Morfológica - vocabulário**

Análise Morfológica é o estudo da estrutura, formação, flexão e classificação das palavras (Cunha, 1982).

O analisador morfológico identifica palavras ou expressões isoladas em uma sentença, sendo este processo auxiliado por delimitadores (pontuação e espaços em branco, por exemplo). As palavras identificadas são classificadas de acordo com seu tipo de uso ou, em linguagem natural, a categoria gramatical (Abreu, 2002).

- **Análise Sintática - gramática**

Análise Sintática é uma técnica empregada no estudo da estrutura de uma sentença (Cunha, 1982).

É um passo importante para o entendimento (semântica) de uma sentença em LN. Somente vocábulos não garantem o entendimento de uma sentença, é importante que a sua estrutura sintática seja analisada (LIMA, 1997). Por exemplo:

A sentença: “sorrindo Maria João achou“.

Possui vocábulos corretos, porém sinteticamente errada e sem formar um significado claro.

Já a sentença: “Maria achou o João sorrindo”.

Está sintática e morfológicamente correta, o que leva ao entendimento da sentença, ou seja, ela atende as regras gramaticais e ao vocabulário da língua portuguesa.

A análise sintática de uma oração em português deve levar em conta os seguintes sintagmas: termos essenciais (sujeito e predicado), termos integrantes (complementos verbal e nominal) e termos acessórios (adjunto adverbial, adjunto adnominal e aposto). A análise do período, por sua vez, deve considerar o tipo de período (simples ou composto), sua composição (por subordinação, por coordenação) e a classificação das orações (absoluta, principal, coordenada ou subordinada) (Abreu, 2002).

As seguintes gramáticas são as mais utilizadas para analisadores sintáticos (LIMA, 1997 e PEREIRA, 1980):

- gramáticas livres de contexto: são muito úteis no que tange à descrição de gramáticas em linguagem natural, permitindo a representação de linguagens com pequeno grau de complexidade. No entanto, a dificuldade em expressar dependências simples (concordância verbal e nominal) constitui um dos maiores problemas para sua utilização no tratamento da língua natural. Abordagens puramente livres de contexto (não suporta a concordância ou a contextualização em gênero, número e grau, por exemplo) não são suficientemente poderosas para captar a total complexidade das linguagens naturais, porém, devido a sua simplicidade, são muito utilizadas na prática (Abreu, 2002); e

- gramáticas sensíveis ao contexto: os problemas de dependência expressos anteriormente (concordâncias) são resolvidos nesta classe de gramática. Para implementar essas gramáticas utiliza-se uma notação denominada como *Definite Clause Grammar* (DCG).

Este trabalho ilustra um analisador sintático implementado com gramática livre de contexto.

- **Análise Semântica**

Análise Semântica é o estudo do significado das palavras e sentenças (Cunha, 1982).

Na língua portuguesa podemos ter palavras, componentes da palavra e frases com vários significados, portanto, a questão da representação do significado apresenta diversas dificuldades (Abreu, 2002).

Como exemplo podemos destacar (Abreu, 2002):

- componentes de uma palavra. Ex: “texto”, “hipertexto”. O adjetivo “hiper” significa que o “hipertexto” contém textos;

- questão da ambigüidade. Ex: “Manga”. O substantivo “manga” tanto pode significar manga de uma camisa, como pode significar a fruta manga; e

- diferenciação entre significado e sentido. Ex: “Casa”, “minha casa”. Podemos verificar que o substantivo casa tem o significado diferenciado quando acrescentamos o pronome possessivo “minha”, pois assim estamos referindo a casa de uma determinada pessoa.

Após o analisador sintático reagrupar as estruturas das palavras e o analisador morfológico identificar as palavras individualmente, o analisador semântico analisa o sentido das estruturas das palavras (Abreu, 2002).

## **GRAMÁTICAS LIVRE DE CONTEXTO (CFG)**

As CFG foram inicialmente desenvolvidas por lingüistas para representar linguagens naturais (UFRN, 2003). São gramáticas de fácil implementação, porém não conseguem tratar uma linguagem natural em toda a sua complexidade

As CFG são gramáticas com regras da forma  $A \rightarrow \beta$ , onde  $\beta$  é uma seqüência qualquer de símbolos não terminais e terminais, possivelmente vazia, A é um símbolo não terminal (Russel, 2004), como ilustra o exemplo a seguir.

