



**UNIVERSIDADE ESTADUAL DO NORTE DO PARANÁ**  
**CAMPUS LUIZ MENEGHEL**

**JUNIOR CESAR ALEXANDRE**

**CONSTRUÇÃO DE UMA FERRAMENTA PARA  
RECONHECIMENTO DE AÇÕES A PARTIR DE TEXTOS  
EM LINGUAGEM NATURAL**

Bandeirantes

2012

**JUNIOR CESAR ALEXANDRE**

**CONSTRUÇÃO DE UMA FERRAMENTA PARA  
RECONHECIMENTO DE AÇÕES A PARTIR DE TEXTOS  
EM LINGUAGEM NATURAL**

Trabalho de Conclusão de Curso  
apresentada à Universidade Estadual do  
Norte do Paraná – *campus* Luiz Meneghel –  
como requisito para aprovação do Curso de  
Sistemas de Informação.

Orientador: Glauco Carlos Silva

Bandeirantes

2012

**JUNIOR CESAR ALEXANDRE**

**CONSTRUÇÃO DE UMA FERRAMENTA PARA  
RECONHECIMENTO DE AÇÕES A PARTIR DE TEXTOS  
EM LINGUAGEM NATURAL**

Trabalho de Conclusão de Curso  
apresentada à Universidade Estadual do  
Norte do Paraná – *campus* Luiz Meneghel –  
como requisito para aprovação no Curso de  
Sistemas de Informação.

**COMISSÃO EXAMINADORA**

---

Prof. Me. Glauco Carlos Silva  
UENP – *Campus* Luiz Meneghel

---

Prof. Graduado. Estevan Braz Brandt Costa  
UENP – *Campus* Luiz Meneghel

---

Prof. Me. Ederson Marcos Sgarbi  
UENP – *Campus* Luiz Meneghel

Bandeirantes, 19 de Novembro de 2012.

Dedico este trabalho à minha família que me deu  
oportunidade de hoje estar aqui.

## **AGRADECIMENTOS**

Agradeço à minha mãe que sempre acreditou em meu potencial, ao meu pai que sempre me deu forças pra continuar em frente e nunca desistir, ao meu irmão que sempre se fez presente me animando e ajudando nessa caminhada, não posso deixar de citar os meus amigos que muitas das vezes deram suporte e ajuda e por fim ao meu orientador que ajudou a tornar esse trabalho possível.

*Não se possui o que não  
se compreende.  
(Johann Goethe)*

## RESUMO

Com a atual evolução tecnológica, a uma enorme e crescente preocupação em se criar sistemas que permitem uma maior interação do usuário para com o sistema, reduzindo esforços e aumentando sua eficiência. Um dos fatores que contribui para com essa evolução é a capacidade do sistema de se comunicar com o usuário através do processamento de linguagem natural. Onde uma ação solicitada pelo usuário pode ser atendida sem maiores esforços.

Este trabalho tem como intuito de apresentar uma ferramenta capaz de localizar e identificar a ação expressa dentro de um texto, onde a escrita desse texto tenha sido feita fazendo uso de linguagem natural do Brasil, posteriormente, essa ferramenta, possa ser utilizada para construção de um Sistema de Diálogo Orientado a Tarefa. Este trabalho apresenta também conceitos de Linguagem Natural, *Parser* e Sistema de Diálogo.

**Palavras-chave:** Linguagem Natural, Sistema de Diálogo, Modelo de Tarefa.

## **ABSTRACT**

With the current technological developments, a huge and growing concern in creating systems that allow more user interaction to the system, reducing effort and increasing their efficiency. One factor that contributes to this evolution is the ability of the system to communicate with the user via natural language processing. Where an action requested by the user can be met without major efforts.

This work has the aim of presenting a tool capable of locating and identifying the action expressed in a text, where the text has been written that made making use of natural language in Brazil later this tool can be used to build a system Task Oriented Dialogue. This paper also presents the concepts of Natural Language, Parser and Dialog System.

**Keywords:** Natural Language, Dialogue System, Model Task.



## LISTA DE FIGURAS

Figura 1 - Estruturação Básica de Sistema de Diálogo.....	21
Figura 2 - Estruturação do Modelo de Domínio.....	22
Figura 3 - Estruturação Básica do Modelo Mundo.....	23
Figura 4 - Estruturação Básica do Modelo de Tarefa.....	24
Figura 5 - Tela inicial do <i>NetBeans IDE</i> .....	26
Figura 6 - Tela inicial do <i>IReport</i> .....	27
Figura 7 - Diagrama de Casos de Uso.....	28
Figura 8 - Diagrama de Atividade para Validar Alfabeto.....	29
Figura 9 - Diagrama de Atividade para Cadastramento de Verbo .....	30
Figura 10 - Diagrama de Atividade para Cadastramento de Verbo no Presente.....	30
Figura 11 - Diagrama de Atividade para Cadastramento de Verbo no Passado.....	31
Figura 12 - Diagrama de Atividade para Cadastramento de Verbo no Futuro.....	31
Figura 13 - Diagrama de Sequencia para Validar Alfabeto.....	32
Figura 14 - Diagrama de Entidade Relacionamento.....	33
Figura 15 - Tela da Ferramenta para Validar Alfabeto.....	36
Figura 16 - Tela da Ferramenta para Cadastrar Verbo.....	37
Figura 17 - Tela da Ferramenta para Cadastrar Verbo no Presente.....	38
Figura 18 - Tela da Ferramenta para Cadastrar Verbo no Passado.....	38
Figura 19 - Tela da Ferramenta para Cadastrar Verbo no Futuro.....	39
Figura 20 - Tela da Ferramenta para Cadastrar Objeto.....	39
Figura 21 - Tela da Ferramenta para Emitir Relatórios.....	40
Figura 22 - Relatório dos Objetos que já foram Cadastrados.....	41
Figura 23 - Relatório dos Verbos que já foram Cadastrados.....	42
Figura 24 - Relatório dos Alfabetos já Testados.....	43
Figura 25 - Informações da Ferramenta.....	44

## LISTA DE GRÁFICOS

Gráfico 1 – Porcentagem de Acertos e Erros no Teste Feito.....	47
Gráfico 2 – Porcentagem de Ações e Objetos Encontrados.....	47
Gráfico 3 – Percentual de Ações e Objetos Encontrados.....	50

## SUMÁRIO

1 Introdução .....	13
1.1 Contexto e Delimitação do Trabalho.....	13
1.2 Formulação do Problema.....	13
1.3 Objetivo Geral .....	14
1.3.1 Objetivo Específico .....	14
1.4 Justificativa.....	14
1.5 Métodos e Materiais .....	15
1.6 Organização do Trabalho.....	15
2 Fundamentação teórica.....	16
2.1 Processamento de Linguagem Natural .....	16
2.1.1 Base de Conhecimento .....	16
2.1.1.1 Léxico .....	16
2.1.1.2 Gramática.....	17
2.1.1.3 Conceitual.....	17
2.1.2 Análise Morfológica.....	17
2.1.3 Análise Sintática.....	17
2.1.3.1 Etiquetagem - <i>Tagging</i> .....	18
2.1.3.2 <i>Parsing</i> .....	18
2.1.4 Análise Semântica .....	19
2.1.4.1 Gramática Semântica.....	19
2.1.4.2 Gramática de Caos.....	19
2.1.5 Análise Pragmática .....	20
2.2 Sistema de Diálogo.....	20
2.3 Modelo Domínio .....	22
2.3.1 Modelo de Discurso.....	22
2.3.2 Modelo Mundo .....	23
2.3.3 Modelo de Tarefa.....	24
3 Desenvolvimento.....	25
3.1 Metodologia.....	25
3.2 Modelagem da Ferramenta VALIDAALFABETO.....	27

3.2.1 Levantamento de Requisitos.....	27
3.2.2 Diagrama de Caso de Uso.....	28
3.2.3 Diagrama de Atividade.....	29
3.2.4 Diagrama de Sequencia.....	32
3.2.5 Diagrama de Entidade Relacionamento.....	32
3.3 Analise Sintática.....	33
3.3.1 Métodos.....	33
3.3.1.1 Split.....	34
3.3.1.2 Classe <i>String Tokenizer</i> .....	34
3.3.1.3 Classe <i>Scanner</i> .....	35
3.4 A ferramenta.....	36
3.4.1 Limitações da Ferramenta.....	44
3.4.2 Testes.....	45
4 Considerações Finais.....	51
Referência.....	52

# 1 INTRODUÇÃO

Com a evolução da tecnologia e a sofisticação dos sistemas, é possível ter um diálogo entre usuário e software/hardware mais natural, tendo em vista que uma linguagem mais natural é mais rápida, por parte do usuário, proporcionando assim um maior rendimento dos trabalhos realizados pelo mesmo. Mas ao mesmo tempo esse tipo de linguagem é mais difícil de ser processado e pode acarretar erros na definição do que é solicitado ao sistema realizar.

O propósito desse trabalho é criar uma ferramenta capaz de detectar a ação que o usuário solicita através de um texto e entender como a ferramenta reage quando lhe é solicitado a realizar uma tarefa, quais os mecanismos que o levam a entender o que o usuário está pedindo, as condições em que a tarefa é solicitada e como ela deve ser apresentada ao final de sua execução.

## 1.1 - CONTEXTO E DELIMITAÇÃO DO TRABALHO

Nesse trabalho será abordado como a ferramenta faz a conversão da linguagem natural para uma linguagem em que ele possa abstrair e executar o que se pede. Será visto conceitos de processamento de linguagem natural, conceitos de Sistema de Diálogo (SD), os modelos que compõem um sistema de diálogo, as técnicas de *parser* e a criação de uma ferramenta capaz de identificar a ação que o usuário está solicitando por um diálogo utilizando linguagem natural.

## 1.2 - FORMULAÇÃO DO PROBLEMA

Como forma de melhoria na execução de tarefas, tem-se um sistema de diálogo baseado em tarefa que apresenta um modelo em que o usuário, através de um diálogo simples, natural, possa interagir com o sistema sem dificuldade, mas, como o sistema consegue entender, qual ação é solicitada pelo usuário já que o mesmo faz uso de uma linguagem de baixo nível?

### **1.3 – OBJETIVO GERAL**

Através de um texto em linguagem natural do Brasil a ferramenta proposta deverá analisar a estrutura linguística e reconhecer qual a ação que está descrita no texto digitado, ou arquivo selecionado.

#### **1.3.1 – Objetivos Específicos**

Fazendo uso da abordagem *parsing*, onde se tem como objetivo verificar como é constituído, gramaticalmente e estruturalmente, um texto ou frase e analisando os sintagmas, que é a segmentação de palavras escritas respeitando determinadas condições estruturais, a formar frases que possuem determinada função. Os sintagmas podem ser nominais e verbais, sequencialmente a estrutura de classificação do pronome pessoal, verbo, artigo e nome, será possível a criação de uma ferramenta capaz de identificar as ações que estão sendo expressas dentro de uma entrada dada pelo usuário.

Sendo assim, serão abordados os temas abaixo:

- Definir os métodos que serão utilizados.
- Desenvolver a ferramenta utilizando de PLN e os métodos de análise Sintática (*Parser*) escolhido.

### **1.4 - JUSTIFICATIVA**

Devido à rápida evolução da tecnologia de IA que permite a criação de sistemas inteligentes, e ainda, sistemas especialistas, que no mundo atual auxilia na realização de tarefas desde as mais simplificadas quanto as mais complexas. Dessa forma, são anulados erros mais comuns cometidos quando somente o homem o fizera e que hoje se faz com somente um clique, ou um comando de voz. Esse projeto se justifica podendo representar a intenção do usuário, onde essa função é amplamente utilizada em SD orientados à tarefa.

## **1.5 - MÉTODO E MATERIAIS**

Este trabalho se inicializará com pesquisas bibliográficas sobre o que são sistemas de diálogos, sistemas de diálogos orientados a tarefa, linguagem natural e modelo de execução de tarefas.

Um próximo passo será confecção de uma ferramenta que identifique as ações que estão expressas em um texto ou frase. Esse processo se caracteriza como sendo parte de um sistema de diálogo orientado a tarefa.

## **1.6 - ORGANIZAÇÃO DO TRABALHO**

O trabalho se encontra dividido em Quatro (4) seções: Introdução, Fundamentação Teórica, Desenvolvimento e Conclusão. Na seção de Fundamentação Teórica, serão mostrados os conceitos de linguagem natural, de sistemas de diálogos e os modelos que o compõe. Na seção de desenvolvimento, será abordado como será desenvolvido a ferramenta, sua modelagem e seu funcionamento. Na seção de Conclusões serão visto as conclusões referente a esse trabalho.

## **2 FUNDAMENTAÇÃO TEÓRICA**

### **2.1 Processamento de Linguagem Natural.**

O processo de linguagem natural constitui no reconhecimento da fala do usuário, ou seja, no reconhecimento da estrutura linguística, do nível de interesse do assunto a ser tratado. Tecnicamente, análise de linguagem utilizada no diálogo natural é feita analisando análise morfológica, sintática, semântica e pragmática (Silva, 2012). Que corresponde respectivamente a análise da fala em si (sozinha) e a análise sintática faz o estudo do conjunto das palavras, relação de uma com as outras dentro de uma oração. Seguindo pela análise semântica, que faz a interpretação do conjunto de todas as sentenças feitas no diálogo e finalmente análise pragmática, que faz a análise do conhecimento obtido pelo diálogo realizado.

#### **2.1.1 - Bases de Conhecimento.**

A base de conhecimento é responsável por armazenar o conhecimento e informações adquiridos, e os dados que lhe já fora reunido, para que se possa utilizar eventualmente quando necessário.

As relações de conectividade de conhecimentos e informações obtidas e armazenadas dentro da base de conhecimento podem ser feitas de maneira a gerar novos conhecimentos específicos ou ainda conceitos específicos de determinado domínio.

A base de conhecimento representada de maneira física, geralmente é criada manualmente e as informações e dados são obtidas através de dicionários e enciclopédias, além do domínio específico do sistema.

##### **2.1.1.1 – Léxico.**

O léxico é uma base de conhecimento que tem como função, fazer as análises linguísticas para o processamento da linguagem natural. Analise essa, que é tanto morfológica, como sintática, semântica e pragmática.



