INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO CEARÁ

PRÓ-REITORIA DE ENSINO

COORDENADORIA DE CIÊNCIA DA COMPUTAÇÃO DO CAMPUS ARACATI

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Cynthia Maria Bravo Pinto

UMA ABORDAGEM PARA CONVERSÃO DE CONSULTAS DE BANCO DE DADOS NO IDIOMA PORTUGUÊS PARA CONSULTAS EM SQL

ARACATI-CE

2017

Cynthia Maria Bravo Pinto

UMA ABORDAGEM PARA CONVERSÃO DE CONSULTAS DE BANCO DE DADOS NO IDIOMA PORTUGUÊS PARA CONSULTAS EM SQL

Trabalho de conclusão de curso apresentado à Coordenadoria de Ciência da Computação do Instituto Federal de Educação, Ciência e Tecnologia do Ceará - Campus Aracati como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Área de pesquisa: Inteligência Artificial, Banco de Dados.

Orientadora: Msc. Francisca Raquel de Vasconcelos Silveira

Aracati-CE
2017

Dados Internacionais de Catalogação na Publicação (CIP)

P659a Pinto, Cynthia Maria Bravo.

Uma abordagem para conversão de consultas de Banco de Dados no idioma Português para consultas em SQL./ Cynthia Maria Bravo. – Aracati: IFCE, 2017. 75f.:

Orientador: Profª. Msc. Francisca Raquel de Vasconcelos Silveira.

Monografia (Graduação em Ciência da computação) – IFCE.

1. Processamento da Linguagem Natural (PLN). 2. Interface em Linguagem Natural para Banco de Dados (ILNBD). 3. Linguagem de Consulta Estruturada (SQL).. I. Título.

IFCE/BIBLIOTECA/ARACATI CDD: 005.7

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO CEARÁ
COORDENADORIA DE CIÊNCIA DA COMPUTAÇÃO DO CAMPUS ARACATI
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

CYNTHIA MARIA BRAVO PINTO

Este Trabalho de Conclusão de Curso foi julgado adequado para a obtenção do Grau de Bacharel em Ciência da Computação, sendo aprovado pela Coordenadoria de Ciência da Computação do Instituto Federal de Educação, Ciência e Tecnologia do Ceará - Campus Aracati e pela banca examinadora:

Prof. Msc. Francisca Raquel de Vasconcelos Silveira
Instituto Federal do Ceará – IFCE
Orientadora

Prof. Msc. Silas Santiago Lopes Pereira
Instituto Federal do Ceará - IFCE

Msc. Yrleyjander Salmito Lopes
CPQi Technology Financial Technical Consultant

Aracati, 03 de Maio de 2017

Agradecimentos

Agradeço primeiramente a Deus, por estar me guiando em todas tomadas de decisões, proteção e sabedoria para alcançar meus objetivos.

Agradeço a Professora Msc. Francisca Raquel de V. Silveira, pela parceria, amizade, orientação, oportunidades e ensinamentos dados durante todo o curso. Sendo um exemplo de caráter, respeito, dedicação, empenho e paixão em todas suas ações.

Agradeço aos membros da banca examinadora o Prof. Silas Santiago Lopes Pereira e o Msc. Yrleyjander Salmito Lopes, pela disponibilidade de participação da banca e contribuições pessoais dadas neste trabalho.

A agradeço a todos os amigos e professores do curso de Ciência da Computação, pela motivação, pelos bons momentos e conhecimentos compartilhados durante o curso.

Por fim, gradeço aos meus pais por todo o ensinamento, amor, carinho, dedicação e apoio, que foram dados durante toda a minha graduação. E a minha família por estar sempre presente me apoiando.

Dedico este trabalho ao meu pai por todo amor e apoio que foi me dado, durante o curso. E a minha mãe, que apesar de não estar fisicamente presente sei que está sempre me acompanhando e protegendo.

Resumo

Um dos principais problemas para usuários leigos ou com pouco conhecimento em Linguagem Consulta Estruturada (*SQL*) é a necessidade de conhecimento prévio da programação *SQL*. Uma das abordagens utilizadas para esta problemática é a utilização de Interfaces de Linguagem Natural para Banco de Dados (ILNBDs). Este trabalho realiza um estudo sobre o Processamento da Linguagem Natural e sua aplicação no acesso aos dados contidos em um Banco de Dados (BD). E apresenta uma arquitetura de interface em linguagem natural para acesso a um banco de dados relacional, que objetiva a obtenção de informações em relação à pesquisa de usuários que não tenham conhecimento da linguagem de consulta estruturada e que não sejam familiarizados com os modos de consulta em linguagem em SQL, possibilitando a conversão de uma consulta a banco de dados informada pelo usuário para uma consulta em SQL.

Palavras-chaves: Processamento da Linguagem Natural (PLN). Interface em Linguagem Natural para Banco de Dados (ILNBD). Linguagem de Consulta Estruturada (SQL).

ABSTRACT

One of the main problems for lay users or with little knowledge on Structured Query Language (SQL) is the need to know a prior knowledge of SQL programming. One of the solutions reached for this problem is to use Natural Language Interface to Database (ILNBDs).This work performs a study on the Natural Language Processing (NLP) and its application for the access to the data contained in a database. And it shows an interface architecture in natural language for the access to a relational database, that aims to obtain information in relation to search for users who do not have knowledge of structured query language and who are not familiar with the modes of language query in *SQL*, enabling the conversion of a query to the database, as informed by the user for a query in *SQL.*

Keywords: Natural Language Processing (NLP), Natural Language Interface to Database (ILNBD), Structured Query Language (SQL).

Sumário

Lista de figuras

Lista de tabelas

Lista de abreviaturas e siglas

AT	ATributo
AVG	Função Média do Banco de Dados
BD	Banco de Dados
FN	Funções
IDE	*Integrated Development Environment*
ILNBD	Interface de Linguagem Natural para Banco de Dados
ILNBDs	Interfaces de Linguagem Natural para Banco de Dados
GUIs	Interfaces Gráficas para Usuários
JSP	Java Server Pages
LN	Linguagem Natural
LNO	*Logic NOde* (Nós lógicos)
MAX	Máximo valor
MIN	Mínimo valor
NASA	National Aeronautics and Space Administration
NLP	*Natural Language Processing*

ON *Operation Node* (Nó de Operação matemática)

PROLOG Programação em Lógica

PLN Processamento de Linguagem Natural

QN *Quantity Node* (Nós de quantidade)

SN *Select Node* (Nó referente ao Select)

SGBD Sistema Gerenciador de Banco de Dados

SQL *Structured Query Language* (Linguagem de Consulta Estruturada)

SUM Função soma do banco de dados

TB Tabela

TICs Tecnologias da Informação e Comunicação

UF Usuário Final

UML Linguagem de Modelagem Unificada

VA Valor

XML *Extensible Markup Language*

CAPÍTULO 1

INTRODUÇÃO

Com a evolução das Tecnologias da Informação e Comunicação (TICs), o fluxo de dados teve um aumento gigantesco na sociedade contemporânea. As TICs possuem uma complexa rede de comunicação, na qual as informações são inseridas de maneira rápida, como é o caso de transações bancárias, fotos, vídeos, mensagens, imagens, entre outros, que são serviços utilizados por usuários em seu cotidiano por meio de computadores, tablets e smartphones (PACIEVITCH, 2006).

Porém, os usuários leigos que necessitam realizar processos que examinem esses dados para diversas finalidades podem encontrar uma grande dificuldade para processá-los, pois é preciso um conhecimento prévio da estruturação dos dados no Banco de Dados (BD) e de Linguagem de Consulta Estruturada (SQL – *Structured Query Language*) (SILVA e LIMA, 2007).

Apesar da linguagem SQL tenha seu mérito ao abstrair a estrutura física dos dados e constituir-se de uma linguagem de programação fácil e declarativa, exige de seus usuários um conhecimento sobre o esquema do BD, ou seja, é necessário ter o conhecimento sobre as tabelas, campos, relacionamentos e utilizar uma linguagem para consultas com regras de sintaxe e de semântica rígidas e limitadas (Silva e Lima, 2007).

O Processamento de Linguagem Natural (PLN) é uma área estudada em inteligência artificial que analisam e sintetizam linguagem falada ou escrita através de componentes de software ou hardware (JACKSON e MOULINIER, 2002). A Linguagem Natural (LN) é uma alternativa para consultas em bancos de dados, principalmente para usuários leigos ou com pouco conhecimento em computação. Pois, o usuário precisa apenas do conhecimento do domínio do BD (OLIVEIRA et al.,2009). Neste trabalho, empregamos a conversão da linguagem natural do usuário para uma linguagem de consulta estruturada (*SQL*).

Como disponibilizar uma interface de consulta para sistemas de apoio à tomada de decisões para usuários com pouco ou nenhum conhecimento técnico em

ferramentas ou linguagens de consulta a banco de dados, e que atenda ao requisito de transportabilidade entre domínios, além de apresentar uma arquitetura que possa ser facilmente estendida ou modificada?

A transportabilidade entre domínios é segundo Agosti (2003) é a capacidade de utilização da interface de LN em um número *n* de domínios, ou seja, a interface deve conseguir retornar dados independente do domínio do banco de dados que está sendo realizada a consulta.

O presente trabalho propõe uma solução para o problema de conversão de linguagem natural do usuário para a linguagem de banco de dados *SQL*, apresentado através da definição de uma nova arquitetura de interface que tenha como meio de interação o usuário do sistema, possibilitando a facilidade da vida de um usuário leigo ou com pouco conhecimento em consultas a banco de dados, sem que haja a necessidade de treinamento quanto a linguagem de programação empregada e conseguindo buscar as informações procuradas.

Apesar do PLN está sendo uma área bastante explorada para fins de pesquisas desde os últimos anos, ainda é encontrado um grande déficit de implementações de ferramentas voltadas para o idioma português. Observando essa carência e a necessidade de ter um conversor de LN em português para SQL, este trabalho propõe o desenvolvimento de uma ferramenta para possibilitar que uma consulta de dados informada pelo usuário em sua linguagem natural (em língua portuguesa) seja transformada em uma consulta SQL, sem a necessidade de conhecimento técnico em banco de dados.

1.1 Objetivos

1.1.1 Objetivo Geral

O objetivo deste trabalho é apresentar uma proposta e desenvolver uma ferramenta que possibilite a conversão de uma sentença que representa uma consulta de dados informada pelo usuário em linguagem natural para uma linguagem de consulta estruturada. Este trabalho converterá consultas estruturadas pela cláusula SELECT, FROM e WHERE, baseado em ILNBDs.

1.1.2 Objetivo Específico

• Estudar as técnicas de Processamento da Linguagem Natural aplicadas ao problema de conversão de consultas em LN para SQL;

• Aplicar técnicas computacionais para a conversão de uma sentença em linguagem natural no idioma português para uma sentença em SQL.

• Apresentar uma nova metodologia para a conversão realizada por uma ILNBD.

• Desenvolvimento do protótipo de interface para a proposta apresentada

• Comprovar que o presente trabalho preenche a lacuna de transportabilidade entre domínios de sistemas baseados em sintaxe.

1.2 Organização deste trabalho

Este trabalho está dividido da seguinte forma:

Capítulo 2 - Fundamentação Teórica: é apresentada a fundamentação teórica relacionada ao tema proposto, descrevendo uma visão sobre os trabalhos envolvidos com a utilização da linguagem natural para o acesso a dados em um banco de dados. São discutidas algumas arquiteturas existentes de PLN e os principais problemas propostos para essa área.

Capítulo 3 - Proposta: é detalhada a abordagem proposta neste trabalho para interface em linguagem natural para banco de dados.

Capítulo 4 - Resultados: Neste capítulo são apresentados os resultados obtidos neste trabalho, como também alguns cenários e exemplos de avaliação do sistema proposto.

Capítulo 5 - Conclusões: apresentam-se as considerações finais, que relatam os resultados obtidos com este trabalho, bem como são propostos trabalhos futuros.

CAPÍTULO 2

FUNDAMENTAÇÃO TEÓRICA

Neste capítulo será apresentada a fundamentação teórica para a realização deste trabalho. Iniciando com o estudo de como é realizada a comunicação entre humanos e entre homem e máquina, onde é confirmado que para que haja comunicação é necessário que os envolvidos estejam em um mesmo meio e em uma mesma linguagem. Levando em consideração este contexto é inserido o conceito de interfaces do usuário e posteriormente as interfaces de linguagem natural e sua utilização no processamento de linguagem natural para banco de dados. Além disso, na seção 2.1 é realizada uma explicação sobre as arquiteturas que podem ser feitas em interfaces de linguagem natural e por fim na seção 2.2 são apresentados os trabalhos relacionados.

Para a realização deste trabalho foram estudados conceitos básicos que são importantes, é o caso do próprio ato da comunicação entre os humanos por meio da linguagem. Iniciando o assunto da comunicação entre o usuário e o computador e suas dificuldades para o entendimento absoluto entre eles, pois cada qual tem sua linguagem.

"A linguagem é um dos aspectos fundamentais do comportamento humano e um componente crucial de nossas vidas" [Allen, 1995]. Esta afirmação reflete a importância da linguagem para a interação entre seres humanos, pois quando acontece um processo de comunicação entre os interlocutores ocorre a forma mais importante de expressão e comunicação. Este processo de comunicação do emissor e do receptor que se dar para a realização da troca de informações pode ser verbal e não verbal, em um mesmo meio e em uma mesma linguagem, seja ela escrita, falada ou por sinais.

Uma das grandes formas de comunicação entre os seres humanos é a utilização da linguagem natural. Neste contexto, podem ser utilizados três conceitos básicos sobre linguagem: (i) uma linguagem é um conjunto de signos e símbolos que permitem um grupo social se comunicar e facilita o pensamento e as ações dos

indivíduos (FISCHLER, 1987). Logo, para compartilhar as mesmas informações, é necessário um mesmo veículo ou meio em uma mesma língua para as pessoas que estão realizando a comunicação. Segundo Savadovsky (1988), (ii) a linguagem natural é uma das formas mais humanas de manifestação externa da atividade mental, ou seja, os seres humanos se utilizam da linguagem natural para se comunicarem, seja ela escrita ou falada, ainda identificando que linguagem natural é a troca de informações e discursos estruturados. E por último, Allen (1995) cita que (iii) a linguagem é um dos aspectos fundamentais do comportamento humano e um componente crucial de nossas vidas. Desta forma, é perceptível um padrão entre esses três conceitos aqui apresentados em que a linguagem natural é o meio mais utilizado para a comunicação entre dois ou mais interlocutores, possibilitando assim a troca de informações e de experiências de vida que são compartilhadas, o que resulta no aumento do conhecimento geral do grupo.

