

# Implementação do Sistema Falibras Baseada em Componentes de Software e suas Aplicações

Ivo Augusto Andrade Rocha Calado<sup>1</sup>, Marcelo Nunes Ribeiro<sup>1</sup>, Luis Cláudius Coradine<sup>1</sup>, Orivaldo de Lira Tavares<sup>2</sup>, João Roberto Santos Júnior<sup>1</sup>, Fabio Cunha de Albuquerque<sup>1</sup>, Arturo Hernandez Domingues<sup>1</sup>

<sup>1</sup> Instituto de Computação – Universidade Federal de Alagoas (UFAL)  
Campus A. C. Simões, Rod. BR 104 - N, Km 97, Tabuleiro dos Martins - Maceió - AL, CEP 57072-970. Fone 82 - 32141401

<sup>2</sup> Departamento de Informática – Universidade Federal do Espírito Santo (UFES)  
Campus de Goiabeiras, Av. Fernando Ferrari s/n, Goiabeiras –Vitória – ES, CEP 29060-970. Fone 27 - 33352654

lccoral@gmail.com coradine@tci.ufal.br tavares@inf.ufes.br txithihausen@gmail.com

**Resumo.** Este artigo apresenta uma nova implementação do sistema Falibras, um tradutor do português oral, ou em texto, para Libras, no formato gestual e animado, com uma estrutura baseada em componentes de software. Descreve, também, as facilidades encontradas para a melhoria do desenvolvimento do tradutor devido a essa nova implementação.

**Abstract.** This article presents a new implementation of the system Falibras, a translator of the verbal Portuguese, or in text, for Brazilian Sign Language, in the gestual and livened up format, with a structure based on software components. It describes, also, the easiness found for the improvement of the development of the translator due to this new implementation.

**Palavras-chave:** Falibras, tradução automática, componentes de software, tradutor Português-Libras

## 1. Introdução

O projeto Falibras<sup>1</sup> foi concebido, inicialmente, como um sistema que, ao captar a fala no microfone, exibe, no monitor de um computador, a tradução do que foi dito, em Libras, na forma gestual e animada, em tempo real. É um sistema interativo que busca auxiliar a comunicação entre ouvintes e surdos, com aplicações em projetos de educação especial [CORADINE 2002]. Nesse sentido, o Falibras poderá ser uma ferramenta de grande contribuição na aprendizagem, principalmente, de crianças surdas, podendo possibilitar a integração dessas nas escolas públicas e ajudar no aprendizado cognitivo. Trabalhos similares estão sendo realizados a partir da idéia do Falibras, tais como, por exemplo, o Tlibras [LIRA 2003].

A interpretação de palavras, expressões e orações simples, a partir de uma análise léxica, mapeando palavras de um texto em português para um conjunto de animações numa forma sentencial Libras, foram mostradas em [CORADINE 2004]. Pesquisas vêm sendo desenvolvidas no sentido de aprimorar o Falibras, como no caso da interface tradutora baseada em sintaxe (Falibras-TS), apresentada em [CORADINE 2006]. Uma interface tradutora baseada em memória de tradução (Falibras-MT), foi apresentada em [TAVARES 2005].

---

<sup>1</sup> Submetido e aprovado pelo CNPq e pela Fundação de Amparo à Pesquisa do Estado de Alagoas (FAPEAL).

A abordagem de desenvolvimento de software baseado em componentes tem como objetivo permitir a criação de componentes que possam ser reusados por outras aplicações. O processo de criação desses componentes deve seguir uma especificação bem definida provendo um alto nível de consistência e tornando-os pedaços de software independentes e configuráveis [VITHARANA 2003].

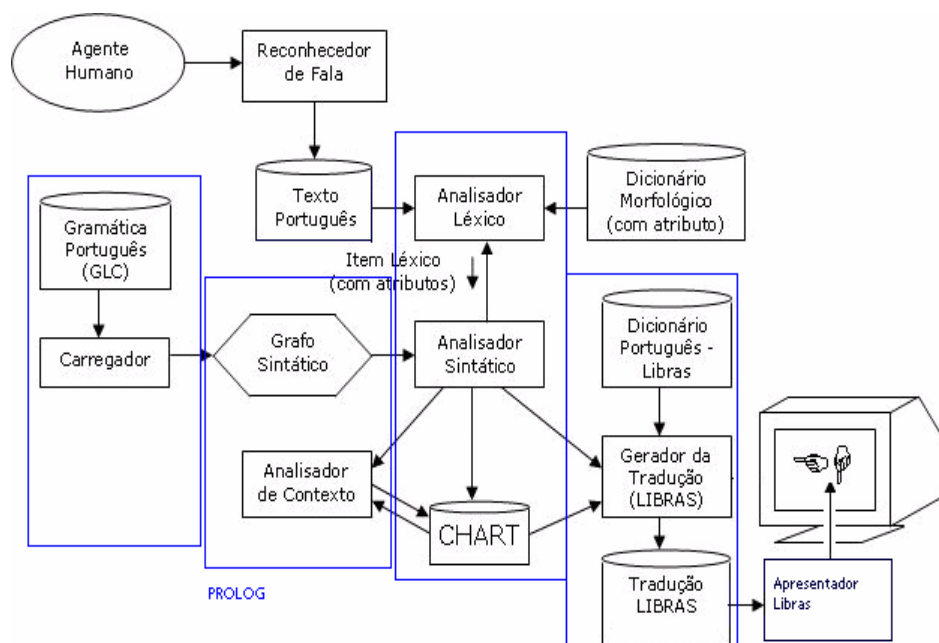
Neste trabalho, busca-se mostrar o aperfeiçoamento do tradutor automático, a partir da modularização do sistema baseado em componentes de software [CALADO 2007], visando uma melhor separação das funcionalidades do sistema provendo independência, de forma que atualizações no sistema possam ser feitas em diferentes contextos. Essa propriedade permitiu que fossem adicionados novos módulos para inclusão de novos recursos, como a redução de ambigüidade e transferência sintática [RIBEIRO 2007], por exemplo.