### **2.1.1.2– Gramática.**

A base gramatical é responsável por conter todas as regras gramaticais do diálogo feito através da linguagem natural, observando a estrutura sintática do diálogo.

### **2.1.1.3 – Conceitual.**

A base de conhecimento conceitual fornece uma descrição do mundo em que o modelo está inserido a fim de ilustrar as propriedades, objetivos, relações, entre outros eventos de maneira organizada e estruturada hierarquicamente, relacionando conceitos entre si e fornecendo novos conceitos do domínio específico.

### **2.1.2 – Análises Morfológicas.**

Segundo Silva (2012) análise morfológica tem como função analisar a estrutura da palavra em si, não somente a frase, nem o sentido que ela expressa, mas sim a estrutura da palavra que a compõe, seus sufixos, prefixos e demais características técnicas que lhe é atribuído e que a forma. Dentro do processo computacional, essa análise é feita com auxílio de uma tabela que apresenta os sufixos e os prefixos que constitui a palavra isolada. Com essa técnica é possível encontrar o padrão (radical) de palavras que estão alterados estruturalmente dentro do diálogo realizado, padronizando assim a mesma e permitindo a continuidade da análise para sua sintaxe.

### **2.1.3 – Análise Sintática.**

A combinação de várias palavras de classes diversas pode ser combinada a formar frases. Nessa análise é averiguada e estudada o conjunto das palavras que formam uma frase, oração sendo o parser (programa ou parte de um programa de computador que serve para analisar a estrutura gramatical de uma entrada), parte fundamental desse processo.

### 2.1.3.1 - Etiketagem – *Tagging*.

Ao se falar em etiquetagem de uma frase, atribuímos o nome de sintagmas. (sintagmas é um conjunto de elementos que confere sua classificação de acordo com a característica do núcleo da oração) (Silva 2012).

Os sintagmas são classificados como sendo Nominal (SN) onde o núcleo constitui substantivo ou mais de um, Verbal (SV) que o núcleo é um verbo ou uma locução verbal, Preposicional (SP) que seu início é constituído de uma preposição e Sintagma Adverbial (SAdv) que se caracteriza por seu núcleo ser advérbio ou ainda uma locução adverbial.

A classificação das frases se inicia com SN e SV, que são depois classificadas por SP e SAdv e por seguinte, classificados como verbos, pronome.

Para ser realizada essa classificação, é necessário implementação de parser, que são programas ou apenas um componente de um programa de computador que serve para analisar a estrutura gramatical de uma entrada. Parsers sintáticos podem ser divididos e classificados em três categorias, os que fazem uso de regras ou lexicalistas que usam de dicionários e enciclopédias para testar a formulação das frases em uma linguagem natural, os que fazem uso de estatísticas ou probabilidades, onde se utiliza da contabilidade das palavras que podem ser usadas mais frequentemente e ainda os híbridos ou funcionalistas que usam dos dois outros tipos apresentados acima, onde lhe é atribuído ainda à funcionalidade de uma técnica que é conhecida como TBL (Transformation Based Learning) que é o Aprendizado Baseado em Transformação, que confere a tarefa de listar todas as regras em ordem, onde a cada contato de um novo diálogo, a regra que melhor se aplicar é inserida na lista de regras aprendidas, advérbios, conjunções e, demais categorias que fazem parte da língua portuguesa.

### 2.1.3.2 – *Parsing*.

Analisando-se o *parsing*, verifica que ele é constituído por uma gramática que cria uma árvore de sintagmas nominais e verbais. Árvore de sintagmas funciona da seguinte maneira, primeiro se analisa a sintagmas nominais e verbais, logo, segue para a estrutura de classificação do pronome pessoal, verbo, artigo e nome. Estrutura essa que define e caracteriza uma frase ou oração.

Existe ainda o *parsing* probabilístico que usa de ponderações estatísticas para a contagem na probabilidade de se ter um diálogo que usa regras no conjunto de palavras de treinamento. Isso permite sanar os problemas com ambiguidade que possa ocorrer com palavras que podem assumir mais de uma classe gramatical dependendo de onde ela é inserida na frase. Mas ainda sim, existe um problema no *parsing* probabilístico, que é o peso em que cada classe gramatical deve assumir, podendo possuir o mesmo peso para construções diferentes. Uma solução para esse problema esta na utilização dos indicadores lexicais em que se busca identificar diferentes ramificações dos diversos sintagmas.

#### **2.1.4 – Analise Semântica.**

Na análise semântica, se faz uso de alguns modelos léxicos que ao serem relacionados, atribuindo significado as palavras ou expressões.

Após análise morfológica, a análise semântica é embutido nas arvores de parsers, que são chamados de anexos semânticos, que fazem a classificação das palavras. Para tal, ainda é necessário à utilização de técnicas como as chamadas redes semânticas, gramáticas semânticas e ainda modelos conexionistas.

##### **2.1.4.1 – Gramática Semântica.**

A gramática semântica é utilizada para amenizar os problemas de ambiguidade que possa eventualmente ocorrer. Tendo como função, a atribuição de regra a cada frase, o que dificulta a utilização da mesma em diferentes domínios. Em sua construção, a gramática semântica é feita através de símbolos que representam à semântica.

##### **2.1.4.2 – Gramática de Caos.**

Com a gramática de caos, é possível obter as características de uma dada sentença e a obtenção de informações que correspondem ações e eventos como o ator, a ação, o objeto a origem entre outros tipos de informações que se encontra dentro de uma frase, que pode ser de fácil compreensão, simples, ou de difícil compreensão, frases mais complexas que possuem varias ações e eventos dentro de si.

### **2.1.5 – Análise Pragmática.**

Na análise pragmática, é construído o conhecimento sobre o diálogo depois de ser analisado a sua morfologia, sintaxe e semântica. Nessa análise o erro de ambiguidade é corrigido, mas deve ser levado em consideração que a alteração do mesmo pode atribuir um novo significado a sua estrutura.

Não somente a ambiguidade, mas outros fatores podem alterar o sentido da frase, como o a ordem das palavras. Ainda existe o caso das conjunções adverbiais, que dão um novo sentido se mal utilizados no diálogo.

Análise pragmática visualiza todo o conteúdo do diálogo, não somente uma frase, pois o mesmo pode não atribuir resultados significativos para sua compreensão. Existindo ainda, outras características que influenciam no processo, como exemplo os tipos de pronomes, advérbios e demais outros. Jain, Mital, Kumar, Mukerjee e Raina propuseram um sistema capaz de identificar e gerar um diálogo quando o usuário faz uso de anáfora.

## **2.2 – Sistemas de Diálogo**

Segundo Mourão (2005), um sistema de diálogo é um sistema que possibilita o usuário e sistema a ter uma interação onde seus objetivos, sejam eles quais forem dentro do contexto em que o sistema foi criado para fazer, sejam alcançados com o mínimo de esforço através de um diálogo simples.

Sistemas de Diálogo vem com uma proposta de executar ações solicitadas pelos usuários através de um diálogo que possibilita o usuário se comunicar com o sistema de uma forma mais natural, o que contribui para um melhor desempenho na execução de determinada tarefa, pois o mesmo não fica limitado a uma comunicação estreita e sistemática, agilizando assim o processo de produção e aumentando a sua produtividade.

Em um diálogo, se deve atentar para a estrutura intencional e atencional. Roman 2001 diz que uma estrutura intencional corresponde ao nível de intenção, que, se pode observar dentro de um discurso, diálogo e que a estrutura atencional se refere ao nível que se pode observa de atenção que ocorre durante o diálogo.

O Sistema de Diálogo tem como estruturação básica seis componentes, que seria o reconhecedor de fala, interpretador de linguagem natural, gestor de diálogo, gestor de tarefa,

gerador de linguagem natural e gerador de fala, que respectivamente seria o mecanismo que reconhece a fala dada pelo usuário, o interpretador é o mecanismo que, identifica as ações dentro do diálogo, o gestor tem a função de estabelecer se existem informações coerentes e suficientes para realizar a ação desejada pelo usuário, o gestor de tarefas tem como objetivo executar as tarefas solicitadas pelo usuário, o gerador de linguagem natural assume o papel de formular a resposta da ação que o usuário solicitou e o gerador de fala tem como objetivo transmitir a resposta formulada para o usuário.

Abaixo, a Figura 1 mostra a estruturação do SD conforme descrito.

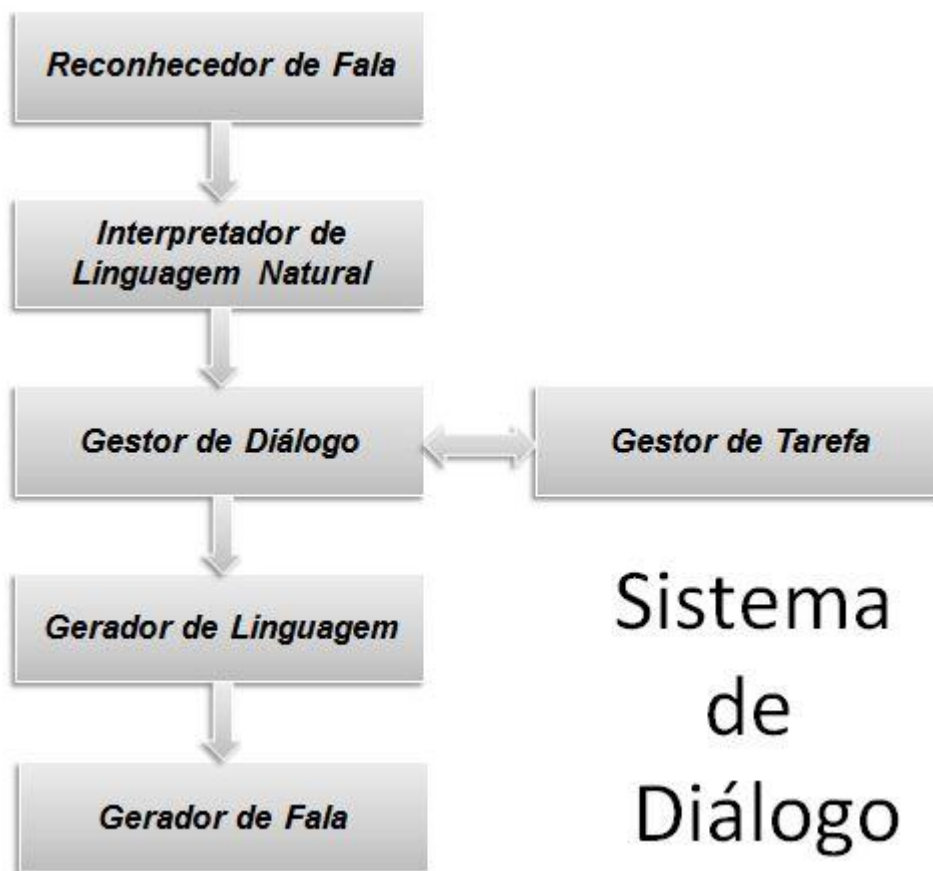


Figura 1 – Estruturação Básica de Sistema de Diálogo

Dentre esses componentes, existe o modelo de domínio, o qual será abordado no próximo tópico.

## 2.3 – Modelo de Domínio.

O modelo de domínio, segundo Filipe é a representação do conhecimento do mundo em que o sistema está inserido. Seria o modelo do ambiente para qual o sistema foi criado. O pleno funcionamento do sistema depende do quão adequado o seu domínio está montado. As tarefas corresponderam somente ao conhecimento de que o domínio está munido.

Geralmente o modelo de domínio se encontra fora do Sistema de Diálogo executando as tarefas solicitadas pelo usuário (Ferreira e Filipe)

A Figura 2 a seguir ilustra quais os modelos que constitui o modelo de domínio.

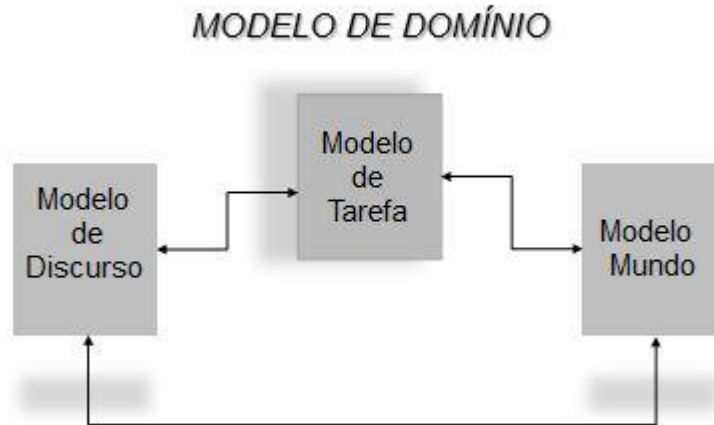


Figura 2 – Estruturação do Modelo de Domínio

### 2.3.1– Modelo de Discurso.

Dentro do modelo de discurso é que se encontram os conceitos referentes ao domínio do sistema. Conceitos esses que identificam o que o sistema pode fazer e que podem ser utilizados para estabelecer uma comunicação com o usuário.

Segundo Filipe, o conhecimento munido no modelo de domínio é especialmente representado no modelo de discurso, podendo-se assim dizer que o suporte do modelo de domínio é o modelo de discurso.

Dentro do modelo de discurso se encontra também o descritor linguístico e descritor semântico, que correspondem respectivamente à ferramenta que faz análise sintática das palavras que o usuário utilizou no diálogo com o sistema, esse processo consiste em analisar a estrutura das palavras e o descritor semântico é a ferramenta que analisa a estrutura da frase em si, do que ela significa.

### 2.3.2 – Modelo Mundo

No modelo mundo se descreve as características que constituem o domínio do sistema.

O modelo mundo é responsável por identificar o ambiente em que o sistema se encontra, quais as características que definem como o sistema irá se portar diante de determinada tarefa. Em sua estrutura, o modelo mundo apresenta dois componentes, que são a Hierarquia de tipos e o mediador, o qual a hierarquia de tipos é responsável pela descrição das classes que compreendem dentro do domínio, dentro da realidade em que o sistema está inserido. O mediador é responsável pela descrição dos artefatos que fazem parte da classe dentro do domínio.

A Figura 3 ilustra a formação básica do modelo mundo.