Sejam os seguintes símbolos:

s – Sentença;

sn – Sintagma Nominal;

sv – Sintagma Verbal;

sa – Sintagma Adjetival;

sp – Sintagma Preposicional;

suj – Sujeito;

adj – Adjetivo;

adv – Advérbio;

det – Determinante;

v – Verbo;

n – Substantivo;

prep – Preposição;

pi – Pronome Interrogativo; e

pp- Pronome Possessivo.

Dados esses símbolos, podemos escrever uma gramática livre de contexto que descreve a estrutura sintática de um pequeno subconjunto do Português (Garcia, 2005).

$s \rightarrow sn, sv$

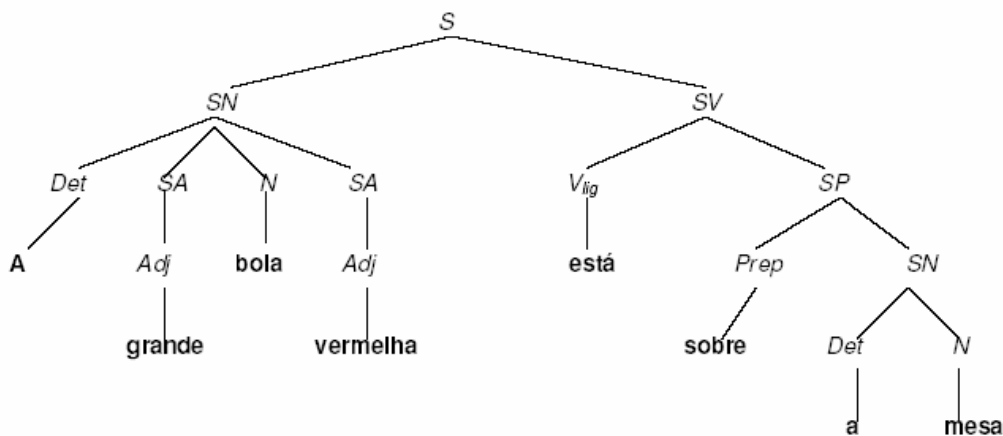
$sn \rightarrow det, n$

sn → det, sa, n  
 sn → det, n, sa  
 sn → det, sa, n, sa  
 sa → adj, sa  
 sa → adj  
 sv → v, sp  
 sp → prep, sn  
 v → é, está...  
 det → o, a, os, as, um...  
 prep → em, sobre, para...  
 adj → grande, pequeno, ...  
 n → bola, mesa, João, Maria, ...

Com essa gramática podemos derivar sentenças como “A grande bola vermelha está sobre a mesa” como na Figura 2 (Garcia, 2005).

Um problema imediato com esta gramática é a concordância (contexto de concordância em número). Essa gramática gera sentenças do tipo “Os grande bola vermelhas estão sobre as mesa”. Deve-se, portanto, incluir as concordâncias de gênero e número (Garcia, 2005).

A figura 2 mostra a árvore de derivação que é gerada pela CFG, onde estão definidas a estrutura gramatical da sentença (gramática) e a classificação das palavras (vocabulário) (Garcia, 2005).



**Figura 2.** Árvore de derivação para “A grande bola vermelha está sobre a mesa”.



Dentre as diversas aplicações do CFG podemos destacar (UFRN, 2003) que a sua principal aplicação é na análise sintática de linguagens de programação, pois não são necessárias concordâncias em gênero, número e grau, ou seja, a gramática é simples e restrita.

## **IMPLEMENTANDO UMA CFG PARA ANÁLISE DE CONSULTAS EM UM BD**

Nas aplicações tradicionais, as consultas em um BD são, geralmente, muito limitadas. Assim, o usuário só tem a possibilidade de realizar consultas pré-definidas, a menos que o usuário defina uma cláusula SQL o que não é aconselhado para usuários leigos. Com a implementação de um tradutor de LN, o usuário poderá criar suas próprias consultas em linguagem natural, assim o tradutor verificará a sentença e a transformará em uma consulta SQL.

