

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIAS DA COMPUTAÇÃO
(Bacharelado)

**FERRAMENTA DE APOIO A IDENTIFICAÇÃO DE EVENTOS
UTILIZANDO LINGUAGEM NATURAL**

TRABALHO DE CONCLUSÃO DE CURSO SUBMETIDO À UNIVERSIDADE
REGIONAL DE BLUMENAU PARA A OBTENÇÃO DOS CRÉDITOS NA
DISCIPLINA COM NOME EQUIVALENTE NO CURSO DE CIÊNCIAS DA
COMPUTAÇÃO — BACHARELADO

RICARDO TOMELIN

BLUMENAU, JUNHO/2001

2001/1-61

FERRAMENTA DE APOIO A IDENTIFICAÇÃO DE EVENTOS UTILIZANDO LINGUAGEM NATURAL

RICARDO TOMELIN

ESTE TRABALHO DE CONCLUSÃO DE CURSO, FOI JULGADO ADEQUADO
PARA OBTENÇÃO DOS CRÉDITOS NA DISCIPLINA DE TRABALHO DE
CONCLUSÃO DE CURSO OBRIGATÓRIA PARA OBTENÇÃO DO TÍTULO DE:

BACHAREL EM CIÊNCIAS DA COMPUTAÇÃO

Prof. Everaldo Artur Grahl — Orientador na FURB

Prof. José Roque Voltolini da Silva — Coordenador do TCC

BANCA EXAMINADORA

Prof. Everaldo Artur Grahl

Prof. Marcel Hugo

Prof. Wilson Pedro Carli

AGRADECIMENTOS

Agradeço aos meus pais, que não pouparam esforços na busca de uma educação melhor para seus filhos, apoiando, incentivando e acreditando que o conhecimento é uma das chaves para o sucesso.

Meus sinceros agradecimentos ao meu professor orientador, Everaldo Artur Grahl, pela dedicação, incentivo, apoio e principalmente amizade durante a elaboração deste trabalho.

Agradeço aos meus colegas de classe, pelo apoio incondicional, horas de estudo e amizade durante a realização deste curso.

Obrigado a todos!

SUMÁRIO

AGRADECIMENTOS	III
LISTA DE FIGURAS	VI
LISTA DE QUADROS	VII
RESUMO	VIII
ABSTRACT	IX
1 INTRODUÇÃO	1
1.1 ORIGEM.....	1
1.2 OBJETIVO	3
1.3 ORGANIZAÇÃO DO TEXTO	3
2 ANÁLISE ESSENCIAL DE SISTEMAS	4
2.1 HISTÓRICO	4
2.2 MODELO ESSENCIAL.....	6
2.2.1 MODELO AMBIENTAL	6
2.2.2 MODELO COMPORTAMENTAL.....	7
2.3 MODELO DE IMPLEMENTAÇÃO.....	8
2.4 VANTAGENS DA ANÁLISE ESSENCIAL SOBRE A ANÁLISE ESTRUTURADA	9
2.5 EVENTOS	10
2.5.1 ANÁLISE DE EVENTOS	10
2.6 LISTA DE EVENTOS.....	12
2.6.1 EVENTO ORIENTADO POR FLUXO DE DADOS.....	14
2.6.2 EVENTO ORIENTADO POR TEMPO	15
2.6.3 EVENTO ORIENTADO POR CONTROLE.....	16
2.7 FERRAMENTAS CASE DE SUPORTE A ANÁLISE ESSENCIAL DE SISTEMAS	16
3 LINGUAGEM NATURAL	19
3.1 ASPECTOS IMPORTANTES.....	20
3.2 INTERPRETAÇÃO DA LINGUAGEM NATURAL.....	21
3.3 ESTRUTURA DA FRASE.....	22
3.4 GRAMÁTICAS	22
3.4.1 GRAMÁTICA SENSÍVEL AO CONTEXTO.....	23
3.4.2 GRAMÁTICA LIVRE DE CONTEXTO	23

3.4.3	GRAMÁTICAS REGULARES.....	24
3.5	CARACTERÍSTICAS DO FORMALISMO GRAMATICAL	24
3.6	AMBIGÜIDADE SINTÁTICA.....	24
3.7	ANÁLISE SINTÁTICA DE FRASES.....	25
3.8	CONSTRUÇÃO DE UM ANALISADOR SINTÁTICO.....	29
3.9	TRABALHOS CORRELATOS QUE UTILIZAM LINGUAGEM NATURAL	30
4	DESENVOLVIMENTO DA FERRAMENTA.....	33
4.1	ESPECIFICAÇÃO.....	33
4.2	IMPLEMENTAÇÃO.....	39
4.2.1	TÉCNICAS E FERRAMENTAS UTILIZADAS	39
4.3	FUNCIONAMENTO DA FERRAMENTA	45
4.4	MONTAGEM DOS EVENTOS	51
4.5	RESULTADO E DISCUSSÃO	51
5	CONCLUSÃO.....	55
5.1	SUGESTÕES.....	55
	ANEXO 1: AJUDA DA FERRAMENTA.....	57
	ANEXO 2: ESTUDO DE CASO	65
	REFERÊNCIAS BIBLIOGRÁFICAS.....	68

LISTA DE FIGURAS

1: ABORDAGENS DA ANÁLISE ESSENCIAL	6
2: RELAÇÃO ENTRE ANÁLISE ESSENCIAL E ANÁLISE ESTRUTURADA	10
3: DEMONSTRATIVO DE EVENTO, ESTÍMULO E RESPOSTA	11
4: EXEMPLO DE EVENTO, ESTÍMULO E RESPOSTA	12
5: LISTA DE EVENTOS NO <i>SYSTEM ARCHITECT</i>	17
6: LISTA DE EVENTOS NO CASE WG-AE	17
7: TELA DE DIGITAÇÃO DE EVENTOS NO CASE WG-AE.....	18
8: EVENTO E DFD NA FERRAMENTA DE APOIO A ANÁLISE ESSENCIAL	18
9: EXEMPLO DE ÁRVORE SINTÁTICA	28
10: DIÁLOGO COM O SISTEMA ELIZA	31
11: COMPONENTES DO SISTEMA.....	32
12: DIAGRAMA DE FUNCIONAMENTO.....	33
13: DIAGRAMA DE CONTEXTO	35
14: DFD GERADO PARA VERIFICAR A ORTOGRAFIA.....	35
15: DFD PARA CADASTRAR PALAVRAS	36
16: DFD PARA GERAR A LISTA DE EVENTOS.....	36
17: MER LÓGICO.....	37
18: MER FÍSICO	37
19: PRIMEIRA PARTE DO FLUXOGRAMA	38
20: SEGUNDA PARTE DO FLUXOGRAMA	39
21: ESTRUTURA DA TABELA DE DICIONÁRIO	40
22: ESTRUTURA DA TABELA DE VERBOS	40
23: ESTRUTURA DA TABELA DE SUBSTANTIVOS	41
24: TELA PRINCIPAL DA FERRAMENTA	45
25: TELA DE MANUTENÇÃO DO DICIONÁRIO	46
26: CADASTRAMENTO DE UMA PALAVRA	46
27: CADASTRAMENTO DE UM VERBO	47
28: CADASTRAMENTO DE UM SUBSTANTIVO	47
29: FRASES EXISTENTES NO ESTUDO DE CASO.....	48
30: SELEÇÃO DE FRASES QUE SERÃO UTILIZADAS.....	49
31: EVENTOS IDENTIFICADOS NO ESTUDO DE CASO.....	50
32: EVENTOS SIMILARES	50

LISTA DE QUADROS

1: EVOLUÇÃO HISTÓRICA DA ANÁLISE DE SISTEMAS	4
2: LISTA DE EVENTOS TEXTUAL	13
3: LISTA DE EVENTOS NA FORMA DE TABELA	14
4: ANÁLISE SINTÁTICA COM O MÉTODO <i>TOP-DOWN</i>	28
5: UTILIZAÇÃO DO MÉTODO <i>BOTTOM-UP</i>	29
6: EVENTOS IDENTIFICADOS	34
7: ALGORITMO DO ANALISADOR SINTÁTICO	41
8: PARTE INICIAL DO ANALISADOR SINTÁTICO	42
9: TRECHO DO CÓDIGO DO ANALISADOR SINTÁTICO	43
10: VERIFICAÇÃO ORTOGRÁFICA NO <i>MICROSOFT WORD</i>	44
11: DIRETÓRIO TEMPORÁRIO DO SISTEMA OPERACIONAL	44
12: ESTUDO DE CASO DO CLUBE ESPORTIVO	45
13: TESTE COM O ESTUDO DE CASO DO CLUBE ESPORTIVO	52
14: ESTUDO DE CASO DO CONSULTÓRIO MÉDICO	52
15: TESTE COM O ESTUDO DE CASO DO CONSULTÓRIO MÉDICO	52
16: ESTUDO DE CASO DO ALUNO	53
17: TESTE COM O ESTUDO DE CASO DO ALUNO	53
18: ESTUDO DE CASO DA INSTALADORA ELÉTRICA	53
19: TESTE COM O ESTUDO DE CASO DA INSTALADORA	54
20: PERCENTUAL DE ACERTO DOS TESTES	54

RESUMO

A análise essencial de sistemas foi desenvolvida na década de oitenta e está sendo cada vez mais utilizada na especificação de sistemas. Seu ponto principal é a definição da lista de eventos. Vários autores definem a lista de eventos como a essência de um sistema. Partindo deste ponto de vista optou-se em facilitar esta identificação de eventos. Neste trabalho tem se como objetivo principal a construção de um software de apoio à identificação destes eventos, onde foram utilizadas técnicas de processamento de linguagem natural. A partir de um estudo de caso informado à ferramenta, faz-se a identificação automática dos eventos.

ABSTRACT

The essential system analysis was developed in the eighties and it's being more and more used in the system specification. Its main issue is the definition of the events rol. Several authors define the events rol as the essence of a system. Starting at this concept, the option was trying to turn the events identification easier. In this study has as its main goal the creation of a support software for identifying these events, where was used techniques from the natural language processing. From one case, once the tool is informed, the identification of events shall occur by itself.

1 INTRODUÇÃO

1.1 ORIGEM

No mercado existem várias abordagens para efetuar a análise de sistemas, entre elas encontra-se a análise essencial de sistemas. A análise essencial é uma abordagem para o desenvolvimento de sistemas que apresenta um grau de abstração independente de restrições tecnológicas.

“A análise essencial propõe um particionamento do sistema em eventos, assim um sistema é construído para responder a estímulos. A cada estímulo o sistema deve reagir produzindo uma resposta pré-determinada. Um evento pode ser definido informalmente como um acontecimento do mundo exterior e que requer do sistema uma resposta. Um estímulo é o ativador de uma função; é a forma como um evento age sobre o sistema, e é a consequência do fato de ter ocorrido um evento externo. Um evento externo é um acontecimento independente que ocorre fora do sistema e provoca um estímulo que faz com que uma função seja executada dentro do sistema. Uma resposta é um resultado gerado pelo sistema devido à ocorrência de um evento” (Pompilho, 1994).

Utilizando-se a análise essencial, a principal preocupação é com a lista de eventos. Localizar os eventos não é fácil e exige bastante atenção do analista uma vez que existem eventos implícitos e eventos explícitos. Eventos explícitos são identificados claramente na leitura de um texto, já os eventos implícitos são mais difíceis de serem localizados, podendo este estar implícito em outro evento.

“A maneira de descrever os eventos requer algumas considerações. Todo evento deve ser descrito por única sentença, geralmente uma estrutura frasal que atende as necessidades de descrição dos eventos” (Pompilho, 1994).

Considerando-se a dificuldade de se localizar e identificar eventos, através de métodos convencionais como a leitura de textos, optou-se em utilizar uma das técnicas da Inteligência Artificial como apoio a este processo. Entre várias alternativas existentes optou-se pelo processamento de linguagem natural.

Um evento definido de forma incorreta pode futuramente provocar um erro de especificação, onde se pode gastar muito tempo para corrigir o problema. Foi por esse motivo que se optou por desenvolver um protótipo de ferramenta para auxiliar o analista de sistemas na definição da lista de eventos.

Este trabalho não pretende ser um passo final para a especificação de um sistema. Em trabalhos futuros pode-se utilizar a lista de eventos identificada neste trabalho para gerar os próximos passos da especificação de um sistema, como diagrama de contexto e diagrama de fluxo de dados (DFD). Um trabalho nesta linha foi desenvolvido por Pereira (1992), onde o usuário informa a lista de eventos e a ferramenta cria o DFD para cada evento.

“A linguagem natural é uma das formas mais humanas de manifestação externa da atividade mental. Na comunicação homem-máquina em linguagem natural nosso problema será implementar alguma forma de expressão de interpretação automática da expressão da pessoa” (Savadovsky, 1989).

Para Rich (1988), “A compreensão da linguagem natural é difícil. Ela requer tanto conhecimento lingüístico de uma língua em particular que estiver sendo utilizada quanto conhecimento do mundo relacionado ao tópico que estiver sendo discutido”.

O mesmo Rich (1988) define compreensão como “compreender uma coisa é transformá-la de uma representação em outra que tenha sido escolhida para corresponder a um conjunto de ações disponíveis que poderiam ser realizadas; e onde tenha sido designado mapeamento, de modo que, para cada evento, uma ação apropriada seja realizada”.

Várias outras aplicações já foram desenvolvidas utilizando-se o processamento de linguagem natural (PLN). Pode-se observar alguma destas como o processo desenvolvido por Manfro (2000) para gerar um modelo de dados e a ferramenta desenvolvida por Gomez (1999) que serve para fazer a geração semi-automática de um modelo Entidade Relacionamento.

1.2 OBJETIVO

O objetivo principal do trabalho foi desenvolver uma ferramenta para auxiliar o analista de sistemas na definição da lista de eventos, utilizando para isto o processamento de linguagem natural.

Os objetivos específicos do trabalho são:

- a) analisar as potencialidades da linguagem natural na especificação de sistemas;
- b) especificar e implementar um protótipo de ferramenta que dê suporte a definição dos eventos.

1.3 ORGANIZAÇÃO DO TEXTO

O trabalho é composto por cinco capítulos, descritos a seguir.

O primeiro capítulo apresenta uma introdução, incluindo origem, objetivos e organização do trabalho.

O segundo capítulo apresenta a técnica de Análise Essencial, eventos e algumas ferramentas que dão suporte à Análise Essencial.

O terceiro capítulo apresenta a técnica de processamento de linguagem natural, incluindo alguns tipos de gramática, análise sintática de frases, desenvolvimento de um *parser* e alguns sistemas já desenvolvidos que utilizam linguagem natural.

O quarto capítulo apresenta a especificação e implementação do software de apoio à identificação de eventos

O quinto capítulo apresenta as conclusões e sugestões deste trabalho.

2 ANÁLISE ESSENCIAL DE SISTEMAS

2.1 HISTÓRICO

A Análise Essencial de Sistemas foi proposta em 1984 por McMenamin e Palmer com o livro *Análise Essencial de Sistemas*. Também é conhecida como Análise por Eventos ou Análise Estruturada Moderna, assim denominada por Yourdon (1990).

Em 1987 os conceitos de Análise Essencial foram estendidos com a publicação do livro *Developing Systems With No Pain* (desenvolvimento de sistemas sem complicação) por Paul Ward.

Segundo Barbieri (1994), “a idéia dessa técnica que originou alterações metodológicas na Análise Estruturada clássica é o particionamento por eventos. Basicamente em vez da análise funcional clássica por decomposição de funções, o trabalho sugere a garimpagem de eventos e objetos”.

Para Pompilho (1994), “A técnica de Análise Essencial pode ser encarada como uma bem-sucedida evolução da Análise Estruturada”.

A evolução histórica da Análise de Sistemas pode ser observada no quadro 1.

QUADRO 1: EVOLUÇÃO HISTÓRICA DA ANÁLISE DE SISTEMAS

Modelo	Período	Abordagem	Ferramentas
Convencional ou Tradicional	~55 até ~75	Funcional	Textos Fluxogramas
Estruturado Chris Gane / 1979 Yourdon / 1979	A partir de 75	Funcional	Diagrama de Fluxo de Dados (DFD) Especificação dos processos
		Dados	Diagrama de Estrutura de dados (DED) Normalização Dicionário de dados
Essencial McMenamin, Stephen M. & Palmer, John F.	A partir de 84	Funcional	Diagrama de Contexto (DFD) Tabela de Eventos DFD por eventos Diagrama de Estrutura
		Dados	Diagrama Entidade Relacionamento Normalização
		Dinâmica	Diagrama de Transição de Estado (DTE)

Fonte: Grahl (2001)

O objetivo da Análise Essencial de Sistemas é identificar a essência do sistema, através da definição da lista de eventos. “Os eventos constituem o cerne de um sistema, a sua razão de existir” (Pompilho, 1994).

Segundo McMenamim (1984) “A essência de um sistema é um conjunto intangível de idéias. Para se ter um sistema que execute atividades essenciais, a essência tem que ser encarnada em objetivos físicos. O termo encarnação é utilizado para representar a materialização de um conceito”.

O mesmo McMenamim (1984) define essência como “todas as características de um sistema de respostas planejadas que existiriam se o sistema fosse implementado independente da tecnologia que fosse utilizada”.

Utilizando Análise Essencial, o primeiro passo para a especificação de um sistema é a definição da lista de eventos. Os eventos são a pedra fundamental do sistema.

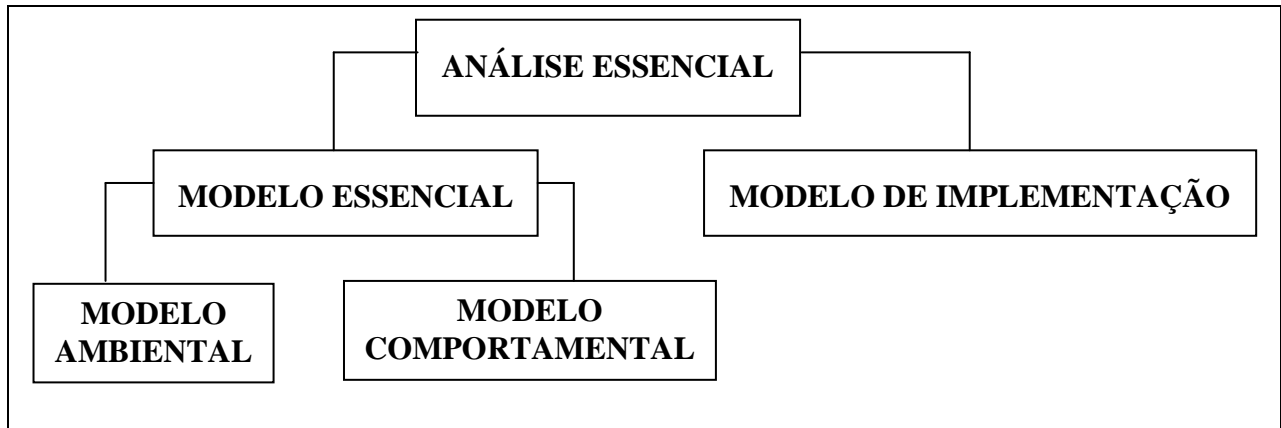
Ao definir a lista de eventos, deve-se ter cuidado. Todos os eventos definidos serão utilizados na especificação e implementação do sistema. Um evento definido de forma equivocada torna as especificações incorretas e difíceis de serem compreendidas, além disso, pode tornar difícil a manutenção do sistema.

Segundo Pompilho (1994), “um evento pode ser definido informalmente como um acontecimento do mundo exterior que requer do sistema uma resposta”.

A Análise Essencial aborda três perspectivas do problema (funções, dados e controle), e possui dois graus de abstração: modelo essencial e modelo de implementação. O modelo essencial é composto de dois componentes: modelo ambiental e modelo comportamental, como pode ser visto na figura 1.

A figura 1 mostra a subdivisão da Análise Essencial e suas abordagens.

FIGURA 1: ABORDAGENS DA ANÁLISE ESSENCIAL



Fonte: Pompilho (1994)

2.2 MODELO ESSENCIAL

O modelo essencial apresenta um grau de abstração independente de restrições tecnológicas. Os autores da análise estruturada denominam de modelo lógico proposto.

Para Pompilho (1994), “é necessário conhecer a verdadeira essência do sistema não importando saber de sua implementação nem mesmo que tipo de hardware ou software será utilizado”.

“O modelo essencial indica o que o sistema deve fazer, mencionando o mínimo possível (de preferência nada) sobre como o sistema será implementado” (Yourdon, 1990).

O modelo essencial é composto por dois componentes: modelo ambiental e modelo comportamental.