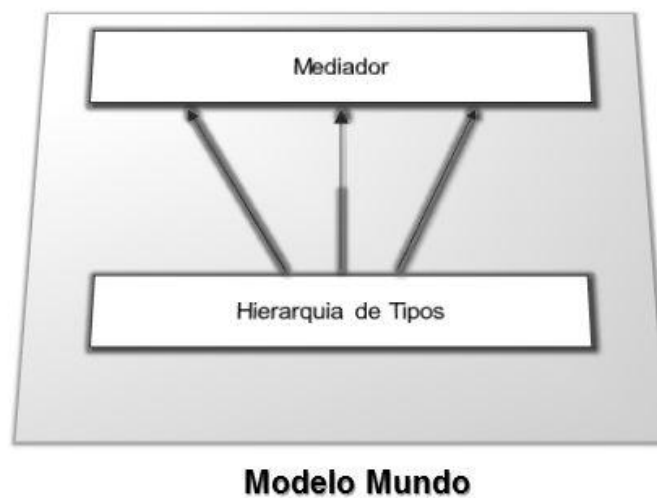


Figura 3 - Estruturação Básica do Modelo Mundo.

### 2.3.3 – Modelo de Tarefa

No modelo de tarefa, são encontrados a classificação das tarefas que o sistema, dentro de seu domínio, possa realizar.

Para a realização dessas tarefas, o modelo precisa estar a par da situação em sua volta. Utilizando como exemplo uma “janela”, dado um comando para o sistema fechar a janela, o modelo precisa verificar se a janela do quarto já não esta fechada, se já estiver o sistema deve emitir uma resposta dizendo que é impossível realizar o comandando, essa abordagem é o que se chama de estado do mundo.

A Figura 4 mostra basicamente como é formada o modelo de tarefas.

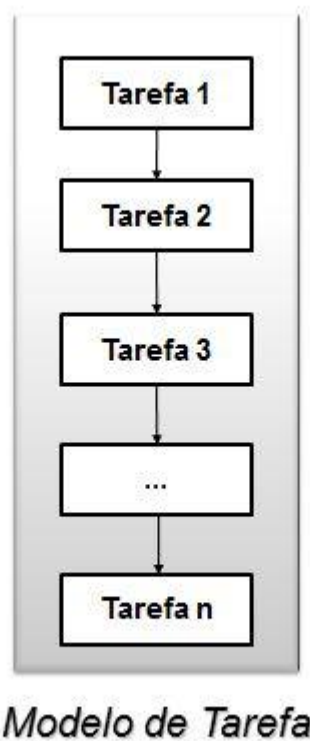


Figura 4 – Estruturação Básica do Modelo de Tarefa

Definido a tarefa solicitada pelo usuário, o sistema a realiza e mais uma vez se inicia um novo diálogo.



### 3 - Desenvolvimento.

A ferramenta proposta objetiva-se comparar um alfabeto de entrada, analisar palavra por palavra, validando este alfabeto e verificando qual é a ação do texto informado qual sua respectiva finalidade. Como por exemplo: - Um alfabeto de entrada “comprarei um carro.”, a ferramenta detectará e validará que o alfabeto contém um verbo + objeto, onde o verbo “comprar” será a ação e o “carro” será a finalidade (objeto), assim apresentando que o objetivo do alfabeto é “comprar”, e o objeto é “carro”.

Desta forma, adotaremos análise sintática (*parser*), onde teremos nosso alfabeto de entrada, regras e critérios para validar se o alfabeto informado é válido. De acordo com as inserções, as a ferramenta.

#### 3.1 - Metodologia

Para concepção da ferramenta, será utilizado o IDE (*Integrated Development Environment*) *NetBeans*. A escolha do mesmo deu-se por ser um ambiente de desenvolvimento multiplataforma, orientado a objeto, totalmente escrita na linguagem Java, hoje uma das linguagens mais exigidas e buscadas pelo mercado, além suportar linguagem orientada a objetos, a IDE suporta também varias linguagens de programação, como C, C++, *Ruby*, PHP, XML e até HTML. A figura 05 representa a tela inicial do *NetBeans* na versão 7.2, o qual foi utilizada.

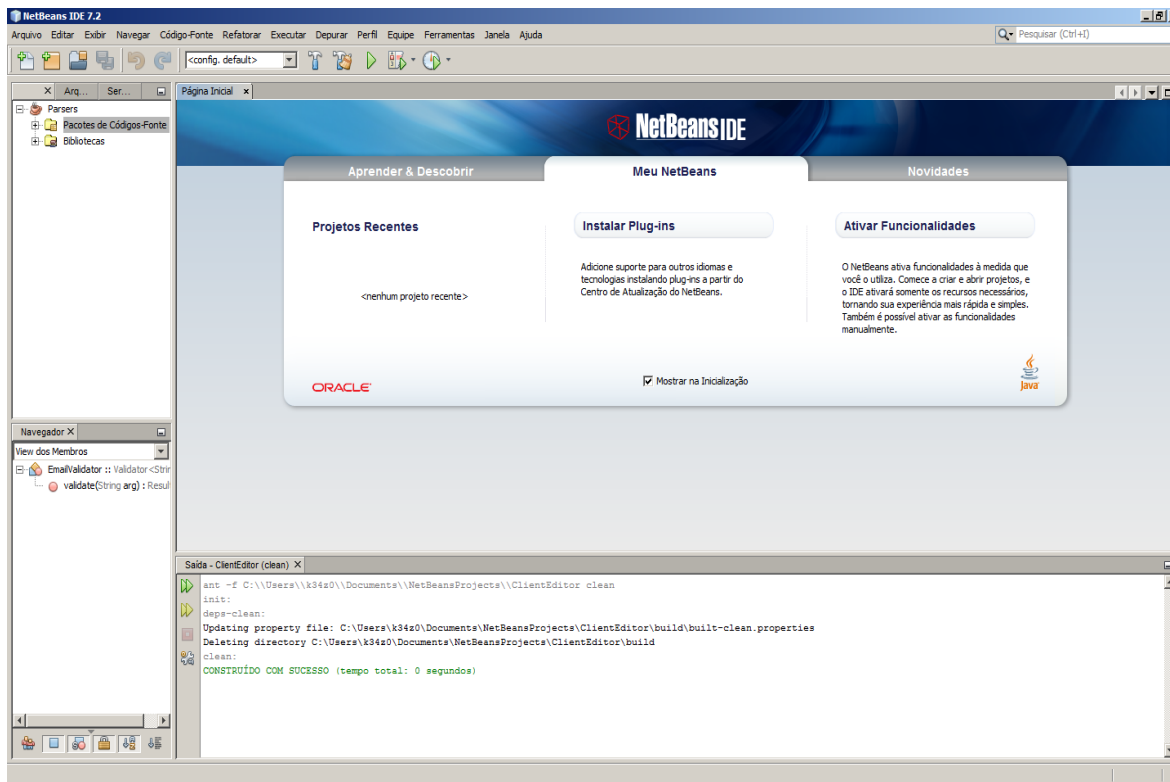


Figura 5 – Tela inicial do *NetBeans IDE*

O MySQL é um sistema de gerenciamento de banco de dados (SGBD), que utiliza a linguagem SQL (Linguagem de Consulta Estruturada, do inglês *Structured Query Language*) como interface. É atualmente um dos bancos de dados mais populares, com mais de 10 milhões de instalações pelo mundo. O SGBD suporta praticamente qualquer plataforma atual, além da compatibilidade com diversos drives como ODBC, JDBC e .NET e também módulos para diversas linguagens de programação como o JAVA o Delphi, C, C++, PHP dentre outros. Uma das vantagens de se utilizar esse SGBD é por ser um *software livre*, por sua facilidade de uso e por exigir pouco de *hardware*. (MySQL. In: WIKIPÉDIA, a enciclopédia livre. [S.l.]: Wikipédia Foundation, 2006. Disponível em: <<http://pt.wikipedia.org/wiki/MySQL>> Acesso em 18 de Setembro de 2012 as 21:00)

Para a construção de relatórios foi utilizado a ferramenta IReport, que vem com a promessa de facilitar a sua emissão por possuir uma interface gráfica desenvolvida em Swing. Ele dispõe de importantes ferramentas para desenvolver relatórios complexos e demorados. Mesmo sem nenhum conhecimento das bibliotecas do JasperReports, é possível criar relatórios muito interessantes através ferramenta e aprender a entender as *tags XML*

utilizadas no JasperReports.(Lima, Bruno R. Tutorial de IREPORT. Disponível em: <<http://javafree.uol.com.br/artigo/3154/Tutorial-de-IREPORT.html>>. Acesso em 19 de Setembro de 2012 as 16:00). A figura 06 ilustra a tela principal do Ireport.

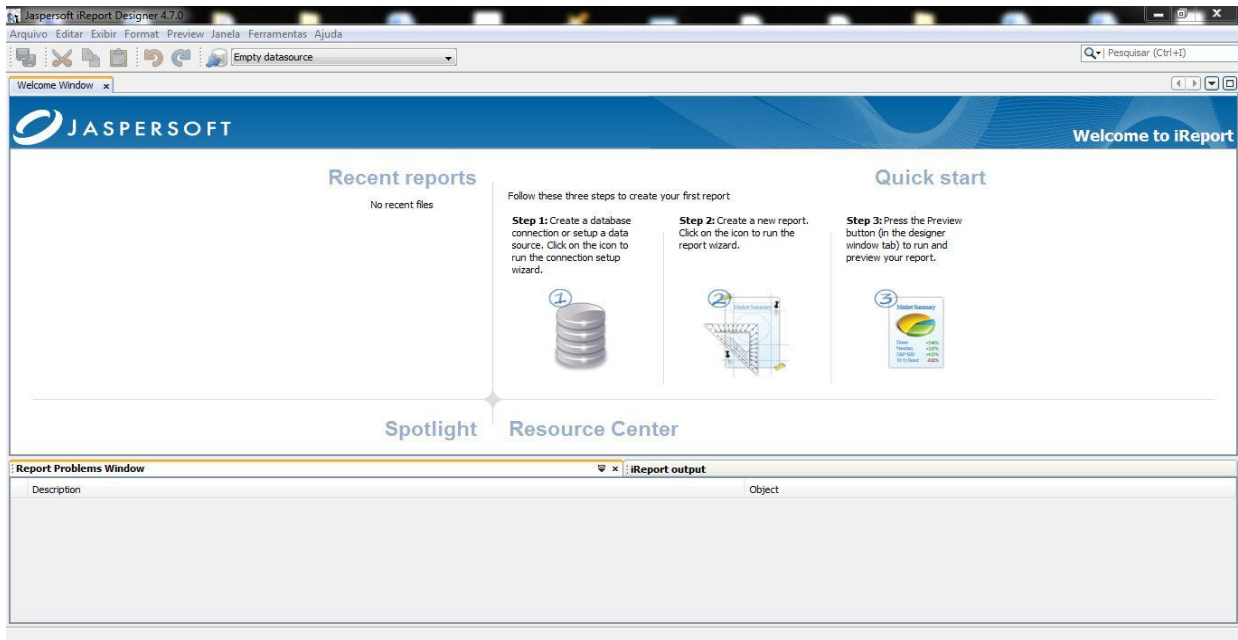


Figura 6 – Tela inicial do *IReport*

## 3.2 - Modelagem da Ferramenta VALIDAALFABETO

Nesta sessão será apresentada a modelagem da ferramenta proposta segundo os critérios definidos do UML, representando através dos diagramas de caso de uso, diagrama de atividades, de sequência e da entidade relacionamento feitos através da ferramenta ASTAH. Além do levantamento de requisitos, visto a seguir.

### 3.2.1– Levantamento de Requisitos.

A ferramenta proposta deve atender as seguintes exigências:

- Receber um texto ou frase validando seu alfabeto e identificar a ação expressa dentro do mesmo, além de identificar o objeto dessa ação.
- Possuir a opção de receber o texto dentro de um arquivo de extensão .txt.
- Exibir mensagens, de quando a ação não for passível de identificação, para o usuário realizar o cadastramento do mesmo na base de dados.
- Apresentar abas que separem as ações que podem ser executadas na ferramenta, como validação do alfabeto, cadastramento de verbo, cadastramento dos verbos conjugados, cadastramento dos objetos, emissão de relatórios e dados sobre a ferramenta, para que o usuário consiga se situar dentro da ferramenta com facilidade.
- Exibir os relatórios de todos os verbos, objetos e textos e alfabetos já validados.

### 3.2.2 - Diagrama de caso de uso.

Através do levantamento de requisitos foi possível identificar os casos de uso da ferramenta, visto na figura 7.

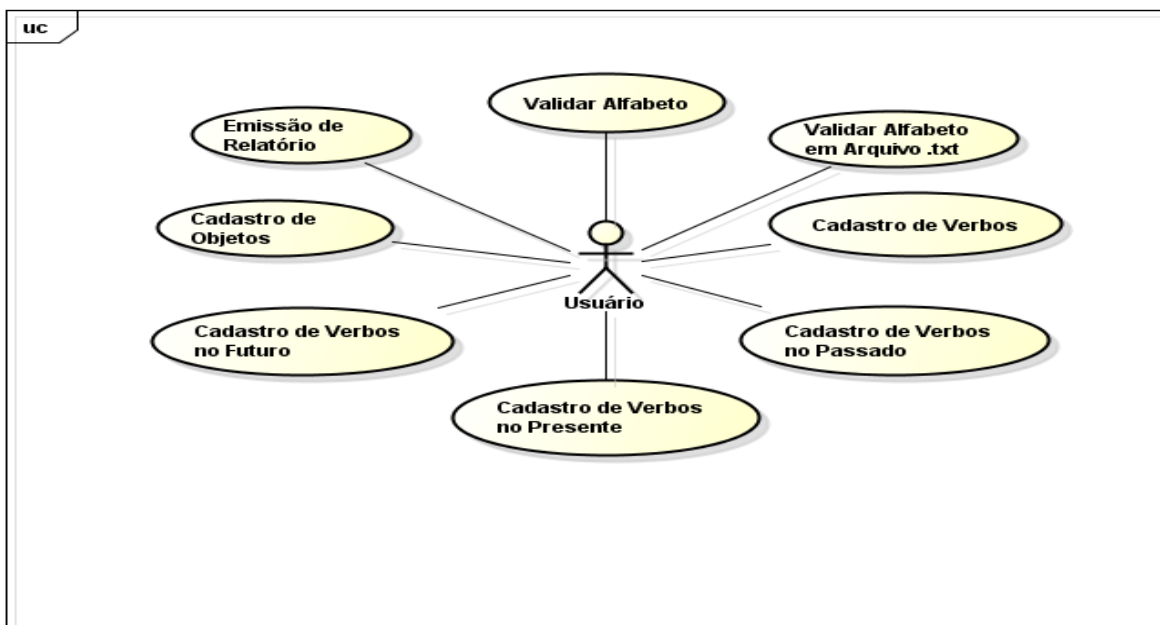


Figura 7 – Diagrama de Casos de Uso.