Um dispositivo necessário para que haja comunicação entre interlocutores, pessoas ou entidades é conhecido como interface do usuário. Para Thro (1991), uma interface é um local para encontro ou interação entre pessoas, a fim de construir uma comunicação e trocar informações pelo meio. Segundo Crane (1993), a utilização de reconhecimento de voz e de escrita manual derrubaram as barreiras de teclados, mouses e Interfaces Gráficas para Usuários (GUIs). Então, uma interface é uma ferramenta que serve como limite comum a interlocutores, garantindo uma conexão capaz de transmitir mensagens, conteúdo ou qualquer fonte de interesse para ambas entidades envolvidas, podendo ser a nível de escrita ou reconhecimento de voz.

A comunicação com o computador foi sempre uma barreira difícil de ultrapassar, pois o computador não possui a capacidade por si só de entender e usar a mesma linguagem que os usuários utilizam. Essa limitação proporcionou estudos voltados para o desenvolvimento de técnicas que fossem capazes de ensinar o computador, métodos de aprendizagem para entender a linguagem humana. Um desses métodos é conhecido como interface de linguagem natural (OLIVEIRA et al.,2009).

O interesse em utilizar sistemas capazes de entender objetivos dos usuários em sua própria linguagem surgiu juntamente com os próprios sistemas. Allan Turing ligava a inteligência dos computadores com a capacidade em lidar com a linguagem natural, concluindo assim que o processo de entendimento entre usuário e máquina vem sendo pensado desde o advento dos computadores.

A linguagem natural fornece instruções para o computador, de tal modo que o computador compreenda a linguagem que foi utilizada para a implementação. Como é difícil reconhecer frases em linguagem natural, a interface (devido a uma série de operações de programação) possibilita reconhecimento de palavras da sentença informada, dependendo da finalidade do software. Com o tempo a interface tornou-se uma das melhores técnicas para tornar o sistema mais intuitivo, logo em vez de tentar lembrar os comandos o usuário entra com o que ele quer que aconteça (OLIVEIRA et al.,2009).

Utilizar linguagem natural na interface acarreta algumas dificuldades, tal como o tratamento de ambiguidade que pode acontecer em qualquer linguagem, por exemplo com a palavra da língua portuguesa banco, na frase “Vá ao banco Gabriela”, onde Gabriela tanto pode ir à agência bancária, quanto ao local de sentar. O uso de restrições pode ser aplicado para resolver a ambiguidade, necessitando que o usuário aprenda quais estruturas são aceitáveis, tornando a linguagem natural uma linguagem de comandos.

A interface de linguagem natural quando realizada com sucesso apresenta as seguintes vantagens:

- É fácil de ser entendida pelo usuário, devido a estrutura e o vocabulário serem conhecidos pelo usuário;
- A mesma linguagem pode ser utilizada para várias aplicações;
- Deve existir pouco problema entre a troca de informações e aplicações;
- Permite considerável flexibilidade em executar os passos de uma tarefa.

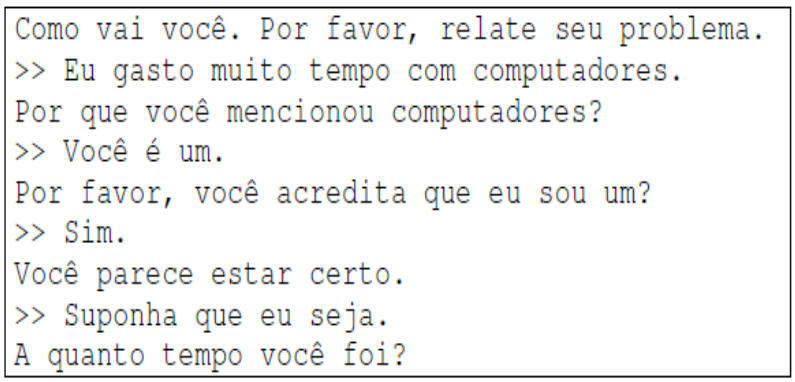
Segundo Anick e Peter (1993), com o crescimento de tamanho e importância das bases de dados, os computadores necessitam ter meios de interpretar a linguagem natural. Dessa forma, um usuário poderá encontrar alguma palavra em sua pesquisa na base de dados, sem se preocupar com seu significado exato – palavras estas que podem ser termos de comandos de busca, como também palavras a pesquisar, permitindo-lhe realizar suas pesquisas através do vocabulário que lhe é conhecido.

A Interface de Linguagem Natural para Banco de Dados (ILNBD) é um sistema que permite ao usuário obter informações contidas em banco de dados através do uso de comandos ou perguntas escritas em linguagem natural, como por exemplo português. Atualmente esses comandos são traduzidos em alguma linguagem formal de acesso a banco de dados, sendo SQL a mais utilizada. A área de acesso a banco de dados foi o primeiro grande sucesso no desenvolvimento de

aplicações que utilizam o processamento automático de linguagem natural (RUSSEL e NORVIG, 2013). Naquela época, final dos anos 60, pesquisadores acreditavam que a tarefa de desenvolver tais interfaces era plausível e que o esforço despendido em pesquisas iria trazer grandes benefícios ao demonstrar a utilidade das interfaces como ferramenta para usuários em geral, e acumular experiência para aplicações mais complexas que fizessem uso de processamento de linguagem. Embora o escopo das ILNBDs seja limitado pelas restrições de linguagem e de domínio de discurso, ainda se faz necessário confrontar questões como transportabilidade entre domínios, identificação das limitações do sistema por parte dos usuários e problemas linguísticos que normalmente ocorrem na maioria dos sistemas que trabalham na interpretação de linguagem natural. Dentre esses problemas, encontramos várias formas de ambiguidade introduzidas por modificadores e conjunções, utilização de sentenças incompletas e mal formuladas (ANDROUTSOPOULOS et al., 1995).

As primeiras prototipagens das ILNBDs iniciaram no final dos anos 60 com o programa ELIZA desenvolvido por Weizenbaum (1966), para varrer a frase do usuário a procura de palavras-chave, que ao serem detectadas disparam uma mensagem padrão ou utilizam parte da entrada para conseguir formar uma sentença de resposta, conseguindo passar a ilusão de que o programa possui inteligência e é capaz de conseguir a conversão humana. Uma imagem do programa é mostrada na Figura 1 a seguir.

Figura 1. Imagem da ferramenta Eliza



Fonte: WEIZENBAUM, 1966.

O grande marco nessa área de linguagem natural aconteceu com o desenvolvimento do LUNAR (*the Lunar Science Natural Language Information*

System), para a NASA (*National Aeronautics and Space Administration*), um sistema experimental desenvolvido para ajudar nos estudos de geólogos que tinham a missão de realizar o acesso, comparação e a avaliação dos dados obtidos de rochas lunares e a composição do solo coletados na missão Apollo-11. Embora o LUNAR não tenha sido utilizado em um ambiente operacional real, suas pesquisas tiveram grande influência na maioria dos sistemas subsequentes (RUSSEL e NORVIG, 2013; ALLEN, 1995), pois ajudava aos projetistas a pesquisarem os problemas que deveriam ser aprimorados no desenvolvimento de um sistema que utilizasse uma linguagem natural como forma de interação homem-máquina.

A maior motivação para a pesquisa e o desenvolvimento de ILNBDs como ferramentas de consulta são suas vantagens, embora no que se refira a sua aplicação prática estas vantagens ainda não tenham alcançado seu máximo potencial. A principal vantagem é que usuários não necessitam aprender uma linguagem de comunicação artificial ou conhecer modelos lógicos dos sistemas gerenciadores de banco de dados a fim de elaborar suas consultas. Outras características que tornam uma ILNBD uma boa estratégia interface perfeita para usuários em geral são: a facilidade de formulação de perguntas que denotam negação ou quantificação e a utilização de expressões resumidas ou incompletas cujo significado é extraído do contexto do discurso (REIS et al.,1997). A ILNBD ideal seria aquela que pudesse oferecer o uso de linguagem natural sem restrições, porém, no atual estado da arte, quando utilizamos a expressão "linguagem natural", estamos fazendo referência a dialetos que restringem a linguagem natural livre (SAVADOVSKY, 1988). Logo, ainda existe a necessidade de conhecer as funcionalidades e limitações de uma interface dessa natureza.

As ILNBDs apresentam certas desvantagens em relação aos outros tipos de interfaces, dentre elas temos: o usuário não possui plena compreensão das limitações da linguagem e semânticas impostas às suas consultas; quando uma pergunta é rejeitada, não é exposto com clareza se a mesma está fora do âmbito da linguagem do sistema ou fora do modelo conceitual (ANDROUTSOPOULOS et al., 1995); as interfaces que se apresentam como de propósito geral requerem longas etapas de configuração antes de serem utilizadas para uma aplicação particular; alto custo de desenvolvimento em virtude da escassez de ferramentas profissionais robustas e integradas aos ambientes computacionais, e desenvolvedores qualificados; alto custo de evolução e extensão desses sistemas ao longo do seu ciclo de vida.

2.1 Arquiteturas das ILNBDs

Esta seção apresenta os modelos e as diferenças entre as arquiteturas ILNBDs. Os softwares que fazem uso do processamento de linguagem natural aplicados a banco de dados adequam determinada arquitetura para suas finalidades. Existem basicamente quatro modelos de arquiteturas:

a) Sistemas baseados em Comparação de Padrões;

b) Sistemas baseados em Sintaxe;

c) Sistemas baseados em Gramática Semântica;

d) Sistemas baseados em Representação Intermediária.

a) Sistemas baseados em Comparação de Padrões

Esta arquitetura foi usada por alguns dos primeiros sistemas que faziam uso de uma interface em linguagem natural. Não utiliza analisadores sintáticos e nem gramáticas durante a interpretação de uma sentença e destina-se a aplicações com um conjunto pequeno de intenções. Sua implementação é fácil e o processamento de uma sentença é feito através do uso de um conjunto de padrões ou palavras-chave. A principal desvantagem está em sua simplicidade, o que não resulta em bons resultados ao usuário (ALLEN, 1995).

Como exemplo de interfaces que utilizam essa arquitetura podemos citar o programa ELIZA (WEIZENBAUM, 1966). Porém, muitos sistemas atuais utilizam variações do uso dessa técnica, tornando esta abordagem menos superficial e alcançando bons resultados.

b) Sistemas Baseados em Sintaxe

Em sistemas baseados em sintaxe, o mapa entre a árvore gramatical e a representação semântica é um processo de somente dois passos. Primeiro uma árvore é feita usando a gramática no analisador sintático, e depois, um módulo de mapeamento associado a cada nó da árvore semântica a qual corresponde diretamente a consulta atual (Agosti, 2003).

As ILNBDs baseadas em sintaxe utilizam essencialmente a sintaxe para construir suas semânticas e suas regras gramaticais, ou seja, a ILNBD cria um modo de realizar a conversão por meio de regras de consulta. Segundo Androutsopoulos et al. (1995) a tendência é existir dificuldade de aplicação prática, logo, questões

como transportabilidade entre domínios e utilização de diferentes tipos de bancos de dados não podem ser contempladas através do uso dessa arquitetura.

c) Sistemas baseados em Gramática Semântica

A arquitetura abstrata dos sistemas baseados em gramática semântica é uma extensão da arquitetura dos sistemas baseados em sintaxe. Sua diferença está na estrutura da gramática utilizada. Essa nova gramática, conhecida como gramática semântica, consiste em analisar o sentido das palavras que foram agrupadas pelo analisador sintático (AGOSTI, 2003).

A vantagem da utilização desse tipo de gramática é que sua estrutura é concebida a partir de restrições semânticas. Dessa forma, o sistema pode assegurar que quando uma consulta for declarada válida pelo analisador sintático, uma sentença equivalente na linguagem de consulta do banco de dados poderá ser gerada.

Sistemas baseados nessa arquitetura alcançam bons resultados quando o domínio da aplicação é relativamente limitado. No entanto, a reutilização da interface para outra aplicação exige a definição de uma nova gramática semântica.

d) Sistemas Baseados em Representação Intermediária

Esses sistemas adotam uma arquitetura que é muito aceita em ILNBDs devido ao uso de uma linguagem de representação intermediária que possui um módulo de lógica independente do domínio do banco de dados. Outro módulo é a análise semântica que gera a linguagem intermediária (ANDROUTSOPOULOS, 1995).

A sentença em linguagem natural informada pelo usuário passa pelos analisadores léxico, sintático e suas regras gramaticais para a geração da estrutura sintática que servirá de entrada para o analisador semântico, que através de suas regras semânticas que gera a estrutura de frase intermediária (NASCIMENTO, 2001). Os detalhes dessa frase intermediária variam de acordo com as características dos sistemas. Antes desta frase ser enviada para o último módulo que é a geração da consulta *SQL* e o acesso ao banco de dados, a frase intermediária passa por uma validação de sua expressão e quando validada é passada para um módulo chamado tradutor para linguagem de banco de dados que fará a conversão da mesma para uma linguagem que é suportada pela maioria dos Sistemas Gerenciadores de Banco de dados (SGBD). A expressão no idioma do

banco de dados é então executada pelo SGBD que buscará atender ao pedido do usuário e será mostrado o resultado que foi solicitado (ANDROUTSOPOLUS, 1995). A grande vantagem dessa abordagem é a modularidade empregada. Nessa arquitetura, podemos destacar dois módulos principais. O primeiro corresponde à parte linguística e o segundo à interface com banco de dados. Essa característica favorece a transportabilidade da interface em diferentes níveis. Edite (REIS et al., 1997), um sistema que responde perguntas sobre informações turísticas, é um exemplo atual de ILNBD que utiliza essa arquitetura.

2.2 Trabalhos Relacionados

Na literatura de Inteligência Artificial, ainda existem questões de como realizar a comunicação entre homem e máquina (RUSSEL e NORVIG, 2013). Como tentativa de preencher essa lacuna, o Processamento de Linguagem Natural visa sintetizar linguagem escrita ou falada. Os trabalhos apresentados a seguir utilizam de PLN em Banco de Dados com a linguagem natural em português e são comparados a este trabalho a nível de facilidade de uso da interface para o usuário, acesso a diversos domínios de banco de dados e a implementação da interface.