2.2.1 MODELO AMBIENTAL

O modelo ambiental define a fronteira entre sistema e o resto do mundo, também é importante para definir as interfaces entre o sistema e o ambiente. É composto por três componentes: declaração de objetivos, diagrama de contexto e lista de eventos.

A declaração dos objetivos é uma declaração breve e concisa do sistema, e é voltada para quem não está diretamente envolvido no desenvolvimento do sistema. Pode ter uma ou mais sentenças. Entretanto o ideal seria um único parágrafo, pois não se destina a dar uma descrição detalhada do sistema.

O próximo passo é a criação do diagrama de contexto, onde algumas das perguntas levantadas na declaração dos objetivos começam a ser respondidas.

Para Yourdon (1990), o diagrama de contexto realça diversas características importantes do sistema:

- a) as pessoas, organizações ou sistemas com que o sistema irá se comunicar;
- b) os dados que o sistema recebe do mundo exterior e que devem ser processados;
- c) os dados produzidos pelo sistema e enviados para o mundo exterior;
- d) os limites entre o sistema e o resto do mundo.

O terceiro passo é a lista de eventos, é uma lista narrativa dos “estímulos” que ocorrem no mundo exterior e que o sistema deve responder. A lista de eventos será vista nos itens subseqüentes.

2.2.2 MODELO COMPORTAMENTAL

O modelo comportamental define o comportamento das partes internas do sistema necessárias para interagir com o ambiente. O modelo comportamental é o modelo interno do sistema. “Preocupa-se em mostrar quais as ações que o sistema deve executar para responder adequadamente aos eventos previstos no modelo ambiental, que é seu ponto de partida” (Pompilho, 1994).

O modelo comportamental possui três componentes:

- a) DFD's particionados por evento;
- b) DFD preliminar;
- c) DFD de nível zero.

DFD's particionados por evento: também é conhecido como DFD para cada evento ou DFD de resposta aos eventos. Para cada evento deve-se representar os processos, entidades externas, depósitos de dados, respostas e fluxos. Este processo deve ser iniciado somente quando a lista de eventos estiver completa.

DFD preliminar: é obtido integrando-se os DFD's para cada evento em um DFD único.

DFD de nível zero: é também conhecido como diagrama de contexto, é o primeiro DFD, onde é representado um único processo com o nome do sistema, entidades externas, fluxos e respostas.

O modelo comportamental possui dois métodos envolvidos: modelagem de dados e modelagem funcional.

A modelagem funcional é utilizada para o detalhamento do processo cujo aspecto principal é desenvolver um modelo de como funcionam os processos do sistema.

A modelagem de dados é utilizada para identificar quais são os dados utilizados pelo sistema e como são manipulados por ele, e em consequência como são utilizados.

O resultado do modelo essencial (modelo ambiental e modelo comportamental), consiste nas atividades e dados essenciais do sistema e no modelo dinâmico que este deve ter para interagir adequadamente com o ambiente externo.

2.3 MODELO DE IMPLEMENTAÇÃO

O modelo de implementação é derivado do modelo essencial e apresenta um grau de abstração completamente dependente das restrições tecnológicas. Em Análise Estruturada é conhecido como modelo físico.

Pompilho (1994) cita que “Neste modelo são colocadas todas as características tecnológicas, importando saber se parte de sua implementação será manual ou automatizada, que tipo de hardware ou software vai ser usado”.

Segundo Belmiro (1993), “O modelo de implementação deve conter uma descrição completa do que o sistema deve fazer para satisfazer o usuário”.

A Análise Essencial não recomenda a elaboração do modelo físico atual, utilizado na Análise Estruturada, somente em casos específicos com o objetivo de auxiliar o entendimento do sistema a ser desenvolvido.

“A proposta da Análise Essencial acelera sobremaneira a velocidade da especificação de um sistema – uma das queixas da Análise Estruturada é quanto ao tempo consumido na fase de especificação” (Pompilho, 1994).

Segundo Yourdon (1990), o modelo de implementação abrange quatro aspectos:

- a) a alocação do modelo essencial a pessoas versus máquinas;
- b) detalhes da interação homem/máquina;
- c) suplementares atividades manuais que podem vir a ser necessárias;
- d) restrições operacionais que o usuário deseja impor ao sistema.

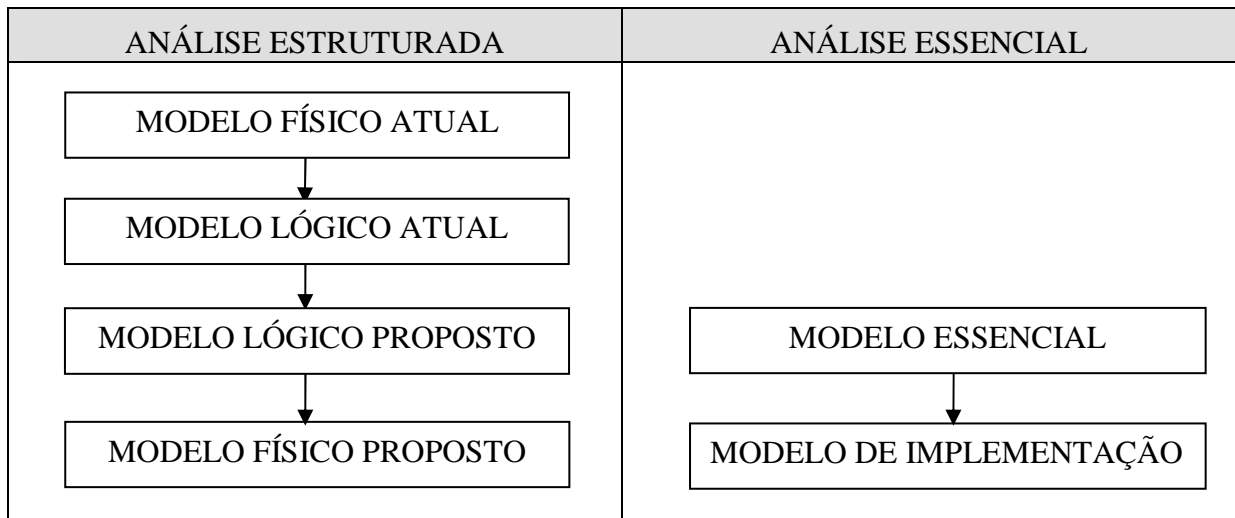
2.4 VANTAGENS DA ANÁLISE ESSENCIAL SOBRE A ANÁLISE ESTRUTURADA

Pode-se observar as seguintes vantagens da Análise Essencial em relação à Análise Estruturada, identificadas por Pompilho (1994):

- a) a Análise Essencial começa pelo modelo essencial, o que equivale na Análise Estruturada a começar diretamente pelo modelo lógico proposto. Portanto já economiza duas etapas de modelagem na Análise Estruturada;
- b) a Análise Estruturada aborda duas perspectivas do sistema, função e dados, ao passo que a Análise Essencial aborda três perspectivas – função, dados e controle, sendo, portanto mais completa;
- c) na Análise Estruturada, o particionamento é feito através da abordagem *top-down*, enquanto a Análise Essencial propõe outra forma de particionamento a qual é baseada nos eventos e tem demonstrado ser muito mais efetiva que a abordagem *top-down*, pois torna mais fácil a identificação das funções e entidades que compõe o sistema;
- d) a Análise Essencial permite a construção dos modelos de dados e de funções concomitantemente, o que garante a correspondência entre os dois modelos.

Uma relação entre os modelos elaborados pela técnica de Análise Essencial e pela Análise Estruturada pode ser observada na figura 2.

FIGURA 2: RELAÇÃO ENTRE ANÁLISE ESSENCIAL E ANÁLISE ESTRUTURADA



Fonte: Pompilho (1994)

2.5 EVENTOS

Segundo Pompilho (1994), “A análise essencial propõe um particionamento do sistema em eventos, assim um sistema é construído para responder a estímulos. A cada estímulo o sistema deve reagir produzindo uma resposta pré-determinada. Um estímulo é o ativador de uma função, é a forma como um evento age sobre o sistema. É a consequência do fato de ter ocorrido um evento externo. Um evento externo é um acontecimento independente que ocorre fora do sistema e provoca um estímulo que faz com que uma função seja executada dentro do sistema”.

2.5.1 ANÁLISE DE EVENTOS

Um sistema pode ser entendido como uma caixa preta, que a partir de certos estímulos, produz as respostas apropriadas.

Não se deve confundir um evento com um estímulo por ele provocado, também não se deve confundir um evento com uma função a ser ativada pela sua ocorrência. Para evitar este tipo de confusão, o observador deve-se colocar na posição de quem está fora do sistema olhando para ele (Pompilho, 1994).

Segundo McMenamim (1984), “o primeiro problema é que os analistas, algumas vezes acham que uma atividade dentro do sistema é um evento”.

Evento é um acontecimento do mundo exterior, que requer uma resposta do sistema. Já estímulo é a consequência da ocorrência do evento, é o que chega ao sistema e ativa a execução de uma função.

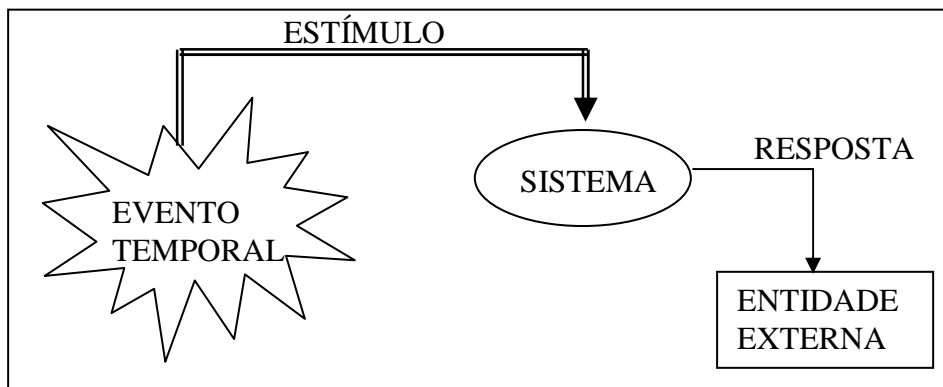
Pompilho (1994) define uma resposta a um evento como “um resultado gerado pelo sistema devido à ocorrência de um evento. Uma resposta é sempre o resultado da execução de alguma função interna do sistema como consequência do reconhecimento pelo sistema de que o evento ocorreu”.

Segundo Pompilho (1994), uma resposta pode ser decorrente de:

- a) um fluxo de dados saindo do sistema para uma entidade externa;
- b) uma mudança de estado em algum depósito de dados (o que equivale a uma inclusão, exclusão ou modificação de algum registro de um arquivo);
- c) um fluxo de dados saindo de uma função para ativar outra.

A figura 3 mostra um evento que faz com que o sistema apresente uma resposta, sem que nenhuma entidade externa envie um fluxo de dados ao sistema. Este exemplo foi desenvolvido por Pompilho (1994).

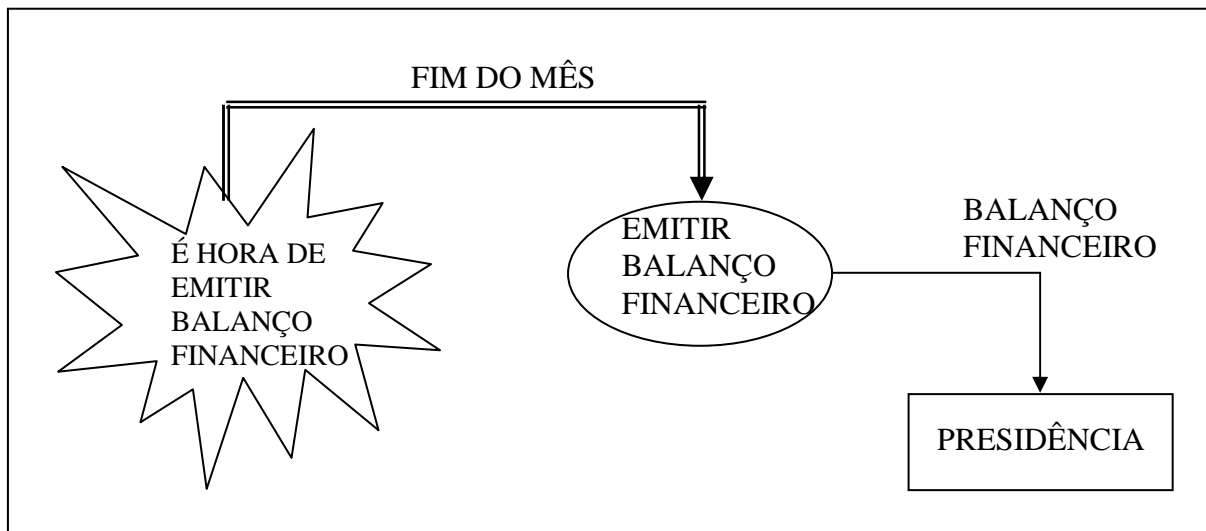
FIGURA 3: DEMONSTRATIVO DE EVENTO, ESTÍMULO E RESPOSTA



Fonte: adaptado de Pompilho (1994)

Um exemplo pode ser visto com o evento: é hora de emitir o balanço financeiro. Este exemplo foi desenvolvido por Pompilho (1994), e pode ser observado na figura 4.

FIGURA 4: EXEMPLO DE EVENTO, ESTÍMULO E RESPOSTA



Fonte: adaptado de Pompilho (1994)

2.6 LISTA DE EVENTOS

Para Belmiro (1993), “a lista de eventos é uma lista narrativa dos estímulos que ocorrem no mundo exterior, e aos quais o sistema deve responder. O principal objetivo é derivar a matéria-prima para a construção do modelo essencial, consistindo nos eventos a que o sistema responde”.

Segundo Bellin (1993), “a lista inicial de eventos deve ser tão completa quanto possível, comece a relacionar tudo o que você e o usuário puderem identificar”.

As finalidades de um sistema são atender a determinadas necessidades, que são decorrentes de eventos que ocorrem no mundo exterior ao sistema. A construção da lista de eventos está intrinsecamente ligada às finalidades de um sistema.

Ao se definir a lista de eventos, existem eventos explícitos e eventos implícitos. Eventos explícitos são identificados claramente na leitura de um texto, já os eventos implícitos são mais difíceis de serem localizados, podendo este estar implícito em outro evento.

Para Shiller (1992), há três maneiras de obter eventos:

- a) do nada: quando se constrói uma lista de eventos do nada, é difícil separar eventos de suas respostas, uma solução aceitável é listar os dois. O refinamento pode vir mais tarde;
- b) colhendo-os de um documento específico do usuário: este documento é de formato livre e funciona como a primeira versão do usuário para a especificação de sistema. Este documento conterá muitos eventos. Mas devido à sua natureza não estruturada, o analista deve ler nas entrelinhas para descobrir os eventos implícitos;
- c) colhendo-os do sistema atual: se um sistema atual existe, é possível derivar eventos se as atividades forem bem definidas. Esses eventos derivados podem então ser examinados para uma possível inclusão na lista de eventos oficial.

A lista de eventos pode seguir dois formatos: lista textual ou tabela. Para ambos os formatos, os eventos são classificados da seguinte maneira:

- a) (F) – evento orientado por fluxo de dados;
- b) (T) – evento orientado por tempo (temporal);
- c) (C) – evento orientado por controle.

No formato de lista enumerada os eventos são descritos um abaixo do outro e numerados sequencialmente como pode ser observado no quadro 2.

QUADRO 2: LISTA DE EVENTOS TEXTUAL

- | |
|--|
| <ul style="list-style-type: none"> 1) cliente entrega pedido (F); 2) cliente cancela pedido (F); 3) cliente envia pagamento (F); 4) fornecedor solicita cadastramento (F); 5) é hora de emitir relatório de vendas (T); 6) é hora de verificar pedidos em atraso (T); 7) direção autoriza pagamento de fornecedor (C); 8) segundo cheque sem fundo é emitido (C); 9) nível de ressurgimento é atingido (C). |
|--|

Fonte: Pompilho (1994)

Outra forma de se apresentar a lista de eventos é na forma de uma tabela. As colunas da tabela são as seguintes:

- a) número do evento;
- b) nome do evento;
- c) tipo de evento;

- d) estímulo;
- e) ação;
- f) resposta.

A mesma lista de eventos identificada anteriormente, é apresentada no quadro 3 no formato de tabela, como foi escrita por Pompilho (1994).

QUADRO 3: LISTA DE EVENTOS NA FORMA DE TABELA

Nº DO EVENTO	NOME EVENTO	TIPO DE EVENTO	ESTÍMULO	AÇÕES	RESPOSTAS
(1)	Cliente entrega pedido	(F)	Pedido	Registrar pedido	(Pedido registrado)
(2)	Cliente cancela pedido	(F)	Pedido de cancelamento	Cancelar pedido	(Pedido cancelado)
(3)	Cliente envia pagamento	(F)	Cheque de pagamento	Emitir recibo de pagamento	Recibo de pagamento (fatura paga)
(4)	Fornecedor solicita cadastramento	(F)	Pedido de cadastramento de fornecedor	Cadastrar fornecedor	(Fornecedor cadastrado)
(5)	É hora de emitir relatório de vendas	(T)	(a hora de emitir relatório de vendas)	Emitir relatório de vendas	Relatório de vendas
(6)	É hora de verificar pedidos em atraso	(T)	(a hora de verificar pedidos em atraso)	Verificar pedidos em atraso	Pedidos em atraso
(7)	Direção autoriza pagamento de fornecedor	(C)	Autorização de pagamento	Gerar pagamento	Pagamento de fornecedor (duplicata paga)
(8)	2º cheque sem fundo é emitido	(C)	(Informação de que é o 2º cheque sem fundo)	Registrar cheque no Serviço de Proteção ao Crédito	Relatório para o Serviço de Proteção ao Crédito - SPC (cheque no SPC)
(9)	Nível de ressuprimento é atingido	(C)	(Informação de que foi atingido o nível de ressuprimento)	Emitir encomenda ao fornecedor	Encomenda de material ao fornecedor (encomendada)

Fonte: adaptado de Pompilho (1994)

2.6.1 EVENTO ORIENTADO POR FLUXO DE DADOS

Os eventos orientados por fluxo de dados são eventos em que o estímulo é a chegada ao sistema de um fluxo de dados enviado por uma entidade externa.

Segundo Pompilho (1994), “todo evento orientado por fluxo de dados é reconhecido por um fluxo de dados que chega ao sistema, mas nem todo fluxo de dados que chega ao sistema serve de estímulo relativo a um evento. Pode ser apenas uma informação complementar para a execução de alguma função do sistema”.

Para Pompilho (1994) a notação de evento orientado por fluxo de dados fica assim:

Sujeito + verbo transitivo (voz ativa) + complemento verbal.

Exemplo: O cliente coloca o pedido (F).

O sujeito é o “cliente”, o verbo transitivo é “colocar” e o complemento verbal é “pedido”.

Bellin (1993) define este evento como evento “não temporal”. O padrão utilizado é:

Interveniente + ação transitiva + objetivo.

Exemplo: usuário digita pedido (F).

O interveniente é o “usuário”, a ação transitiva é “digita” e o objetivo é “pedido”.

Barbieri (1994) definiu o seguinte padrão para os eventos de fluxo:

Entidade externa + verbo + objeto.

Um exemplo para este padrão pode ser: usuário solicita empréstimo de livro (F).

2.6.2 EVENTO ORIENTADO POR TEMPO

Os eventos orientados por tempo, conhecidos como eventos temporais, são eventos em que o estímulo não é a chegada de um fluxo de dados ou de um fluxo de controle, mas a informação de ter decorrido um determinado intervalo de tempo.

Segundo Pompilho (1994), “pode haver fluxos de dados complementares associados ao evento temporal. Para um evento temporal existir, o sistema deve ser responsável por registrar a passagem do tempo. Um evento temporal ocorre pelo complemento da passagem de algum intervalo de tempo”.

Um bom formato para um evento temporal é:

É hora de + verbo no infinitivo + complemento verbal.

Exemplo: é hora de emitir relatório (T).

O verbo no infinitivo é “emitir” e o complemento verbal é “relatório”

Bellin (1993) define este tipo de evento como evento “condicional”. O padrão utilizado é: “Momento de” + resposta do sistema.

Exemplo: Momento de tocar o sino (T).

2.6.3 EVENTO ORIENTADO POR CONTROLE

Os eventos de controle são eventos que o estímulo é a chegada de um fluxo de controle no sistema, enviado por uma entidade externa ou por uma função interna do sistema. Não é através da chegada de um fluxo de dados que o sistema toma conhecimento da ocorrência de um evento.