Esta possibilidade do usuário criar suas próprias consultas é exatamente o ponto forte da LN, onde pode-se criar consultas de acordo com a necessidade sem a limitação das consultas já pré-definidas pelo programador.

A aplicação proposta neste trabalho pretende implementar um analisador sintático livre de contexto, na linguagem de programação PROLOG, para consultar um banco de dados de currículo que tratará algumas das possíveis consultas sobre esse BD.

- **A Linguagem PROLOG**

A linguagem PROLOG foi criada em 1972 por Colmerauer e Roussel, sendo sua primeira versão feita em Fortran por Battani e Meloniem 1973. Desde então tem sido utilizada para aplicações relacionadas à IA, como compreensão de linguagem natural, automação de projetos, sistemas especialistas, dentre outros (Pallazo, 1997).

Algumas versões (Arity Prolog) implementam diretamente as gramáticas livres de contexto e as gramáticas de cláusulas definidas (LIMA, 1997).

- **O Banco de Dados de Currículos da FMG**

O BD de Currículos da Faculdade Metodista Granbery é acessível via Web, onde os alunos cadastram seus currículos, consultam as vagas disponíveis, e também entram em contato com as instituições. As instituições podem se cadastrar no sistema, disponibilizar vagas, consultar os currículos dos alunos e entrar em contato com os alunos.

Na figura 3, podemos verificar que, atualmente, a consulta de currículos feita pelas instituições são pré-definidas pelo programador, ou seja, apesar de o usuário não precisar usar SQL (que está embutido na interface gráfica) ele fica limitado à rígidas opções de consulta.

Faculdade Metodista Granbery  
SAAP - Módulos auxiliares

Manutenção Banco de Currículos Estágio Supervisionado Relatórios

Home :: [Relação de Currículos](#) :: Pesquisa de Currículo

**Pesquisa de Currículo**

Dados Pessoais  
Faixa Etária: 18 - 25 Sexo: Masculino Estado Civil: Casado  
Logradouro: Bairro:  
Cidade: Estado: --Selecione--

Formação Acadêmica  
 Graduação  Especialização  Mestrado  Doutorado  
Cursos: ADMINISTRAÇÃO Período Inicial: --Selecione-- Período Final: --Selecione-- Ano Formatura: Semestre Formatura: Primeiro Semestre

Dados do Currículo  
Conhecimento e Habilidades (Pesquisa por palavra chave): Cursos Extras (Pesquisa por palavra chave):  
Tempo Experiência (Mês) Idiomas (Pesquisa por palavra chave):

Pesquisar Cancelar

Coordenador: ALESSANDRÉIA MARTA DE OLIVEIRA Curso: SISTEMAS DE INFORMAÇÃO Tempo de geração: 0.246 sec

**Figura 3.** Tela de Consulta ao BD de Currículos da FMG

O BD de Currículos da FMG armazena informações sobre: instituição, alunos, currículos, cursos, histórico profissional, idioma, vagas de estágio, coordenadores de estágio e professores.

- **Gramática e Vocabulário**

A seguir, apresenta-se algumas consultas identificadas para serem realizadas no BD de currículos, sua definição gramatical e seu vocabulário.

1: Quais são as instituições cadastradas?

**Gramática:**

s → sn, sv

sn → suj

suj → pi

sv → v, sn

sn → det, n, adj

**Vocabulário:**

pi (Indefinido) → Quais

v → são

det → as

n → instituições

adj → cadastradas

2: Quais são os alunos cadastrados?

**Gramática:**

s → sn, sv

sn → suj

suj → pi

sv → v, sn

sn → det, n, adj

**Vocabulário:**

pi (indefinido) → Quais

v → são

det → os

n → alunos

adj → cadastrados

3: Quais são os currículos disponibilizados pelos alunos?

**Gramática:**

s → sn, sv

sn → suj

suj → pi

sv → v, det, n, adj, sp

sp → prep, n

**Vocabulário:**

pi (indefinido) → Quais

v → são

det → os

n → currículos

adj → disponibilizados

prep (contração per + artigo os) → pelos  
n → alunos

4: Quais são as vagas disponibilizadas pelas instituições?