### 3.2.3 – Diagrama de Atividade.

Os diagramas a seguir ilustram os passos para execução de determinadas ações.

**Validar um texto:** onde o usuário deve inserir uma frase dentro do campo texto ou ainda selecionar a opção de validar um arquivo onde ele pode escolher um arquivo de extensão .txt para validar a frase ou texto descrito no arquivo e em seguida clicar sobre o botão de validação de arquivo que retornará uma mensagem dizendo qual a ação foi localizada e qual objetivo dessa ação encontrada. A figura 8 ilustra o diagrama de atividade da ação de validar um texto.

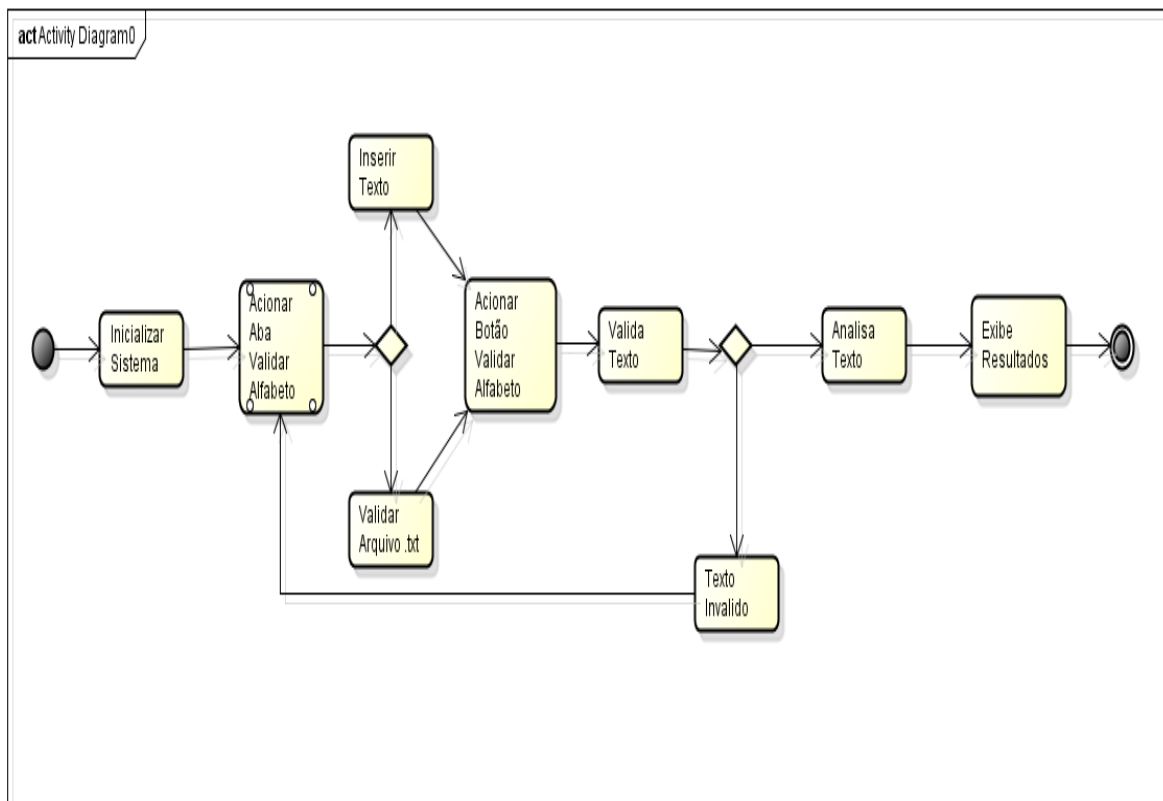


Figura 8 – Diagrama de Atividade para Validar Alfabeto.

**Cadastramento de Verbo:** Onde o usuário deve acionar a aba de cadastramento, inserir o verbo desejado e clicar sobre o botão “cadastrar”. A figura 9 ilustra o diagrama de atividade da ação de cadastramento de verbo.

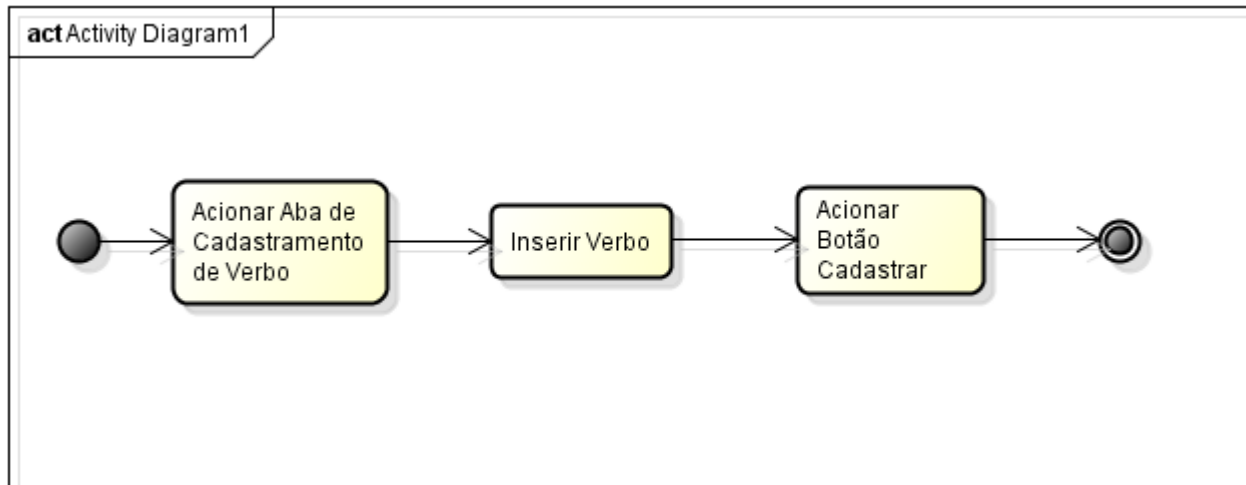


Figura 9 – Diagrama de Atividade para Cadastramento de Verbo.

**Cadastramento de Verbo conjugado no Tempo Presente:** O usuário deve acionar a aba de cadastramento de verbo presente, selecionar o verbo a ser conjugado no tempo presente, inserir a sua conjugação no campo texto e acionar o botão cadastrar. A figura 10 ilustra o diagrama de atividade da ação de cadastramento de verbo conjugado no tempo presente.

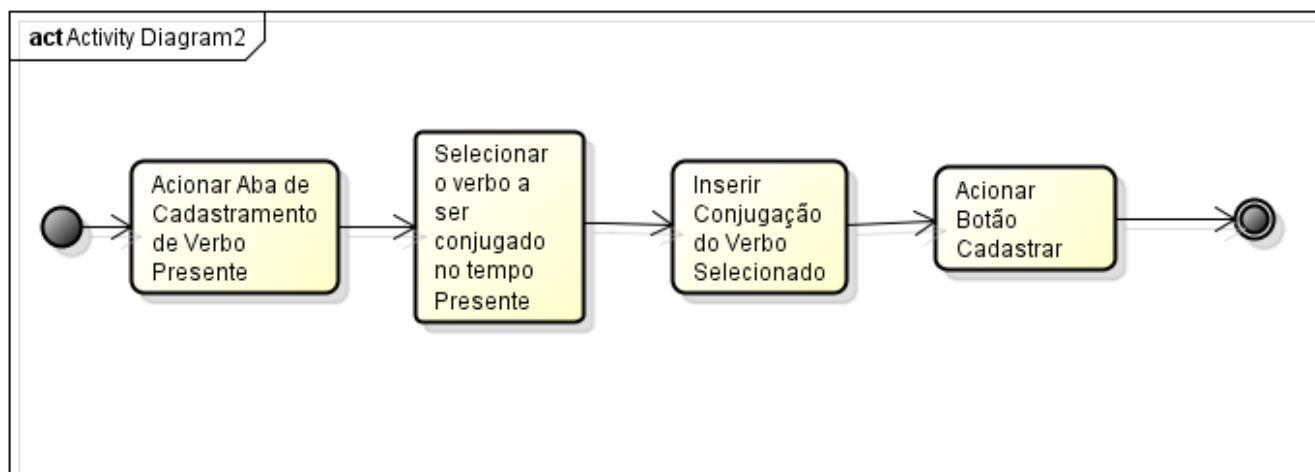


Figura 10 – Diagrama de Atividade para Cadastramento de Verbo no Presente.

**Cadastramento de Verbo conjugado no Tempo Passado:** Segue o mesmo roteiro do cadastramento do verbo conjugado no tempo presente mas nesse caso o verbo é conjugado no tempo passado. A figura 11 ilustra o diagrama de atividade da ação de cadastramento de verbo conjugado no tempo passado.

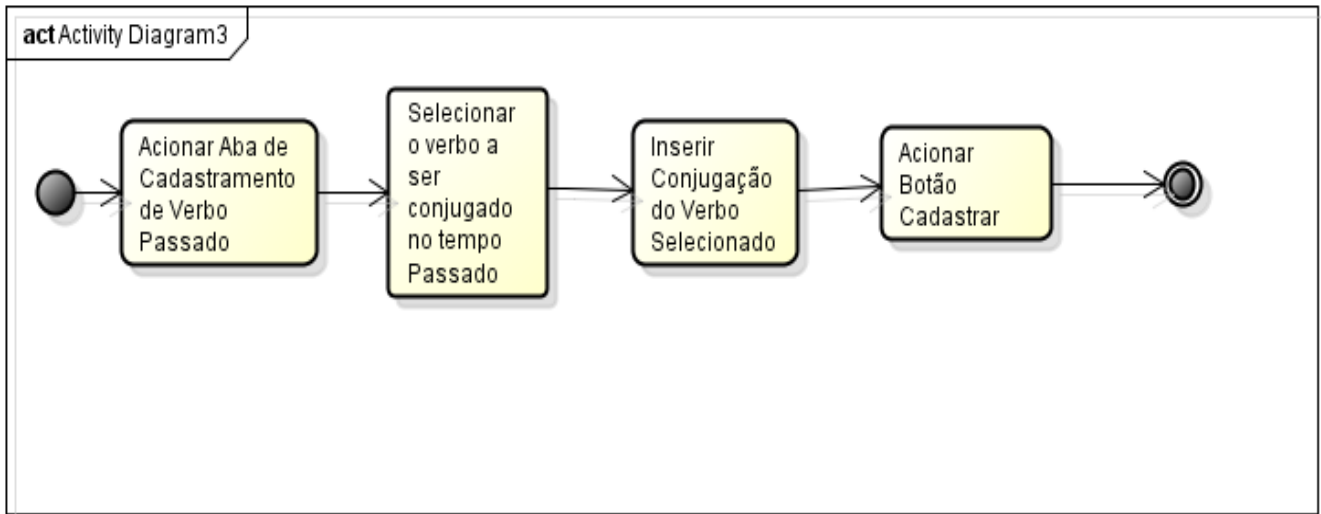


Figura 11 – Diagrama de Atividade para Cadastramento de Verbo no Passado.

**Cadastramento de Verbo conjugado no Tempo Futuro:** Segue o mesmo roteiro do cadastramento do verbo conjugado no tempo presente e no tempo passado, mas nesse caso o verbo é conjugado no tempo futuro. A figura 12 ilustra o diagrama de atividade da ação de cadastramento de verbo conjugado no tempo futuro.

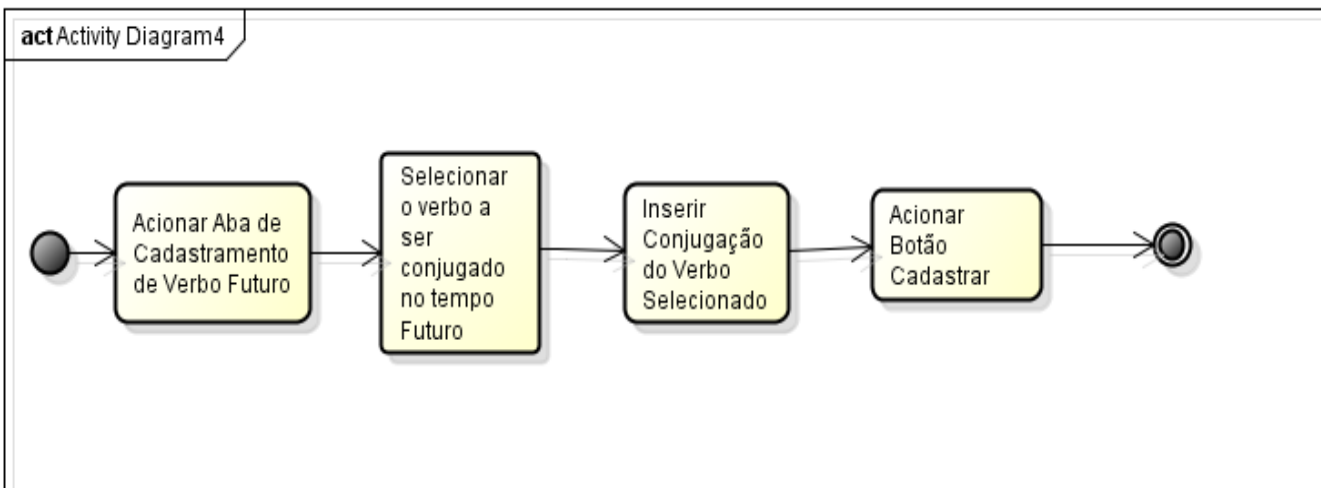


Figura 12 – Diagrama de Atividade para Cadastramento de Verbo no Futuro.

### 3.2.4 – Diagrama de Sequência.

O diagrama de sequência ilustra a mesma ação de validar um texto descrito no diagrama de atividade. A figura 13 ilustra o diagrama de sequência da validação de um texto.