“O sistema pode ou não produzir uma resposta externa a um evento orientado por controle” Pompilho (1994).

Para Pompilho (1994), “um evento orientado por controle é aquele em que uma função só é ativada se uma determinada variável binária (fluxo de controle) está com um determinado valor”.

O formato para este evento pode ser:

- a) sujeito + verbo transitivo na voz ativa + complementos verbais. Exemplo: diretoria autoriza o pagamento da fatura (C);
- b) sujeito + verbo na voz passiva. Exemplo: oitavo cheque é emitido (C).

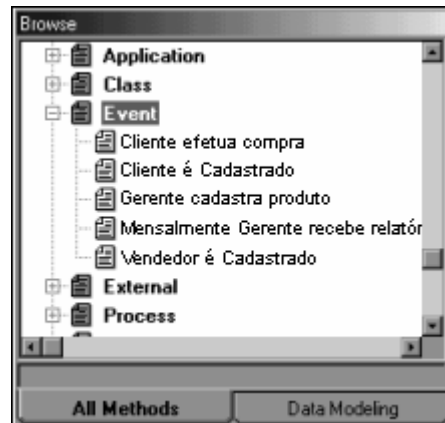
No exemplo a, o sujeito é “diretoria”, o verbo transitivo na voz ativa é “autoriza” e os complementos verbais são “pagamento da fatura”.

2.7 FERRAMENTAS CASE DE SUPORTE A ANÁLISE ESSENCIAL DE SISTEMAS

No mercado existem algumas ferramentas que dão suporte à Análise Essencial. Uma dessas é o *System Architect* desenvolvido para o sistema operacional *Windows* e que dá suporte às principais técnicas da Análise Essencial, como lista de eventos, diagrama de contexto, diagrama entidade relacionamento e diagrama de fluxo de dados.

A figura 5 mostra a lista de eventos no *System Architect*

FIGURA 5: LISTA DE EVENTOS NO SYSTEM ARCHITECT



Outra ferramenta CASE (*Computer Aided Software Engineering*) é o WG-AE desenvolvida para o sistema operacional MS-DOS pelo IBPI (Instituto Brasileiro de Pesquisa em Software). Esta é uma ferramenta CASE voltada somente para a Análise Essencial. Uma versão educacional desta ferramenta acompanha o livro de Pompilho (1994). Por se tratar de uma versão educacional, seu funcionamento é limitado permitindo apenas sete eventos e sete entidades externas por projeto. O WG-AE oferece suporte a várias fases da Análise Essencial, como: diagrama de fluxo de dados, dicionário de dados, diagrama de contexto, modelo comportamental e lista de eventos.

A figura 6 mostra a tela da lista de eventos no WG-AE.

FIGURA 6: LISTA DE EVENTOS NO CASE WG-AE

NRO.	DESCRIÇÃO SUCINTA DO EVENTO	TIPO	ESTÍMULO	
1	Cliente entrega pedido	Fluxo	Pedido	Ped
2	Cliente Cancela Pedido	Fluxo	Pedido de Canc...	Ped

EDE Editor de Eventos W6

Na figura 7 pode-se observar a tela de digitação dos eventos no WG-AE.

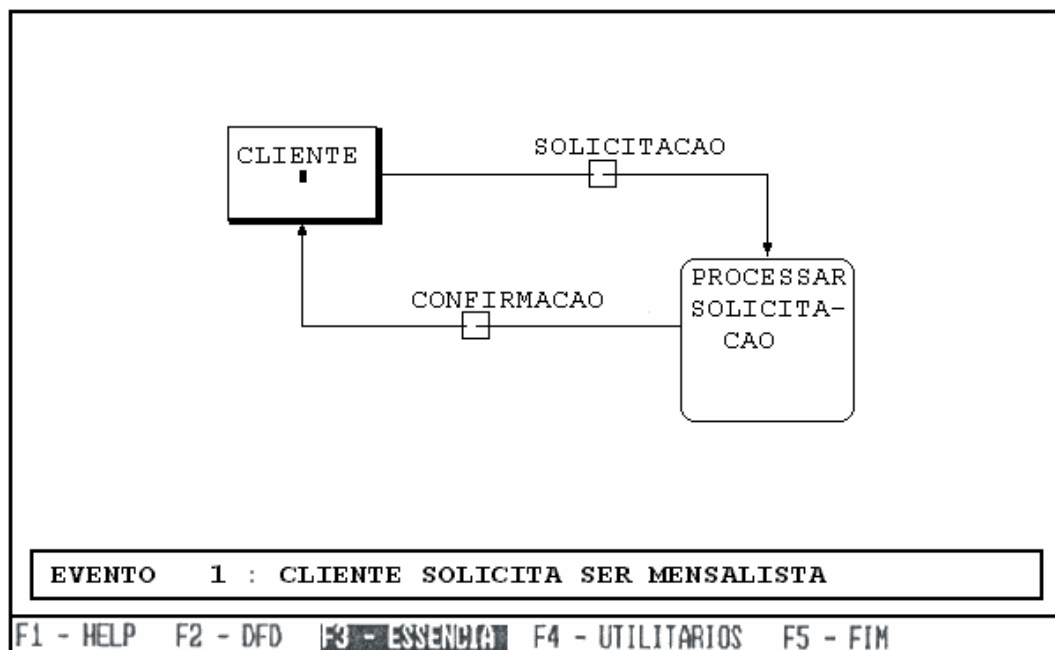
FIGURA 7: TELA DE DIGITAÇÃO DE EVENTOS NO CASE WG-AE

Atributos do Evento	
Evento	: Cliente Entrega Pedido
Tipo	: <input type="text" value="FLUXO"/>
Estímulo	: <input type="text" value="Pedido"/>
Resposta	: <input type="text" value="Pedido Entregue"/>
Saída	: <input type="text" value=""/>
Origem	: <input type="text" value=""/>
Destino	: <input type="text" value=""/>
Tipo de Origem	: <input type="text" value="ENTIDADE"/> Tipo de Destino: <input type="text" value="ENTIDADE"/>
Continua Inserindo	: <input type="text" value="SIM"/>
<input type="button" value="Ok"/>	

Outra ferramenta que apóia a Análise Essencial foi desenvolvida num trabalho de conclusão de Curso de Ciências da Computação por Pereira (1992). Esta ferramenta possui um editor de diagrama de fluxo de dados (DFD) e um editor de Análise Essencial.

A figura 8 mostra um evento e o DFD para este evento na ferramenta desenvolvida por Pereira (1992).

FIGURA 8: EVENTO E DFD NA FERRAMENTA DE APOIO A ANÁLISE ESSENCIAL



Fonte: adaptado de Pereira (1992)

3 LINGUAGEM NATURAL

A maneira mais comum das pessoas se comunicarem no dia-a-dia é através da linguagem. Segundo Allen (1995), “a linguagem é um dos aspectos fundamentais do comportamento humano e um componente muito importante para a vida do ser humano. Da forma escrita, serve como registro em longo prazo para a próxima geração. Da forma falada, serve como nossos meios primários de coordenar nosso comportamento no dia-a-dia para com as outras pessoas”.

Para Rabuske (1995), “o entendimento da linguagem natural é difícil, pois requer conhecimentos de lingüística e do domínio do discurso. Estes conhecimentos, em boa dose, o ser humano os adquire naturalmente, desde tenra idade. À medida que a pessoa se desenvolve intelectualmente, acumula mais e mais conhecimentos e experiências que vão enriquecendo a possibilidade de comunicação”.

O mesmo Rabuske (1995) define o objetivo básico da linguagem natural como “possibilitar que as pessoas se comuniquem, e isto só é plenamente atingido se elas se entenderem”.

O processo de comunicação, embora seja relativamente fácil, envolve o uso de várias regras gramaticais pertencentes à língua em uso.

As aplicações da linguagem natural podem ser divididas em dois grupos (Allen, 1995): baseadas em texto e baseadas em diálogo.

As aplicações baseadas em texto envolvem o processo de texto escrito, como livros, jornais, relatórios, manuais, textos digitados e assim por diante. Um exemplo deste tipo de aplicação pode ser: pesquisar livros existentes em uma biblioteca.

As aplicações baseadas em diálogo envolvem a comunicação homem-máquina, na maioria das vezes através do idioma falado, mas também pode existir interação via teclado. Um exemplo deste tipo de aplicação pode ser: o atendimento automatizado ao consumidor via telefone.

Segundo Hübner (1992), o processamento de linguagem natural (PLN) é uma área de estudos da Inteligência Artificial (IA) que tem por objetivo dotar as interfaces de

computadores da capacidade de comunicar-se com seu usuário na língua deste. Para atingir este fim existem as seguintes sub-áreas de estudo no PLN:

- a) conversão de fala em texto;
- b) compreensão do texto (Compreensão de Linguagem Natural – CLN);
- c) geração de fala (conversão de texto em voz);
- d) tradução de textos;
- e) correção de textos.

3.1 ASPECTOS IMPORTANTES

Para ser possível representar uma linguagem natural, é muito importante definir-se um padrão comum de comunicação que no caso da língua é a própria gramática, que segue um padrão definido e não contempla características particulares de um dialeto específico (Allen, 1995).

Segundo Neves (1998), “vale a pena destacar uma série de fatores que interferem, e até mesmo prejudicam, o processo de comunicação relacionados com a fisiologia do corpo humano. Pode-se analisá-las de duas formas distintas: do ponto de vista fisiológico e frente as diferentes áreas de estudo relativas à linguagem natural. Esta análise é importante para que se tenha uma visão da complexidade e da relevância da área em estudo no âmbito da computação”.

A linguagem natural possui diferentes áreas de estudo. Segundo Allen (1995), cada área é composta pelo seu próprio conjunto de problemas e possui seus próprios métodos para identificá-los.

Outro aspecto referente à linguagem natural é a necessidade de uma base de conhecimento para o entendimento da sentença. Esta base de conhecimento é adquirida durante toda uma vivência.

Segundo Hübner (1992), a interação entre usuário e um sistema, utilizando linguagem natural, torna-se muito mais simples ao usuário comum, pois o mesmo irá interagir com o sistema utilizando a linguagem da forma que está habituado a comunicar-se. Porém, se durante a interação o usuário tiver que escrever toda uma frase, o uso desta interface tornar-se-á enfadonho. Um recurso para solucionar este problema é o uso de uma interface de voz.

Para Allen (1995), o entendimento da fala por um computador é muito mais difícil do que a escrita, por englobar, além dos recursos de comunicação escrita, também formas de representar o som considerando todas as suas variáveis (ruídos, ambigüidades no sinal de áudio, etc).

3.2 INTERPRETAÇÃO DA LINGUAGEM NATURAL

Segundo Rabuske (1995), “a compreensão de frases em linguagem natural sempre recebeu maior atenção que sua geração. Entender uma frase e gerar uma frase são aparentemente processos iguais e há quem confunda. Mas as duas tarefas têm diferenças fundamentais. A compreensão da linguagem é vista especialmente como uma tarefa de decomposição e análise, enquanto que a geração é vista como uma tarefa de projeto e construção”.

Segundo Savadovsky (1989), “para nós, interpretar linguagem natural será transformar sua forma para a de outra linguagem, adequada para ser usada pela máquina, mantendo constante a interação original da pessoa”.

O mesmo Savadovsky (1989) apresenta o seguinte exemplo:

“Sílvia gosta de Fernando”

Pode-se dizer que a frase foi interpretada se internamente for gerada uma estrutura S tal que se possa mostrar que S estabelece uma relação ordenada R entre os dois nomes, X e Y, sendo que X denota o indivíduo Sílvia, e Y denota o indivíduo Fernando e R denota o verbo gostar (Savadovsky, 1989).

“A dificuldade de interpretar as frases está na forma da linguagem interna da máquina que escolhermos. Como outras linguagens, ela pode ser mais ou menos expressiva, e a sua escolha deverá ser um dos pontos mais importantes da implementação do interpretador.” (Savadovsky, 1989).

Para Schildt (1989), “quando implementarmos um processador de linguagem natural, somos tentados a restringir os tipos de sentenças que o processador entenderá a um subconjunto da linguagem natural. Se você restringir a gramática que um processador pode

aceitar, sua tarefa tornar-se-á muito mais simples, e, se feita corretamente a restrição será quase imperceptível”.

3.3 ESTRUTURA DA FRASE

A palavra é a unidade básica da linguagem. Uma frase é composta por uma cadeia de palavras.

Uma definição de frase é dada por Silva (1986) “costuma-se entender por frase a expressão verbal de um pensamento, ou seja, todo enunciado suficiente por si mesmo para estabelecer comunicação. Por meio dela pode-se expressar juízos, descrever ações, estados ou fenômenos transmitir apelos, ordens ou emoções”.

Segundo Cunha (1972) as frases podem ser formadas:

- a) de uma só palavra. Exemplo: Chove. Atenção! Silêncio!;
- b) de várias palavras, entre as quais inclui ou não um verbo. Exemplo: Preste atenção! Noite linda!.

Para Savadovsky (1989), a sintaxe da linguagem é o estudo das regras que determinam quais cadeias de palavras podem formar frases. Cadeias sintaticamente corretas são fortes candidatas a frases. Uma frase é a unidade da linguagem que comunica um pensamento ou intenção de uma pessoa. O conjunto de regras que determinam as cadeias sintaticamente válidas é a gramática da linguagem.

O mesmo Savadovsky (1989) define informalmente gramática como “um conjunto de regras que agrupam palavras em porções bem definidas de frase, que recebem nomes próprios e as agrupam entre si para formar frases”.

Exemplos sobre análise de frases e árvore sintática de frases serão abordados nos itens subseqüentes.

3.4 GRAMÁTICAS

A gramática pode ser definida como um conjunto limitado de regras a partir da qual pode-se gerar um número infinito de frases.

Para Savadovsky (1989), “a investigação das diferentes classes de gramáticas e seu poder de gerar linguagens usando métodos matemáticos precisos foi muito avançada pelo trabalho de N. Chomsky, um dos mais importantes pesquisadores da lingüística”.

É atribuída a Noam Chomsky uma hierarquia gramatical que pode ser vista a seguir.

3.4.1 GRAMÁTICA SENSÍVEL AO CONTEXTO

A gramática sensível ao contexto é também denominada como gramática do tipo 1. Neste tipo de gramática são geradas linguagens em que a sentença pode ser reconhecida por uma máquina computacional determinística (Rabuske, 1995). Uma dessas máquinas é conhecida como “Máquina de Turing”, desenvolvida em 1936 por Alan Turing.

Segundo Menezes (1998), “a Máquina de Turing é aceita como uma formalização de um procedimento efetivo (algoritmo ou função computável), ou seja, uma seqüência finita de instruções, as quais podem ser realizadas mecanicamente em tempo finito”.

Para Rabuske (1995), nas regras de produção deste tipo de gramática, o lado direito consiste na reprodução do lado esquerdo, expandindo apenas um único símbolo.

3.4.2 GRAMÁTICA LIVRE DE CONTEXTO

A gramática livre de contexto também é denominada como gramática do tipo 2. A característica principal deste tipo de gramática é que possui do lado esquerdo de suas regras de produção apenas um símbolo não-terminal.

Para Rabuske (1995), “as gramáticas livres de contexto servem bastante bem para descrever linguagens de programação, sendo particularmente úteis na construção de compiladores. Para linguagem natural, porém são um tanto fracas. O lado esquerdo só admite símbolo não-terminal. Isto significa que a expansão deste símbolo não consegue retratar as influências que sofre no contexto”.

Rabuske (1995) afirma que “a fraqueza das gramáticas livre de contexto não implica em proibição de uso em linguagem natural”.

3.4.3 GRAMÁTICAS REGULARES

As gramáticas regulares são também denominadas como gramática do tipo 3. Este tipo de gramática se caracteriza por ter no lado direito, apenas um único símbolo terminal, ou então um símbolo terminal seguido de apenas um único símbolo não terminal (Rabuske, 1995). O mesmo Rabuske (1995) justifica que este tipo de gramática tem utilidade reduzida na implementação de linguagem natural.

3.5 CARACTERÍSTICAS DO FORMALISMO GRAMATICAL

Segundo Rabuske (1995), anteriormente foram apresentados alguns tipos de gramáticas que podem ser usadas para a implementação da linguagem natural. Tê-los à mão ou conhecê-los, no entanto não basta. É também necessário saber escolher o mais apropriado para a aplicação pretendida.

Segundo Rabuske (1995), para efetuar a escolha adequada, o leitor deverá observar se a ferramenta apresenta algumas virtudes das quais destacam-se as seguintes:

- a) perspicácia: a ferramenta deve apresentar clareza. O que é expresso a respeito da língua deve ser diretamente visível em sua forma. No caso das gramáticas livres de contexto, por exemplo, cada regra segue determinado modelo de escrita. Com a análise de alguns poucos casos, consegue-se ter uma noção do problema que se terá pela frente com o seu uso;
- b) não direcionalidade: quando se escreve uma regra para uma gramática livre de contexto, não se sabe em que sentido será usada. Pode ser para gerar ou explicar, em processo *top-down* ou *bottom-up*. De modo geral ferramentas em que a direção de atuação é livre, são uma melhor escolha;
- c) ter correspondência entre a estrutura sintática e seu significado;
- d) ter recursos que sustentem a habilidade de distinção entre possíveis significados que podem ser atribuídos ao que é expresso.

3.6 AMBIGÜIDADE SINTÁTICA

A ambigüidade sintática ocorre quando uma frase pode ter mais de uma árvore de derivação. Isto pode acontecer quando a frase tem mais de um verbo e não fica claro qual é o

verbo principal da frase. Pode-se observar o duplo significado de interpretação nas frases abaixo definidas por Savadovsky (1989):

- a) Pedro viu Maria passeando;
- b) apague todos os arquivos usando *del*.

O mesmo Savadovsky (1989) demonstra a seguir como essas duas frases podem ser interpretadas: “Na primeira frase, Pedro estava passeando e viu Maria, ou Pedro viu Maria passeando, na televisão talvez ? E na segunda, é para apagar qualquer arquivo que esteja usando *del* (o que quer que seja isto), ou para usar *del* para apagar os arquivos” (Savadovsky, 1989).

Outras possibilidades para a ambigüidade sintática são demonstradas por Savadovsky (1989):

- a) no caso da ambigüidade sintática, ela é consequência direta das regras de gramática e é previsível;
- b) no segundo exemplo, apague todos os arquivos usando *del*, a análise semântica poderia talvez concluir que os arquivos não usam algo no mundo em consideração, e então sobraria a análise de que *del* deve ser usado para apagar os arquivos.

Podem existir casos em que a ambigüidade sintática não pode ser resolvida com a análise sintática ou semântica das frases. Nestes casos o que se pode fazer é consultar o usuário para que ele esclareça as dúvidas sobre o assunto em questão.

3.7 ANÁLISE SINTÁTICA DE FRASES

A análise sintática das frases tem como objetivo mostrar como as frases se relacionam com as palavras e entre si. Para fazer esta avaliação será utilizada a gramática livre de contexto.

Rich (1988) define análise sintática como “seqüências lineares de palavras são transformadas em estruturas que mostram como as palavras se relacionam entre si. Algumas seqüências de palavras poderão ser rejeitadas se violarem as normas da linguagem que regem a combinação de palavras”.

Para não precisar escrever os nomes completos das partes que constituem uma frase, serão utilizadas as abreviaturas, como os lingüistas fazem no estudo das frases.

A seguir, pode-se observar uma lista de abreviaturas utilizadas para criar as regras para a análise sintática:

- a) F = Frase;
- b) SN = sintagma nominal;
- c) ADV = advérbio;
- d) ADJ = adjetivo;
- e) ART = artigos;
- f) SUBS = substantivo;
- g) SV = sintagma verbal;
- h) VERBO = verbo;
- i) PREP = preposição;
- j) SP = sintagma preposicional.

A palavra sintagma e seus tipos apresentados acima são explicados por Silva (1986).

“O sintagma consiste num conjunto de elementos que constituem uma unidade significativa dentro da oração e que mantém entre si relações de dependência e de ordem”. (Silva, 1986).

O sintagma nominal (SN) pode ter como núcleo um nome ou pronome substantivo (pessoal, demonstrativo, indefinido, interrogativo, possessivo ou relativo) (Silva, 1986).

O sintagma preposicional (SP) é constituído de uma preposição seguida de um SN (Silva, 1986).

O sintagma verbal (SV), um dos elementos básicos da oração, pode apresentar configurações diversificadas. O SV pode ser representado apenas pelo núcleo, isto é, o verbo ou pelo verbo acompanhado de um ou mais elementos precedidos ou não de preposição (Silva, 1986).