**Gramática:**

s → sn, sv

sn → suj

suj → pi

sv → v, det, n, adj, sp

sp → prep, n

**Vocabulário:**

pi (indefinido) → Quais

verbo → são

det → as

n → vagas

adj → disponibilizadas

prep (contração per + artigo os) → pelas

n → instituições

5: Quais instituições não possuem vagas disponíveis?

**Gramática:**

s → sn, sv

sn → suj

suj → pi, n

sv → adv, v, sn

sn → n, adj

**Vocabulário:**

pi (indefinido) → quais

n → instituições

adv (negação) → não

v → possuem

n → vagas

adj → disponíveis

6: Quais alunos não possuem currículo cadastrado?

**Gramática:**

s → sn, sv

sn → suj

suj → pi, n

sv → adv, v, sn

sn → n, adj

**Vocabulário:**

pi (indefinido) → Quais

n → alunos

adv (negação) → não

v → possuem

n → currículo

adj → cadastrado

7: Quais cursos não possuem vagas disponíveis?

**Gramática:**

s → sn, sv

sn → suj

suj → pi, n

sv → adv, v, sn

sn → n, adj

**Vocabulário:**

pi (indefinido) → Quais

n → cursos

adv (negação) → não

v → possuem

n → vagas

Adj → disponíveis

8: Quais alunos não possuem curso extra em seu currículo?

**Gramática:**

s → sn, sv

sn → suj

suj→ pi, n

sv→ adv, v, sn, sp

sn→ n, adj

sp→ prep, pp, n

**Vocabulário:**

pi (indefinido) → quais

n→ alunos

adv (negação) → não

v→ possuem

n→ curso

adj→ extra

prep→ em

pp→ seu

n→ currículo

9: Quais alunos não possuem Histórico Profissional em seu currículo?

**Gramática:**

s→ sn, sv

sn→ suj

suj→ pi, n

sv→ adv, v, sn, sp

sn→ n, adj

sp→ prep, pp, n

**Vocabulário:**

pi (indefinido) → quais

n→ alunos

adv (negação)→ não

v→ possuem

n→ histórico

adj→ profissional

prep→ em

pp→ seu

n→ currículo

10: Quais alunos não possuem idioma em seu currículo?

**Gramática:**

s → sn, sv

sn → suj

suj → pi, n

sv → adv, v, sn, sp

sn → n

sp → prep, pp, n

**Vocabulário:**

pi (indefinido) → Quais

n → alunos

adv (negação) → não

v → possuem

n → idioma

prep → em

pp → seu

n → currículo

- **Implementação no PROLOG**

Definida a gramática e o vocabulário das consultas, o próximo passo é implementar a análise sintática no PROLOG que mapeia diretamente as CFG, através da criação das árvores de derivação, testando se as sentenças são válidas ou não, ou seja, se atendem às regras gramaticais e ao vocabulário definidos. Caso a sentença for válida, o PROLOG retornará “YES” e, se a sentença não for válida, o PROLOG retornará “NO”.

A figura 5 representa a gramática implementada no Arity PROLOG. Já a figura 6 representa o vocabulário implementado no PROLOG.

Após a implementação da gramática e do vocabulário foram feitos testes no PROLOG sobre as consultas definidas.

```

File Edit Buffers Info Debug Switch Help
ROW:00001 COL:001 INSERT E:\ARITY32\ARITY32\BIN\MONOGRFIA.ARIBUFFER 1
s-->sn,sv.
sn-->suj.
suj-->pi.
sv-->v,sn.
sn-->det,n,adj.

s-->sn,sv.
sn-->suj.
suj-->pi.
sv-->v, det, n, adj, sp.
sp-->prep,n.

s-->sn,sv.
sn-->suj.
suj-->pi,n.
sv-->adv,v,sn.
sn-->n,adj.

s-->sn,sv.
sn-->suj.
suj-->pi,n.
sv-->adv,v,sn,sp.
sn-->n,adj.
sp-->prep,pp,n.

s-->sn,sv.
sn-->suj.
suj-->pi,n.
sv-->adv,v,sn,sp.
sn-->n.
sp-->prep,pp,n.