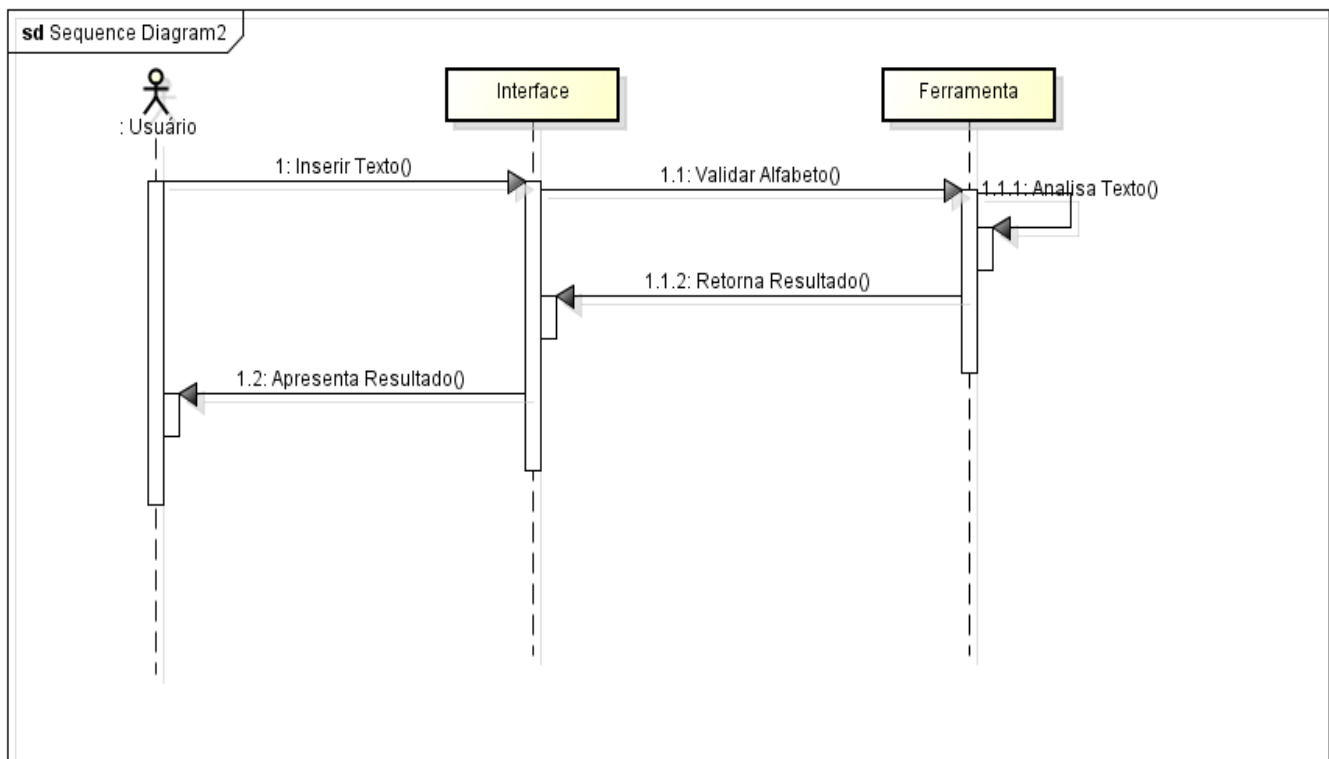


Figura 13 – Diagrama de Sequencia para Validar Alfabeto.

### 3.2.5 - Diagrama Entidade Relacionamento.

O diagrama mostrado pela figura 14 ilustra a estrutura das tabelas da ferramenta com seus níveis de relacionamentos.



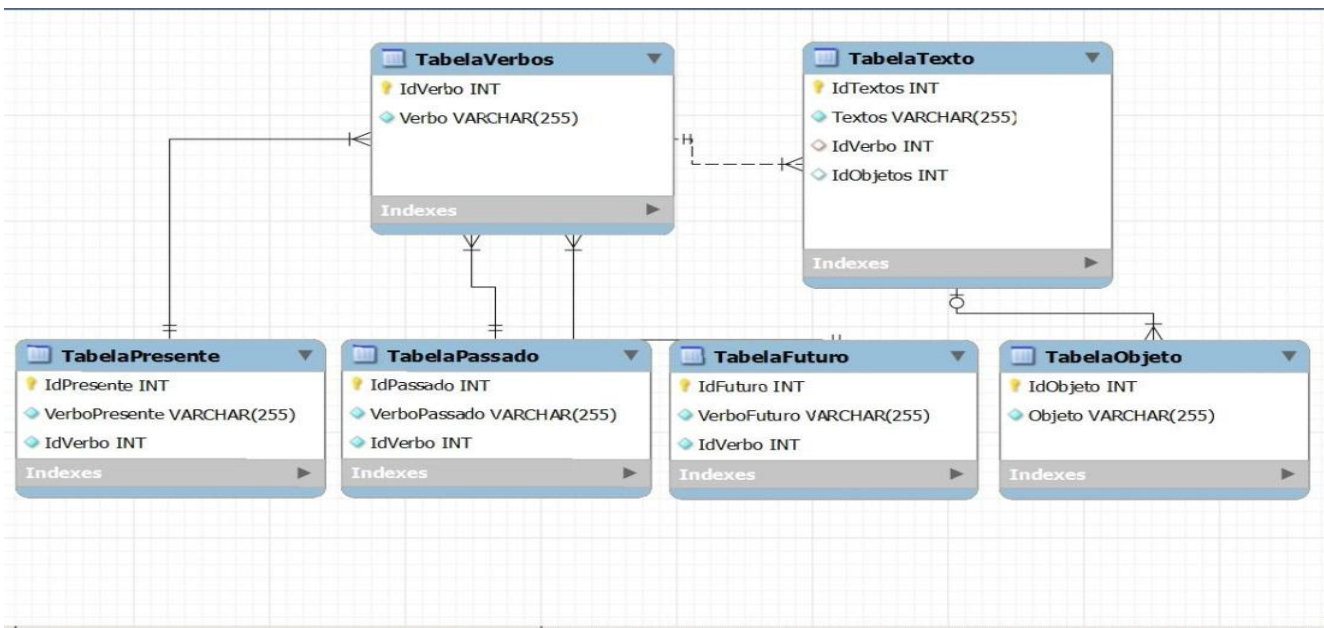


Figura 14 – Diagrama de Entidade Relacionamento.

### 3.3 - Análise Sintática.

#### 3.3.1 - Métodos:

Teremos duas formas de entrar com o alfabeto: Via texto ou leitura de um arquivo. Teremos como alimentar nossa base de dados pela sugestão (caso a ferramenta ainda não tenha cadastrado a situação) e outra clicando na aba cadastro e adicionar seus respectivos verbos e objetos.

A ferramenta será capaz de emitir relatórios contendo dados os verbos e suas conjugações, os objetos e também as frases e textos já escritos no banco de dados. Para isso usamos a ferramenta IReport que permite a criação desses relatórios mais facilmente, além de possuir extensões necessárias para o mesmo.

Para realizar a operação de validação do alfabeto foi utilizado técnicas encontradas no site da *Oracle*, como *Split*, *Tokenizer*, *Scanner* vistas na próxima sessão.

### 3.3.1.1 *Split*.

Segundo o material da Oracle, o método *split* se divide em torno de partidas da expressão regular incluídos na matriz resultante, ou seja, o método quebra uma *String* em várias substrings a partir de um caractere definido.

Tomemos como exemplo a *String* “frutas”, que recebe os valores “abacaxi, laranja, morango”, e que o caractere definido seja “,” utilizando de um array de três posições, conforme o esquema abaixo:

```
String frutas = "abacaxi,laranja,morango";  
String array [ ] = new String [3];  
array = frutas.split(",");
```

Utilizando o seguinte trecho:

```
system.out.println(frutas[0]);  
system.out.println(frutas[1]);  
system.out.println(frutas[2]);
```

Iremos obter o seguinte resultado:

```
Abacaxi  
Laranja  
Morango
```

### 3.3.1.2 Classe *StringTokenizer*.

A classe *StringTokenizer* permite a quebra de uma *String* em *Tokens* através de um delimitador escolhido.

Um *token* é uma sequência de caracteres que são separados por um delimitador, ou seja, seu tamanho depende do caractere definido como delimitador. Esse método pode ser utilizado para fazer a varredura de linhas em um arquivo *.txt*.

### 3.3.1.3 - Classe *Scanner*.

Segundo o material da Oracle, um *Scanner* pode analisar textos simples primitivos e *string* usando expressões regulares.

O *Scanner* quebra o texto introduzido em *Tokens* através de um delimitador, que por padrão corresponde espaços em branco. Este método é idêntico aos dois outros métodos visto acima, mudando a estrutura de como o código é introduzido.

```
Entrada String = "1 peixe 2 peixe vermelho peixe azul peixe";
Scanner s = new Scanner (entrada) useDelimiter ("\\s * peixe \\s *");
System.out.println (s.nextInt ());
System.out.println (s.nextInt ());
System.out.println (s.next ());
System.out.println (s.next ());
s.close ();
```

Imprime o seguinte resultado:

```
1
2
vermelho
azul
```

Os métodos *next* e *hasnext* têm como função pular para a próxima entrada de *token* depois do delimitador.

Conforme for atribuído o delimitador, a possibilidade de serem devolvidos símbolos vazios. Tendo como exemplo o delimitador de espaço, ao ser utilizado o “[ ]+” o método

retornará os casos em que mais de uma vez o delimitador foi acionado sequencialmente, por exemplo quando o usuário digitar sem perceber dois espaços sequenciais. Diferente de quando é usado somente o delimitador “[ ]” que retornará toda vez ao se deparar com um espaço.

### 3.4 – A Ferramenta

A ferramenta possui sete telas, a qual o usuário pode se localizar pelas abas, respeitando uma ordem de uso, onde a tela principal é “carregada” primeira.

Ao começar pela tela principal temos duas opções para efetuar a validação do texto. Na primeira opção, é possível validar o alfabeto escrevendo no campo texto e clicando sobre o botão validar. Na segunda opção, é possível carregar o texto que está dentro de um arquivo de extensão .txt, Conforme apresenta a figura 15.

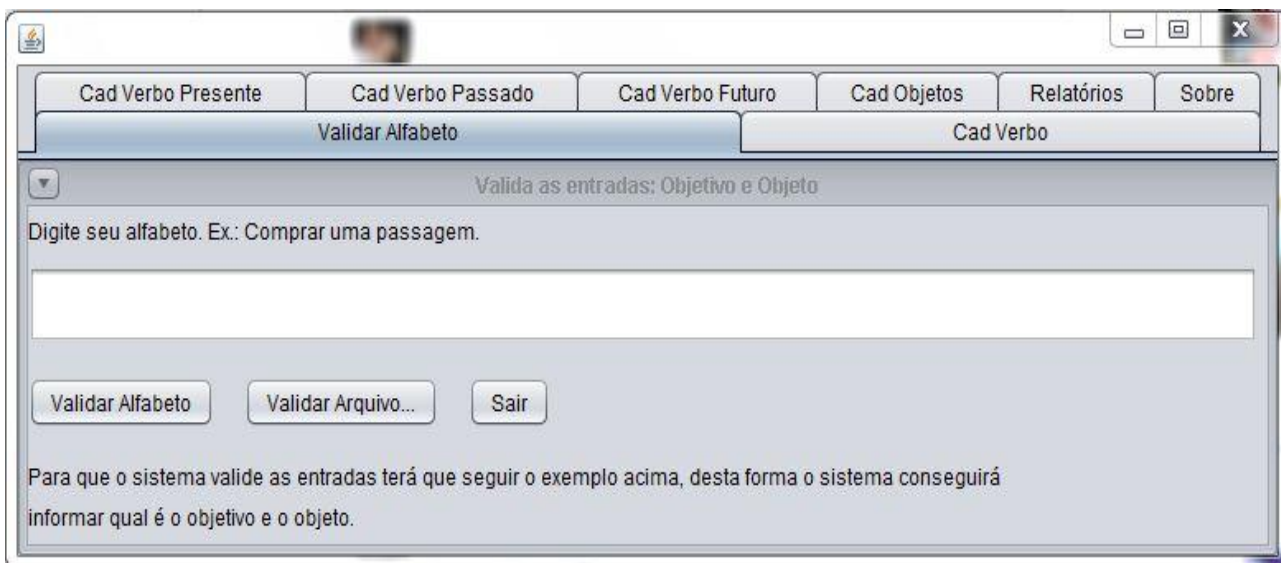


Figura 15 – Tela da Ferramenta para Validar Alfabeto.

Ao deparar com uma frase onde a ferramenta não consiga encontrar seu objetivo, se abre uma caixa de diálogo que solicita ao usuário cadastrar o verbo que está presente no texto ou frase informado, juntamente com suas conjugações e os objetos existentes no texto.

Sendo dessa maneira que se torna possível localizar as ações e o objetivo da frase, munindo a ferramenta com informações que podem ser úteis para uma próxima validação.

A figura 16 mostra como se dá o cadastramento do verbo.