Para fazer a análise sintática das frases serão usadas as seguintes regras. Onde as regras estão no formato (X → Y), e lê-se (X gera Y):

- a) F → SN SV;
- b) SN → ART SUBS;
- c) SN → ADJ SUBS;
- d) SN → SUBS;
- e) SN → PREP SUBS;
- f) SN → ADV ADJ SUBS;
- g) SN → ADV SUBS;
- h) SV → VERBO SN;
- i) SV → VERBO;
- j) SV → VERBO PREP;
- k) SV → VERBO ADV;
- l) SV → SV SP;
- m) SP → PREP SN.

Para Winston (1988), devido ao uso de setas é comum pensar-se em usar as regras de forma gerativa. Isto é, a partir de uma cadeia de apenas um símbolo que consiste de F, a primeira regra pode-se usar para reescrever F como SN SV; a segunda regra pode então ser usada para reescrever SN; e assim por diante, um passo de cada vez, até que seja produzida uma cadeia de apenas símbolos terminais.

Winston (1988) explica também como reescrever uma frase a partir das regras vistas anteriormente, “o procedimento para reescrever uma frase é simples: investigue a cadeia de símbolos vigente da esquerda para a direita até que um símbolo não terminal seja encontrado; substitua esse símbolo não terminal usando uma regra (se houver mais de uma regra aplicável, escolha uma ao acaso); e repita até que não reste nenhum não terminal”. Como pode ser cansativo escrever o desenvolvimento da cadeia após cada passo, esta pode ser mostrada na forma de uma árvore denominada de árvore de derivação.

No quadro 4, pode-se observar a análise sintática utilizando o método *top-down* para a frase “o vaqueiro tange a boiada”. No método *top-down*, parte-se da primeira regra substituindo-se o lado direito da regra pelo seu lado esquerdo até formar a frase.

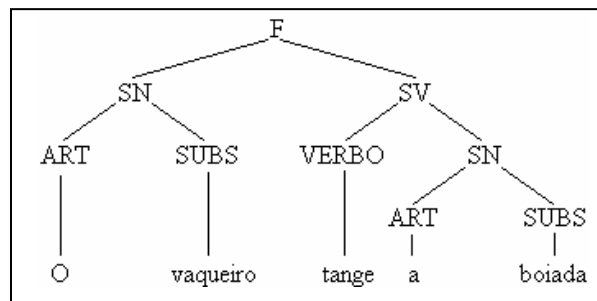
QUADRO 4: ANÁLISE SINTÁTICA COM O MÉTODO *TOP-DOWN*.

F
SN SV
ART SUBS VERBO SN
O vaqueiro tange ART SUBS
O vaqueiro tange a boiada

Outra forma para representar a aplicação das regras da gramática livre de contexto é através das árvores sintáticas. A árvore de derivação é uma estrutura gráfica que representa a estrutura sintática da frase. O método utilizado para criar as árvores sintáticas dos exemplos é conhecido como *top-down*. Para facilitar o entendimento sobre a árvore sintática, será utilizada uma frase onde a árvore não fique muito complexa.

O exemplo de árvore sintática será feito com a mesma frase usada anteriormente “o vaqueiro tange a boiada”, e pode ser observado na figura 9.

FIGURA 9: EXEMPLO DE ÁRVORE SINTÁTICA



Fonte: adaptado de Rabuske (1995)

Outra maneira de construir a árvore de análise sintática é usar as regras da gramática livre de contexto ao contrário. Este método é conhecido como *bottom-up*. No método *bottom-up*, parte-se da frase, substituindo-se as palavras pelos seus tipos e substituindo-se o lado esquerdo das regras pelo seu lado direito até chegar na primeira regra. No quadro 5, pode-se observar um exemplo de árvore sintática utilizando o método *bottom-up*. Será utilizada a mesma frase que foi utilizada nos exemplos anteriores “o vaqueiro tange a boiada”.

“Cada vez que uma substituição é feita , construímos uma parte da árvore de análise sintática que reflete a substituição. Esta árvore de análise sintática é chamada de análise

sintática de baixo para cima porque a construção da árvore de análise sintática parte de baixo com a substituição das palavras e move para cima em direção ao nó F da árvore de análise sintática” Winston (1988).

QUADRO 5: UTILIZAÇÃO DO MÉTODO *BOTTOM-UP*

O vaqueiro tange a boiada
ART vaqueiro tange a boiada
ART SUBS tange a boiada
SN tange a boiada
SN VERBO a boiada
SN VERBO ART boiada
SN VERBO ART SUBS
SN VERBO SN
SN SV
F

3.8 CONSTRUÇÃO DE UM ANALISADOR SINTÁTICO

Ao se construir um analisador sintático, também conhecido como *parser*, deve-se observar vários aspectos.

Segundo Oliveira (1999), “nos sistemas de processamento de linguagem natural, o maior problema é a transformação de uma frase potencialmente ambígua em uma não ambígua, a qual será utilizada pelo sistema. Esta transformação é conhecida como *parsing*”.

Winston (1988) explica como fazer a análise sintática utilizando o método *bottom-up*. Em vez de partir da seqüência de símbolos simples F e trabalhar descendo em direção a uma frase, parte-se de uma frase e trabalhamos subindo em direção a um símbolo F. O procedimento básico é simples: quando o lado direito de uma regra é observado, substitua os símbolos envolvidos pelo lado esquerdo; se for observado mais de um lado direito de regra, pegue um ao acaso; se ocorrer impasse, recue para a escolha mais recente e faça outra coisa.

3.9 TRABALHOS CORRELATOS QUE UTILIZAM LINGUAGEM NATURAL

Outros trabalhos já foram realizados nesta linha, porém não são aplicados a Análise Essencial de Sistemas. Como por exemplo, tem-se um processo desenvolvido por Manfro (2000) que utiliza linguagem natural para criar um modelo semântico de dados através de regras para a construção de diagramas provenientes de sentenças em português. Isto foi feito com IDEF1X (*Integration DEFINition for Information Modeling*).

IDEF1X é uma técnica para a modelagem de dados, e tem como base o modelo entidade-relacionamento. É utilizada para produzir um modelo gráfico lógico das informações.

Hübner (1992) desenvolveu um protótipo de ferramenta que trabalha com interface em linguagem natural. Este protótipo é responsável pela comunicação direta com o usuário. O usuário digita uma frase, que é processada e enviada para um simulador semântico e envia o resultado da execução para a interface.

Outro trabalho utilizando linguagem natural foi desenvolvido por Neves (1998). Este trabalho de conclusão de curso identifica verbos e substantivos da língua portuguesa, onde esses verbos e substantivos são candidatos a classes, atributos e métodos na análise orientada a objetos.

Um sistema desenvolvido em 1960 é apresentado por Allen (1995). Este sistema chamado de ELIZA, representa o papel de um terapeuta e o usuário representa o papel de um paciente.

A figura 10 apresenta um diálogo com o sistema ELIZA.

FIGURA 10: DIÁLOGO COM O SISTEMA ELIZA

User:	ELIZA:
Men are all alike.	In what way?
They're always bugging us about something or other.	Can you think of a specific example ?
Well, my boyfriend made me come here.	Your boyfriend made you come here.
He says I'm depressed much of the time.	I am sorry to hear you are depressed.
It's true, I am unhappy.	Do you think coming here will Help you not to be unhappy?
I need some help; that much seems certain.	What would it mean to you if you Got some help?
Perhaps I could learn to get along with my mother.	Tell me more about your family.
My mother takes care of me.	Who else in your family takes care of you?
My father.	Your father.
You are like my father in some ways.	What resemblance do you see?

Fonte: Allen (1995)

Um sistema para a geração semi-automática de um modelo de dados E-R (Entidade-Relacionamento) utilizando linguagem natural é apresentado por Gomez (1999).

O algoritmo implementado pelo gerador de E-R, desenvolvido por Gomez (1999), consiste em duas passagens. Todas as estruturas são examinadas na primeira passagem, onde algumas estruturas podem resultar na criação de relações, entidades ou atributos, outras estruturas não podem causar nenhuma ação pelo gerador e, finalmente outras estruturas têm sido consideradas na segunda passagem, depois que o gerador de E-R tivesse uma chance para juntar possíveis e pertinentes informações sobre outras estruturas. São analisadas gramaticalmente frases informadas pelo usuário.

O *parser* reconhece os complementos verbais, frases preposicionais e outras cláusulas relativas. As estruturas de representação do conhecimento são construídas pelo interpretador semântico, que são entradas para o gerador de E-R, onde são reconhecidas entidades, atributos e relações.

O gerador de E-R possui dois conjuntos de regras, genéricas e específicas. Primeiramente são testadas as regras específicas, se estas não tiverem sucesso, serão testadas as regras genéricas.

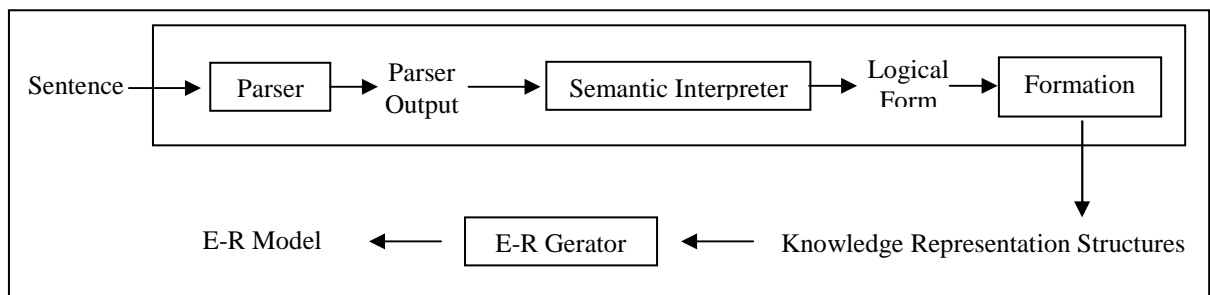
As regras específicas utilizam conceitos verbais e indicam que uma ação específica deve ser executada pelo gerador de E-R, por exemplo, relações que definem relações hierárquicas entre as entidades ou que definem atributos chaves.

As regras genéricas não são ligadas a conceitos verbais e as ações do gerador de E-R são somente baseadas em argumentos.

A técnica desenvolvida foi implementada em *Common Lisp* sendo executado em estações de trabalho. O programa já foi testado com um conjunto de trinta problemas (Gomez, 1999).

A figura 11 mostra como funciona o sistema desenvolvido por Gomez (1999).

FIGURA 11: COMPONENTES DO SISTEMA



Fonte: adaptado de Gomez (1999).

4 DESENVOLVIMENTO DA FERRAMENTA

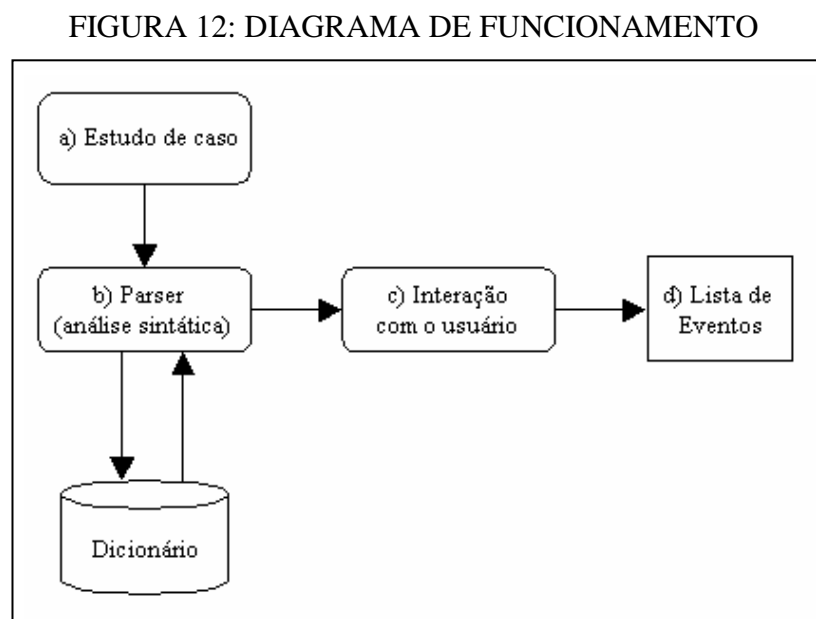
A ferramenta que foi desenvolvida possui as funções básicas de um editor de textos, permitindo assim que seja informado um texto conhecido como um estudo de caso. Necessita também de um dicionário onde se pode cadastrar as palavras utilizadas no estudo de caso. A ferramenta deve possuir um processo para efetuar a análise sintática das frases existentes no estudo de caso e um processo para efetuar a verificação ortográfica do texto. Para efetuar a análise sintática foi utilizada uma das sub-áreas do processamento de linguagem natural conhecido como compreensão de texto ou compreensão de linguagem natural.

O principal objetivo da ferramenta é identificar a lista de eventos presente no estudo de caso que estiver sendo analisado.

4.1 ESPECIFICAÇÃO

A especificação da ferramenta foi elaborada com a ferramenta *CASE Power Designer* 6.1. Para a criação dos fluxogramas foi utilizado o editor gráfico *SmartDraw* 4.0. A abordagem utilizada para a especificação foi a Análise Essencial de Sistemas.

A figura 12 mostra um diagrama do funcionamento da ferramenta.



A seguir pode-se observar a explicação para cada um dos quatro passos apresentado na figura 12:

- a) o usuário informa ao sistema o estudo de caso, este pode ser digitado ou aberto de um arquivo nos formatos txt ou rtf;
- b) quando o usuário aciona a rotina para gerar a lista de eventos, todas as frases do estudo de caso são analisadas sintaticamente pelo *parser*, que faz a interação com o dicionário para reconhecer o tipo de cada palavra. As frases que são reconhecidas sintaticamente não serão utilizadas para fazer a interação com o usuário, são examinadas diretamente;
- c) as frases que não forem reconhecidas sintaticamente pelo *parser* são mostradas para o usuário, e este pode escolher quais as frases que serão utilizadas para se identificar os eventos existentes. Quando dois eventos são similares, o usuário pode escolher quais os eventos que serão utilizados;
- d) a lista de eventos é mostrada para o usuário e este tem a opção de salvar a lista em arquivo texto.

Primeiramente o usuário informa o estudo de caso, que pode ser digitado ou pode ser aberto um arquivo já existente. Quando o estudo de caso já estiver carregado na ferramenta, o usuário pode iniciar o processo para gerar a lista de eventos, que será apresentada para o usuário quando o processo estiver concluído.

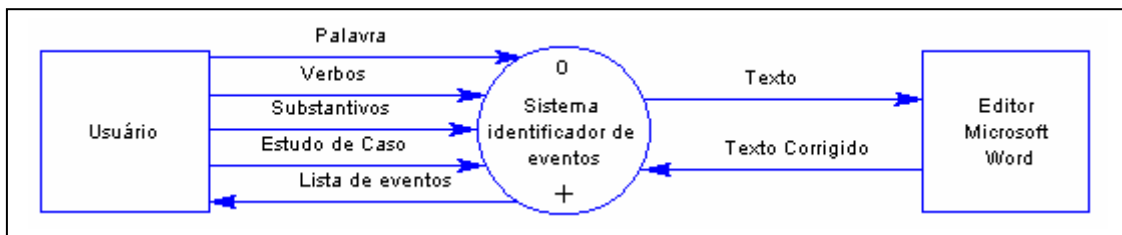
No quadro 6 pode-se observar os eventos que foram identificados para o desenvolvimento da ferramenta.

QUADRO 6: EVENTOS IDENTIFICADOS

Usuário verifica ortografia do texto (F) Usuário cadastra palavras (F) É momento de gerar a lista de eventos (T)
--

A figura 13 mostra o diagrama de contexto desenvolvido na ferramenta CASE *Power Designer*.

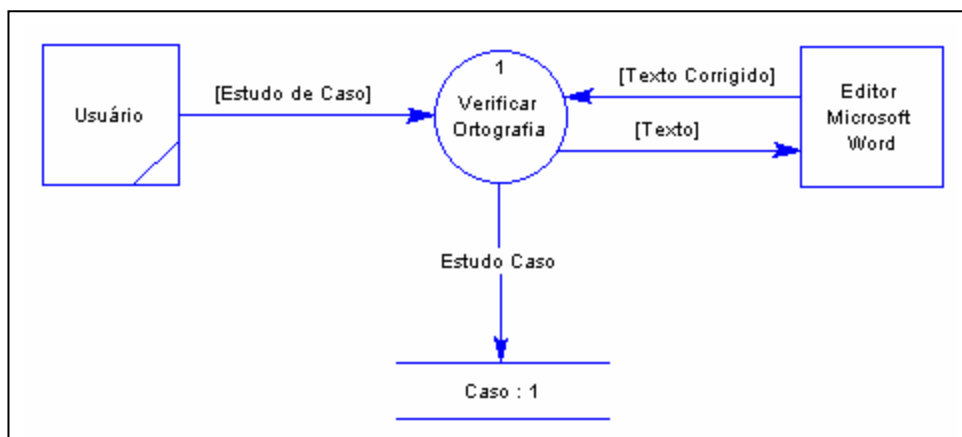
FIGURA 13: DIAGRAMA DE CONTEXTO



A seguir são apresentados os diagramas de fluxo de dados (DFD) para cada evento, seguindo a ordem dos eventos apresentados no quadro 6.

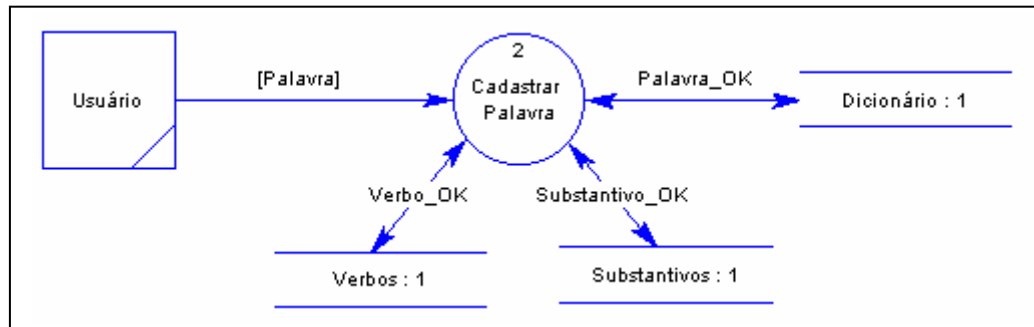
Quando o usuário desejar efetuar a verificação ortográfica do texto, o estudo de caso é enviado para o editor de textos *Microsoft Word* onde é feita a verificação ortográfica. Quando a verificação ortográfica for concluída ou cancelada pelo usuário, o estudo de caso é devolvido para a ferramenta. O DFD que mostra o processo para efetuar a verificação ortográfica pode ser observado na figura 14.

FIGURA 14: DFD GERADO PARA VERIFICAR A ORTOGRAFIA



A figura 15 mostra o DFD gerado para cadastrar as palavras no dicionário

FIGURA 15: DFD PARA CADASTRAR PALAVRAS

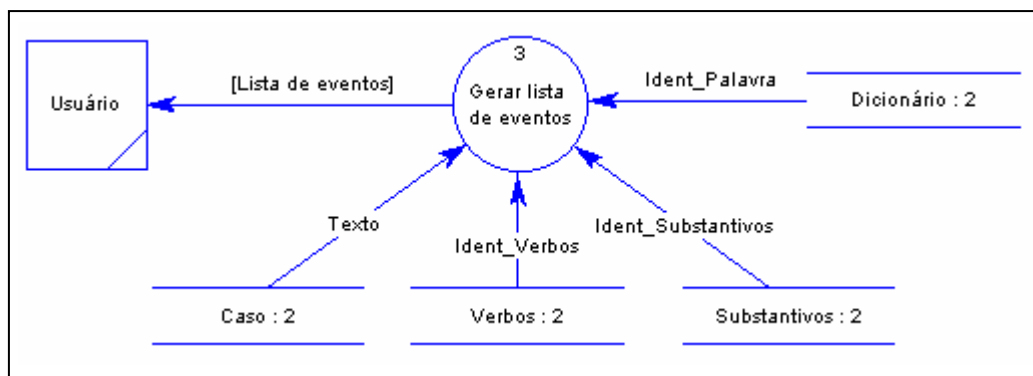


Percebe-se que ao cadastrar uma palavra, se a mesma for um verbo é utilizada uma tabela secundária para cadastrar as flexões deste verbo. Se for um substantivo é utilizada outra tabela secundária para cadastrar as variações deste substantivo.