```

Figura 5. Gramática Implementada no Arity PROLOG

```

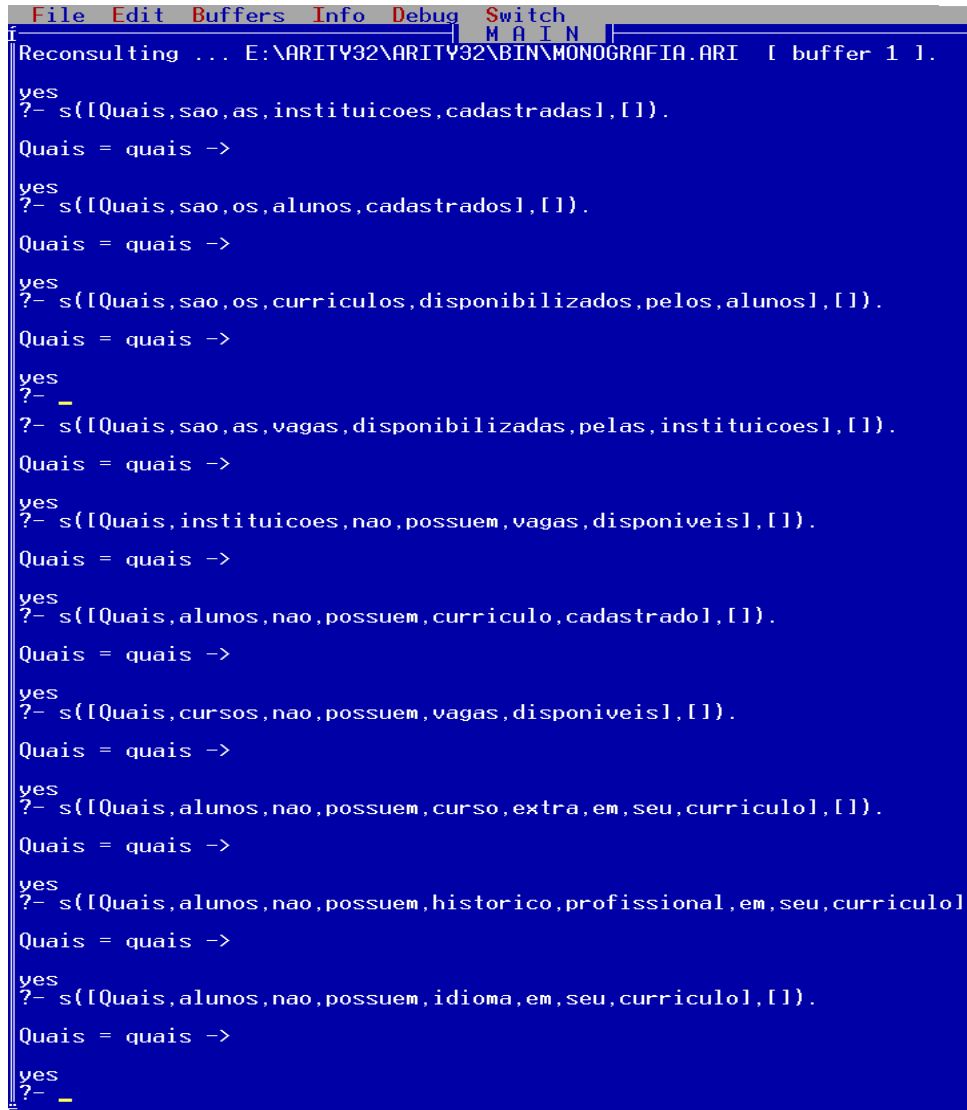
pi-->[quais].
v-->[sao].
v-->[possuem].
det-->[as].
det-->[os].
n-->[instituicoes].
n-->[alunos].
n-->[curriculos].
n-->[curriculo].
n-->[vagas].
n-->[curso].
n-->[cursos].
n-->[historico].
n-->[idioma].
adj-->[cadastradas].
adj-->[cadastrados].
adj-->[cadastrado].
adj-->[disponibilizados].
adj-->[disponibilizadas].
adj-->[disponiveis].
adj-->[extra].
adj-->[profissional].
prep-->[pelos].
prep-->[pelas].
prep-->[em].
adv-->[nao].
pp-->[seu].