Souza e Campos (2006) apresentam uma ferramenta para o acesso ao Banco de Dados relacional, utilizando-se de um Banco de Dados e um tradutor (*parser*) para retornar os dados de uma consulta em forma de tabela de informações sobre a consulta realizada pelo usuário. A abordagem proposta em Souza e Campos (2006) conduz o usuário passo a passo, até a formulação da consulta completa, com isso, a interface consegue a cada interação disponibilizar um conjunto de palavras, nas quais o usuário vai escolhendo uma a uma, até realizar a consulta completa. Para mostrar o desempenho de sua aplicação, a abordagem foi testada por 20 pessoas leigas e em cerca de 85% dos casos de teste foi vista a facilidade de aprendizado e adaptação, provando a usabilidade perante o ambiente de consultas em Banco de Dados. A ferramenta apresentada neste trabalho se difere por dois aspectos: (i) a nível de sentença, o usuário não precisa escolher palavra por palavra para a geração da sentença, o que diminui a quantidade de passos realizados para a formulação da pergunta desejada; (ii) domínio do banco de dados, o trabalho de Souza e Campos (2006) mostra o teste com apenas um domínio de banco de dados, porém deixa claro que sua ferramenta pode ser expandida para outros domínios. A diferença para este trabalho é a utilização de cinco testes de domínios, o que

comprova a transportabilidade e a expansão para mais domínios.

Agosti (2003) apresenta uma interface em Linguagem Natural aplicada para web que independe do gerenciador de Banco de Dados, ou seja, dada uma consulta do usuário em sua língua natural consegue-se buscar em um sistema de banco de dados a resposta e retorna para o usuário uma tabela de informações. A abordagem utilizada é um estudo das formas de acesso ao banco de dados utilizando Processamento de Linguagem Natural, explicando pontos importantes sobre expressões regulares, análise semântica, sintática e léxica, como também modelos gramaticais, que servem como base para a interpretação e conversão da consulta do usuário. A implementação é realizada com *Java Server Pages* (JSP) e Prolog. Deste modo, o que o diferencia com este trabalho é (i) a não necessidade de um usuário administrador para realizar a configuração inicial da ferramenta manualmente para se ter um dicionário de sinônimos manuais e após isto conseguir a interface de linguagem natural para banco de dados. (ii) A interface é desktop para conseguir ser utilizada pelo usuário offline.

Com o trabalho desenvolvido por Perché e Pinheiro (2010) foi criada uma ferramenta capaz de recuperar informações em banco de dados por meio do processamento de linguagem natural. A busca é realizada em passos, cada passo é referente a uma interface visual do sistema, onde a interface inicial é caracterizada pela escolha entre três domínios de banco de dados. Logo após a escolha do domínio do banco de dados, o usuário é enviado para uma outra interface, a fim de selecionar qual tabela será realizada a busca dos dados. A próxima etapa é onde o usuário preenche os campos de palavras e sinônimos no qual será de interesse da consulta, gerando um arquivo XML contendo os sinônimos da tabela e de seus atributos, possibilitando que a aplicação consiga converter a consulta em linguagem natural para linguagem SQL, retornando para o usuário dados de sua requisição. Utiliza-se do SGBD Microsoft SQL Server e do arquivo XML gerado para realizar a conversão da sentença. Utiliza também o conceito de módulo etiquetador desenvolvido por Nunes (2007) para realizar a etiquetagem da sentença, que inicialmente é realizada o reconhecimento e a divisão dos termos da frase e após é gerada uma árvore sintática dos termos. Os termos identificados são enviados para um etiquetador probabilístico, baseado nos conceitos de Qtag (MASON, 2010), que tem a finalidade de auxiliar a conversão em caso de erro, logo, o programa consegue processar a possibilidade de ser determinada classe gramatical. O trabalho foi desenvolvido com a linguagem Java e a IDE Netbeans. Diferencialmente

da abordagem exposta por Perché e Pinheiro (2010), este trabalho apresenta (i) uma interface mais simples de ser utilizada, pois a tela inicial do usuário é o campo de escrita do usuário e no trabalho de Perché e Pinheiro (2010) o usuário perde muito tempo passando de tela em tela até a realização da sentença.

Nantes (2008) utiliza PLN voltado para uma consulta de dados na web, utilizando-se de XML e RDF para adicionar mais semântica ao documento sem necessitar entender sua estrutura. Além disso, possui uma camada muito importante de ontologia para oferecer expressividade necessária para definir restrições complexas e construções que implementam características de *frames* e lógica de descrição. Suas classes são modeladas por Linguagem de Modelagem Unificada (UML) e implementadas utilizando a linguagem Java (tecnologias Servlets e JSP). Sua contribuição para o mundo acadêmico é uma ferramenta que se utiliza de ontologia e PLN para extração de dados na web. O trabalho proposto se difere de Nantes (2008) por ser uma forma simplificada de realizar o processamento de linguagem natural sem a necessidade de abordar ontologia e retirando a abordagem voltada para uma página da web. Além disso o usuário não precisa realizar um passo a passo da sentença que ele quer construir, o que diminui o tempos gasto pelo usuário para a geração dessa sentença.

Dentre os quatro trabalhos citados anteriormente, todos podem ter sua contribuição quanto ao processamento de linguagem natural. Porém o diferencial do trabalho aqui proposto é a transparência para o usuário, o qual precisa apenas digitar sua pergunta e a ferramenta realiza os processos necessários para a conversão, sem a necessidade que o usuário perca tempo para a formulação da questão. Além disso, para a arquitetura proposta deste trabalho (sistemas baseados em sintaxe) o resultado obtido supre sua principal dificuldade que é a transportabilidade entre domínios. O próximo capítulo apresenta a abordagem aplicada neste trabalho, como também o detalhamento de sua estrutura.

CAPÍTULO 3

ABORDAGEM PROPOSTA

Neste capítulo é realizado o explanamento da abordagem proposta deste trabalho. Inicialmente é realizada uma breve introdução da proposta, seguida da visão geral da ferramenta e da detalhada, a fim de uma melhor explicação de como essa proposta está estruturada. Em seguida explicamos o pré-processamento realizado no analisador léxico e suas subseções. Assim como o analisador sintático e a etiquetagem e suas subseções. Por último é feito explicação da geração de consulta, suas regras e a medida de similaridade empregada.

Inicialmente a arquitetura escolhida, dentre as arquiteturas apresentadas no item 2.1, para este trabalho foram os sistemas baseados em sintaxe devido a lacuna desta arquitetura referente a transportabilidade entre os domínios, além de possuir simplicidade na técnica de desenvolvimento que se adequa a implementação realizada no período de seis meses de uma primeira ferramenta que conseguisse realizar o processamento de linguagem natural e conseguisse a transportabilidade entre domínios, no apêndice 1 é visualizada os cinco domínios usados para testes.

A abordagem proposta está pautada especificamente na conversão de uma consulta de banco de dados na língua portuguesa informada pelo usuário para uma consulta em *SQL*, conforme exemplificado na Figura 2, possibilitando o acesso a um grande volume de dados e utilizados para devolver a resposta por meio de consulta *SQL* ao usuário. Além do PLN em banco de dados, a abordagem também envolve o uso de compiladores com funções de interpretação léxica, que analisa a frase de entrada do usuário e o analisador sintático com suas regras de conversão e medida de similaridade adotadas para a geração da consulta em *SQL.*

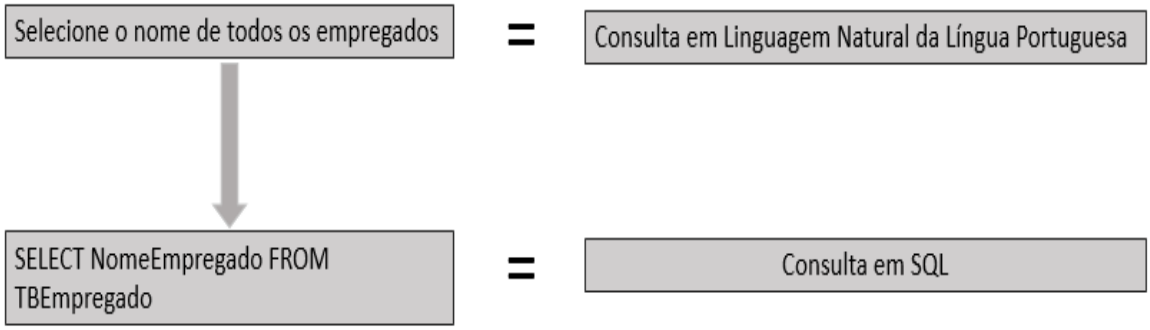


Figura 2. Exemplo de conversão da linguagem natural para a linguagem SQL. Próprio Autor.

3.1 Visão Geral da Abordagem

O intuito deste trabalho é desenvolver uma ferramenta que facilite a consulta de um usuário ao banco de dados, de modo que o mesmo não necessite ter o domínio da linguagem de consulta. Os passos citados abaixo mostram os papéis contidos em cada ramo da abordagem proposta:

Interface do Usuário: Responsável por servir como interface visual entre as sentenças do usuário e o retorno dos dados em forma de consulta SQL.

- Analisador Léxico: Responsável por identificar e etiquetar as palavras (tokens) da sentença ou expressões isoladas. As palavras identificadas são classificadas de acordo com sua categoria gramatical.
- Analisador Sintático: Consiste em criar uma árvore de derivação para cada sentença obtida no analisador léxico e mostrar como as palavras estão relacionadas entre si.
- Conversor SQL: Utilização dos analisadores citados acima para a conversão da consulta em linguagem natural para uma consulta em SQL.
- Banco de dados: A consulta SQL gerada pelo conversor é inserida no banco de dados e realizado o *data mining* para obter os dados requeridos.

A Figura 3 seguir mostra uma visão geral da ferramenta.

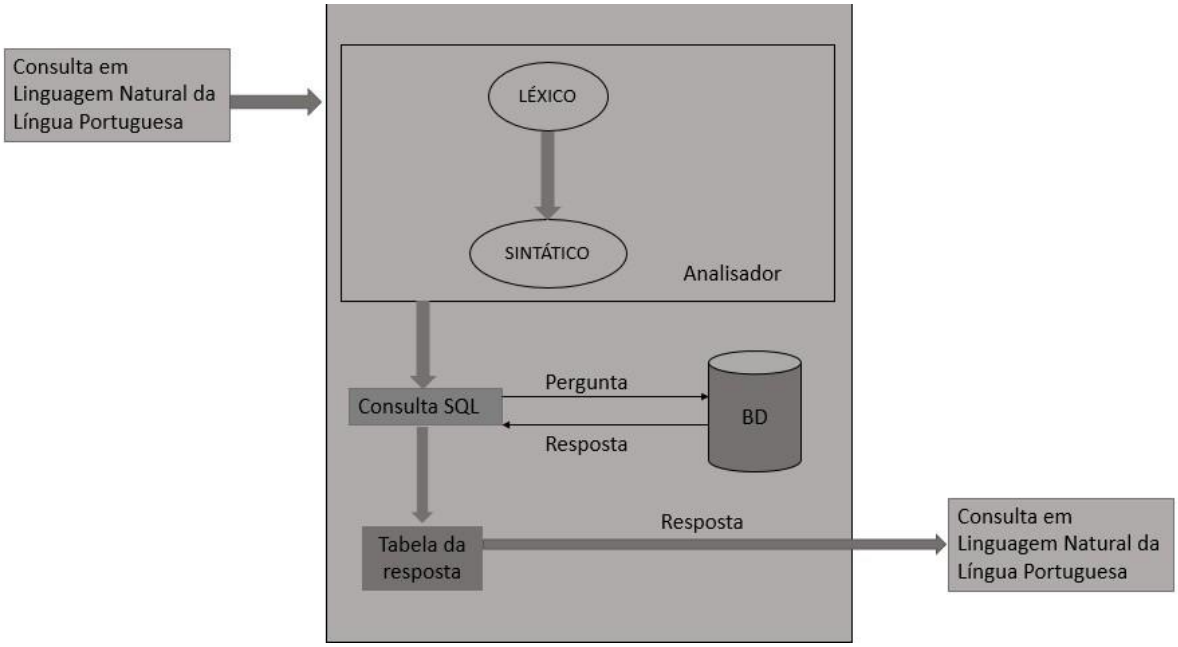


Figura 3: Visão geral da ferramenta. Próprio Autor

Como pode ser visualizado na Figura 3, inicialmente o usuário fornece uma consulta em português na ferramenta; essa consulta é enviada para o analisador, que possui os seguintes módulos:

a) Analisador Léxico

b) Analisador Sintático

c) Geração da Consulta em *SQL*

a) Analisador Léxico

Conforme (TERRA, 1992) a morfologia é uma parte da gramática que estuda a estrutura, processos de formação, flexão e classificação das palavras. Em um sistema de interpretação da linguagem natural isto é feito por um módulo chamado analisador léxico, cuja finalidade é identificar palavras (tokens) ou expressões isoladas em uma sentença.

Em relação à estrutura de formação das palavras, essas podem ser divididas em unidades significativas menores, ou morfemas, cuja manipulação pode gerar diversas variações para um mesmo vocábulo (TERRA, 1992). São exemplos de morfemas: radicais,desinências nominais e verbais, prefixos e sufixos, etc. A manipulação de morfemas é especialmente interessante quando se deseja armazenar palavras de uma linguagem de forma condensada, em que ao invés de se armazenar todos os vocábulos, teriam apenas os morfemas necessários para sua

criação.

Este processo retorna a divisão léxica da sentença, em que a pergunta feita pelo usuário é analisada, retornando cada palavra e a sua classificação gramatical. (i) Neste trabalho o analisador léxico recebe a pergunta do usuário e deve identificar palavras e expressões regulares que estejam isoladas por meio de delimitadores. Desta forma essas palavras após serem analisadas devem ser classificadas dentro de uma categoria gramatical. Esta etapa é importante para o processamento de linguagem natural, pois o computador deve compreender cada uma dessas palavras.

b) Analisador Sintático

A análise sintática em um sistema é feita através do analisador sintático ou *parsing* e depende diretamente da análise léxica, pois após receber as principais palavras presentes na sentença digitada pelo usuário, a análise sintática faz o processo de identificação destas palavras é o caso de sujeito e predicado, complemento verbal e nominal, classificação das orações, entre outros. Este processo é melhor realizado fazendo uso de expressões livres de contexto, pois, em geral, são mais poderosas que as regulares, permitindo a representação de linguagens com certo grau de complexidade.