Para gerar a lista de eventos, as frases são separadas e analisadas individualmente. O primeiro passo é verificar se na frase existe alguma palavra que possa fazer parte do padrão de um tipo de evento, por exemplo, um verbo. Se esta palavra estiver na frase o evento é identificado obedecendo o padrão para cada tipo de evento: fluxo, tempo ou controle. Quando os eventos estiverem identificados, é feito um refinamento para verificar a existência de eventos similares, neste caso é feita uma interação com o usuário para que este possa optar pelos eventos que poderão ser ignorados e aceitos. Quando a lista de eventos estiver pronta, o usuário pode optar por salvar a lista em arquivo texto.

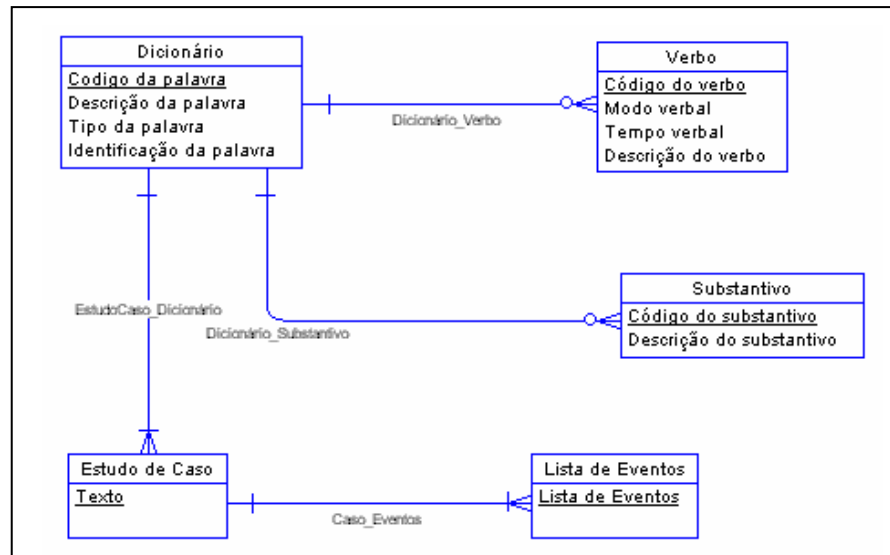
A figura 16 mostra o DFD utilizado para gerar a lista de eventos.

FIGURA 16: DFD PARA GERAR A LISTA DE EVENTOS



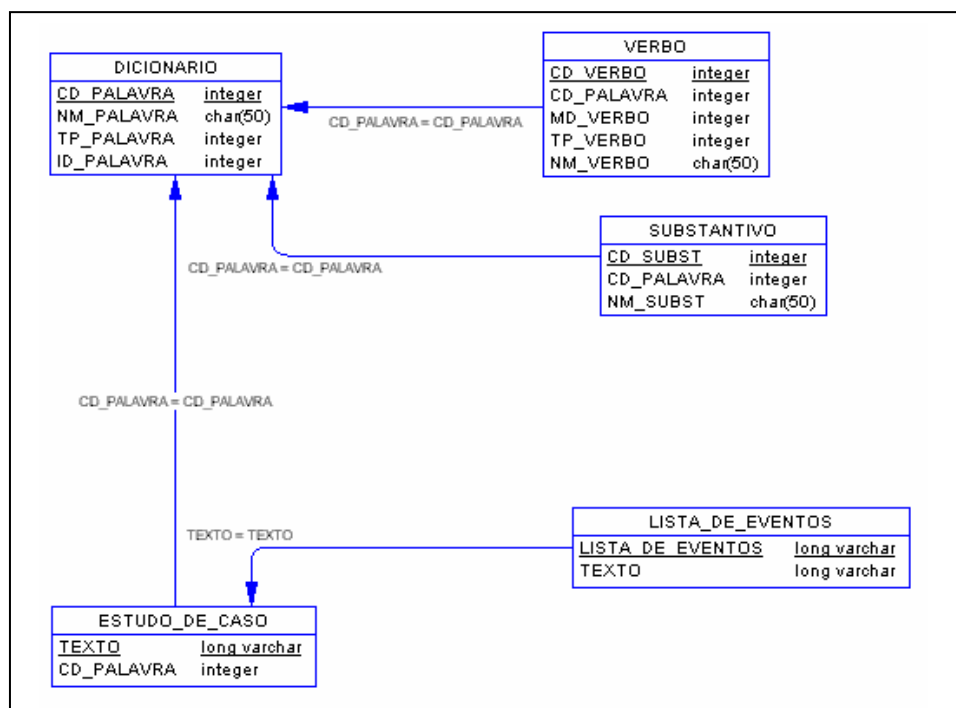
A figura 17 mostra o modelo lógico de entidade e relacionamento (MER) que mostra a relação entre as tabelas e os arquivos de usuário representados pelas entidades estudo de caso e lista de eventos

FIGURA 17: MER LÓGICO



A figura 18 mostra o MER físico, gerado pela ferramenta *CASE Power Designer*.

FIGURA 18: MER FÍSICO



O fluxograma completo sobre o funcionamento da ferramenta pode ser observado nas figuras 19 e 20. Neste fluxograma é representado todo o todo o processo para gerar a lista de eventos.

FIGURA 19: PRIMEIRA PARTE DO FLUXOGRAMA

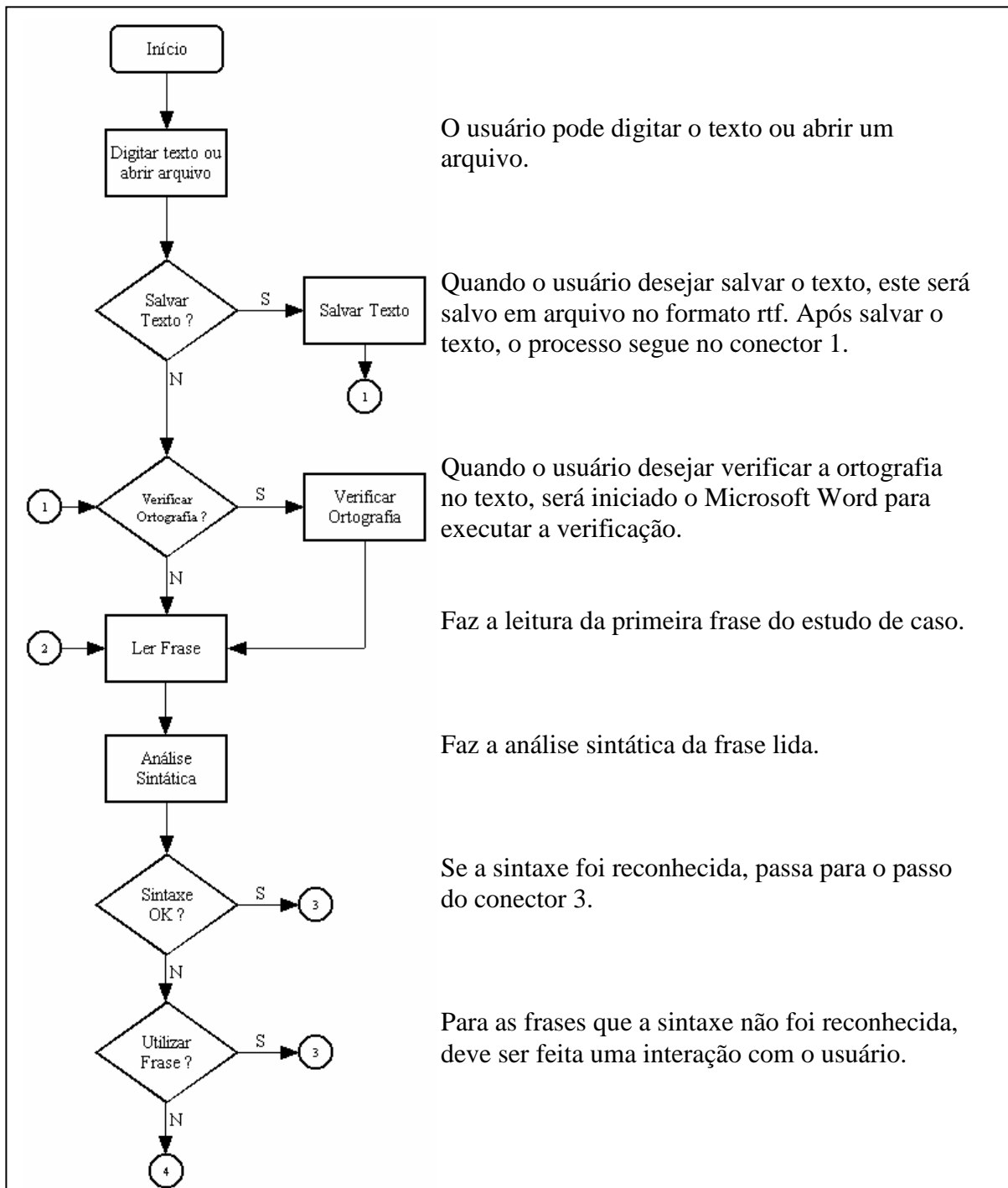
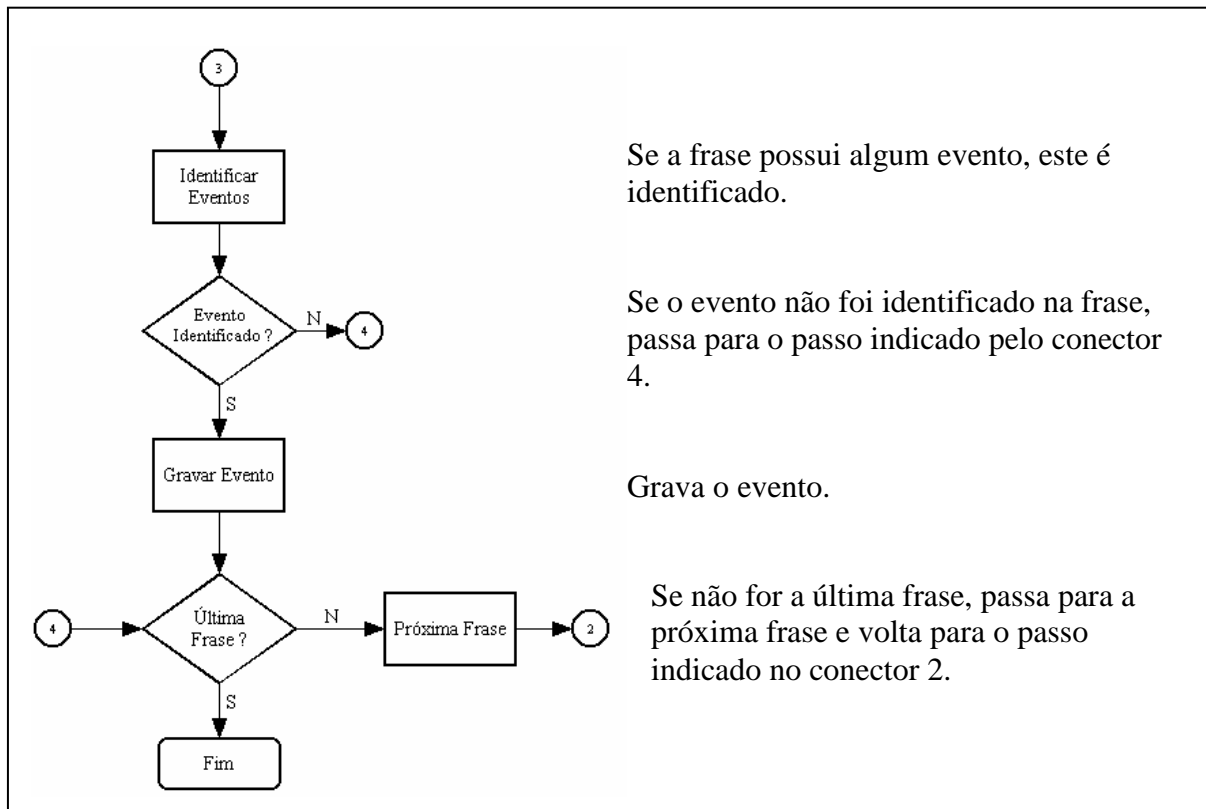


FIGURA 20: SEGUNDA PARTE DO FLUXOGRAMA



4.2 IMPLEMENTAÇÃO

4.2.1 TÉCNICAS E FERRAMENTAS UTILIZADAS

Para a implementação da ferramenta foi utilizada a linguagem de programação *Object Pascal* e o ambiente de desenvolvimento *Delphi 4.0*. Para a criação das tabelas que compõem o dicionário foi utilizado o *Paradox 7.0*.

Nas figuras 21, 22 e 23 pode-se observar a estrutura das tabelas criadas no utilitário *Database Desktop* e que são utilizadas para armazenar as palavras do dicionário.

Para desenvolver o analisador sintático das frases no ambiente de desenvolvimento *Delphi 4.0*, foi utilizada a gramática livre de contexto e o método *bottom-up*, presente no processamento de linguagem natural.

A figura 21 mostra a estrutura da tabela principal do dicionário

FIGURA 21: ESTRUTURA DA TABELA DE DICIONÁRIO

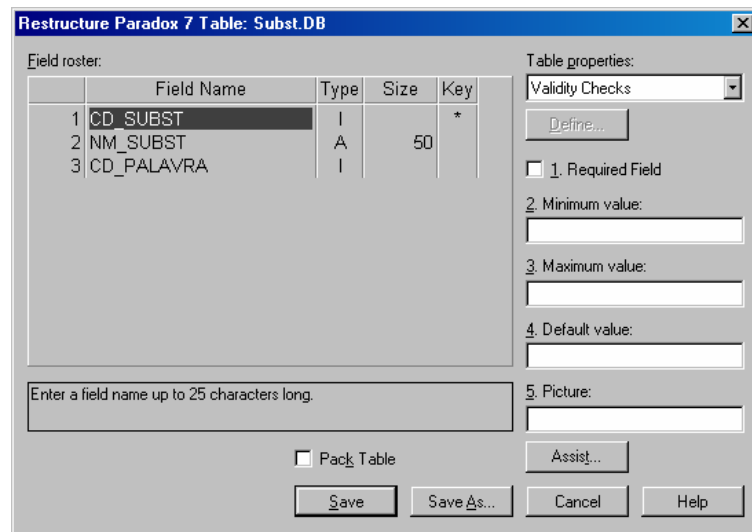
	Field Name	Type	Size	Key
1	CD_PALAVRA	I		*
2	NM_PALAVRA	A	50	
3	TP_PALAVRA	I		
4	ID_PALAVRA	I		

Na figura 22 pode-se observar a estrutura da tabela de verbos. Na figura 23 pode-se observar a estrutura da tabela de substantivos. As outras entidades representadas na figura 17 como estudo de caso e lista de eventos são arquivos que o usuário pode optar por salvar. O arquivo do estudo de caso pode ser salvo a qualquer momento, o arquivo da lista de eventos pode ser salvo quando esta for gerada.

FIGURA 22: ESTRUTURA DA TABELA DE VERBOS

	Field Name	Type	Size	Key
1	CD_VERBO	I		*
2	MD_VERBO	I		
3	TP_VERBO	I		
4	NM_VERBO	A	50	
5	CD_PALAVRA	I		

FIGURA 23: ESTRUTURA DA TABELA DE SUBSTANTIVOS



Para algumas rotinas foi desenvolvido um algoritmo, para utilizá-lo como roteiro durante a implementação. Um dos algoritmos desenvolvidos foi utilizado para implementar o analisador sintático de frases. Este algoritmo pode ser observado no quadro 7.

QUADRO 7: ALGORITMO DO ANALISADOR SINTÁTICO

```

Separar as regras e o resto da frase em duas variáveis
repita
  verificar se existe alguma regra que o lado direito se aplica um
  trecho da frase
  se existir,
    troque esta parte da frase pela regra do lado esquerdo
  se não existir então
    separe uma palavra da frase
    troque a palavra separada pelo seu tipo especificado no dicionário
  se a frase for igual a primeira regra 'F' então
    frase está correta
    pare
  senão
    frase não esta correta
até (((não existir mais palavras a serem trocadas) e (regras a serem
testadas)) ou (frase ser igual a primeira regra 'F'))

```

Uma parte do código fonte do analisador sintático de frases pode ser observada nos quadros 8 e 9.

QUADRO 8: PARTE INICIAL DO ANALISADOR SINTÁTICO

```

// Duas strings para trabalhar com a frase.
Frase      := Trim(LowerCase(Frase)) + ' .';
FraseRegra := (Copy(Frase, 1, Pos('.', Frase) - 1));
repeat
  // Inicializa as variáveis
  Regra := '';
  I := 0;
  Inc(Tentativas);
  // Copia as regras da frase. a frase está em minúsculo e as regras em
  // maiúsculo
  for I := 1 to Length(FraseRegra) do
  begin
    if FraseRegra[I] in ['A'..'Z'] then
      Regra := Regra + FraseRegra[I]
    else Regra := Regra + ' ';
  end;
  // Tira os espaços no inicio e no final
  Regra := Trim(Regra);

  // testa a regra com lado direito
  if Regra <> '' then
  begin
    I := 0;
    Troca := False;
    // Percorre todas as regras
    for I := 1 to 12 do
    begin
      // Verifica se a regra existe na frase
      PosicaoRegra := Pos(Direito[I], Regra);
      // Se existir, troca as regras
      if PosicaoRegra > 0 then
      begin
        // Não está no inicio da frase
        if PosicaoRegra > 1 then
          Regra := Copy(Regra, 1, PosicaoRegra - 1) + Esquerdo[I]
        else
          // A regra está no inicio da frase
          if PosicaoRegra = 1 then
            Regra := Esquerdo[I] + Copy(Regra, Length(Direito[I]) + 1,
              Length(Regra));
          // Indica que houve troca de regras
          Troca := True;
          Break;
        end;
      end;
    end;
    // Monta a frase com as regras
    FraseRegra := Regra + ' ' + Frase;
  end;
end;

```

QUADRO 9: TRECHO DO CÓDIGO DO ANALISADOR SINTÁTICO

```

// Se não trocou regras, troca as palavras
if not Troca then
begin
  Palavra := Copy(Frase, 1, Pos(' ', Frase) - 1);
  if (Palavra <> '.') and (Palavra <> '') then
  begin
    // Localiza a palavra no dicionário e retorna o tipo da palavra
    if DMPrincipal.QueryVerbo.Locate('NM_VERBO', Palavra,
    [loCaseInsensitive]) then // Procura no dicionário de verbos
      TipoPalavra := 7
    else
      if DMPrincipal.QuerySubst.Locate('NM_SUBST', Palavra,
      [loCaseInsensitive]) then
        // Procura no dicionário de substantivos
        TipoPalavra := 6
      else
        if DMPrincipal.QueryPalavra.Locate('NM_PALAVRA', Palavra,
        [loCaseInsensitive]) then // Procura no dicionário principal
          TipoPalavra :=
            DMPrincipal.QueryPalavra.FieldName('TP_PALAVRA').AsInteger;

    // Passa para as regras o tipo da palavra
    case TipoPalavra of
      0 : Regra := Regra + ' ' + 'ADJ ';
      1 : Regra := Regra + ' ' + 'ADV ';
      2 : Regra := Regra + ' ' + 'ART ';
      3 : Regra := Regra + ' ' + 'PREP ';
      6 : Regra := Regra + ' ' + 'SUBS ';
      7 : Regra := Regra + ' ' + 'VERBO ';
    end;
  end;
end;
end;

```

Outras rotinas são mais fáceis de implementar e não foi preciso desenvolver um algoritmo para utilizá-lo como roteiro. Um exemplo de uma destas rotinas pode ser observado no quadro 10, onde é apresentada a rotina que utiliza o editor de textos *Microsoft Word* para efetuar a verificação ortográfica no estudo de caso.

QUADRO 10: VERIFICAÇÃO ORTOGRÁFICA NO *MICROSOFT WORD*

```

procedure TFEstudoCaso.VerificaOrtografia;
var
  Word: Variant;
  Dir : string;
begin
  // Obtem o diretório temporário
  Dir := Temporario;
  // Cria um objeto do tipo Word.Basic
  Word:=GetOrCreateObject('Word.Basic');
  // faz um arquivo novo - Menu + Submenu
  Word.FileNew;
  // Mostra o word
  Word.AppShow;
  // Insere o texto no word
  Word.Insert(MemoEstudoCaso.Lines.Text);
  // Verifica a ortografia
  Word.ToolsSpelling;
  // Seleciona todo o texto
  Word.EditSelectAll;
  // Passa para a aplicação o texto do word
  MemoEstudoCaso.Lines.Text := Word.Selection;
  // Salva o texto num arquivo para não mostrar a mensagem de salvar
  Word.FileSaveAs(Dir + 'Teste.doc');
  // Sai do word
  Word.FileExit;
  // Passa a aplicação para primeiro plano
  Application.BringToFront;
  // Apaga o arquivo temporário
  DeleteFile(Dir + 'Teste.doc');
end;

```

Para criar o arquivo temporário utilizado na verificação ortográfica, foi necessário desenvolver uma função para obter o diretório temporário utilizado pelo sistema operacional. O quadro 11 mostra o código fonte desta função.