```

Figura 6. Vocabulário Implementado no PROLOG



A figura 7 mostra os testes feitos no PROLOG, onde os resultados foram “YES”; portanto, a sentença foi validada.



```
File Edit Buffers Info Debug Switch
M A I N
Reconsulting ... E:\ARITY32\ARITY32\BIN\MONOGRAFIA.ARI [ buffer 1 ].
yes
?- s([Quais,sao,as,instituicoes,cadastradas],[ ]).
Quais = quais ->
yes
?- s([Quais,sao,os,alunos,cadastrados],[ ]).
Quais = quais ->
yes
?- s([Quais,sao,os,curriculos,disponibilizados,pelos,alunos],[ ]).
Quais = quais ->
yes
?- s([Quais,sao,as,vagas,disponibilizadas,pelas,instituicoes],[ ]).
Quais = quais ->
yes
?- s([Quais,instituicoes,nao,possuem,vagas,disponiveis],[ ]).
Quais = quais ->
yes
?- s([Quais,alunos,nao,possuem,curriculo,cadastrado],[ ]).
Quais = quais ->
yes
?- s([Quais,cursos,nao,possuem,vagas,disponiveis],[ ]).
Quais = quais ->
yes
?- s([Quais,alunos,nao,possuem,curso,extra,em,seu,curriculo],[ ]).
Quais = quais ->
yes
?- s([Quais,alunos,nao,possuem,historico,profissional,em,seu,curriculo],[ ]).
Quais = quais ->
yes
?- s([Quais,alunos,nao,possuem,idioma,em,seu,curriculo],[ ]).
Quais = quais ->
yes
?- -
```

**Figura 7.** Teste no PROLOG das consultas com resultado “YES”

A figura 8 mostra os testes feitos no PROLOG, onde os resultados foram “NO”; portanto, a sentença não foi validada, ou seja, não reconhecida pela gramática definida

```
File Edit Buffers Info Debug Switch
M A I N
?- s([Quais, sao, as, empresas, cadastradas],[ ]).
no
?- s([Quais, instituicoes, nao, possui, vagas, disponiveis],[ ]).
no
?- s([Quais, as, intituicoes, cadastradas],[ ]).
no
?-
```

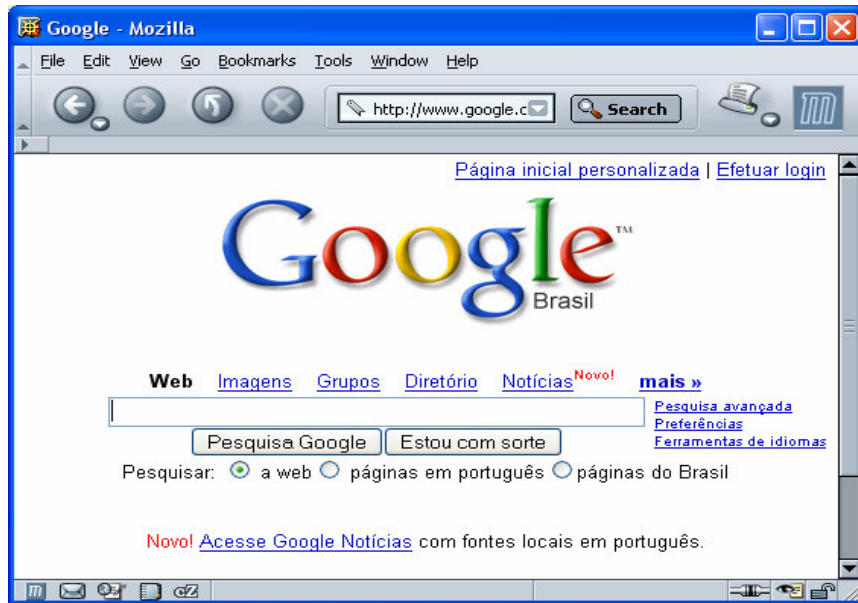
**Figura 8.** Teste no PROLOG das consultas com resultado “NO”

Pode-se verificar, na figura 8, que as sentenças não foram validadas porque a palavra “empresas” na primeira sentença e a palavra “possue” na segunda sentença não existem no vocabulário definido na figura 6. A terceira sentença não foi válida, porque não atendeu as regras gramaticais definidas na figura 5, além do erro ortográfico em “intituicoes”. Para sanar esses problemas a gramática teria que ser expandida e um corretor ortográfico automático poderia ser implementado.