A análise sintática de uma oração em português deve levar em consideração diversos tipos de sintagmas. Para Savadovsky (1988), sintagmas são subdivisões intuitivas de orações de uma linguagem natural em que se percebe um significado claro. Cada sintagma tem uma palavra principal, que é chamada núcleo sintagmático e outras palavras dependentes desse núcleo. Recursivamente, as palavras que acompanham o núcleo podem formar outros sintagmas.

Logo após a análise léxica é obtido um grupo de tokens, para que o (ii) analisador sintático realize a análise da estruturação dos tokens em nós e use um conjunto de regras para preencher as informações destes nós, os nós preenchidos com informações pelo analisador sintático são enviados para o módulo de geração da consulta *sql.*

c) A geração da consulta em *SQL*.

Os nós etiquetados do analisador sintático são enviados para a geração da

consulta em linguagem estruturada. Posteriormente, a ferramenta faz a consulta no banco de dados e recebe os dados como resposta a consulta inicial realizada pelo usuário. Após esse conjunto de passos, os dados obtidos são mostrados para o usuário, seguido de sua pergunta em linguagem natural e a mesma pergunta em SQL, como pode ser visualizado na figura 3.

Uma visão mais detalhada da proposta deste trabalho é mostrada na figura 4, porém para um melhor entendimento de como é processada a frase inicial, iremos utilizar a nomenclatura de pré-processamento para os passos que são feitos no analisador léxico e o termo etiquetagem para os processos do analisador sintático, pois esta nomenclatura está de acordo com as funções do que cada analisador desempenha.

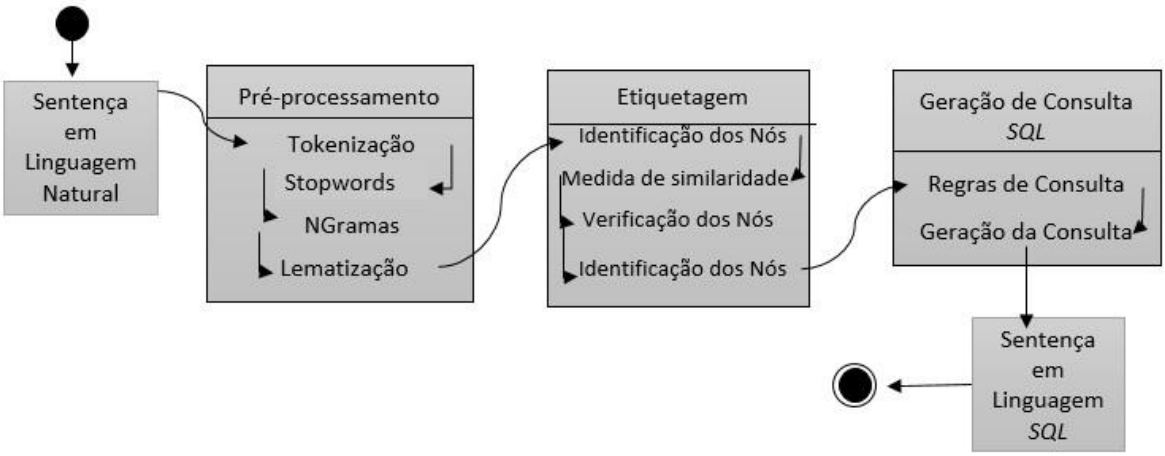


Figura 4. Visão detalhada da ferramenta. Próprio Autor.

Na figura acima, mostra uma outra forma de visualizar a divisão dos passos desta ferramenta detalhadamente. A abordagem proposta para realizar a conversão de uma consulta em linguagem natural para uma consulta em SQL é concebida em três módulos: (i) Pré-processamento, (ii) Etiquetagem e (iii) Geração de consulta SQL. Inicialmente, o usuário fornece uma sentença em linguagem natural que é enviada para o primeiro módulo da interface, o pré-processamento. Nessa etapa, a sentença é dividida em termos, utilizando os seguintes passos: (i) Tokenização, (ii) Formação de n-gramas, (ii) Stopwords e (iii) Lematização. Após essa etapa, a sentença pré-processada é enviada para a etiquetagem, para a inserção dos dados nos nós (dados estes que servirão para a conversão da linguagem natural para a

SQL), passando pelas fases de: (i) Verificação do nó nas cláusulas próprias do banco de dados e (ii) Verificação do nó para termos de tabela, atributo e valor. Para complementar essa interface, o módulo de geração de consultas *SQL*, onde os nós são recebidos e a linguagem natural é convertida em linguagem de *SQL*, que é dividido em: (i) Regras de Consulta e (ii) Geração da Consulta. Todos os passos citados anteriormente serão explicados detalhadamente nas seções a seguir.

3.2 Pré-Processamento

Após o usuário informar a sentença que representa uma consulta a banco de dados em língua portuguesa, a ferramenta realiza a leitura da entrada e inicia o pré-processamento que passará por quatro etapas: (i) tokenização, (ii) formação de n-gramas, (iii) *stopwords* e (iv) lematização. Essas etapas são realizadas com o intuito de deixar a sentença separada em termos para ser realizada a conversão para uma consulta SQL, seguindo as demais etapas da abordagem proposta.

3.2.1 Tokenização

Para conseguir realizar o processo de conversão da sentença de entrada informada pelo usuário, é necessário o tratamento da frase em três passos de pré-processamento. O primeiro desses passos é a tokenização, que divide a sentença em tokens (palavras delimitadas por espaço ou pontuação). Um exemplo da tokenização é visto na figura 5 abaixo, em que o usuário informa como entrada uma frase e, logo após, essa frase é separada em tokens.

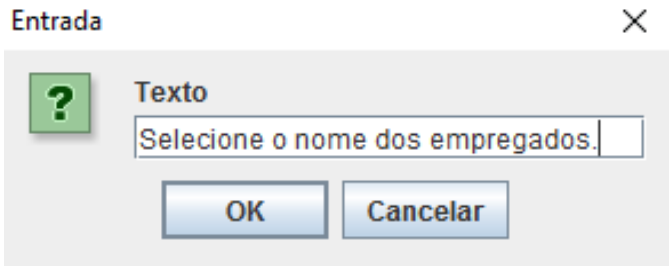


Figura 5: Exemplo da tela de entrada do sistema. Próprio autor.

Nesse exemplo, o usuário entra com a sentença “Selecione o nome dos empregados”. Essa frase é tokenizada, ou seja, é realizada uma varredura na frase,

a fim de delimitar cada palavra através de espaços ou pontuações.

O tratamento dos tokens tem diferentes enfoques conforme sua finalidade, como é o caso de seu uso para a área da segurança, de extração de palavras-chave em notícias, uso em tradutores linguísticos, entre outros. Neste trabalho os tokens servem para encontrar cada palavra da sentença, independente se existe alguma informação relevante. Um exemplo da saída desta frase é mostrado na figura 6, onde cada palavra é identificada a partir do delimitador espaço. Caso existisse alguma pontuação, ela seria salva como token. Neste caso, as palavras encontradas são: “Selecionar” “o” “nome” “de” “todos” “os” “empregados”.

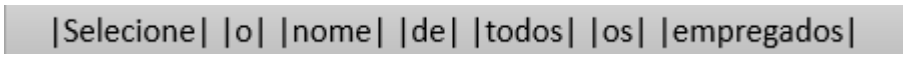


Figura 6: Exemplo da frase do sistema. Próprio Autor.

A lista com os tokens identificados da sentença informada pelo usuário é enviada para a segunda etapa de pré-processamento, a formação dos n-gramas.

3.2.2 Formação de N-Gramas

Segundo Sarmento (2011) o texto não é um simples amontoado aleatório de palavras. A ordem da colocação das palavras no texto é que produz o significado. Isso pode indicar que as palavras estão relacionadas, diretamente por composicionalidade ou afinidade, ou indiretamente por semelhança entre os termos, ou seja, a combinação dos termos (tokens) no texto dá origem aos n-gramas.

Os n-gramas como o próprio nome diz é um conjunto *n* de gramas de uma frase, onde *n* é o valor da quantidade máxima de combinações entre os tokens. Essa quantidade máxima é definida pelo programador da ferramenta, dependendo de sua problemática e contexto. Neste trabalho foi escolhido até o número três de n-gramas, ou seja, a ferramenta aqui apresentada é capaz de fazer combinações de até três tokens. Esses 3-gramas são chamados de trigramas. Um exemplo do que acontece com a frase de entrada do usuário é vista na figura 7.

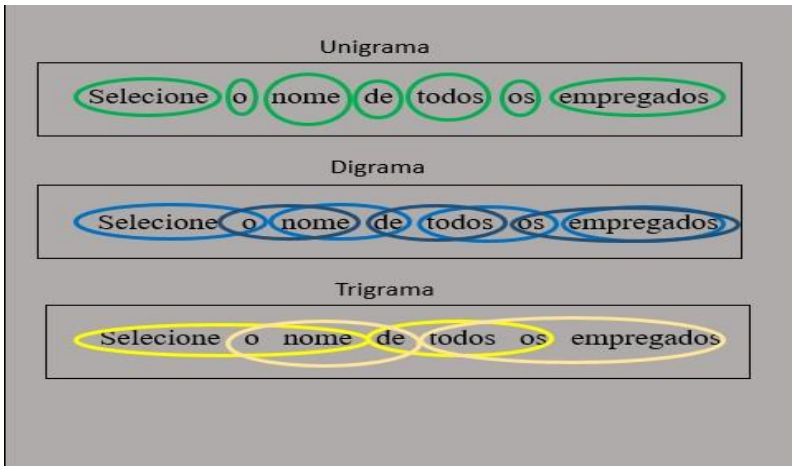


Figura 7: Exemplo da formação dos n-gramas. Próprio Autor.

A figura 7 demonstra a lógica utilizada para a combinação dos n-gramas. O primeiro passo é o teste de todas as palavras individualmente. Essa verificação e combinação de cada palavra é chamada unigrama; no segundo passo, as palavras são agrupadas de duas em duas, sendo esse processo chamado de diagramas, e assim sucessivamente até um número *n* de combinações (n-gramas). Logo, como saída desse exemplo temos os seguintes n-gramas mostrados na figura 8. Vale ressaltar que a imagem serve como ilustração para a formação dos n-gramas, cujos termos formados (unigrama, bigrama e trigrama), são enviados para a próxima etapa, a extração das stopwords.

Unigrama
Selecione
o
nome
de
todos
os
empregados

Bigrama
Selecione o
o nome
nome de
de todos
Todos os
os empregados

Trigrama
Selecione o nome
o nome de
nome de todos
de todos os
Todos os empregados

Figura 8: Exemplo de saída da formação dos n-gramas. Próprio autor.

3.2.3 Stopwords

Até este terceiro passo, o pré-processamento identifica as palavras da sentença com a ajuda dos delimitadores, obtendo assim os tokens que foram usados pelos n-gramas, como tentativa de identificar os possíveis termos do texto que formam uma expressão ou que tenham um sentido ao juntar até três palavras.

Stopwords podem ser traduzidas do inglês para o português como “palavras de parada”. No processamento de linguagem natural se refere às palavras que podem ser consideradas irrelevantes no contexto da tarefa de PLN realizada, ou seja, são termos, geralmente com elevada recorrência, que podem ser removidos por possuir baixo valor semântico. Um conjunto de stopwords tipicamente é composto por palavras das classes gramaticais: artigos, preposições, conjunções, pronomes, advérbios ou, em alguns casos, números e pontuação.

Neste projeto, as stopwords escolhidas são artigos, conjunções, pronomes, pontuação e preposições. Caso seja encontrada uma n-grama selecionada pela etapa de "formação de n-gramas" que tenha no início ou no fim uma stopword, essa n-grama é excluída, por não caracterizar um termo relevante. Sendo assim, após a verificação das *stopwords* nos n-gramas apresentados na Figura 8, temos como resultado os seguintes n-gramas apresentados na Figura 9.

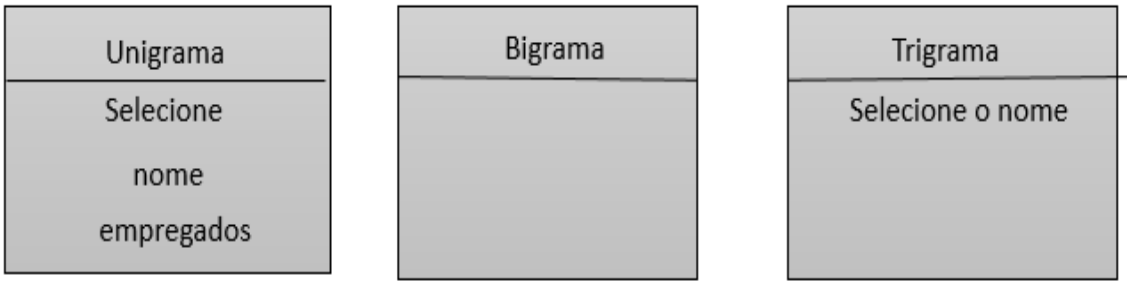


Figura 9: Exemplo de saída de stopwords.

Na Figura 9 acima temos os n-gramas resultantes após serem excluídos os n-gramas que iniciam ou terminam com *stopwords.* Percebe-se que nesse exemplo a coluna de bigramas ficou vazia, porém em outros exemplos é possível encontrar nome de tabelas, atributos ou valores, como por exemplo: Carga horária de professores, onde carga horária será identificada como um bigrama. Após a verificação das stopwords, essa lista é enviada para o último passo do pré-

processamento, a lematização.

3.2.4 Lematização

A lematização é a última etapa do pré-processamento. Nessa fase ocorre a redução canônica dos termos que passaram pelos passos anteriormente citados, ou seja, os verbos conjugados em um tempo verbal são reduzidos a sua forma no infinitivo e os adjetivos e substantivos a forma masculina singular.