QUADRO 11: DIRETÓRIO TEMPORÁRIO DO SISTEMA OPERACIONAL

```

function TFEstudoCaso.Temporario: string;
// Esta função tem a finalidade de informar o diretório temporário do
// sistema operacional.
var
  TempDir: array[0..255] of Char;
begin
  GetTempPath(255, @TempDir);
  Result := StrPAs(TempDir);
end;

```

4.3 FUNCIONAMENTO DA FERRAMENTA

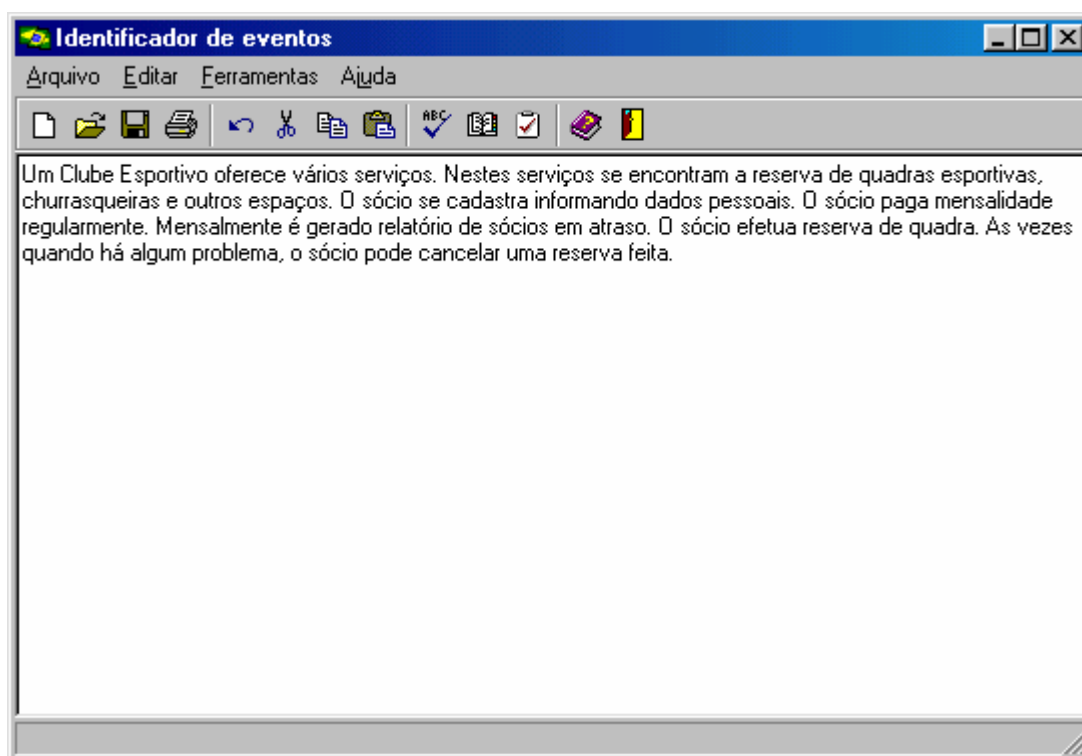
A seguir pode-se observar a funcionalidade da ferramenta desenvolvida. O estudo de caso utilizado pode ser observado no quadro 12.

QUADRO 12: ESTUDO DE CASO DO CLUBE ESPORTIVO

Um Clube Esportivo oferece vários serviços. Nestes serviços se encontram a reserva de quadras esportivas, churrasqueiras e outros espaços. O sócio se cadastra informando dados pessoais. O sócio paga mensalidade regularmente. Mensalmente é gerado relatório de sócios em atraso. O sócio efetua reserva de quadra. Às vezes quando há algum problema, o sócio pode cancelar uma reserva feita.

A figura 24 mostra o estudo de caso aberto na ferramenta. Esta tela é apresentada em tamanho reduzido e seu arquivo de ajuda (*help*) pode ser observado no anexo 1 deste trabalho.

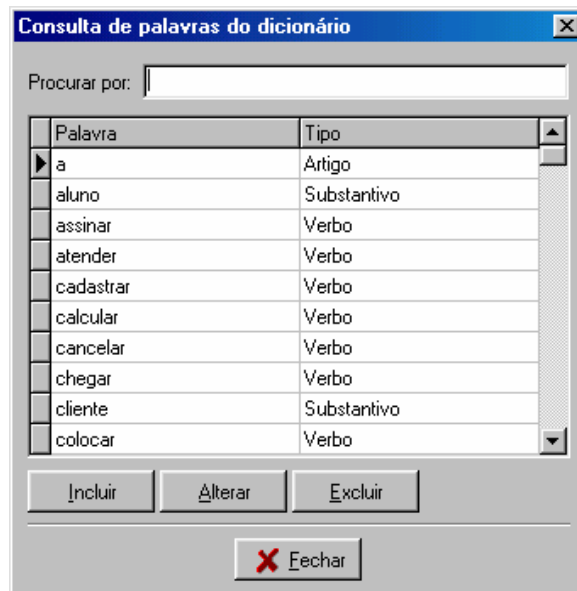
FIGURA 24: TELA PRINCIPAL DA FERRAMENTA



Na figura 25 pode-se observar a tela de manutenção de dicionário. Esta tela permite incluir, alterar e excluir palavras do dicionário. Esta tela possui um campo de procura. Para

cada tecla pressionada neste campo, a lista de palavras é percorrida para localizar a palavra desejada. Esta tela mostra também uma grade com a palavra e seu tipo.

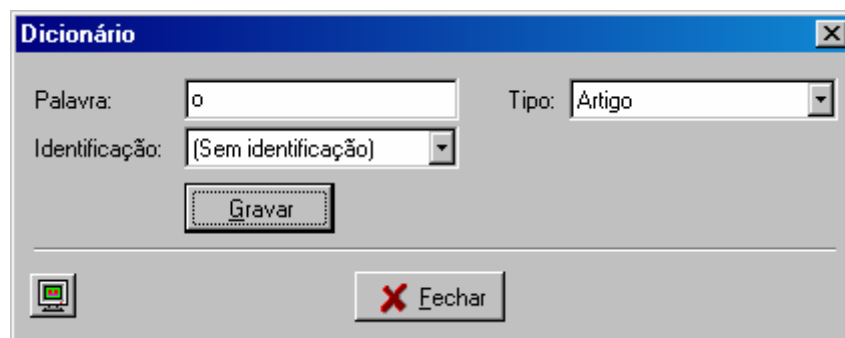
FIGURA 25: TELA DE MANUTENÇÃO DO DICIONÁRIO



A figura 26 mostra o cadastramento de um artigo. O campo identificação pode ser utilizado quando for cadastrada uma palavra que indique a passagem de tempo, por exemplo, “mensalmente”. Quando o foco estiver sobre o campo tipo, pode-se pressionar a tecla F1, para abrir a ajuda de cadastramento de palavras. Quando for pressionado o botão existente no canto inferior esquerdo com o desenho de um monitor, a tela é preparada para o cadastramento de uma nova palavra.

Quando o tipo da palavra for verbo ou substantivo, ao pressionar o botão gravar, a tela é alterada para que se possa cadastrar verbos e substantivos.

FIGURA 26: CADASTRAMENTO DE UMA PALAVRA



A figura 27 mostra um verbo sendo cadastrado. Nesta tela pode-se escolher o modo verbal (indicativo ou subjuntivo) e o tempo verbal (presente, pretérito ou futuro).

FIGURA 27: CADASTRAMENTO DE UM VERBO

The screenshot shows a window titled "Dicionário" with a blue header and a close button (X). The main area contains the following elements:

- Palavra:** A text input field containing "vender".
- Tipo:** A dropdown menu set to "Verbo".
- Identificação:** A dropdown menu set to "(Sem identificação)".
- Gravar:** A button to save the entry.
- Modo:** Two radio buttons: "Indicativo" (selected) and "Subjuntivo".
- Tempo:** Three radio buttons: "Presente" (selected), "Pretérito", and "Futuro".
- Verbo:** An empty text input field.
- Adicionar:** A button to add the verb to the list.
- Alterar:** A button to edit the entry.
- Excluir:** A button to delete the entry.
- Fechar:** A red button with a close icon and the text "Fechar".

On the right side, a list of verb conjugations is displayed in a text area:

```
vendo  
vendes  
vende  
vendemos  
vendeis  
vendem
```

A figura 28 mostra o cadastramento de um substantivo.

FIGURA 28: CADASTRAMENTO DE UM SUBSTANTIVO

The screenshot shows a window titled "Dicionário" with a blue header and a close button (X). The main area contains the following elements:

- Palavra:** A text input field containing "aluno".
- Tipo:** A dropdown menu set to "Substantivo".
- Identificação:** A dropdown menu set to "(Sem identificação)".
- Gravar:** A button to save the entry.
- Substantivo:** An empty text input field.
- Adicionar:** A button to add the noun to the list.
- Alterar:** A button to edit the entry.
- Excluir:** A button to delete the entry.
- Fechar:** A red button with a close icon and the text "Fechar".

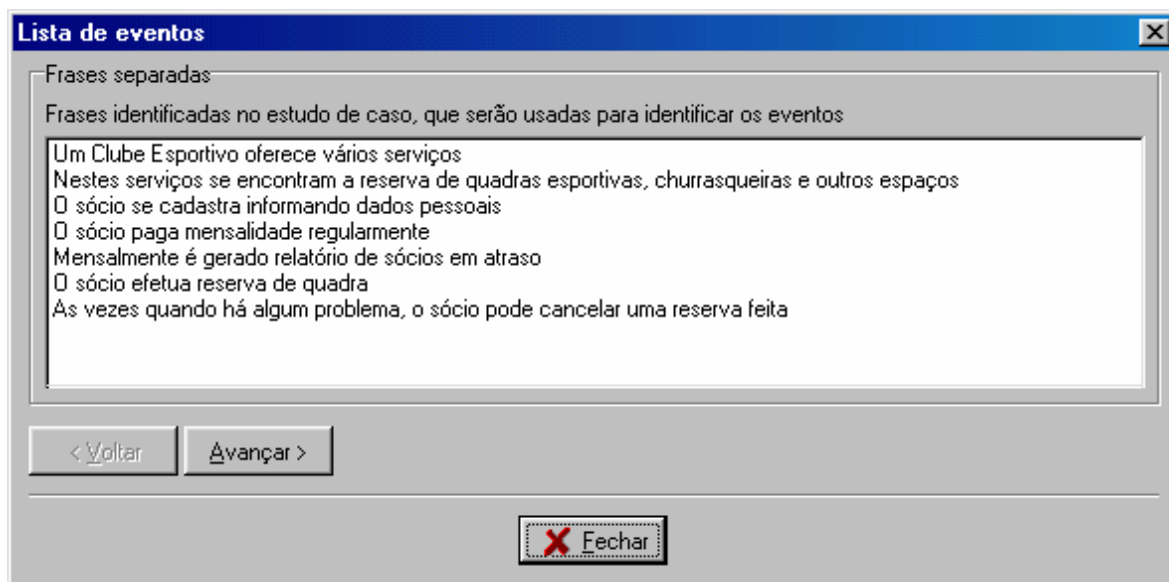
On the right side, a list of noun variations is displayed in a text area:

```
aluna  
alunos  
alunas
```


Quando se optar por identificar os eventos existentes no estudo de caso, inicialmente é feita uma verificação de todas as palavras do estudo de caso. As palavras que ainda não estão cadastradas no dicionário são apresentadas ao usuário para que seja efetuado o seu cadastro.

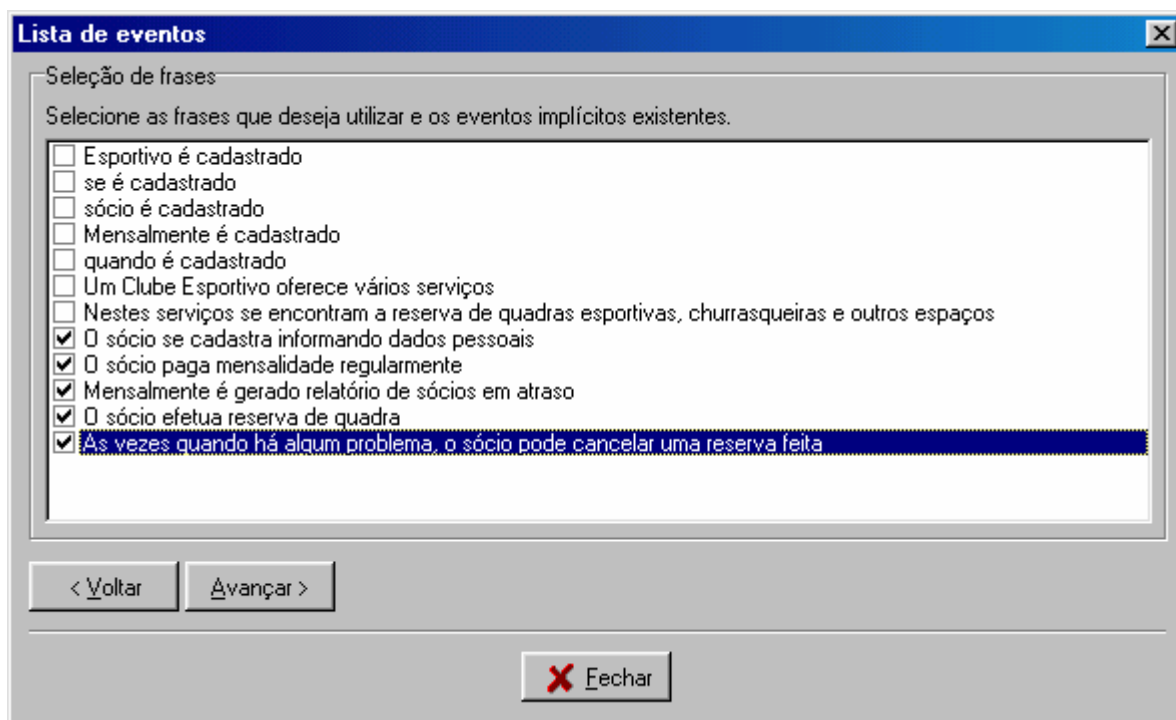
A tela de identificação de eventos possui três partes. O desenho das telas foi reduzido para que o processo possa ser ilustrado. A primeira parte mostra as frases identificadas no estudo de caso. Para separar uma frase, o primeiro passo é localizar a posição do ponto final. Depois de localizada a posição, a frase é copiada do início até a posição do ponto final. Esta parte inicial pode ser observada na figura 29.

FIGURA 29: FRASES EXISTENTES NO ESTUDO DE CASO



A segunda parte do processo de identificação dos eventos é a seleção das frases que serão utilizadas para se identificar os eventos (figura 30). As frases do estudo de caso que foram identificadas na análise sintática de frases não serão mostradas nesta parte do processo. Nesta tela também são apresentadas algumas frases que podem representar possíveis eventos implícitos. Isto é feito quando se identifica um substantivo antes do primeiro verbo de cada frase.

FIGURA 30: SELEÇÃO DE FRASES QUE SERÃO UTILIZADAS

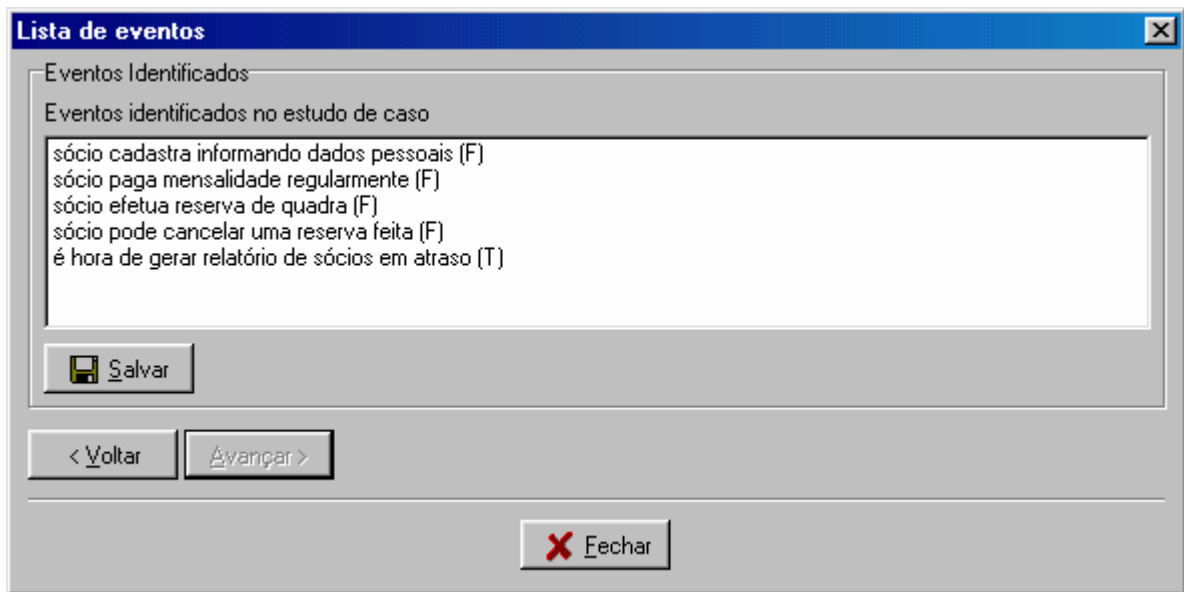


A terceira e última parte do processo de identificação de eventos, apresenta os eventos identificados no estudo de caso. Nesta parte do processo, pode-se salvar a lista de eventos em arquivo no formato texto. Quando um evento é localizado, este já é classificado dependendo de seu tipo. Para classificar os eventos são utilizados os seguintes símbolos:

- a) (F): evento de fluxo;
- b) (T): evento temporal;
- c) (C): evento de controle.

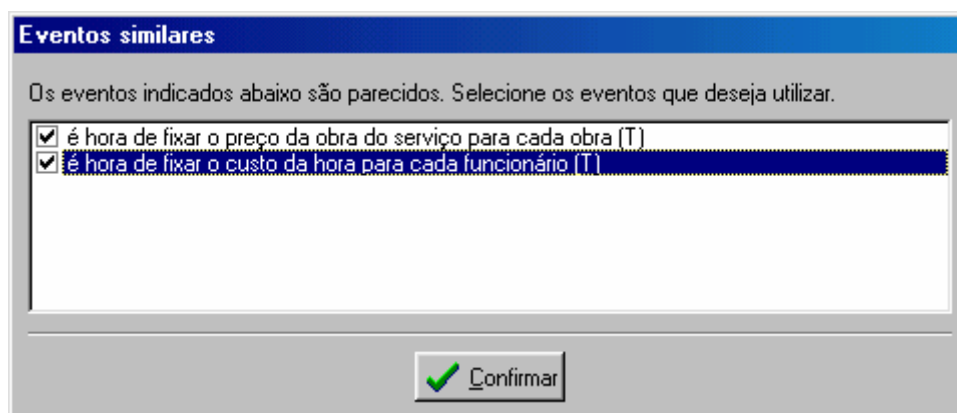
A lista de eventos identificados pode ser observada na figura 31.

FIGURA 31: EVENTOS IDENTIFICADOS NO ESTUDO DE CASO



Quando existirem dois eventos similares com a maioria de suas palavras iguais, é apresentada para o usuário uma tela onde se pode escolher os eventos a serem utilizados. Os eventos selecionados nesta tela serão adicionados na lista de eventos. Utilizando-se como exemplo os eventos existentes na figura 32 apresentada a seguir, pode-se observar que existem 8 palavras iguais nas duas frases. A maior frase possui 13 palavras. Neste caso seria necessário apenas 7 palavras iguais para que esta tela seja exibida. Os eventos apresentados nesta tela não fazem parte do estudo de caso apresentado no quadro 12. Esta tela pode ser observada na figura 32.

FIGURA 32: EVENTOS SIMILARES



4.4 MONTAGEM DOS EVENTOS

Para cada tipo de evento é utilizado um processo diferente que pode ser observado a seguir.

Para montar um evento de fluxo, primeiramente é verificado se na frase existe no mínimo um verbo e dois substantivos, um antes e um depois do verbo para seguir o formato do evento. Este tipo de evento é montado no formato: substantivo antes do verbo + primeiro verbo + complemento verbal. Neste caso o complemento verbal é toda a parte restante da frase existente depois do primeiro verbo.