## CONSIDERAÇÕES FINAIS

Este trabalho procurou mostrar, de uma maneira simples, que a linguagem natural pode trazer benefícios para implementações de aplicações inteligentes, com maior interação com o usuário, principalmente no caso e consultas a BD. A flexibilização das consultas é a principal vantagem da utilização da LN, livrando o usuário, principalmente os leigos, de ter que dominar linguagens formais de consulta como o SQL, ou de ter que estar limitado à rígidas opções de consultas pré-determinadas.

Uma outra tendência do PLN é a implementação de aplicações de interfaces multimodais, que utilizam a junção de técnicas de manipulação direta (consultas pré-definidas) e a LN (Kirner, 2004 e Cohen, 1992). Portanto, pode-se ter *softwares* que possuem consultas e técnicas já pré-definidas, porém com uma interação inteligente e uma maior flexibilidade com a utilização do PLN. Exemplos de interfaces multimodais são os sites de busca na internet (figura 9), porém ainda sem capacidades de interação inteligente, ou seja, não possuem capacidades de PLN.



**Figura 9.** Interface multimodal do buscador Google

## **AGRADECIMENTOS**

Agradecemos à Faculdade Metodista Granbery pelo apoio à pesquisa, com seus inerentes benefícios ao processo ensino-aprendizagem.

## **REFERÊNCIAS BIBLIOGRÁFICAS**

- ABREU, F.D.E. **Processamento da Linguagem Natural: princípios básicos e a implementação de um analisador sintático de sentenças da língua portuguesa.** Revista de Ciência da Informação. Rio de Janeiro. n. 5. Maio 2002.
- BECHARA, E., **Moderna Gramática Portuguesa.** Rio de Janeiro: Lucerna, 1999.
- CUNHA, C.F., **Gramática da Língua Portuguesa,** 7. ed. Rio de Janeiro: Fename, 1982.
- COHEN, P. R. **The Role of Natural Language in a Multimodal Interface.** Proceedings of the ACM Symposium on User Interface Software and Technology. 1992.
- GARCIA, J.L.R., **Linguagens Livres de Contexto e Autômatos de Pilha,** Monografia de Final de Curso. Curso Bacharel em informática. Universidade Federal do Piauí, 2005.
- KINER, C. **Realidade Virtual: Conceitos e Tendências.** São Paulo, 2004.
- LIMA, S. M. B. **Implementando um Tradutor de Linguagem Natural para a Linguagem Legal.** Dissertação de Mestrado, UNICAMP, 1997.

PALLAZO, L.M.A. **Introdução à Programação PROLOG**. Pelotas: UCPEL, n. 1. 1997.

PEREIRA, F. C. N. e WARREN, H. D. **Definite Clause Grammars for Language Analysis – a Survey of the Formalism and a Comparison with Augmented Transition Networks**.

Artificial Intelligence, vol. 13. North-Holland Publishing Company, 1980.

CONPET Programa Nacional de Racionalização do uso dos Derivados de Petróleo e do Gás Natural. **Robô Virtual da Petrobrás**. Disponível em <http://www.inbot.com.br/ed/>. Último acesso em 01 de abril de 2007.

RUSSEL, Stuart J., **Inteligência Artificial**. Rio de Janeiro: Elsevier, n. 2. 2004.

SANTOS, D. **Introdução ao Processamento de Linguagem Natural Através das Aplicações**. Revista de Ciência da Informação. Rio de Janeiro. n. 1. Fevereiro 2002.

UFRN. **Gramáticas Livre de Contexto**. Disponível em: <http://www.consiste.dimap.ufrn.br/~david/enseignement/2003.1/dim0330> Último Acesso em: 30 out. 2005.