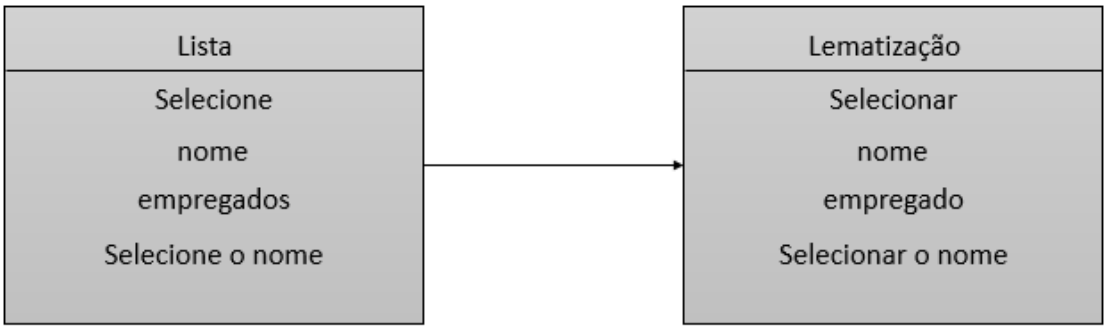


Figura 10: Exemplo de lematização

A Figura 10 mostra a saída da sentença inicial, passada pelos processos de tokenização, validação de *stopwords*, formação de n-gramas e por último a lematização, que reduziu os verbos para o infinitivo e substantivos e adjetivos para um masculino singular, sem acentuações. Contudo, com os testes realizados no decorrer da implementação, foi vista necessidade de exceção ao se tratar das reduções de um pequeno conjunto de palavras, é o caso de português, matemática, ciência da computação, álgebra, carga horária, entre outras. Caso sofressem a lematização tornariam portugue, matematico, ciencio da computaçao, algebro, cargo horario e ao comparar com tabelas, atributos ou valores não seriam encontrados. Então palavras que se encaixem nesta exceção apenas são retiradas as acentuações e conserva sua escrita sem lematizar para o masculino singular. Esse conjunto de palavras tratadas no pré-processamento é enviado para o próximo passo, a etiquetagem.

3.3 Etiquetagem

Nesta parte, cada nó é um n-grama da sentença, quer seja unigrama, bigrama ou trigrama. Portanto, a quantidade de nós será a mesma de n-gramas encontrados na sentença. Utilizaremos esses nós até o final do processo de descoberta e geração da consulta *SQL*. Cada nó possuirá os seguintes parâmetros:

- palavra: nome do n-grama da sentença.
- tipoDoNo: refere-se a identificação do nó (AT, ON, SN, entre outros) quanto a sua etiquetagem, ou seja, o nó pode ser atributo ou de operação ou select respectivamente (a explicação para que serve a identificação do nó é feita nas subseções 3.3.1 e 3.3.2), tal como, a palavra aluno que será identificada como uma palavra que corresponde a tabela aluno terá tipoDoNo TB.
- paraSQL: O campo paraSQL refere como está a escrita daquela palavra do nó no *sql*, tal como, a palavra aluno, que no item anterior foi identificada como um nó TB, pois a palavra corresponde a uma tabela. Em seu campo paraSQL será identificada como o nome que está no *sql*, ou seja, tbAluno.
- nomeTabela: este parâmetro é utilizado na etiquetagem e se refere ao nome da tabela do banco de dados, caso o nó se refira a uma tabela.
- nomeAtributo: este parâmetro é utilizado na etiquetagem e se refere a palavras que sejam considerados atributos da frase.

Com a identificação desses parâmetros, o nó possui as informações que serão tratadas para os próximos processos. O próximo passo é a verificação se o nó corresponde a algum termo da nomenclatura da estruturação do *SQL*.

O processo de etiquetagem feito através do analisador sintático criado neste projeto que tem a finalidade de encontrar nós na frase que possam ser identificados como possuindo alguma informação importante para a conversão, pode ser dividido da seguinte maneira: (i) verificação se o nó em questão faz parte de algum termo de nomenclatura da estruturação do *SQL*, ou seja, se o termo está presente em alguma palavra própria do *SQL*, é o caso de *select, where*, entre outros; e, por último, (ii) caso os nós não pertencem a palavras-chave do *SQL* é verificado se ele faz parte da estrutura do banco de dados, ou seja, é tabela, atributo, valor ou restrição. Uma melhor visualização de como é realizada a etiquetagem, pode ser vista na imagem 11.

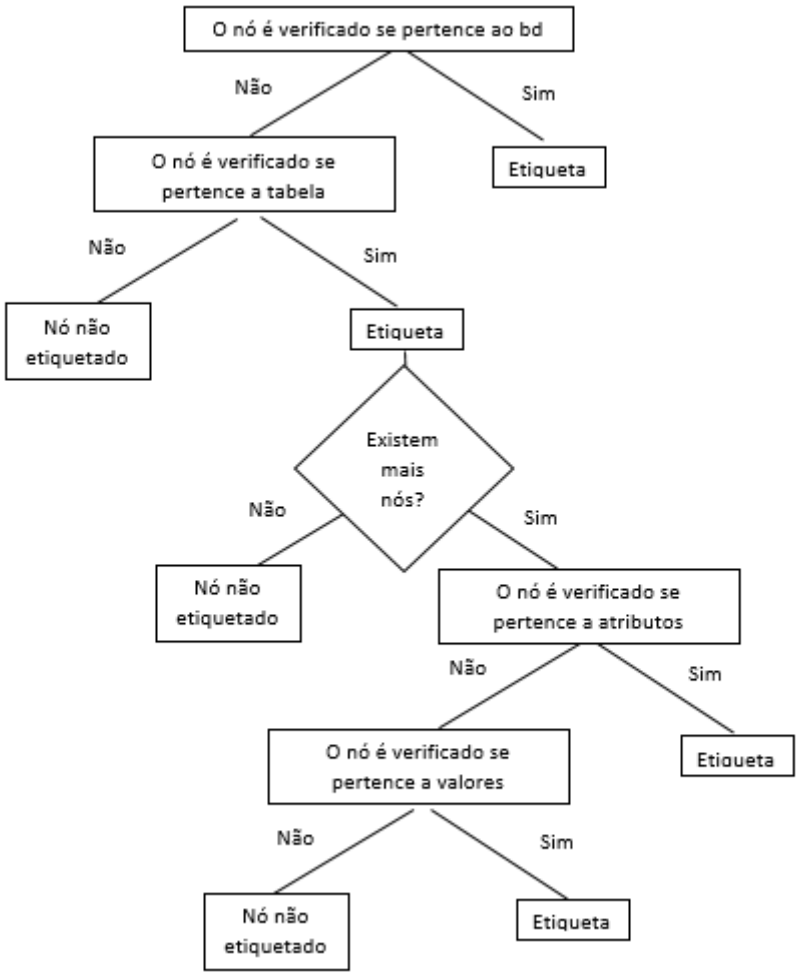


Figura 11: Árvore geral da tomada de decisão da etiquetagem.

A imagem 11 mostra de forma geral a ordem seguida para a etiquetagem dos nós. Inicialmente todos os nós são comparados para ver se pertencem a cláusula select, caso pertençam são etiquetados (detalhes desta etiquetagem são vistos na subseção 3.3.2). Caso não pertençam os nós são enviados para a verificação com as tabelas do banco de dados, se os nós ao serem comparados venham a coincidir com o nome de uma tabela estes armazenam as informações desta etiquetagem (detalhes desta etiquetagem são vistos na subseção 3.3.2). O próximo passo é verificar se ainda existem nós para serem etiquetados, pois se o usuário digitar apenas uma palavra e esta for etiquetada como TB, a interface consegue gerar uma consulta para trazer todos os dados desta tabela. Então caso ainda existam nós, eles são enviados para verificar se pertencem a atributos, se esta comparação

retornar verdadeira o nó é etiquetado (detalhes desta etiquetagem são vistos na subseção 3.3.2) como AT. Porém caso ainda existam nós que não forem etiquetados como AT, passam por uma verificação se estes nós são valores de atributos, caso sejam são etiquetados (detalhes desta etiquetagem são vistos na subseção 3.3.2) como VA.

3.3.1 Verificação do nó nas cláusulas próprias do BD

Esta etapa verifica se um nó é um termo presente na cláusula de consulta do BD. Na imagem 12 é possível visualizar como o processo de etiquetagem para cláusulas do BD é realizada. O nó é comparado as palavras referentes ao *select*, caso seja verdadeira a comparação, o nó é etiquetado como tipoDoNo SN (*Select Node*) e é preenchido o campo paraSQL com a palavra *select.* Caso contrário o nó é enviado para a comparação com as palavras referentes as operações de igual ou maior ou menor ou maior e igual ou menor igual, caso comparação seja aceita o tipoDoNo é etiquetado como ON (*Operation Node*) e o campo paraSQL com o símbolo da operação. Também é comparado a nós referentes a funções do *sql* cujo tipoDoNo é preenchido com FN (*Function Node*) se a comparação der verdadeira e o campo paraSQL é preenchido com suas siglas MAX ou MIN ou AVG ou SUM ou Count e após os nós são comparados com funções quantitativas e preenchido o tipoDoNo com QN (*Quantity Node*) e o campo paraSQL com suas siglas ALL ou ANY ou Each e por último se o nó pertence aos operadores lógicos do *sql* é inserido em seu campo tipoDoNo a sigla LNO (*Logic NOde*) e no seu campo paraSQL é preenchido com AND ou OR ou NOT. Abaixo está a tabelas das palavras consideradas como correspondentes das cláusulas do bd com as palavras em português. E por último a tabela 1 mostra os sinônimos das palavras próprias do banco de dados utilizados para a etiquetação do nó.

Figura 12: Árvore da tomada de decisão da etiquetagem para cláusulas do BD. Próprio autor

A Tabela 1 mostra as palavras de conversão utilizadas para as cláusulas do banco de dados e seus sinônimos.

Tabela 1: Exemplo das palavras de conversão para as cláusulas do BD. Próprio Autor

	Exemplo de conversão
SQLSelect	Retorne, liste, mostre, selecione, apresente, traga, exiba, são
SQLOperaçãoIgual	Igual, mesmo, idêntico, semelhante, identico
SQLOperaçãoMaior	Maior, superior, depois, após, apos
SQLOperaçãoMenor	Menor, inferior, antes, anterior
SQLOperaçãoMaiorOuIgual	Maior ou igual
SQLOperaçãoMenorOuIgual	Menor ou igual
SQLOperaçãoDiferente	Diferente, oposto, distinto
SQLLogicoAnd	e
SQLLogicoOr	ou
SQLLogicoNot	Não, não

Após a etiquetação de nós referentes às cláusulas do BD, os outros nós são enviados para o próximo passo, que é a verificação dos nós quanto a tabela, atributos e valores.

3.3.2 Verificação do nó para termos de tabela, atributos e valores.

O próximo passo é verificar se o nó corresponde a um elemento da estrutura do banco de dados, ou seja, descobrir se o nó pertence a tabela, atributo ou valor do banco de dados, a Figura 13 apresenta a árvore de decisão para a etiquetagem dos nós referentes a tabelas.

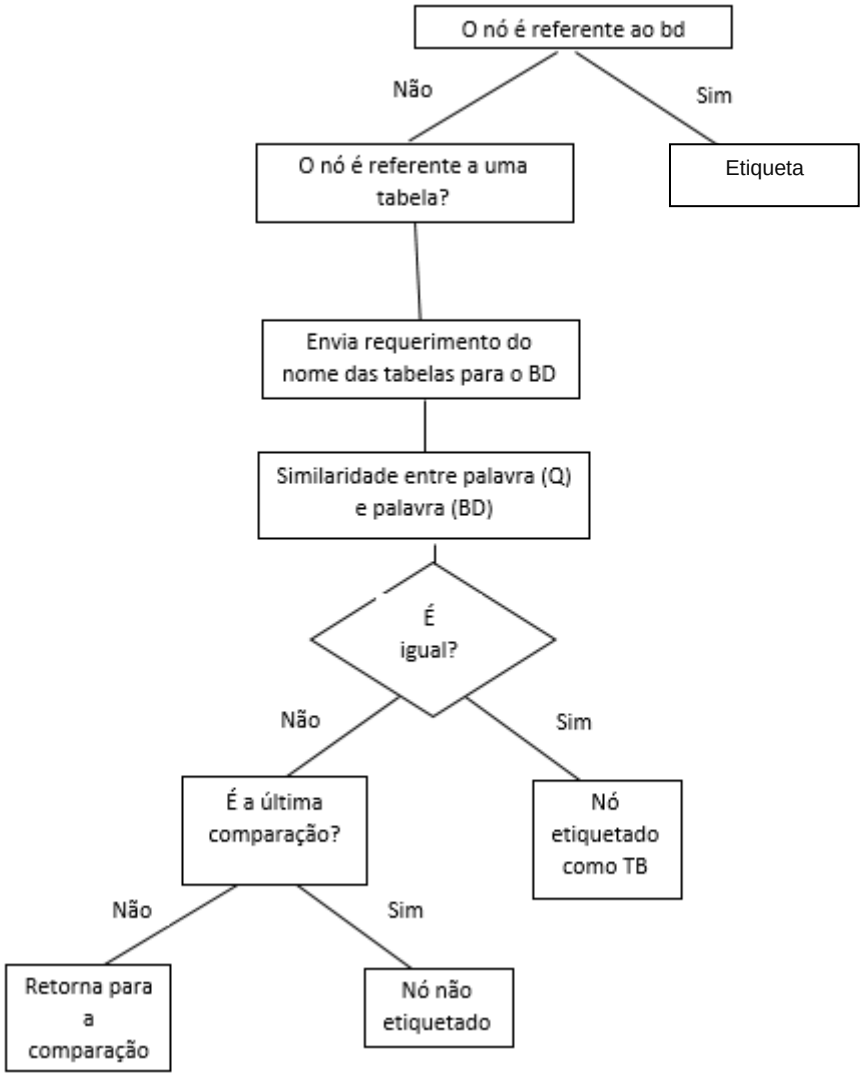


Figura 13: Árvore da tomada de decisão da etiquetagem para tabelas. Próprio autor.

A figura 13 demonstra inicialmente que o nó está vindo de uma negação para cláusulas do bd e após é verificado se este nó encaixa em tabelas, para isso é

realizada uma requisição para os nomes das tabelas do banco de dados. Possuindo a palavra do nó e a palavra da tabela do bd é realizada uma comparação para verificar se a palavra (Q) está contida na palavra (BD), caso exista o nó é etiquetado como tipoDoNo TB. Caso a comparação seja falsa é verificado se a palavra (Q) ja foi comparada com todos os nomes de tabelas, caso não seja a última comparação retorna para a comparação de similaridade para ser comparada com outra tabela, porém se for a última o nó não foi etiquetado como tabela.

Porém para as palavras (Q) e (BD) conseguirem realizar as comparações houve a necessidade de realizar alguns tratamentos, com a finalidade de deixar as palavras mais simples para a comparação:

- Primeiro tratamento: Para encontrar a tabela, tanto as palavras (Q) quanto as palavras (bd) passam por .toLowerCase(); que garante que todas as palavras estejam no minúsculo.
- Segundo tratamento: As palavras (bd) ainda passam por duas fases desse segundo tratamento que utiliza a função .replaceAll, para caso apareça o “_”, seja retirado e no canto adiciona “” (vazio) e caso apareça a palavra “tb” seja retirado e no canto adiciona “”(vazio).