Para montar um evento temporal, primeiramente é verificado se na frase existe uma palavra que está identificada no dicionário como tempo e pelo menos um verbo. Este tipo de evento é montado no formato: é hora de + verbo + complemento verbal. Para este tipo de evento, o complemento verbal é o restante da frase existente após o verbo.

Para montar um evento de controle, a frase deve possuir pelo menos um verbo e um substantivo antes do verbo. Este tipo de evento é montado no formato: substantivo + verbo + complemento. O complemento verbal é todo o trecho da frase existente depois do verbo.

4.5 RESULTADO E DISCUSSÃO

Foram realizados alguns testes de utilização da ferramenta desenvolvida. Os estudos de caso foram obtidos de notas de aula do professor Everaldo Artur Grahl na disciplina de Análise e Projeto de Sistemas I da FURB. Um processo realizado para cada estudo de caso novo ou modificado é o cadastramento das palavras que ainda não existem no dicionário. Um dos estudos de caso que foram testados é apresentado no quadro 12. A seguir, pode-se observar no quadro 13 um comparativo com os eventos identificados pela ferramenta e os eventos existentes no estudo de caso.

QUADRO 13: TESTE COM O ESTUDO DE CASO DO CLUBE ESPORTIVO

Eventos existentes	Eventos Identificados pela ferramenta
Sócio cadastra informando dados pessoais (F)	Sócio cadastra informando dados pessoais (F)
Sócio paga mensalidade regularmente (F)	Sócio paga mensalidade regularmente (F)
Sócio efetua reserva de quadra (F)	Sócio efetua reserva de quadra (F)
Sócio pode cancelar uma reserva feita (F)	Sócio pode cancelar uma reserva feita (F)
É hora de gerar relatório de sócios em atraso (F)	É hora de gerar relatório de sócios em atraso (T)
Serviço é cadastrado (F)	

Com o resultado do estudo de caso apresentado no quadro 12, somente o evento implícito “serviço é cadastrado (F)”, não foi identificado. Os demais eventos foram identificados corretamente, totalizando 83% de acerto.

No quadro 14 pode-se observar o estudo de caso do consultório médico que também foi utilizado nos testes.

QUADRO 14: ESTUDO DE CASO DO CONSULTÓRIO MÉDICO

Um consultório médico oferece serviços de atendimento médico. Os médicos são cadastrados junto a secretaria do consultório. As especialidades são cadastradas pelos médicos. Um paciente quando chega a clínica primeiramente se cadastra. Quando o paciente necessita solicita uma consulta. O paciente pode cancelar uma consulta. Mensalmente o gerente do consultório recebe um relatório com os pacientes mais atendidos.

O resultado do estudo de caso apresentado no quadro 14 é mostrado no quadro 15.

QUADRO 15: TESTE COM O ESTUDO DE CASO DO CONSULTÓRIO MÉDICO

Eventos existentes	Eventos identificados pela ferramenta
Médico é cadastrado (F)	Médicos são cadastrados junto a secretaria do consultório (F)
Médico cadastra especialidades (F)	Especialidades são cadastradas pelos médicos (F)
Paciente é cadastrado (F)	Paciente chega a clínica primeiramente se cadastra (F)
Paciente solicita consulta (F)	Paciente necessita solicita uma consulta (F)
Paciente cancela consulta (F)	Paciente pode cancelar uma consulta (F)
É hora de receber um relatório com os pacientes mais atendidos (T)	É hora de receber um relatório com os pacientes mais atendidos (T)
Serviço é cadastrado (F)	

No estudo de caso apresentado no quadro 14, alguns eventos implícitos não foram identificados e outros foram identificados com um formato mais detalhado totalizando 85,70% de acerto.

Outro teste foi realizado com o estudo de caso apresentado no quadro 16.

QUADRO 16: ESTUDO DE CASO DO ALUNO

O sistema de matrícula da FURB permite a consulta de dados sobre os alunos. Os alunos se cadastram na divisão acadêmica. Os alunos podem consultar suas notas. O professor informa notas de alunos. A secretaria da FURB cadastra as disciplinas. Semestralmente é gerado boletim de notas.

O resultado do estudo de caso apresentado no quadro 16, pode ser observado no quadro 17.

QUADRO 17: TESTE COM O ESTUDO DE CASO DO ALUNO

Eventos existentes	Eventos identificados pela ferramenta
Aluno é cadastrado (F)	Alunos cadastram na divisão acadêmica (F)
Aluno consulta notas (F)	Alunos podem consultar suas notas (F)
Professor informa notas (F)	Professor informa notas de alunos (F)
Secretaria cadastra disciplinas (F)	Secretaria cadastra as disciplinas (F)
É hora de gerar boletim de notas (T)	É hora de gerar boletim de notas (T)

No estudo de caso apresentado no quadro 16, foram identificados todos os eventos, alguns num formato mais detalhado, totalizando 100% de acerto.

No quadro 18 pode-se observar um estudo de caso mais complexo que foi utilizado nos testes.

QUADRO 18: ESTUDO DE CASO DA INSTALADORA ELÉTRICA

A instaladora elétrica black-out está pretendendo informatizar o seu controle de serviços prestados. Ela atende várias obras onde são realizados serviços pelos seus funcionários. Semanalmente os funcionários enviam o papel com as informações sobre os serviços realizados. Ao final do mês é efetuado o pagamento do salário com o envio do contra-cheque. O cálculo do salário é feito a partir das horas trabalhadas no mês e o custo da hora. Mensalmente o gerente fixa o custo da hora para cada funcionário. Ao final do mês é emitido um extrato de cobrança para cada obra. Mensalmente o gerente fixa o preço da obra do serviço para cada obra. Semestralmente o gerente recebe um comparativo da evolução do preço médio das obras e custo médio da hora dos funcionários.

O resultado do estudo de caso apresentado no quadro 18, pode ser observado no quadro 19.

QUADRO 19: TESTE COM O ESTUDO DE CASO DA INSTALADORA

Eventos existentes	Eventos identificados pela ferramenta
ao final do mês é efetuado o pagamento do salário (F)	mês efetuado o pagamento do salário com o envio do contracheque (F)
é momento de calcular o salário (F)	salário feito a partir das horas trabalhadas no mês e o custo da hora (F)
é momento de emitir extrato de cobrança para cada obra (F)	mês emitido um extrato de cobrança para cada obra (F)
é hora de enviar o papel com informações sobre serviços realizados (T)	é hora de enviar o papel com as informações sobre os serviços realizados (T)
é hora de fixar o custo da hora para cada funcionário (T)	é hora de fixar o custo da hora para cada funcionário (T)
é hora de fixar o preço da obra do serviço para cada obra (T)	é hora de fixar o preço da obra do serviço para cada obra (T)
é hora de receber o comparativo da evolução dos preços (T)	é hora de receber um comparativo da evolução do preço médio das obras e custo médio da hora dos funcionários (T)
funcionário é cadastrado (F)	funcionário é cadastrado (F)
serviço é cadastrado (F)	
obra é cadastrada (F)	

No estudo de caso apresentado no quadro 18, os eventos implícitos não foram identificados. Os eventos de fluxo que foram identificados, ficaram com um formato um pouco extenso e seu início ficou um pouco confuso por existirem vários substantivos e vários verbos nas frases. Os eventos temporais foram identificados corretamente. Neste teste o acerto foi de 80%.

O quadro 20 mostra o percentual de acerto dos testes realizados. Percebe-se que em média o grau de acerto foi de 79,80%

QUADRO 20: PERCENTUAL DE ACERTO DOS TESTES

Estudo de caso	Percentual de acerto
Clube esportivo	83%
Consultório médico	85,70%
Aluno	100%
Instaladora Elétrica	80%
Média	87,18%

No anexo 2 encontra-se um estudo de caso mais complexo que foi testado pela ferramenta. Nele constam os eventos existentes no estudo de caso e os eventos identificados pela ferramenta.

5 CONCLUSÃO

A ferramenta desenvolvida tem a função de auxiliar o analista de sistemas na definição da lista de eventos existente na Análise Essencial de Sistemas. Nos testes realizados e que foram apresentados, o percentual de acerto foi de no mínimo 80% considerado por alguns autores um percentual bastante aceitável.

Pode-se também concluir que a técnica de Processamento de Linguagem Natural possui várias potencialidades na especificação de sistemas e que podem ser realizados vários trabalhos nesta área, como os trabalhos já desenvolvidos por Neves (1998), e Gomez (1999) que já foram observados.

Este trabalho pode ser considerado como uma expansão dos trabalhos desenvolvidos por Pereira (1992) e Neves (1998), unindo a Análise Essencial de Sistemas com o Processamento de Linguagem Natural para auxiliar na especificação de sistemas.

Outro aspecto que pode ser observado é o didático. Este trabalho pode auxiliar os alunos que cursam as disciplinas de análise de sistemas na aprendizagem da Análise Essencial de Sistemas.

A ferramenta desenvolvida possui algumas limitações, que podem ser observadas:

- a) a análise sintática das frases ficou um pouco limitada, devido a utilização da gramática livre de contexto e do conjunto de regras utilizado que abrange um conjunto pequeno de frases;
- b) na análise sintática, o *parser* não consegue retornar e escolher outra regra quando há algum impasse ao tentar provar uma regra, isso é considerado necessário por vários autores. Mas existem métodos que necessitam ser pesquisados onde as regras são organizadas para que não haja nenhum impasse ao se provar as regras no *parser*.

5.1 SUGESTÕES

Uma sugestão seria realizar a troca da gramática utilizada. Como já foi visto a gramática livre de contexto não é a mais adequada para ser utilizada na análise sintática das frases. Pode-se também permitir que a pessoa que utiliza a ferramenta crie suas próprias

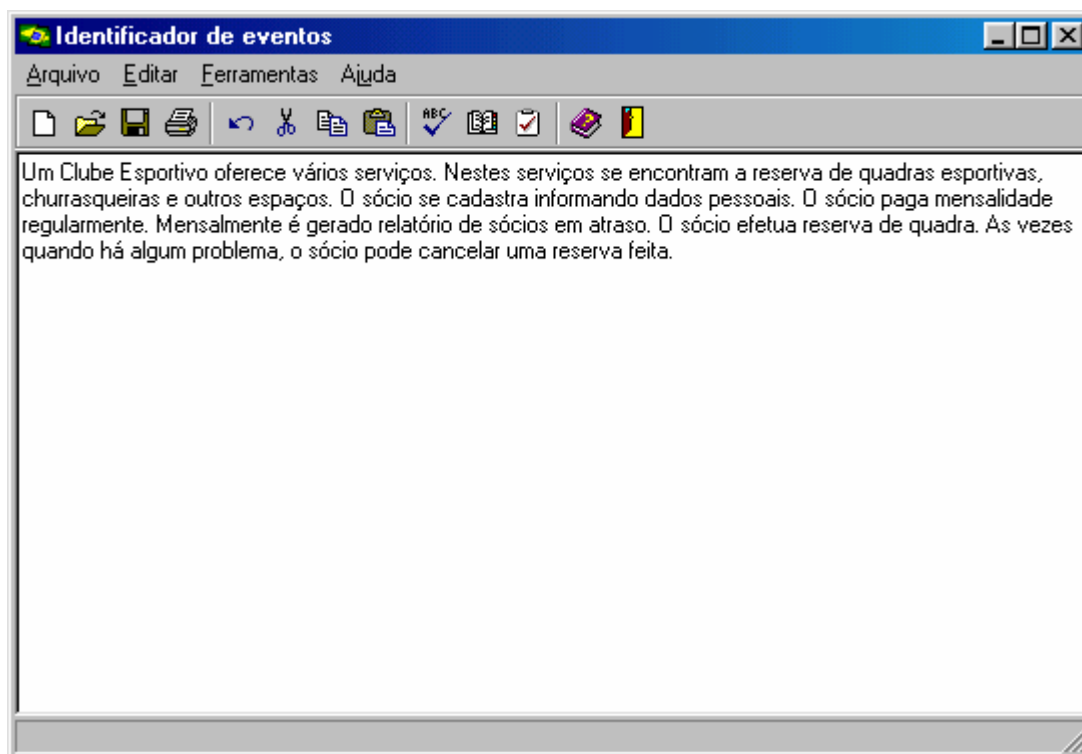
regras para a análise sintática, podendo desta forma o *parser* abranger um conjunto maior de frases.

Com a lista de eventos já identificada, pode-se realizar um trabalho que gere o diagrama de contexto e os DFD para cada evento, auxiliando ainda mais na especificação de sistemas.

Outra sugestão que pode ser realizada neste trabalho é implementar um melhor analisador léxico e um analisador semântico, para identificar o significado de cada frase.

ANEXO 1: AJUDA DA FERRAMENTA

Abaixo serão explicados os botões da tela principal.



Novo: inicia um novo documento;

Abrir: abre um arquivo no formato txt ou rtf;

Salvar: salva o arquivo que está sendo editado no formato rtf;

Imprimir: imprime o texto atual;

Desfazer: desfaz a última ação realizada;

Recortar: recorta o texto selecionado;

Copiar: copia o texto selecionado;

Colar: cola o conteúdo da área de transferência no texto;

Verificar Ortografia: verifica a ortografia do documento;

Dicionário: para fazer a manutenção no dicionário;

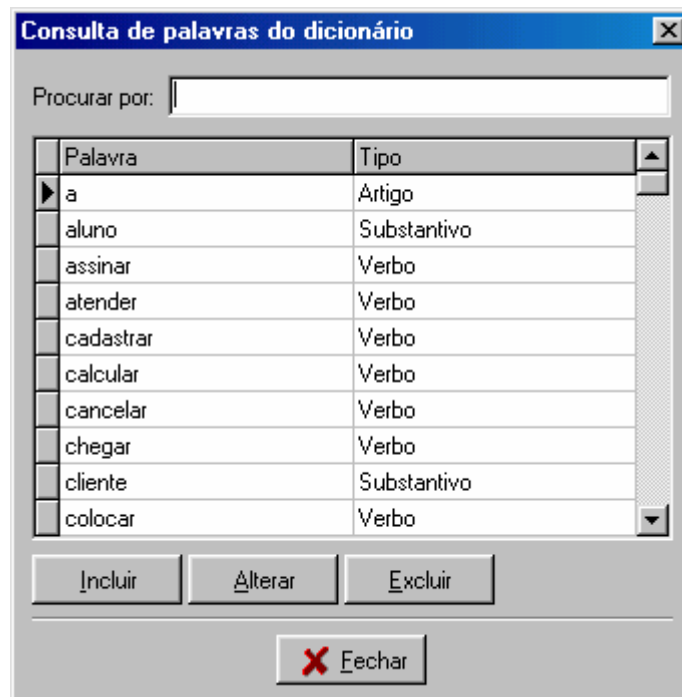
Lista de Eventos: faz a análise sintática das frases e localiza os eventos no texto;

Ajuda: ajuda do editor;

Sair: sai do editor.

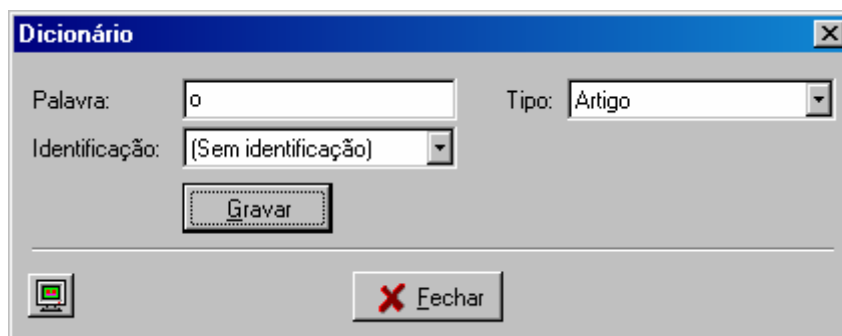
Cadastrar Palavras

A tela de cadastro de palavras possui uma tela de navegação que pode ser observada na figura abaixo.



Ao selecionar uma palavra já cadastrada, pode-se altera-la ou exclui-la, pressionando os respectivos botões. Se for necessário cadastrar uma nova palavra, é necessário pressionar o botão incluir. Ao se alterar uma palavra a tela que será exibida dependerá do tipo da palavra (verbo, substantivo, etc.).

Ao se incluir uma palavra é aberta a tela abaixo (formato padrão).



A palavra será digitada no campo palavra, seu tipo (verbo, advérbio, substantivo, artigo, etc) será escolhido no campo tipo, e sua identificação (sem identificação, tempo) será escolhida no campo identificação.

Quando uma palavra for do tipo verbo, a tela é modificada, para cadastrar as conjunções verbais, o novo formato da tela pode ser visto abaixo.

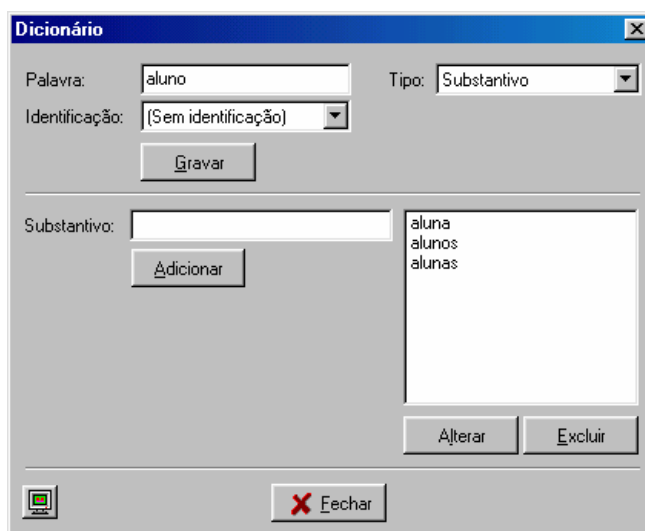
The screenshot shows a window titled "Dicionário" with a blue title bar. It contains the following elements:

- Input fields for "Palavra:" (containing "vender") and "Tipo:" (a dropdown menu showing "Verbo").
- An "Identificação:" dropdown menu showing "(Sem identificação)".
- A "Gravar" button.
- A "Modo" section with radio buttons for "Indicativo" (selected) and "Subjuntivo".
- A "Tempo" section with radio buttons for "Presente" (selected), "Pretérito", and "Futuro".
- A "Verbo:" input field.
- A list box on the right containing the conjugated forms: "vendo", "vendes", "vende", "vendemos", "vendeis", and "vendem".
- Buttons for "Adicionar", "Alterar", and "Excluir" below the list box.
- A "Fechar" button with a red 'X' icon at the bottom center.

Nesta tela é informado o modo (indicativo ou subjuntivo) e o tempo verbal (presente, pretérito ou futuro). Ao se cadastrar uma conjunção verbal, esta será exibida na caixa da direita. Para alterar ou excluir uma conjunção verbal, basta selecionar a palavra desejada na lista e pressionar o botão de exclusão ou alteração. Se for uma exclusão será necessário confirmar a exclusão. Se for uma alteração a palavra será exibida no campo verbo e o botão "adicionar" passa a ser identificado como "gravar".

Quando se terminar de cadastrar uma conjunção verbal e se deseja cadastrar outra palavra independente do tipo, basta pressionar o botão inferior esquerdo com o desenho de um monitor que a tela volta para o seu formato padrão.

Se a palavra que se deseja cadastrar for um substantivo, a tela será modificada, esta tela pode ser observada abaixo.



Ao se adicionar um substantivo, a palavra adicionada será exibida na tela da direita. o seu funcionamento é semelhante ao cadastro de conjunções verbais visto anteriormente.

Para obter ajuda sobre cada tipo de palavra, basta pressionar a tecla F1 quando a lista de tipo de palavras estiver selecionada. Os tipos de palavras existentes podem ser observadas abaixo:

Adjetivo

Adjetivo é a palavra que se junta a um substantivo para atribuir ao ser que este representa a posse de determinada qualidade, propriedade ou estado. A seguir pode-se observar alguns exemplos de adjetivo:

limpo, branco, preto, bonito, feliz.

Advérbio

A função do advérbio é exprimir circunstância e assim modificar o sentido da expressão a que se junta. O advérbio acrescenta-se geralmente a um verbo, mas há advérbios que podem afetar o significado de um adjetivo, de um advérbio ou de uma frase inteira. Nas frases abaixo, os advérbios aparecem sublinhados.

Era tarde demais.

Hoje as coisas são diferentes.