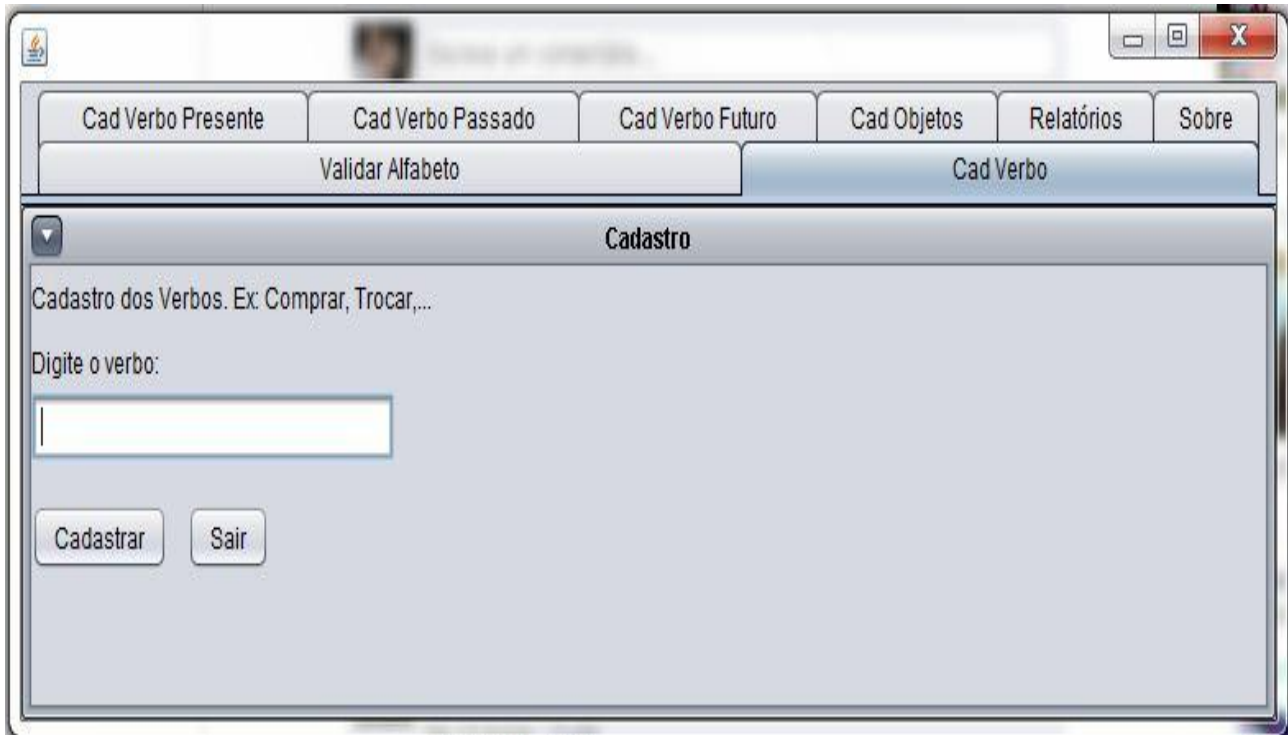


Figura 16 – Tela da Ferramenta para Cadastrar Verbo.

É importante destacar que todas as telas apresentam um exemplo de como proceder, caso o usuário não saiba como realizar alguma operação. Apresenta também o botão de sair, que irá fechar o programa a qualquer momento que o usuário desejar. Além de destacar a aba que corresponde a tela ao qual o usuário estiver.

Após o cadastramento do verbo, o usuário pode cadastrar suas conjugações no estado “presente”, “passado” e “futuro”. Existem muitos estados de conjugações do verbo, mas as mais comuns de se ouvir informalmente são as das três opções mencionadas acima. A figura 17 corresponde a tela de cadastramento do verbo conjugado no presente.

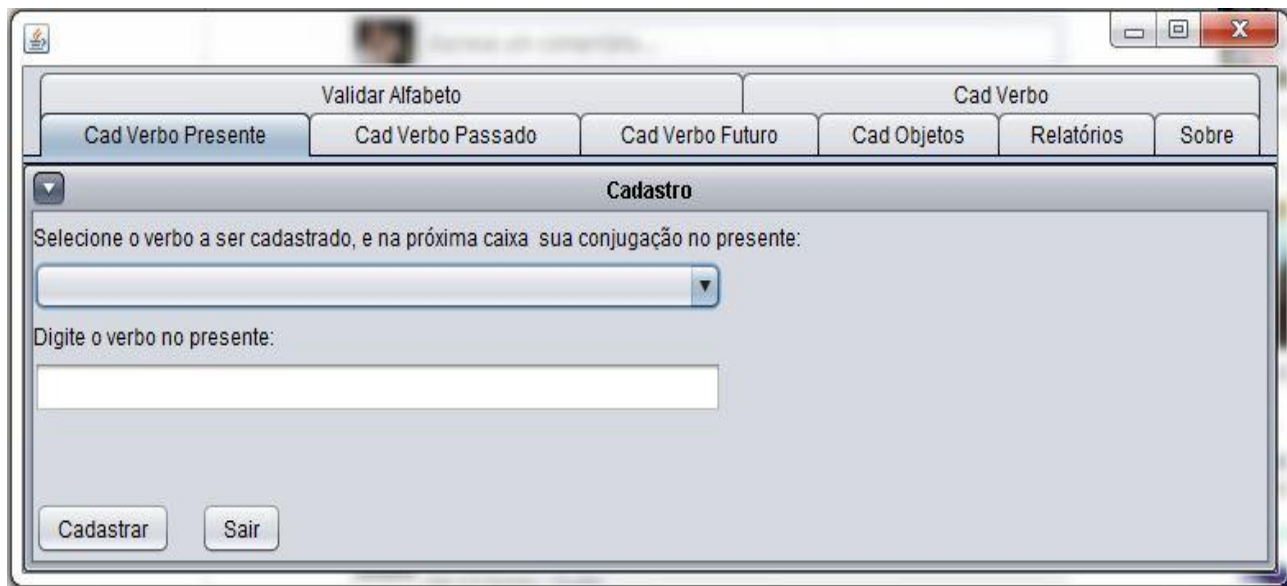


Figura 17 – Tela da Ferramenta para Cadastrar Verbo no Presente.

A tela de cadastramento apresenta uma componente *Drop Down*, onde ao ser clicado abre um leque de verbos cadastrados, onde possibilita que o usuário escolha o verbo para se fazer a sua conjugação no estado presente que é escrita pelo usuário no campo texto e depois cadastrado.

As figuras 18 e 19 representam as telas do cadastramento do verbo no passado e no futuro, que apresentam as mesmas características da tela de cadastramento do verbo presente visto na figura 17

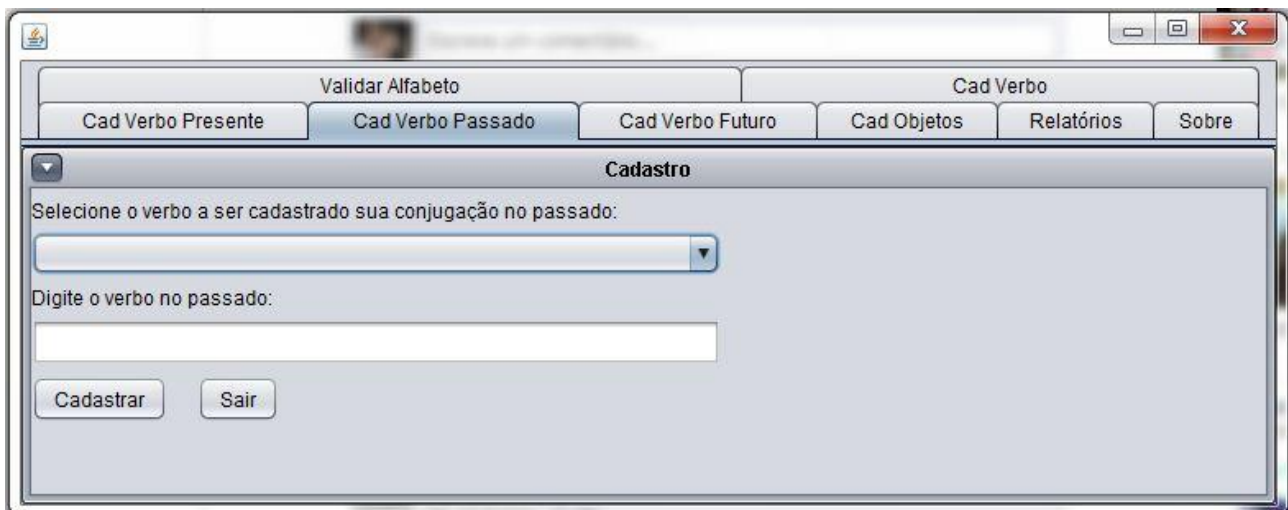


Figura 18 – Tela da Ferramenta para Cadastrar Verbo no Passado

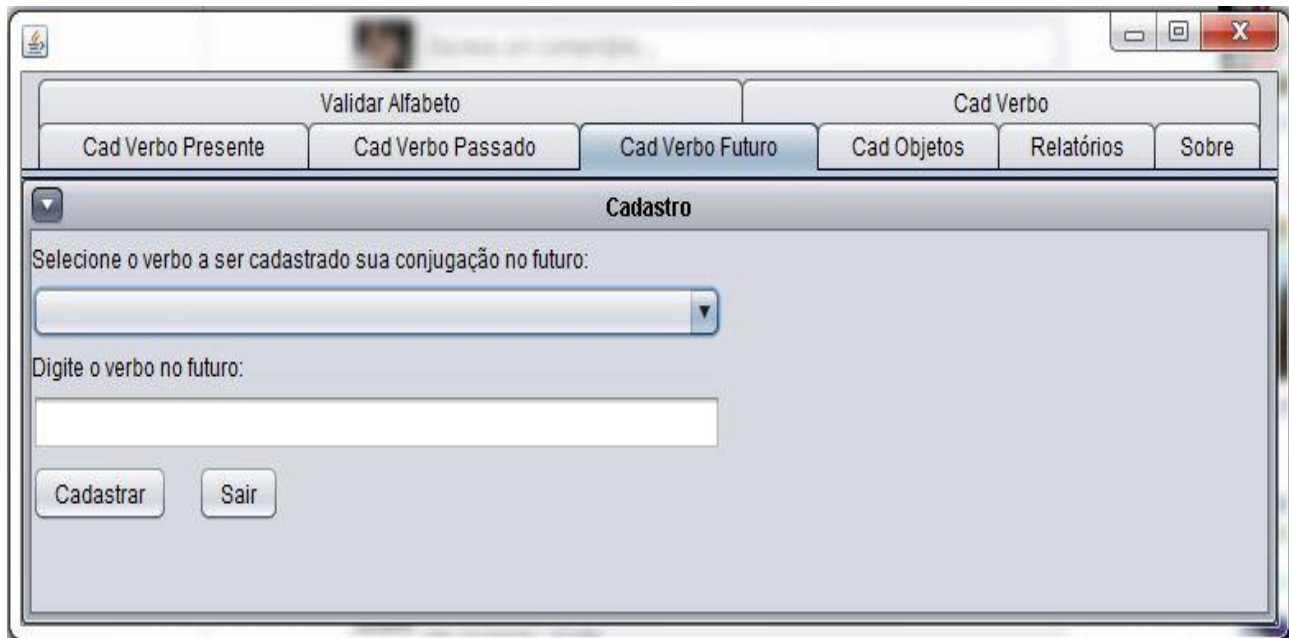


Figura 19 – Tela da Ferramenta para Cadastrar Verbo no Futuro.

Existe ainda a tela de cadastramento de objetos, onde podem ser armazenados os objetos que não foram encontradas dentro de uma frase ou texto. A figura 20 representa a tela de cadastramento de objetos.

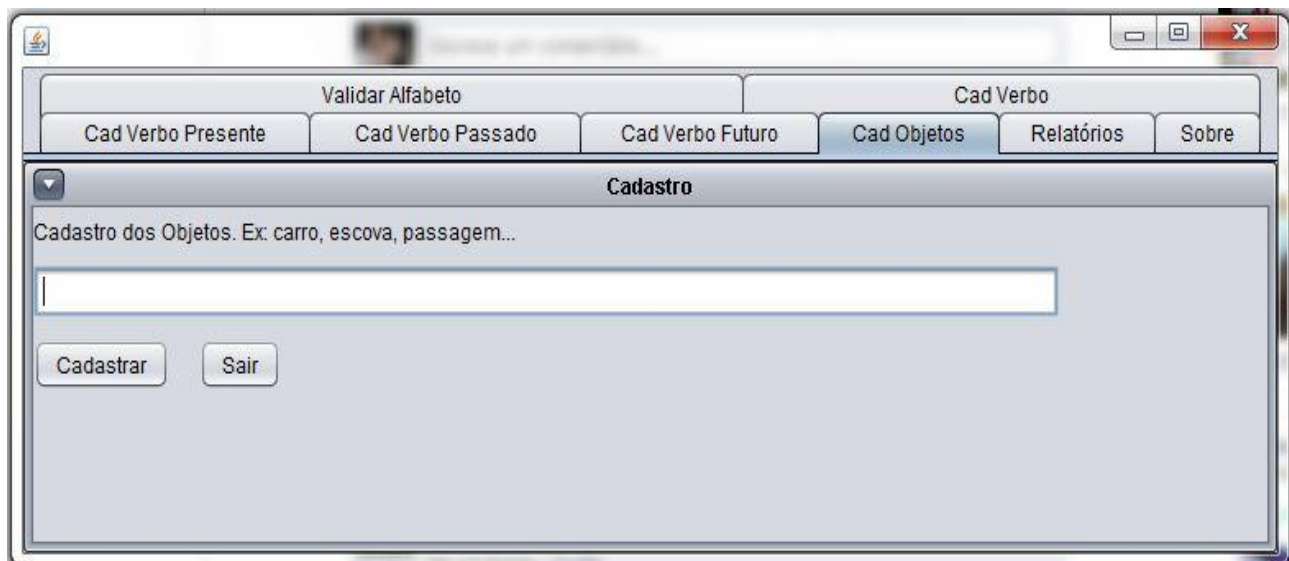


Figura 20 – Tela da Ferramenta para Cadastrar Objeto.

Essa tela possui a mesma estrutura da tela de cadastramento de verbo, que possui um campo texto para o usuário digitar o objeto desejado e localizado na frase.

A ferramenta apresenta também uma tela de relatório. Que emite a relação dos verbos, objetos e textos validados. A figura 21 representa a tela de relatórios.