Vale ressaltar que caso duas tabelas contenham a palavra (Q) é utilizada uma medida de similaridade para conseguir estabelecer de qual tabela é aquela palavra. Essa medida verifica a quantidade de caracteres da palavra (Q) e verifica a quantidade de caracteres das tabelas. Logo, em seguida a menor diferença da quantidade de caracteres entre a palavra (Q) e a palavra (BD) é escolhida como a tabela daquele nó, tal como, obtém-se o nó com a palavra aluno e o banco de dados retorna duas tabelas que contém a palavra aluno, que são aluno e alunomatricula (levando em consideração que ambas as palavras já passaram pelos tratamentos citados acima). A contagem para a palavra do nó dá cinco, após é realizada a contagem das tabelas, a primeira tabela aluno também possui cinco caracteres, já a segunda tabela possui 14 caracteres, então a primeira tabela é escolhida por conter a mesma quantidade de caracteres da palavra do nó e possuir menos caracteres da segunda tabela.

Caso nenhum nó seja etiquetado como TB em tipoDoNo, a interface retorna para o usuário uma janela com o campo para refazer a frase de entrada. Caso existam mais nós para serem etiquetados que não tenham entrado como nós referentes às tabelas os nós são verificados para atributos (AT) do banco de dados ou valores de atributos (VA). Como as etapas da etiquetagem dos nós para AT ou VA são muito parecidas, será apresentada a árvore de decisão dos atributos na figura 14 e a árvore de decisão dos valores está no apêndice B.

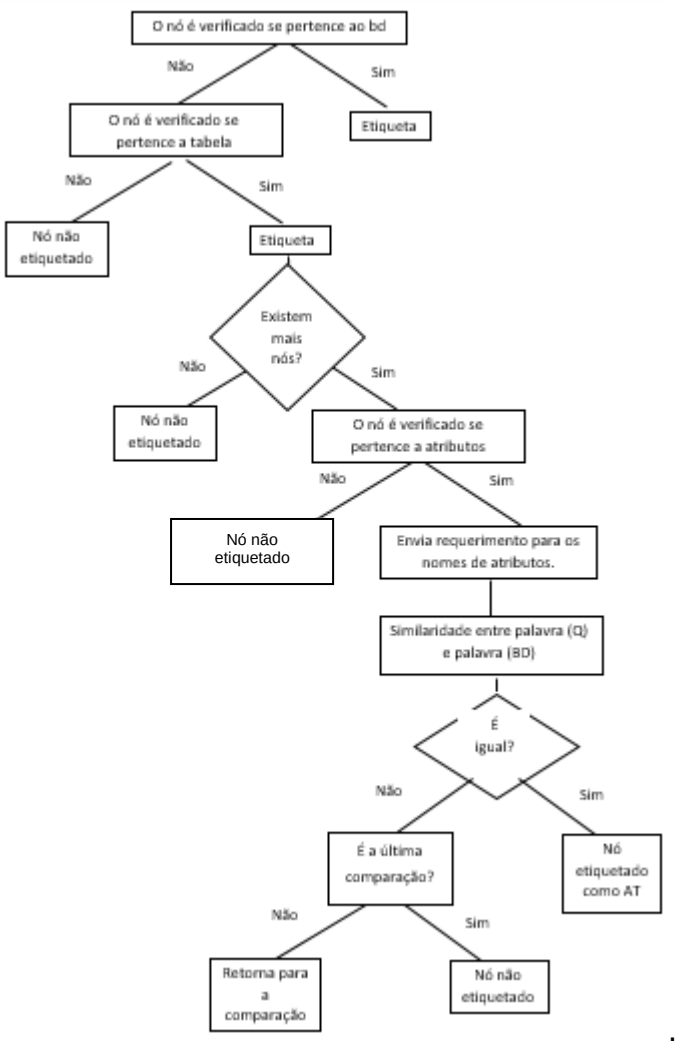


Figura 14: Árvore da tomada de decisão da etiquetagem para atributos. Próprio autor.

Para encontrar na frase de entrada do usuário palavras que correspondem a atributos ou valores do BD é necessário ter no mínimo um nó que corresponda a uma das tabelas do banco de dados. Posteriormente é enviada uma requisição para o envio dos atributos daquela tabela encontrada, de tal modo que a busca para encontrar atributos restringe apenas a uma tabela. Estes dados (nó e atributos) são enviados para comparação, a palavra do nó e a palavra vinda do BD passam pelos mesmos tratamentos dos nós da tabela (.toLowerCase e replaceAll). Após este conjunto de tratamentos as palavras são efetivamente comparadas e caso a palavra do nó esteja contida em palavra do bd e é atribuída a etiqueta para o nó.

Por último para encontrar os valores que são restrições da cláusula *where* do *SQL*, é enviada a palavra que está no nó para ser comparada com os dados dos atributos trazidos da tabela etiquetada. E passa pelos mesmos tratamentos para a identificação dos atributos. Com uma diferença, porventura o usuário não digitou um nó correspondente a atributos, porém digitou um nó que é identificado como valor de atributo, a interface consegue procurar em todos os valores do banco de dados e etiquetar de qual nó se refere o valor e qual é seu atributo, tal como, o usuário digita a seguinte frase “Selecione o empregado Henrique ”, a interface etiqueta a palavra

selecione como referente ao nó *select* e tipo SN, o nó com a palavra empregado é etiquetado como um nó de tipo TB e pertencente a tabela tbEmpregado e o nó Henrique será pesquisado como atributo, caso não seja encontrado é pesquisado em valores e caso exista Henrique o nó é etiquetado com o nome de seu atributo EMP_NOME.

Concluindo este processo de etiquetagem, temos a saída da identificação de cada nó com todos os parâmetros completos como exemplo dos itens 3.3.1 e 3.3.2 e estão prontos para serem enviados para desenvolver a consulta em *SQL*.

Etiquetagem do nó - Selecione	
palavra	Selecionar
tipoDoNo	SN
paraSQL	Select
verificado	1
nomeTabela	null
nomeAtributo	null

Etiquetagem do nó – nome	
palavra	nome
tipoDoNo	AT
paraSQL	EMP_NOME
verificado	1
nomeTabela	tbEmpregado
nomeAtributo	EMP_NOME

Figura 15: Exemplo da saída da etiquetagem dos nós.

A Figura 15 exibe a etiquetagem de dois nós da frase de entrada do usuário “Selecione o nome de todos os empregados”. Cada informação dos nós é importante para as regras de consulta (Seção 3.4).

3.4 Geração de Consulta SQL

Na etiquetagem cada nó tem como saída a sua etiqueta, trazendo os valores que serão necessários para a geração da consulta. Porém, antes de gerar a consulta, é necessário realizar uma análise dos nós, para decidir se os dados da frase são suficientes para conseguir gerar a consulta ou se o usuário digitou alguma sentença fora do padrão aceito de frase, conforme detalhado a seguir.

3.4.1 Regras de Consulta

As regras de consulta são regras seguidas para que a ferramenta retorna os dados pedidos pelo usuário a partir da entrada de sua frase. A primeira

regra de consulta é que o usuário deve conhecer o domínio do banco de dados que está acessando. A segunda é que a sentença do usuário não tenha erros de português. E a terceira é que o usuário digite uma palavra que possa ser identificada como tabela, tal como, a palavra empregados na frase “Retorne o nome dos empregados”, do exemplo que estamos adotando desde o início do trabalho. Nas próximas subseções são pontuados lados positivos e negativos desta interface e, logo após, é explicado detalhadamente como é realizada a conversão da frase inicial.

3.4.1.1 Pontos Positivos

1. Interface consegue a conversão com a identificação mínima de uma tabela – Caso o usuário digite apenas uma palavra e esta seja considerada uma tabela, a interface consegue estabelecer uma conversão retornando todos os dados desta tabela.
2. Interface não interage com o usuário, de tal modo que o usuário não perca tempo realizando uma série de passos até chegar na consulta – Para o processo de conversão a ferramenta procura criar uma sentença válida a partir da frase do usuário, sem que haja a necessidade do usuário ter que escolher passo a passo cada palavra que ele quer, ou seja, a ferramenta consegue fazer uma conversão automática, seguindo as regras de conversão da consulta.
3. Interface com o tempo de resposta aceitável – A interface consegue resolver o questionamento do usuário em um tempo considerado rápido.
4. Interface consegue reconhecer atributos e valores ocultos pelos usuários - Usuário não precisa digitar precisamente um valor de um atributo tal qual é encontrado no banco de dados. A ferramenta é capaz de etiquetar e retornar uma consulta, como pode ser visto no exemplo 5 do item 4.2, onde a palavra Aracati refere ao atributo cidade que está oculto no texto.
5. São identificadas datas de nascimento – Para a identificação de datas é necessário que o usuário use o formato dd/mm/yyyy.

3.4.1.2 Pontos Negativos

Apesar da boa quantidade de consultas válidas nesta interface, ainda são

encontrados alguns problemas que devem ser tratados, é o caso dos pontos abaixo:

1. Sentença do usuário não deve ter erros de português – Caso o usuário digite errado ou escreva errado uma palavra, e caso a interface não consiga identificar a etiquetagem ideal para este nó, a interface retorna uma mensagem para o usuário avisando que no nó da palavra foi visto um erro e pede para refazer a frase.
2. Não são identificadas frases onde não são encontradas palavras que remetam a tabelas na hora da identificação – A ferramenta aqui apresentada, consegue converter frases de até uma tabela, caso não seja identificado nenhum nó como tabela, é retornado para o usuário que a conversão não é aceita.

3.4.2 Geração da Consulta

No fim dos dois primeiros módulos a interface tem como resposta a identificação e etiquetagem dos nós. Esta lista de dados dos nós é enviada para o módulo de geração da consulta para realizar o processo de conversão da sentença do usuário para a consulta em SQL.

O passo inicial para a criação da consulta é a verificação da ordem da frase de entrada do usuário, onde só são vistos os nós etiquetados. Com isso é possível descartar os nós que passaram pelo lematizador, porém não foi encontrado nenhum vínculo com palavras próprias do *SQL,* ou com tabelas, atributos ou valores dos atributos. Restando somente nós com informação para a consulta.

O segundo passo é a comparação dos nós restantes com a sua localização na frase inicial. A frase inicial começa a ser verificada até encontrar um nó etiquetado como tabela, ao encontrar este nó, todos os nós antes são enviados para realizar a primeira parte da consulta *SQL*. Envolvendo desde a cláusula *select* até o preenchimento da cláusula *from* com o nome da tabela.

O terceiro passo é a formação da segunda parte da consulta *SQL*, que faz referência as condições da cláusula *where.* Caso a frase não contenha condições, enviamos a frase realizada no passo 2 para a consulta no *SQL*. Caso contrário, uma estrutura de repetição verifica o restante da frase após o nó etiquetado como TB, ao encontrar nós etiquetados como atributos e valores. No final os dados dos nós

etiquetados são adicionados na cláusula *where* e é enviada a frase completa para a consulta. A figura 16 abaixo mostra como é inserida a estruturação da consulta em *SQL*.

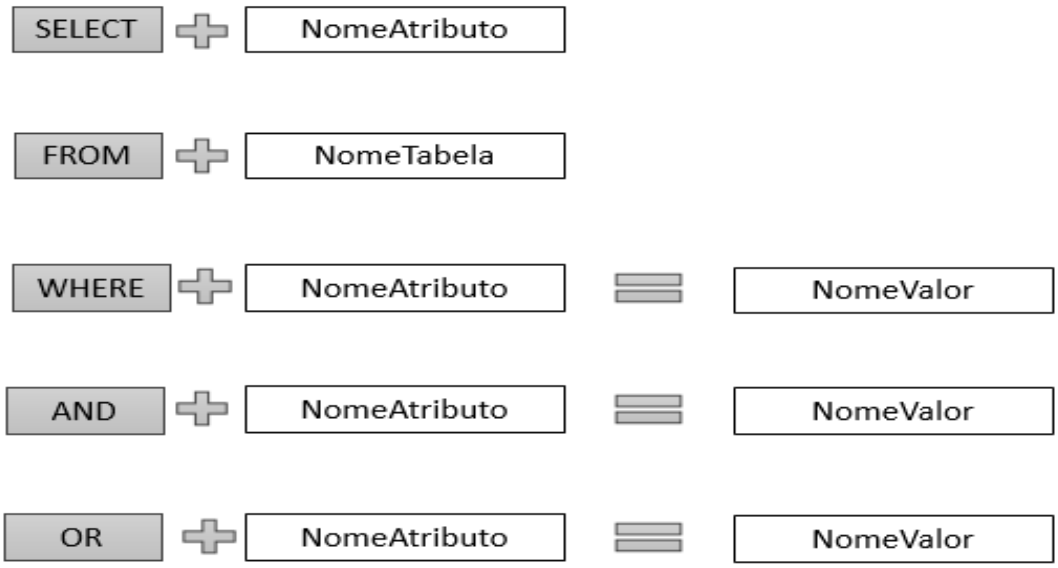


Figura 16: Estruturação da consulta em *sql*. Próprio autor.

Como pode ser visto na figura 16, temos palavras fixas do *SQL* (entende-se como palavras fixas o conjunto de palavras próprias da consulta, como: *select, from, where, and* e *or*). Sendo assim dependendo da etiquetagem do nó, como mostrado nas subseções 3.3.1 e 3.3.2, os nós são inseridos em seus respectivos locais.

CAPÍTULO 4

RESULTADOS

4.1 Arquitetura do Modelo dos Experimentos

No capítulo 3 foi desenvolvida e detalhada a proposta deste trabalho, que visa o desenvolvimento de uma interface de linguagem natural para usuários que não tenham conhecimento para a construção de consultas de banco de dados em SQL, devido à falta de entendimento da linguagem de programação ou da estrutura do banco de dados. Este capítulo aborda os experimentos realizados para geração de consultas em SQL a partir da abordagem proposta neste trabalho, como também traz dados sobre os resultados dessas consultas realizadas em três domínios diferentes de banco de dados.