Artigo

Artigo é uma pequena palavra que se antepõe ao substantivo para individualizar o que ele representa, de maneira determinada ou de maneira vaga. O artigo indefinido deixa o indivíduo a distância; o definido aproxima-o, identificando-o como determinado ser. Abaixo pode-se observar exemplos de artigos.

De maneira determinada, é o artigo definido: o, a, os, as.

De maneira vaga, é o artigo indefinido: um, uma, uns, umas.

Conjunção

As conjunções servem para ligar orações ou ligar palavras ou grupo de palavras da mesma função. as conjunções podem ser simples, formadas de uma só palavra ou compostas, formadas de duas ou mais palavras. Abaixo pode-se observar exemplos de conjunções:

Simples: e, nem, ou, mas, porque.

Compostas: ainda que, assim como, uma vez que.

Preposição

Preposição é a palavra que serve de ligação entre duas outras, estabelecendo uma relação de dependência da segunda para com a primeira. Como exemplos de preposição temos: a, de, em, este, dela, no, na.

Pronome

Pronome é a palavra que serve para designar o ser sem o nomear, ou simplesmente representá-lo como elemento do discurso, ou para fazer referência a ele juntando-lhe uma idéia de posse ou de posição no espaço ou no tempo ou uma significação vaga, indefinida ou ainda para recordar uma oração, ou ser empregado para introduzir uma pergunta. Como exemplo de pronomes pode-se observar: eu, nós, me, lhe, te, ele.

Substantivo

Substantivo é a palavra que representa o ser; o ser que existe como realidade sensível ou é assim imaginado, ou o ser que existe como abstração: pessoas, animais, coisas, fenômenos,

atos, conceitos, qualidades. Como exemplos de substantivos temos: cliente, aluno, cidade, homem.

Verbo

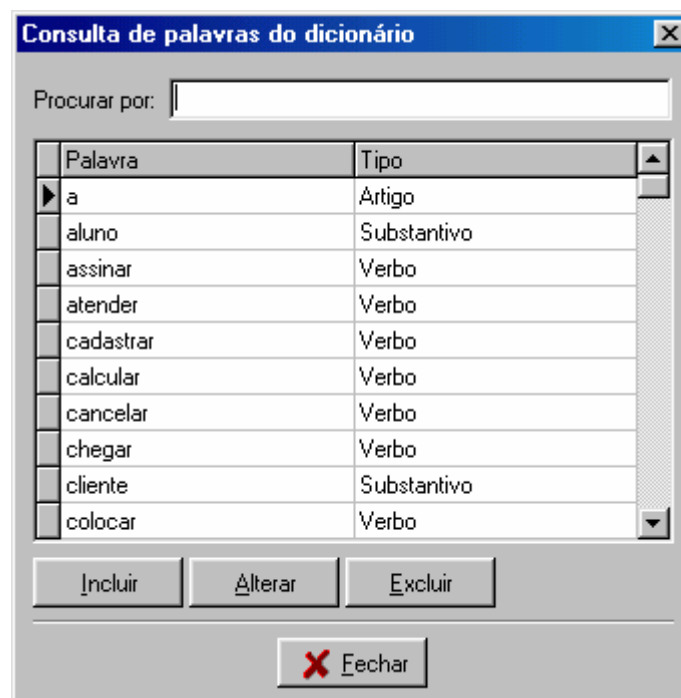
O verbo é a palavra mais importante do discurso. Exprime a ação, o estado, o fenômeno, mas exprime-os através de variados acidentes que são: o modo, o tempo, a pessoa, o número e a voz. Como exemplos de verbos pode-se observar: colocar, pedir, fazer.

-- Outro --

Outros tipos de palavras que não estão disponíveis na tela de cadastro.

Consultar Palavras

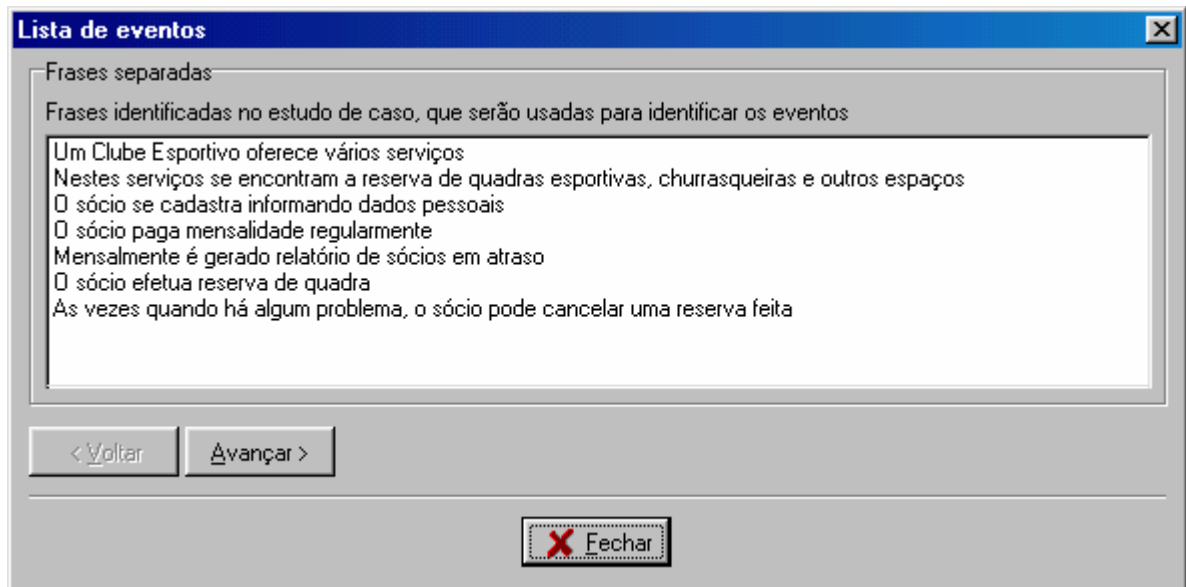
Para se consultar uma palavra, será exibida a tela abaixo



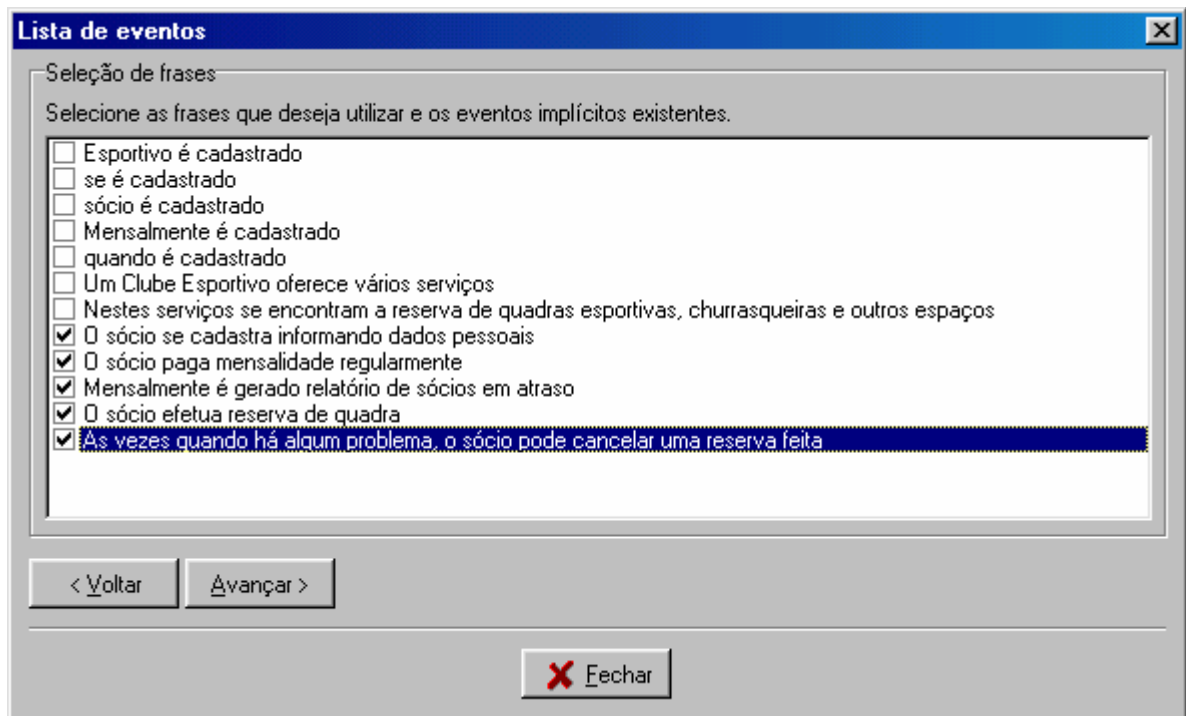
O campo procurar por é sensível ao contexto, ao se digitar algo neste campo a busca é feita na lista abaixo onde é exibida a palavra e seu tipo. Esta tela é usada também para incluir, alterar e excluir palavras do dicionário.

Lista de Eventos

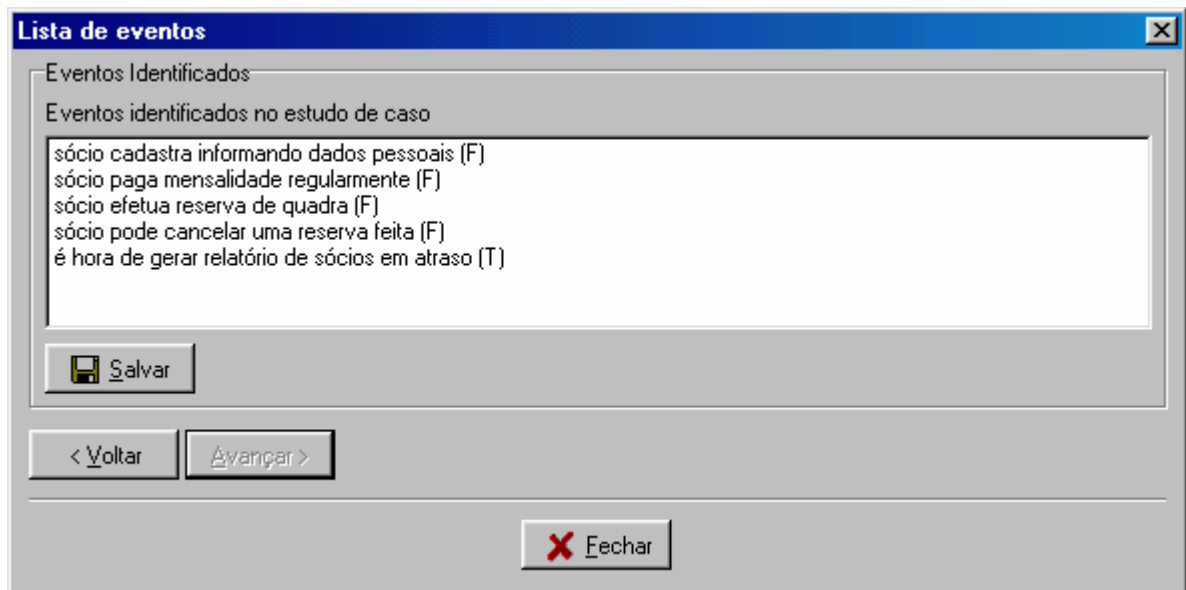
Este processo é dividido em três partes. A primeira parte são as frases identificadas no estudo de caso atual, como se pode observar na figura abaixo.



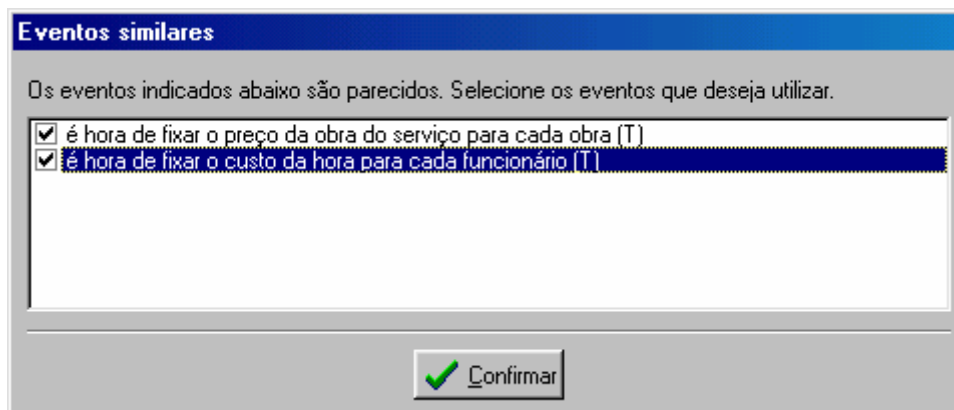
A segunda parte são as frases que serão utilizadas para identificar os eventos. Nesta tela, pode-se selecionar os eventos implícitos existentes no estudo de caso. Esta tela é apresentada em tamanho reduzido no exemplo abaixo.



A terceira parte apresenta a lista com os eventos identificados. Nesta parte final do processo, pode-se salvar a lista de eventos.



Quando dois eventos possuem muitas palavras iguais, é apresentada uma tela com os dois eventos para se escolher quais dos eventos deseja-se utilizar. Um exemplo desta tela pode ser observado na figura abaixo.



ANEXO 2: ESTUDO DE CASO

Uma entidade ambientalista resolveu decidir criar um banco de dados de informações de pesca para disponibilizar dados a pescadores, entidades de pesquisa e a comunidade em geral.

Foi realizado um Censo onde foram coletadas as seguintes informações:

- a) Dados sobre embarcações : proprietário, nome da embarcação, comprimento, inscrição na capitania dos portos, ano construção. O proprietário poderá ser uma pessoa física ou jurídica, no primeiro caso, serão armazenados seu nome, endereço, CPF, apelido, município, e no segundo caso, nome, endereço, CNPJ, município.
- b) As embarcações podem ser para pesca artesanal ou industrial. Quando for barco para pescaria industrial, devem ser armazenados dados sobre aparelhos, tais como: sonar, radar, navegador; capacidade de estocagem, se possui tanque de isca. No caso da pesca artesanal devem ser informados o tipo de material do casco e o tipo de propulsão do barco (motor, vela, remo, etc.).
- c) Para cada embarcação serão armazenados os diversos tipos de petrechos de pesca utilizados (rede, caniço, etc.) e também o tipo de conservação do pescado (refrigerado, sem refrigeração , etc.)

Cabe aos pesquisadores o cadastramento das espécies de animais encontrados na área marítima considerada pelo sistema (litoral Catarinense). Os pesquisadores constataram que seria mais fácil se os animais fossem também cadastrados pelos seus nomes vulgares, assim, para cada gênero de animal, foram classificadas as espécies e estas espécies, por sua vez, podem possuir vários nomes vulgares.

O gerente de fiscalização cadastra no sistema todas as portarias referentes aos períodos de defeso de uma determinada espécie, informando a espécie e o período em que a mesma não poderá ser capturada (data inicial e final).

Os fiscais vão informar os dados coletados sobre as pescas, que foram anotados nos pontos de desembarque, registrando a data, hora, embarcação e para cada espécie capturada,

será registrado seu nome vulgar, quantidade em quilos e petrecho utilizado para captura. Ao informar um nome vulgar, o sistema irá automaticamente associá-lo a espécie correspondente.

Os pescadores, empresas e demais interessados, terão a possibilidade de realizar consultas através da Internet sobre as portarias para poderem ter conhecimento dos períodos de defeso. Poderão parametrizar a consulta por espécie de animal, nome vulgar, região, período. A região será um conjunto de municípios do litoral catarinense. Estas regiões poderão ser criadas conforme a necessidade de novos agrupamentos.

Sobre as pescas realizadas (desembarques), o sistema disponibilizará somente estatísticas aos usuários em geral, informando, para cada espécie a quantidade total pescada por período (ano/mês).

Eventos existentes no estudo de caso:

- a) Proprietário é cadastrado (F)
- b) Pesquisador é cadastrado (F)
- c) Fiscal é cadastrado (F)
- d) Proprietário informa dados pessoais (F)
- e) Proprietário informa dados sobre embarcação (F)
- f) Gerente cadastra portaria referente ao período de defeso (F)
- g) Fiscal informa dados coletados sobre as pescas (F)
- h) Empresa realiza consultas pela internet (F)
- i) Região é cadastrada (F)

Eventos identificados pela ferramenta:

- a) proprietário é cadastrado (F)
- b) pesquisador é cadastrado (F)
- c) fiscal é cadastrado (F)
- d) região é cadastrado (F)
- e) pesquisadores constataram que seria mais fácil se os animais fossem também cadastrados pelos seus nomes vulgares, assim, para cada gênero de animal, foram

classificadas as espécies e estas espécies, por sua vez, podem possuir vários nomes vulgares (F)

- f) fiscalização cadastra no sistema todas as portarias referentes aos períodos de defeso de uma determinada espécie, informando a espécie e o período em que a mesma não poderá ser capturada (data inicial e final) (F)
- g) fiscais vão informar os dados coletados sobre as pescas, que foram anotados nos pontos de desembarque, registrando a data, hora, embarcação e para cada espécie capturada, será registrado seu nome vulgar, quantidade em quilos e petrecho utilizado para captura (F)
- h) empresas terão a possibilidade de realizar consultas através da Internet sobre as portarias para poderem ter conhecimento dos períodos de defeso (F)
- i) regiões poderão ser criadas conforme a necessidade de novos agrupamentos (F)

REFERÊNCIAS BIBLIOGRÁFICAS

ALLEN, James. **Natural language understanding**. Redwood City: The Benjamin/Cummings Publishing Company, 1995.

BARBIERI, Carlos. **Modelagem e administração de dados**. Rio de Janeiro: Infobook, 1994.

BELMIRO, João N. **Metodologias de desenvolvimento de sistemas**. São Paulo: Érica, 1993.

BELLIN, David; SUCHMAN, Susan. **Manual de desenvolvimento de sistemas estruturados**. São Paulo: Makron Books, 1993.

CUNHA, Celso. **Gramática do português contemporâneo**. Belo Horizonte: Bernardo Álvares S.A, 1972.

GOMEZ, Fernando; SEGAMI, Carlos; DELAUNE, Carl. A system for the semiautomatic generation of E-R models from natural language specifications. **Data & knowledge engineering**. [S.l.], n. 29, p. 57-81, 1999.

GRAHL, Everaldo Artur. **Publicação eletrônica** [mensagem pessoal]. Mensagem recebida por <egrahl@furb.br> em 09 fev. 2001.

HÜBNER, Jomi Fred. **Interface em linguagem natural**. 1992, 54 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau.

MANFRO, Paulo Renato. Usando linguagem natural para gerar um modelo de dados em IDEF1X, 1., 2000, Blumenau, **Artigo submetido à avaliação ao IX SEMINCO – Seminário de Computação**. Florianópolis: UFSC, 2000.

MCMENAMIN, Stephen M.; PALMER, John F. **Análise essencial de sistemas**. São Paulo: McGraw-Hill, 1991.

MENEZES, Paulo Blauth. **Linguagens formais e autômatos**. Porto Alegre: Sagra Luzzatto, 1998.

NEVES, Cloves das. **Protótipo de um identificador de verbos e substantivos da língua portuguesa**. 1998, 52 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau.

OLIVEIRA, Fabio Abreu Dias de. **Processamento de linguagem natural: princípios básicos e a implementação de um analisador sintático de sentenças da língua portuguesa**, Porto Alegre, dez. 1999. Disponível em: <<http://www.inf.ufrgs.br/proctpar/disc/cmp135/trabs/992/Parser/parser.html>>. Acesso em: 14 abr. 2001.

PEREIRA, Sérgio Luiz. **Um protótipo de ferramenta de apoio a análise essencial de sistemas**. 1992, 60 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau.

POMPILHO, S. **Análise essencial**. Rio de Janeiro: Infobook, 1994.

RABUSKE, Antonio Renato. **Inteligência artificial**. Florianópolis: Editora da UFSC, 1995.

RICH, Elaine. **Inteligência artificial**. São Paulo: McGraw-Hill, 1988

SAVADOVSKY, Pedro. **Introdução ao projeto de interfaces em linguagem natural**. São Paulo: SID Informática, 1989.

SCHILD, Herbert. **Inteligência artificial utilizando linguagem C**. São Paulo: McGraw-Hill, 1989.

SHILLER, Larry. **Excelência em software**. São Paulo: Makron Books, 1992.

SILVA, Maria Cecília de Souza e; KOCK, Ingedore Grunfeld Villaça. **Linguística aplicada ao português: sintaxe**. São Paulo: Cortez, 1986.

YOURDON, Edward. **Análise estruturada moderna**. Rio de Janeiro: Campus, 1990.

WINSTON, Patric Henry. **Inteligência artificial**. Rio de Janeiro: LTC-Livros Técnicos e Científicos, 1988.