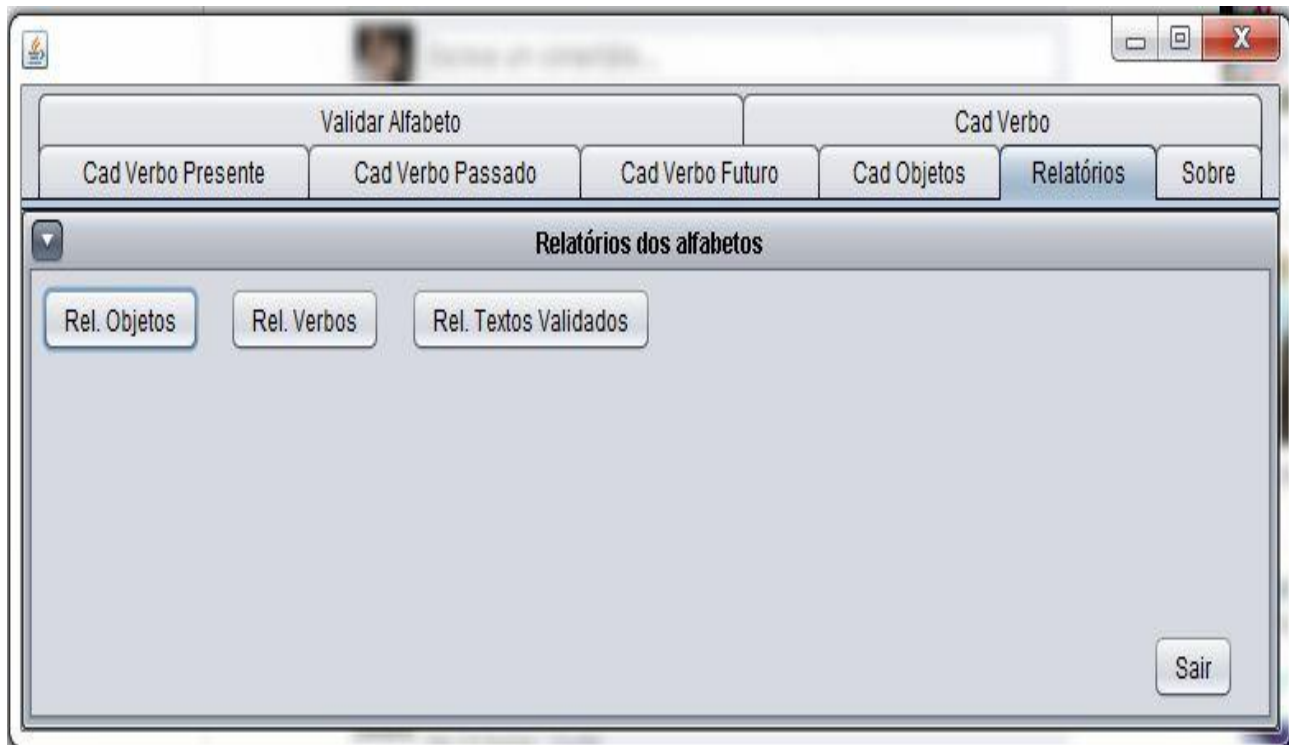


Figura 21 – Tela da Ferramenta para Emitir Relatórios.

Como mencionado no decorrer desse projeto, utilizamos da ferramenta *IReport* para emitir relatório onde pode ser observados os verbos, objetos e textos já inseridos na ferramenta. Conforme pode ser observado nas figuras 22, 23 e 24, os relatórios podem ser impressos ou salvos em arquivo (*.jrprint*, *.pdf*, *.doc*, *.odt*. entre outras extensões).



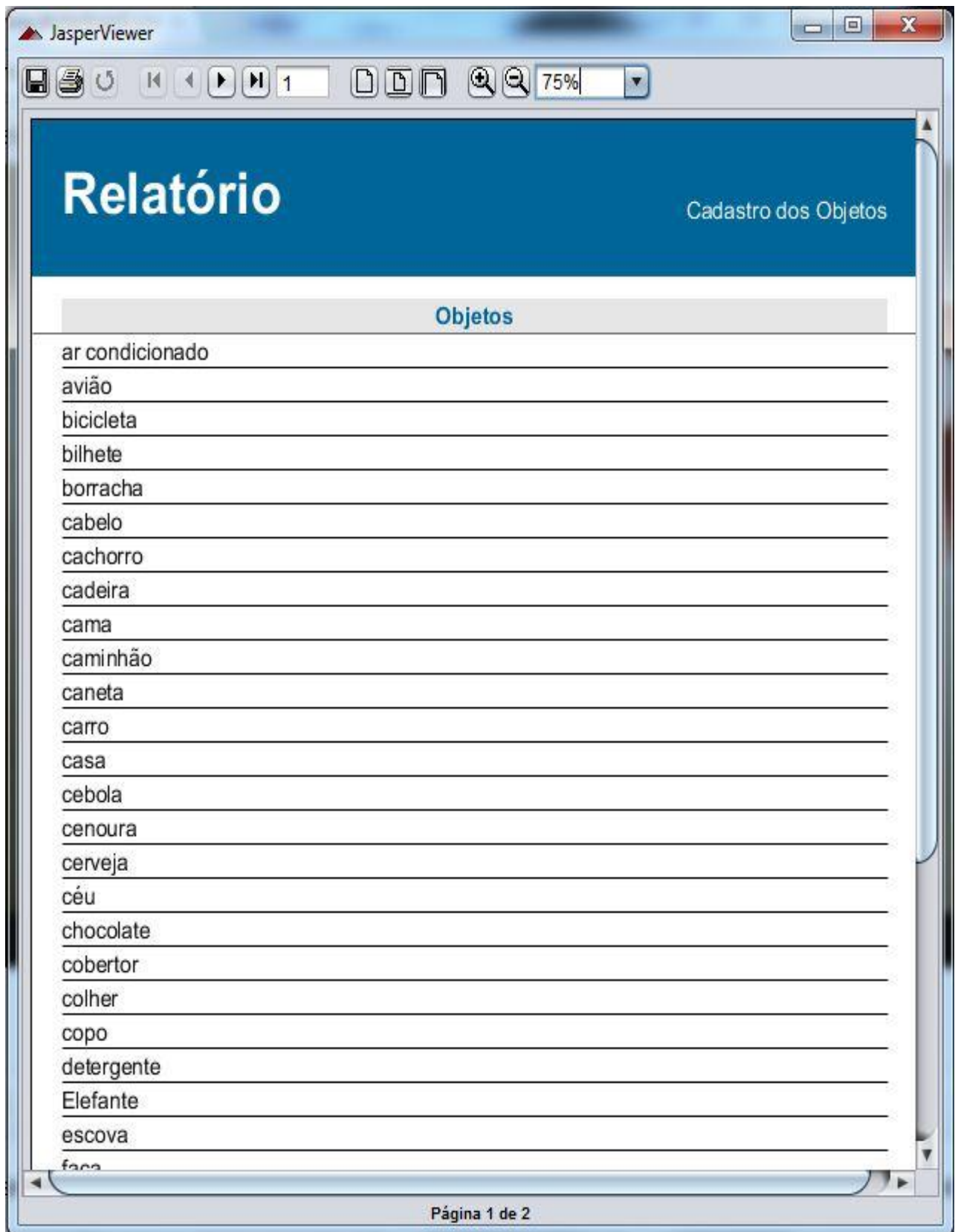


Figura 22 – Relatório dos Objetos que já foram Cadastrados.

The screenshot shows a JasperViewer window displaying a report. The report has a blue header with the title 'Relatório' and the subtitle 'Verbos Cadastrados'. The main content is a table listing verbs and their conjugations. The table is organized into groups for each verb, with each group containing four rows: the verb name, the future tense, the present tense, and the past tense. The verbs listed are: abrir, alterar, amar, andar, aprender, and bater. The past tense for 'alterar' is 'Alterei' and for 'amar' is 'Amei', both starting with a capital letter. The footer of the report indicates 'Página 1 de 5'.

Verbo	abrir
VerboFuturo	abrirei
VerboPresente	abri
VerboPassado	abri
Verbo	alterar
VerboFuturo	alterarei
VerboPresente	altere
VerboPassado	Alterei
Verbo	amar
VerboFuturo	amarei
VerboPresente	ame
VerboPassado	Amei
Verbo	andar
VerboFuturo	andarei
VerboPresente	andei
VerboPassado	andei
Verbo	aprender
VerboFuturo	aprenderei
VerboPresente	aprendi
VerboPassado	aprendi
Verbo	bater
VerboFuturo	baterei

Figura 23 – Relatório dos Verbos que já foram Cadastrados.

**Relatório** Alfabetos inseridos

---

**Verbo:** [Comprar]

Alfabeto comprar uma passagem  
Objeto [passagem]

Alfabeto fomos comprar uma passagem  
Objeto [passagem]

Alfabeto Comprarei um carro  
Objeto [carro]

Alfabeto fomos comprar uma passagem  
Objeto [passagem]

Alfabeto comprarei um carro  
Objeto [carro]

---

**Verbo:** [Correr]

Alfabeto correr unha  
Objeto [unha]

---

**Verbo:** [brigar]

Alfabeto brigar de faca  
Objeto [faca]

---

**Verbo:** [ler]

Alfabeto li um livro  
Objeto [livro]

---

**Verbo:** [brincar]

Alfabeto brinquei com faca

Página 1 de 3

Figura 24 – Relatório dos Alfabetos já Testados.

E por fim existe uma aba que traz os dados do desenvolvedor, a que organização ele pertence e a versão da ferramenta.

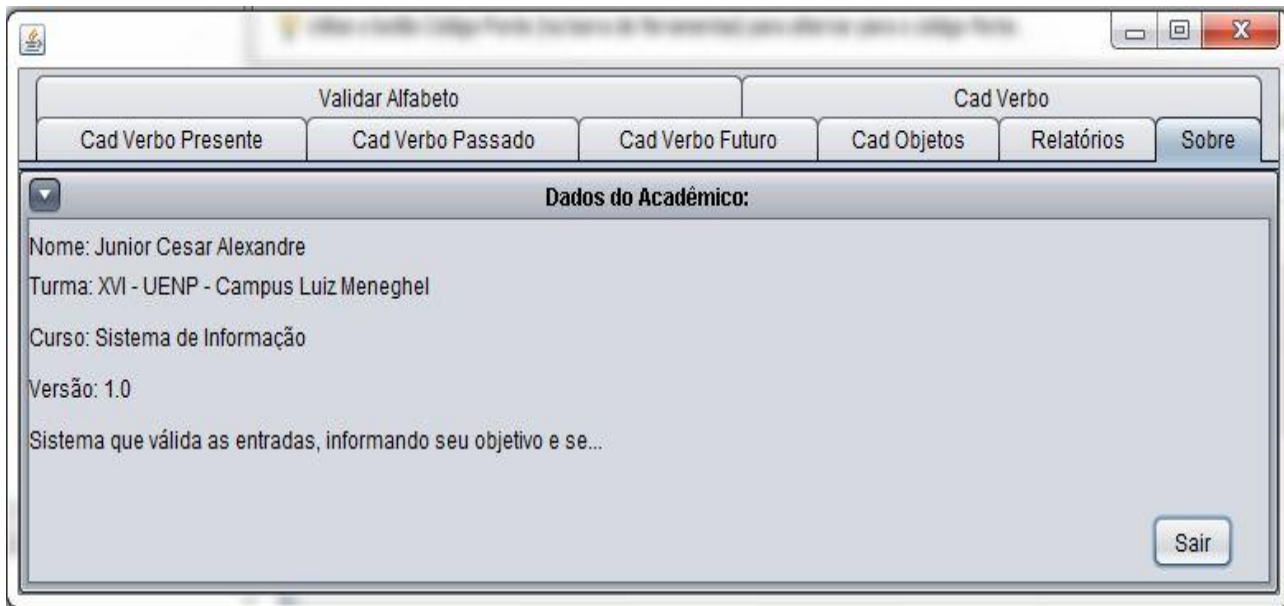


Figura 25 – Informações da Ferramenta.

### 3.4.1 - Limitações da Ferramenta.

Baseados em teste foram identificados algumas limitações da ferramenta proposta, embora não apresentem riscos eminentes para seu desempenho, é possível maximizar o seu funcionamento. A seguir segue as limitações encontradas.

- Possibilitar que a ferramenta aponte erros ortográficos ao ser inseridos um texto, tanto no momento da validação ou quando um verbo ou objeto for cadastrado em sua base.
- Assimilação de palavras tanto escritas no singular como no plural.
- Possibilitar que a ação de validação possa ser utilizada por outra ferramenta ou sistema.

### 3.4.2 Teste.

Aplicando os testes sobre a base de dados schema\_dbparser, utilizando o contexto viajar de avião.

Textos Inseridos:

1. Eu queria viajar de avião.
2. Eu já viajei de avião.
3. Eu irei viajar de avião.
4. Ele viajou de avião.
5. Ela queria de avião viajar.
6. De avião, ela queria viajar.
7. De avião eu viajo.
8. De avião, ele viajará.
9. De avião já viajei.
10. De avião viajarei.

Resultados Obtidos:

- ✓ Aplicado o teste sobre o texto “Eu queria viajar de avião” a ferramenta foi capaz de identificar exatamente a ação e o objetivo do usuário, obtendo um resultado satisfatório.
- ✓ Aplicado o teste sobre o texto “Eu já viajei de avião” a ferramenta foi capaz de identificar exatamente a ação e o objetivo do usuário, obtendo um resultado satisfatório.
- ✓ Aplicado o teste sobre o texto “Eu irei viajar de avião” a ferramenta foi capaz de identificar exatamente a ação e o objetivo do usuário, obtendo um resultado satisfatório.

- ✓ Aplicado o teste sobre o texto “Ele viajou de avião” a ferramenta não foi capaz de identificar a ação do usuário, mas identificou o seu objetivo.
- ✓ Aplicado o teste sobre o texto “Ela queria de avião viajar” a ferramenta foi capaz de identificar exatamente a ação e o objetivo do usuário, obtendo um resultado satisfatório.
- ✓ Aplicado o teste sobre o texto “De avião, ela queria viajar” a ferramenta foi capaz de identificar exatamente a ação e o objetivo do usuário, obtendo um resultado satisfatório.
- ✓ Aplicado o teste sobre o texto “De avião, ela viajou” a ferramenta foi capaz de identificar exatamente a ação e o objetivo do usuário, obtendo um resultado satisfatório.
- ✓ Aplicado o teste sobre o texto “De avião, eu viajo” a ferramenta não foi capaz de identificar a ação do usuário, mas identificou o seu objetivo.
- ✓ Aplicado o teste sobre o texto “De avião já viajei” a ferramenta foi capaz de identificar exatamente a ação e o objetivo do usuário, obtendo um resultado satisfatório.
- ✓ Aplicado o teste sobre o texto “De avião viajarei” a ferramenta foi capaz de identificar exatamente a ação e o objetivo do usuário, obtendo um resultado satisfatório.

O gráfico 1 apresenta o porcentual de acertos e erros dos testes aplicados no contexto viajar de avião.