A realização gráfica desta interface foi desenvolvida na IDE (*Integrated Development Environment)* Eclipse Mars.1, em linguagem Java. A implementação consta com dezesseis classes que codificam a entrada da frase até seus resultados). As consultas aos dados foram realizadas pela conexão do Eclipse com o SGBD (Sistema Gerenciador de Banco de Dados) Microsoft SQL Server Express 2012, o que possibilitou o retorno dos dados. Na fase do pré-processamento foi utilizado o openNLP, uma biblioteca em Java para processamento de linguagem natural em texto, utilizada para a tokenização e etiquetagem morfossintática.

4.2 Descrição dos Experimentos

Antes de disponibilizar o sistema para usuários finais foram feitos alguns

testes para validar o modelo proposto. A seguir são apresentados exemplos que tratam uma consulta em linguagem natural e, a partir da abordagem proposta, é apresentada a solução dada, ou seja, a consulta em SQL após a pergunta do usuário passar pelo conversor. Os exemplos aqui tratados são resultados do projeto e implementação da arquitetura proposta. Vale ressaltar que os exemplos aqui mostrados são realizados com os cinco domínios de bancos de dados testados pela ferramenta, com a finalidade de conseguir alcançar a transportabilidade entre domínios. A representação destes cinco domínios pode ser vista nas tabelas de 1 a 5 no apêndice A.

Os exemplos 1 e 2 apresentam perguntas do usuário realizadas de formas diferentes para a mesma consulta na linguagem SQL, essa resposta será enviada para o banco de dados e terá como retorno os dados resultados da consulta. Essa situação se dá pelo motivo que as perguntas podem ter sido estruturadas pelo usuário de forma transparente, porém foram interpretadas pelo programa com a mesma finalidade, nesses exemplos, trazer todos os dados dos alunos salvos no banco de dados.

Exemplo 1: Utilizado o banco de dados SistemaAcademico
Pergunta: “Retorne os alunos”.
Resposta: SELECT * FROM tbAluno

Exemplo 2: Utilizado o banco de dados SistemaAcademico
Pergunta: “Quais são os alunos cadastrados”.
Resposta: SELECT * FROM tbAluno

A conversão para os exemplos 1 e 2 foi possível devido a forma como foram estruturadas as sentenças informadas pelo usuário. Inicialmente foi encontrada alguma tabela, após realizada a procura de atributos e, por último, possíveis valores dos atributos. Nesses casos, o conversor encontra, como no exemplo 1, que a palavra “retorne” refere-se ao select do SQL e alunos é etiquetado como tabela. Já no exemplo 2, apenas a palavra alunos é etiquetado como tabela, logo configura-se uma situação diferente para o conversor, em comparação ao exemplo 1. A solução para o exemplo 2 que possui um nó etiquetado como ‘TB’ (tabela), concluindo assim que possui algum dado para ser pesquisado, criou-se uma estrutura de consulta onde o select já é campo preenchido pela ferramenta, como também o “*” (obtenção

de todos os atributos de uma determinada tabela) e assim preenche apenas a tabela de onde serão trazidos os dados.

A Figura 17 mostra a resposta obtida do banco de dados no console da ferramenta, a partir da consulta gerada pelos exemplos 1 e 2.

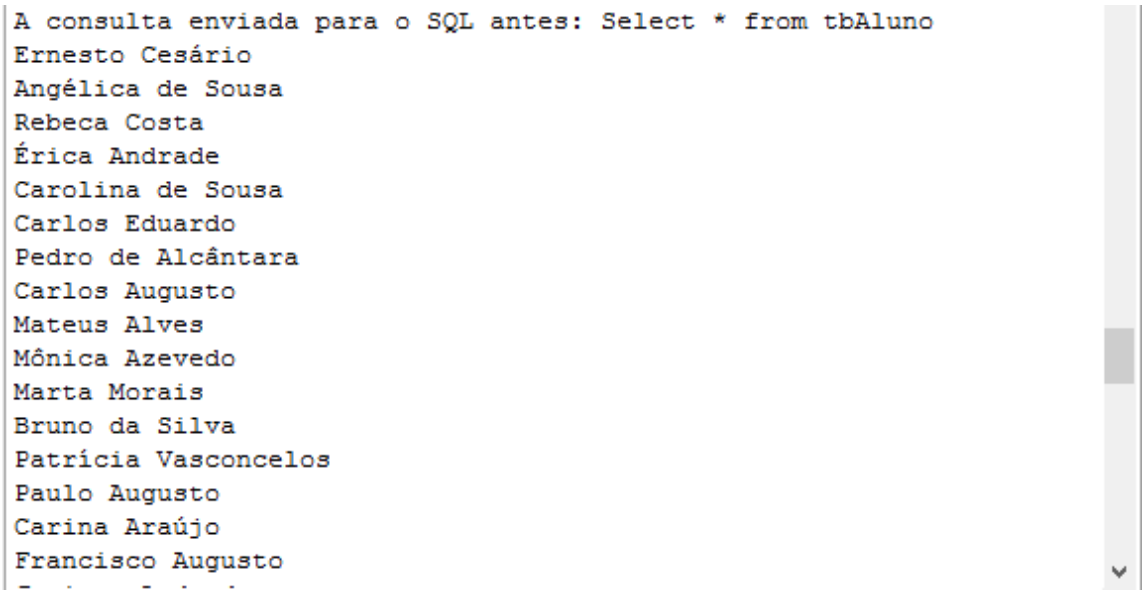


Figura 17: Resultado da consulta dos exemplos 1 e 2. Próprio Autor

Um exemplo da etiquetagem desses nós é visualizado na tabela 2, onde é expresso o tratamento destes nós nas etapas de pré-processamento, etiquetagem e geração da consulta.

Tabela 2: Processamento completo para os exemplos 1 e 2. Próprio autor.

	Retorne os alunos	Quais são os alunos?
Tokenização	Retorne os alunos	Quais são os alunos

NGramas	Retorne os alunos Retorne os os alunos Retorne os alunos	Quais são os alunos Quais são os os alunos Quais são os são os alunos
Stopwords	Retorne alunos Retorne os alunos	alunos
Lematização	Retorne aluno Retorne o aluno	aluno
Verificação do nó BD	Retorne = *SQL* aluno = null Retorne o aluno = null	aluno = null
Verificação de tabelas, atributos e valores	Retorne = *SQL* aluno = tbAluno Retorne o aluno = null	aluno = tbAluno
Geração da consulta *SQL*	Select * from tbAluno	Select * from tbAluno

Exemplo 3: Utilizado o banco de dados Hospital

Pergunta: “Retorne os nomes dos medicos”.

Resposta: SELECT nome

FROM Medico

No exemplo 3, pessoas com conhecimentos básicos em SQL conseguem

determinar que existe um select, um atributo e uma tabela. Porém, para que o computador entenda esta situação, foi implementada uma maneira diferente. Após os nós serem etiquetados (conforme descrito nas seções dada no item 3.3.1 e 3.3.2), o conversor percorre cada nó verificando primeiro se algum nó tem medidas de similaridades (conforme descrito na seção no item 3.3.2) aceitáveis com nome de tabela, após com atributo e por último com valores. Como não existem valores neste exemplo, a consulta é estruturada assim: Inicialmente, o select foi encontrado no nó que contém a palavra “retorne”, depois a palavra “medicos” é comparada com as tabelas do sistema e é etiquetada como Medico e, nessa tabela, é verificado se existe algum atributo que corresponda através das medidas de similaridade aos demais nós da sentença, achando, assim, que nome é identificado como nome. Na classe que desenvolve a construção da sentença em linguagem de programação de banco de dados, verifica que existem três tipos de dados (SN, TB e AT) e que não contém valores. Logo, por não existir valores, a ferramenta sabe que esse atributo não é uma condição do *where.* E, assim, o adiciona como atributo da cláusula select. O resultado é mostrado na figura 18.

```
A consulta Tabela é: Select nome from Medico
João de Oliveira Paiva
Pedro Falcão
Niara Souza de Alencar
Gabriela Perez Moreno
Miguel Flores do Araújo
Tiago Soares da Conceição
```

Figura 18: Resultado da consulta do exemplo 3. Próprio autor

Exemplo 4: Utilizado o banco de dados de loja

Pergunta: “Retorne os clientes da cidade Aracati”.

Resposta: SELECT CLI_NOME

FROM tbAluno

WHERE CLI_CIDADE = ‘Aracati’

O exemplo 4 traz uma combinação dos exemplos anteriormente citados, pois possui um *select,* uma tabela e um atributo do *select*, porém com uma diferença, ao etiquetar os nós é adicionado um nó etiquetado como VA (Valor). Sendo assim, a

ferramenta procura nas tabelas do SQL se existe o valor “Aracati’ em alguma delas. Neste caso, verifica se ‘Aracati’ é um nome de um cliente em tbCliente e assim o atributo identificado como CLI_NOME se encaixa como atributo do *select* e após identifica o outro nó atributo como CLI_CIDADE. O resultado da ferramenta é visto na figura 19.

```
A consulta Tabela é: Select CLI_NOME from tbCliente where CLI_CIDADE = 'Aracati'
Maria de josé
Pedro de Alcantara
Glória do carmo
Caetano Vieira
João Izidoro Alencar
Antonio Cesar Coe Pinto
Flavia de Souza Bravo
```

Figura 19: Resultado da consulta do exemplo 4. Próprio autor.

Exemplo 5: Utilizado o banco de dados Veiculo

Pergunta: “Retorne os veículos azuis ”.

Resposta: SELECT * FROM Veiculo

WHERE cor = ‘Azul’

No exemplo 5, diferentemente do exemplo 4, temos a palavra azuis (que durante a lematização é feita sua conversão para masculino singular), que pode ser identificada pelos usuários como uma cor, porém com a problemática de não possui nenhuma palavra anterior que sirva para a identificação pelo computador de que azul é uma cor. Com isso, é necessário realizar a pesquisa em todos os atributos da tabela que foi reconhecida até que a palavra ‘Azul’ seja encontrada. Obtendo que o termo da sentença veículos se refere a tabela tbVeiculo e que retorne é select e que azul é um valor do atributo cor. O exemplo da resposta de retorno do banco de dados é mostrado na figura 20.

```
------------------------------------------------
Tabela recebida: veiculo
Tamanho da tabela: 1
A frase antes é: Select * from Veiculo
O indice que ta recebendo agora é: 1
O tamanho do armazenamento é : 5
Valor recebido: azul
C4 Palace
------------------------------------------------------
```

Figura 20: Resultado da consulta do exemplo 5. Próprio autor.

Exemplo 6: Utilizado o banco de dados Veiculo

Pergunta: “Retorne o modelo dos veiculos com marca fiat ou citroen ”.

Resposta: SELECT modelo FROM tbVeiculos
WHERE marca = ‘Fiat’ OR
marca = ‘Citroen’

O exemplo 6 serve para mostrar a adição dos operadores lógicos nas consultas, onde a ferramenta ao realizar o mesmo processo do exemplo 5, ao identificar um nó etiquetado como operador lógico deve realizar novamente o processo de identificação das etiquetagens dos nós da sentença. Sendo assim ao identificar os operadores lógicos, a ferramenta chama um método que traz a solução e caso não haja operadores lógicos a ferramenta faz apenas uma restrição da consulta. Caso haja dois ou mais nós etiquetados como operadores lógicos, continua sendo adicionado na consulta *where* o atributo e seu valor, juntamente com a palavra AND (e) ou OR (ou) ou NOT (não). Conseguindo formar a resposta do exemplo 6, que possui uma condição *where* seguida de operador lógico, a figura 21 mostra a resposta enviada do banco de dados do exemplo 6.

```
A consulta Tabela é: Select modelo from Veiculo where marca = 'fiat' OR marca = 'citroen'
Bravo
C4 Palace
C4 Picasso
Siena
C4 Lounge
Punto
linea
Strada
Picasso
Aircross
```

Figura 21: Resultado da consulta do exemplo 6. Próprio autor.

Exemplo 7: Utilizado o banco de dados SistemaHoteleiro

Pergunta: “Retorne o número apartamento e o estado”.

Resposta: SELECT COQ_NUMERO_APARTAMENTO, COQ_ESTADO
FROM tbControleQuarto

O caso visto no exemplo 7 é a problemática de mais de um atributo na

cláusula *select*, já que os exemplos anteriores tratam apenas sentenças com um atributo no *select* e, caso tenham mais de um atributo, a interface adiciona na consulta o último atributo encontrado, realizando um processo de sobrescrita. Para solucionar este problema foi adicionado uma lista que adiciona todos os valores dos atributos e, caso exista mais de um valor nessa lista, a consulta recebe a etiqueta desses nós e adiciona a vírgula após cada atributo com exceção do último. E depois completa com a palavra reservada *from* e adiciona a tabela que foi identificada. A imagem 22 mostra o resultado da consulta do exemplo 7.

```
A consulta Tabela é: Select COQ_NUMERO_APARTAMENTO, COQ_ESTADO from tbControleQuarto
101
102
103
104
105
106
107
108
109
Ocupado
Desocupado
Em Manutenção
Ocupado
Ocupado
Ocupado
```

Figura 22: Resultado da consulta do exemplo 7. Próprio Autor

O exemplo 8 traz todas as problemáticas exibidas nos exemplos de 1 ao 7 e realiza a conversão de maneira adequada. E a imagem 23 mostra o resultado do banco de dados.

Exemplo 8: Utilizado o banco de dados SistemaHoteleiro

Pergunta: “Retorne o valor diaria e o estado do quarto que possuem o numero de pessoas 4 e 5”.

Resposta: SELECT COQ_VALOR_DIARIA, COQ_ESTADO

FROM tbControleQuarto WHERE COQ_NUMERO_PESSOA = 4 AND COQ_NUMERO_PESSOA = 5

```
A consulta Tabela é: select COQ_VALOR_DIARIA, COQ_ESTADO from tbControleQuarto WHERE COQ_NUMERO_PESSOA = 4 AND COQ_NUMERO_PESSOA = 5
520.00
620.00
Ocupado
Desocupado
```

Figura 23: Resultado da consulta do exemplo 8. Próprio autor

Esses exemplos foram realizados com cada dado da tabela que pode ser vista no apêndice A tabelas de 1 a 5, para todos os cinco domínios de banco de dados, que são: Sistema Acadêmico; Controle Hoteleiro; Locadora de veículos; Gerencia de Loja; BD Hospitalar. Em todos estes domínios de banco de dados foram realizadas trinta rodadas de testes de acordo com cada problemática apresentada nos exemplos de 1 a 8, em cada coluna dos BDs. E todos os dados foram retornados de forma correta para cada teste. Os resultados obtidos nos exemplos representam o que pode ser alcançado com a arquitetura proposta neste trabalho. Pode-se observar que existe grande flexibilidade sintática para a seleção e restrição de atributos.

CAPÍTULO 5

CONCLUSÕES

A proposta abordada neste trabalho surgiu da necessidade de se obter informações em linguagem natural para uma determinada aplicação do computador, configurando um contexto entre o usuário que necessita realizar consultas em banco de dados, porém não conhece a linguagem de programação que deve ser aplicada. Uma das soluções criadas para essa problemática é o uso de interfaces de linguagem natural para banco de dados.

Apresentou-se uma proposta e implementação de um sistema em linguagem natural para banco de dados desenvolvido para validação do modelo proposto. Acreditamos ter alcançado um avanço no que tange ao assunto sobre interface em linguagem natural para banco de dados, o usuário tem acesso simples para a entrada de sua frase, sem o uso de menus para a escolha de palavra por palavra o que diminui o tempo gasto para a criação da sentença, utilizando diversos domínios diferentes de banco de dados, o que comprova sua transportabilidade e a ferramenta é voltada para o idioma português brasileiro.

Por fim, implementamos dois padrões de analisadores, um léxico e um sintático, com o objetivo de apresentar a implementação da abordagem desenvolvida para sistemas baseados em sintaxe e, ao mesmo tempo, produzir resultados tangíveis que pudessem exemplificar os propósitos gerais da arquitetura e da transportabilidade entre domínios. O mecanismo para a realização da conversão da frase em linguagem *sql* que é utilizado pelo sistema pressupõe conhecimentos sobre o processamento de linguagem natural. Logo, poderá ser efetuado por qualquer programador. Ao analisarmos os resultados produzidos pelo uso dos padrões sintáticos, salientamos que os mesmos, por terem sido definidos a partir de objetos do modelo conceitual, contribuem para a transportabilidade de domínio e, portanto, poderiam ser utilizados por outras aplicações.

Sugestões para Trabalhos Futuros

O assunto desenvolvido neste trabalho dá margem a outras pesquisas relacionadas à área. A seguir, são relacionadas algumas sugestões que se julgam pertinentes e que não foram abordadas neste trabalho e poderão ser acrescidas no mesmo:

- Implementação de funções agregadas do *SQL* (MAX, MIN, SUM, COUNT, AVG), acrescentando os cálculos matemáticos às consultas.
- Implementação do uso de JOIN entre as tabelas.
- Implementação de subconsultas.
- Adição do dicionário de sinônimos e bases de conhecimento para obtenção de nomes (tabelas ou atributos) semanticamente similares.

Referências

(Agosti, 2003). AGOSTI, Cristiano. Interface em Linguagem Natural para Banco de Dados: uma Abordagem Prática. Monografia. Universidade Federal de Santa Catarina Programa de Pós-Graduação em Ciência da Computação, 2003.

(Allen, 1995) ALLEN, J. Natural Language Understanding. 2nd edition. CA: Benjamim/Cummings, 1995.

(Androutsopoulos et al., 1995) ANDROUTSOPOULOS, I., RITCHIE, G. D., THANISCH, P. Natural Language Interfaces to Databases: An Introduction. DAÍ Research Paper Nº. 709, Dep. of Artificial Intelligence, Edinburgh University, Scotland, UK.

(Anick e Peter, 1993) ANICK, PETER G. Integrating natural language processing and information retrieval in a troubleshooting help desk. In IEEE Expert, p. 9- 17, Dec 1993.

(Crane, H., 1993) CRANE, HEWITT D., RTISCHEV, DIMITRY. Pen and voice unite. In BYTE, p. 98-102, Oct 1993.

(Elmasri e Navathe, 1994) ELMASRI, R., NAVATHE, S. B. Fundamentals of Database, Systems. 2nd edition. CA: Benjamim/Cummings, 1994.

(Fischler, A., 1987) FISCHLER, MARTIN A., FIRSCHEIN, OSCAR. Intelligence: the eye, the brain, and the computer. Menlo Park: AddisonWesley, 1987

(Jackson e Moulinier, 2002). JACKSON, P.; MOULINIER, I. Natural Language Processing for Online Applications: Text Retrieval, Extraction and Categorization.

Amsterdan: John Benjamins Publishing Company. v. 5, 225, 2002.

(Long, B., 1994) LONG, B. Natural Language as an Interface Style. Disponível em:. Acessado em: Setembro,2016.

(Mason, 2010). MASON, Oliver. Qtag, Universidade de Birmingham no Reino Unido, 2007. Disponível em: http://www.english.bham.ac.uk/staff/omason /software/qtag.html. Acessado em: 11/12/2016.

(Nantes, 2006). NANTES, L. M. Desenvolvimento de um sistema baseado em linguagem natural para consultas em banco de dados na Web. 63 p. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Universidade do Oeste Paulista, Presidente Prudente, 2008. Disponível em:<http://fipp.unoeste.br/~chico/FIPP/projetos/projeto2008/Monografia_Nantes_2008.pdf>.

(Nunes, 2007). NUNES, RODRIGO ALVES. Sistema Baseado em Linguagem Natural para a Recuperação de Imagens em um Banco de Imagens. Presidente Prudente, 2007.

(Oliveira et al.,2009) OLIVEIRA Neto, João Mendes de. ; TONIN, S. D. ; Prietch, Soraia Silva . Processamento de Linguagem Natural e suas Aplicações Computacionais. In: I Escola Regional de Informática - Regional Norte 1 (ERIN 2009), 2009, Manaus/AM. Anais do I Escola Regional de Informática - Regional Norte 1 (ERIN 2009): Interação X Computação. Manaus/AM: UEA, UFAM e SBC, 2009. v. 1.

(Pacievitch, 2006) PACIEVITCH. T. Tecnologia da Informação e comunicação. Disponível em: http://www.infoescola.com/informatica/tecnologia-da-informacao-e-comunicacao/

(Perché e Pinheiro, 2010). PERCHÉ, Bruno Pinto. ; PINHEIRO, D.; PEREIRA, D. A. Recuperação de Dados em Banco de Dados por meio da Linguagem Natural. Revista exacta, v. 3, N.º 2, 2010.

(Reis et al., 1997) REIS, P., MATIAS, J., MAMEDE, N. Edite - A Natural Language Interface to Databases: a New Dimension for an Old Approach. Proceeding of the Fourth International Conference on Information and Communication Technology in Tourism, ENTER'97, Edinburgh, Escócia. Springer-

Verlag, 1997.

(Russel & Norvig, 2013). RUSSEL, S. e NORVIG, P. Inteligência Artificial. 3 ed. Rio de Janeiro: Campus, 2013.

(Sarmento, 2011) SARMENTO, Luís. Simpósio Doutoral Linguateca. 2006. Disponível em:http://www.linguateca.pt/documentos/SimposioDoutoral2005.html. Acessado em: out.2016

(Savadovsky, 1988) SAVADOVSKY, P. Introdução ao Projeto de Interfaces em Linguagem Natural.São Paulo: SID Informática, 1988. Citado 3 vezes.

(Shneiderman, B, 1998) SHNEIDERMAN, B. Designing the user interface: strategies for humancomputer interaction. Reading, Addinson-Wesley, 3 ed. 1998

(Silva e Lima, 2007)SILVA, Renato Rocha; LIMA, Sergio Muinhos Barroso. Consultas em Bancos de Dados Utilizando Linguagem atural. Juiz de Fora, 2007.

(Souza e Campos, 2006). SOUZA, Solange N. A.; CAMPOS, E. G. L. ; SANTOS, A. R. D. . Uma ferramenta para Definição de Consultas Baseadas em Papéis e Entidades. Revista IEEE América Latina, v. 4, p. 277-282, 2006.

(Terra, 1992) TERRA, ERNANI. Curso Prático de Gramática. Editora Scipione; São Paulo 1992.

(Thro, E., 1991) THRO, ELLEN. The artificial intelligence dictionary. The Lance. A. Leventhal Microtrend Series. San Marcos: Microtrend, 1991.

(Ullman & Widom, 1997) ULLMAN, J. D., WIDOM, J. A First Course in Database Systems. New Jersey: Prentice Hall, 1997.

(Van der Lans, 1993) VAN DER LANS, R. F. Introduction to SQL. Addison-Wesley, 1993

Apêndices

Apêndice A - Tabelas da arquitetura do banco de dados

As tabelas 1,2,3,4,5 a seguir, apresentam os domínios dos bancos de dados utilizados nos testes desta aplicação, assim como sua divisão na primeira parte com os nomes das tabelas e após os nomes de seus atributos como foram adicionados nos bancos de dados. Vale ressaltar que foram feitos 15 testes diferentes para cada tabela, cada atributo e cada valor.

Apêndice 1 – Tabela referente ao BD de Gerenciamento Hoteleiro

tbControleHospede	COH_CODIGO (PK) COH_NOME_FANTASIA COH_RAZAO_SOCIAL COH_CNPJ (UK) PES_CODIGO
tbControleFuncionario	COF_CODIGO (PK) COF_FUNCAO COF_SALARIO COF_TURNO PES_CODIGO (FK)

tbControleQuarto	COQ_CODIGO (PK) COQ_VALOR_DIARIA COQ_NUMERO_PESSOA COQ_NUMERO_APARTAMENTO COQ_APARTAMENTO_TIPO COQ_LOCALIZACAO COQ_ESTADO
tbContaPagar	COP_CODIGO (PK) COP_DESCRICAO COP_VALOR COP_DATA COP_CATEGORIA
tbContaReceber	COR_CODIGO (PK) COR_DESCRICAO COR_VALOR COR_DATA COR_CATEGORIA
tbFluxoCaixa	FLC_CODIGO (PK) COP_CODIGO (FK) COR_CODIGO (FK)
tbEstoque	EST_CODIGO (PK) EST_PRODUTO_TOTAL PRO_CODIGO (FK)
tbPedidosProdutos	PEP_CODIGO (PK) PEP_TOTAL PEP_QUANTIDADE PEP_TIPO

tbPessoa	PES_CODIGO (PK) PES_NOME PES_EMAIL PES_CPF (UK) PES_TELEFONE PES_RG (UK) PES_NASCIMENTO PES_RUA PES_NUMERO PES_BAIRRO PES_CEP PES_UF PES_CIDADE
tbProdutos	PRO_CODIGO (PK) PRO_QUANTIDADE PRO_NOME PRO_PRECO PRO_FABRICACAO PRO_VALIDADE PRO_FORNECEDOR PRO_CNPJ PRO_TIPO
tbQuartoPedido	QUP_CODIGO (PK) COQ_CODIGO (FK) PEP_CODIGO (FK)
tbLocacao	LOC_CODIGO (PK) LOC_DATA LOC_TEMPO LOC_VALOR_TOTAL COQ_CODIGO (FK) COH_CODIGO (FK) COR_CODIGO (FK) QUP_CODIGO (FK)

Apêndice 2 – Tabela referente ao BD Sistema Acadêmico

tbAluno	ALU_CODIGO (PK) ALU_NOME ALU_MATRICULA ALU_CIDADE ALU_TELEFONE ALU_SEXO CUR_CODIGO (FK)
tbAlunoMatricula	ALD_CODIGO (PK) ALU_CODIGO (FK) ODI_CODIGO (FK)
tbCurso	CUR_CODIGO (PK) CUR_NOME CUR_SIGLA CUR_CARGA_HORARIA CUR_NIVEL
tbDisciplina	DIS_CODIGO (PK) DIS_NOME DIS_CARGA_HORARIA CUR_CODIGO (FK)
tbOfertaDisciplina	ODI_CODIGO (PK) DIS_CODIGO (FK) PRO_CODIGO (FK) ODI_ANO ODI_SEMESTRE ODI_TURNO ODI_CAPACIDADE
tbProfessor	PRO_CODIGO (PK) PRO_NOME PRO_EMAIL PRO_DATA_ADMISSAO

Apêndice 3 – Tabela referente ao BD de loja

tbCliente	CLI_CODIGO (PK) CLI_NOME CLI_CPF (UK) CLI_RUA CLI_NUMERO CLI_BAIRRO CLI_CIDADE CLI_SEXO
tbPedido	PED_CODIGO (PK) PED_NUMERO PED_DATA CLI_CODIGO (FK) VEN_CODIGO (FK)
tbProduto	PRO_CODIGO (PK) PRO_NOME PRO_QUANTIDADE PRO_UNIDADE PRO_PRECO_UNITARIO SEC_CODIGO (FK)
tbProdutoPedido	PRP_CODIGO (PK) PED_CODIGO (FK) PRO_CODIGO (FK) PRP_QUANTIDADE

tbSecao	SEC_CODIGO (PK) SEC_NOME
tbVendedor	VEN_CODIGO (PK) VEN_NOME VEN_CPF (UK)

Apêndice 4 – Tabela referente ao BD hospitalar

Hospital••••	codigo_hospital (PK) cnpj (UK) nome endereco telefone

Pessoa	codigo_pessoa (PK) cpf (UK) nome cns data_nascimento sexo cor endereco telefone
Consulta	codigo_consulta (PK) data hora diagnostico sintomas pessoa_cpf hospital_cnpj (FK)
Receita	receita_codigo (PK) codigo_consulta (FK) data hora descricao •••••••••

Exame	exame_codigo (PK) codigo_consulta (FK) data hora descricao diagnostico
Internacao	codigo_internacao (PK) consulta_codigo_consulta (FK) data hora leito diagnostico_inicial diagnostico_final dias_permanencia tratamento hospital_cnpj (FK)
Evolucao	codigo_evolucao (PK) data hora evolucao

Funcionario	funcionario_codigo (PK) data_inicio horas_plantao cart_trab codigo_pessoa (FK) codigo_hospital (FK)
Medico	codigo_medico (PK) especialização crm (UK) nome cpf
Enfermeiro	codigo_enfermeiro (PK) nome cpf (UK)

Apêndice 5 – Tabela referente ao BD de uma concessionária

Cliente	codigoCliente (PK) cpfCliente(UK) nomeCliente dataNasc sexo endereco telefone
Locacoes	CodigoCliente (PK) horaLocacao dataLocacao codigoVendedor (FK)
Veiculo	marca modelo chassi (PK) cor placa (UK) ano preco

Vendedor	codigoVendedor (PK) cpfVendedor(UK) nomeVendedor dataNasc sexo
Fabricante	nomeFabricante cnpj (PK) quantidadeCarros

Apêndice B – Árvore de tomada de decisão de atributos

A árvore de decisão mostrada neste apêndice é referente as decisões tomadas para a etiquetagem do nó valor.