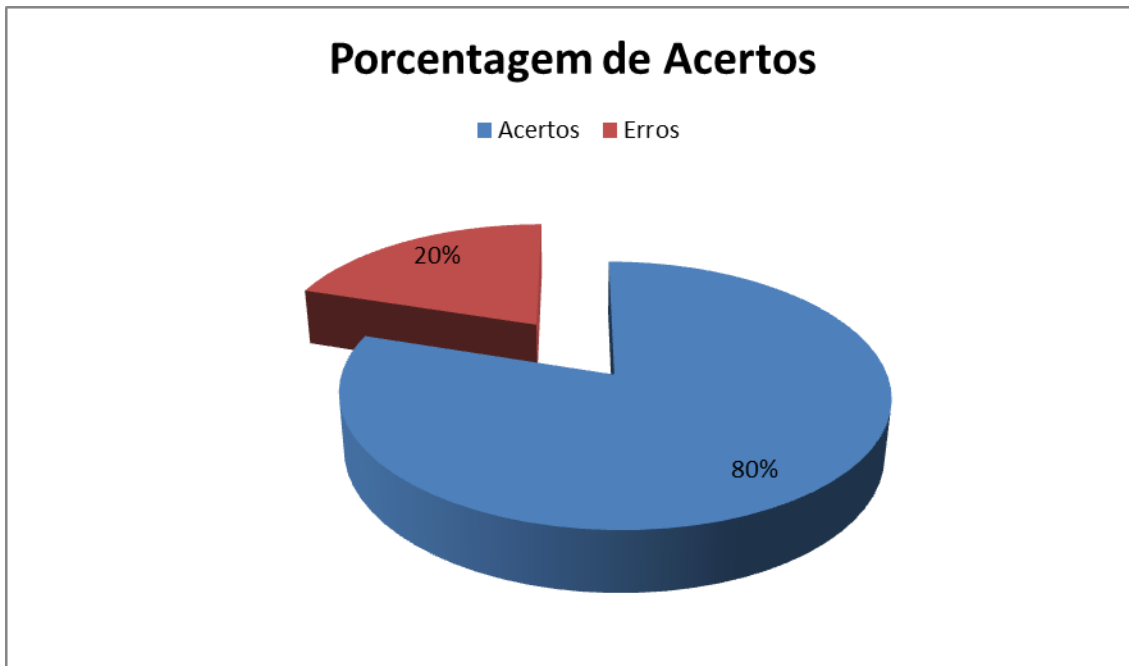


Gráfico 1 – Porcentagem de Acertos e Erros no Teste Feito

Pode ser observado que enquanto os testes feitos na busca de ações dentro dos textos nem sempre resulta em acertos, os objetivos sempre são encontrados, conforme apresenta o gráfico 2.

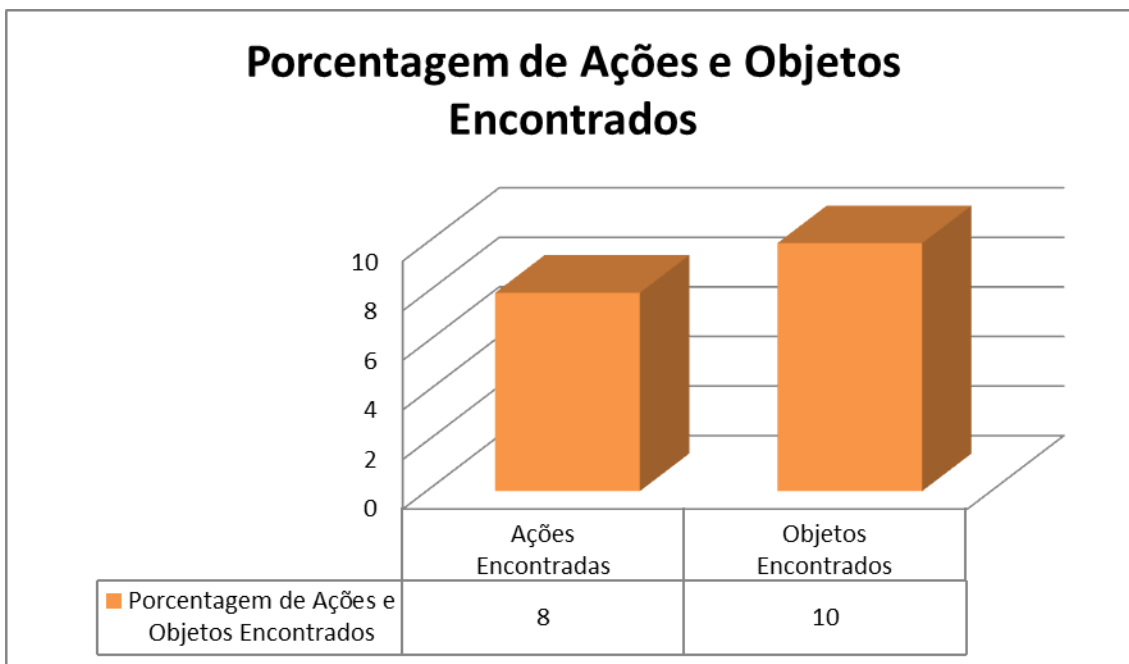


Gráfico 2 Porcentagem de Ações e Objetos Encontrados.

Utilizando o contexto viajar e mudando os objetos

Textos Inseridos:

11. Eu queria viajar de avião.
12. Eu queria viajar de carro.
13. Eu queria viajar de motocicleta.
14. Eu queria viajar de bicicleta.
15. Eu queria viajar de barco.
16. Eu queria viajar de ônibus.
17. Eu queria viajar de van.
18. Eu queria viajar de skate.
19. Eu queria viajar de iate.
20. Eu queria viajar de jatinho.

Resultados Obtidos:

- ✓ Aplicado o teste sobre o texto “Eu queria viajar de avião” a ferramenta foi capaz de identificar exatamente a ação e o objetivo do usuário, obtendo um resultado satisfatório.
- ✓ Aplicado o teste sobre o texto “Eu queria viajar de carro.” a ferramenta foi capaz de identificar exatamente a ação e o objetivo do usuário, obtendo um resultado satisfatório.
- ✓ Aplicado o teste sobre o texto “Eu queria viajar de motocicleta” a ferramenta foi capaz de identificar exatamente a ação mas não identificou o objeto.



- ✓ Aplicado o teste sobre o texto “Eu queria viajar de bicicleta” a ferramenta foi capaz de identificar exatamente a ação e o objetivo do usuário, obtendo um resultado satisfatório.
- ✓ Aplicado o teste sobre o texto “Eu queria viajar de barco” a ferramenta foi capaz de identificar exatamente a ação e o objetivo do usuário, obtendo um resultado satisfatório.
- ✓ Aplicado o teste sobre o texto “Eu queria viajar de ônibus” a ferramenta foi capaz de identificar exatamente a ação e o objetivo do usuário, obtendo um resultado satisfatório.
- ✓ Aplicado o teste sobre o texto “Eu queria viajar de van” a ferramenta foi capaz de identificar exatamente a ação e o objetivo do usuário, obtendo um resultado satisfatório.
- ✓ Aplicado o teste sobre o texto “Eu queria viajar de skate” a ferramenta foi capaz de identificar exatamente a ação e o objetivo do usuário, obtendo um resultado satisfatório.
- ✓ Aplicado o teste sobre o texto “Eu queria viajar de iate” a ferramenta foi capaz de identificar exatamente a ação mas não identificou o objeto.
- ✓ Aplicado o teste sobre o texto “Eu queria viajar de jatinho” a ferramenta foi capaz de identificar exatamente a ação mas não identificou o objeto.

O gráfico 3 apresenta o percentual de acertos ao encontrar as ações e os objetos no contexto viajar, mudando o objeto

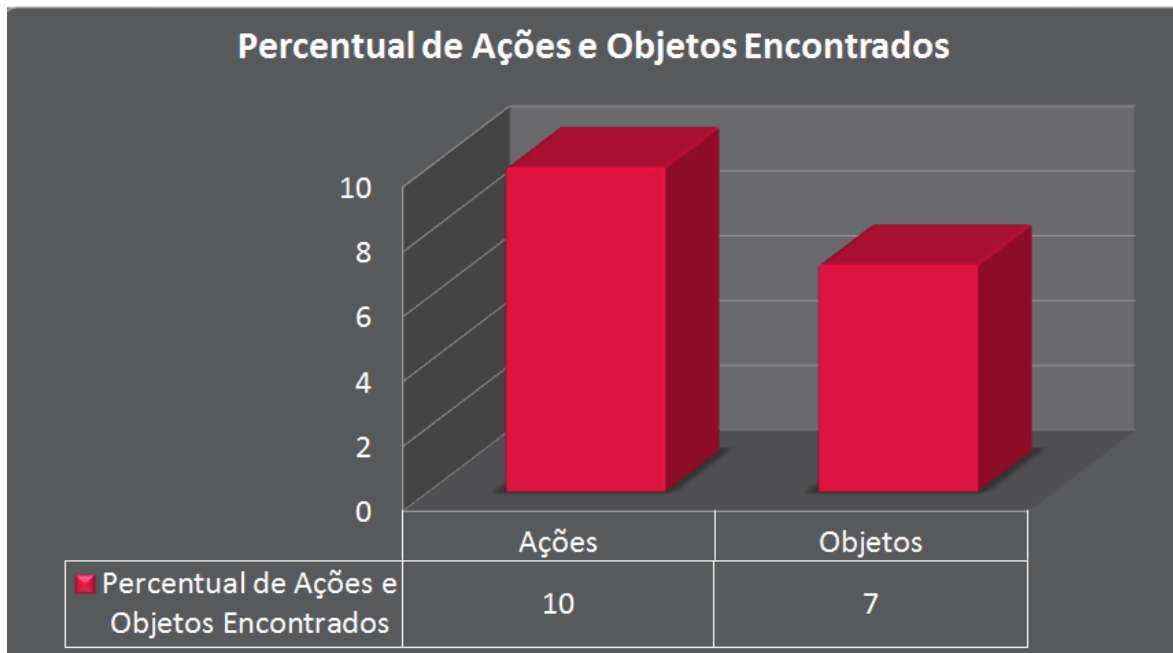


Gráfico 3 Percentual de Ações e Objetos Encontrados.

Com a realização desses testes foi possível observar que a ferramenta possui uma taxa de acertos significativa. A cada nova interação do usuário com a ferramenta e a cada novo dado armazenado dentro de sua base de dados, maior vai ser a taxa de acertos.

## 5 Considerações Finais

Diversas foram às dificuldades encontradas no decorrer do processo, desde os erros quanto à pesquisa teórica a mau funcionamento da ferramenta (erros estes que foram corrigidos). Mas também esse trabalho rendeu inúmeras contribuições como ampliação de conhecimentos, melhor maneira de escrita e maior sensibilidade quanto às exigências acadêmicas.

Com referência ao projeto é possível afirmar que seu objetivo principal, que seria a criação de uma ferramenta que fosse capaz de localizar a ação expressa pelo usuário dentro de um texto ou frase escrito pelo mesmo, baseado também nos testes realizados, pode-se dizer que foi concluído embora possua certas limitações.

Durante a execução deste trabalho, foram identificados alguns quesitos que podem ser melhorados em trabalhos futuros, como:

- Integrar uma dicionário a sua base de dados capaz de informar ao usuário que a sua escrita possui erros ortográficos e aconselha-los a corrigir esses erros antes de executar a validação de um texto.
- Criar um mecanismo que reconheça uma palavra no plural sendo que a mesma já está no banco de dados no singular.
- Criar uma interface mais simples que melhor possa atender as necessidades do usuário
- Criar um mecanismo que informe ao usuário qual são as conjugações que determinado verbo está incompleta.
- Permitir que a ação de validação possa ser integrada a outra ferramenta que necessite dessa função.

O código fonte da ferramenta pode ser encontrado pelo link <<http://www.4shared.com/zip/el6QegD1/validaalfabeto.html>>.

## REFERÊNCIAS

FILIFE, Porfírio Pena; **Integração Dinâmica de Artefatos em Sistemas de Diálogo** (MARÇO 2007).

SILVA, GLAUCO CARLOS; **Um método semiautomático para o Aprendizado do Modelo Conceitual para Diálogos em Português do Brasil** (JUNHO 2012).

MOURÃO, MÁRCIO DUARTE ALBASINI; **Gestão e Representação de Domínio em Sistemas de Diálogo** (SETEMBRO 2005).

FERREIRA, JOÃO A AND FILIFE, PORFÍRIO PENA; **Integração Dinâmica de Artefatos em Sistemas de Diálogos.**

ROMAN, Norton Trevison; **Estudo de Diálogo Orientado à Tarefa usando a Teoria de Multiagentes** (agosto 2001).

JAIN, Prateek and MITAL, Manav Ratan and KUMAR Sumit and MUKERJEE, Amitabha and RAINA Achla M.; **Anaphora Resolution in Multi-Person Dialogues.**

PAULINO, Estela dos Santos and SOUZA, Thaís Barbosa; **Software De Apoio Ao Aprendizado De Trigonometria - OPENSAAT: Modelagem e Prototipação.**

**MySQL.** In: WIKIPÉDIA, a enciclopédia livre. [S.I.]: Wikipédia Foundation, 2006. Disponível em: <<http://pt.wikipedia.org/wiki/MySQL>> Acesso em 18 de Setembro de 2012 as 21:00.

Lima, Bruno R. **Tutorial de IREPORT.** Disponível em: <<http://javafree.uol.com.br/artigo/3154/Tutorial-de-IREPORT.html>>. Acesso em 19 de Setembro de 2012 as 16:00.

Java Practices – **Parser text**, Disponível em: <[www.javapractices.com/topicaction.do?id=87](http://www.javapractices.com/topicaction.do?id=87)>  
Acesso desde: 18/09/2012.

IReport – Disponível em:  
<[suporte.softwell.com.br/marker/mnual/pt/dicas\\_e\\_truques/utilizando\\_oireport.html](http://suporte.softwell.com.br/marker/mnual/pt/dicas_e_truques/utilizando_oireport.html)> Acesso  
em 28/10/2012.

Mysql – Disponível em: <[dev.mysql.com/downloads](http://dev.mysql.com/downloads)> Acesso desde: 18/09/2012.

Netbeans – Disponível em : <[netbeans.org/download](http://netbeans.org/download)> Acesso desde 18/09/2012.

IReport – Disponível em:<[community.jaspersoft.com/wiki/ireport-desingner-v470-download](http://community.jaspersoft.com/wiki/ireport-desingner-v470-download)>  
Acesso em 19/09/2012.