## 2. Materiais e Métodos

### 2.1. O Falibras-TS

A arquitetura do Falibras-TS, baseada em [PINHEIRO 2002], é apresentada na figura 1. Ela é constituída pelas seguintes funções:

1. Analisador léxico: reconhece os itens léxicos do texto fonte e seus atributos, tais como, a classe léxica, o gênero, o número, o tempo (se for verbo);
2. Analisador sintático: reconhece as estruturas sintáticas do texto fonte, a partir dos itens léxicos, usando um grafo sintático;
3. Analisador de contexto: verifica se os atributos das estruturas sintáticas reconhecidas estão corretos para o contexto em que aparecem;
4. Gerador da tradução em Libras: gera a tradução das estruturas sintáticas reconhecidas para Libras, usando um dicionário de traduções.



**Figura 1. Arquitetura do sistema Falibras-TS.**

## **2.2. Desenvolvimento de Software Baseado em Componentes**

O desenvolvimento baseado em componentes, ou a engenharia de software baseada em componentes emergiu no final da década de 90 como uma abordagem baseada no reuso para o desenvolvimento de sistemas de software, como aperfeiçoamento do desenvolvimento orientado a objetos, visando melhorar o reuso de software. As classes de objetos individuais se mostram muito detalhadas e específicas enquanto que os componentes, mais abstratos que elas, podem ser considerados provedores de serviços stand-alone [SOMMERVILLE 2003].

Visualizar um componente como um provedor de serviços enfatiza uma importante característica de um componente que é a independência entre os componentes que precisa obedecer as seguintes diretrizes [GILL 2006]:

- Os componentes publicam sua interface e todas as interações são feitas por meio dessa interface. A interface do componente é expressa em termos de operações parametrizadas e seu estado interno nunca é exposto;

- Como os componentes comunicam-se apenas por suas interfaces é possível mudar a lógica do funcionamento de um componente sem refletir no restante do sistema. Um exemplo desta possibilidade seria no caso de criar um novo tradutor utilizando uma outra tecnologia, mesmo que a lógica do funcionamento do tradutor esteja radicalmente mudada desde que a interface seja mantida nenhuma alteração ao restante do sistema seria necessária;

- Os componentes devem ser tão independentes quanto possíveis. Idealmente, um componente deve se manter isolado, de maneira que não precise de qualquer outro componente para entrar em operação. Um exemplo de onde esta propriedade é aplicada é no caso do componente tradutor que não precisa de nenhum outro componente para funcionar.

## **3. Resultados e Discussões**

Para proceder a implementação desejada, tomando como base a arquitetura anterior do Falibras e o levantamento do estado da arte relacionado ao desenvolvimento baseado em componentes, fez-se, primeiramente, uma análise sobre como o sistema poderia ser remodelado para atender a uma especificação que estivesse de acordo. Ou seja, de que maneira seria possível reestruturar o sistema em novos componentes que apresentassem funcionalidades bem distintas e independentes uma das outras.

Foram identificados inicialmente quatro componentes para fazerem parte da nova arquitetura: o componente tradutor, o componente de persistência, o componente de gerenciamento IO e o componente de gerência.

A seguir faz-se um detalhamento desses componentes desenvolvidos, apresentando a motivação por tal escolha, a estrutura do componente e os ganhos obtidos com essa nova arquitetura.

### **3.1. Componente Tradutor**

O objetivo primordial deste componente é gerenciar todo o processo de tradução de um termo em português para seu correspondente em LIBRAS (Língua Brasileira de Sinais).

Esse componente foi desenvolvido de forma de abstrair a origem dos dados de entrada e também sua saída, após o devido tratamento. Isso se fez interessante devido ao fato de poder haver várias fontes de entrada de dados, como a captação de voz de um usuário, um texto proveniente de uma entrada manual de dados ou até mesmo via uma API para uma chamada de método proveniente de outro elemento do sistema.

Tal fato mostrou-se semelhante para saída de dados, onde seria possível dar diferentes tratamentos ao texto traduzido, como a exibição da animação correspondente, ou na visão mais pura de um tradutor, o texto em Libras, correspondente ao texto de entrada (ambas as funcionalidades atendidas pelo Falibras).

Visto que a área de Processamento de Linguagem Natural (PLN) sempre está em constante mudança, novas técnicas de tradução são desenvolvidas com grande rapidez. Assim, o componente de tradução deveria permitir que novos módulos, que apresentariam implementações de técnicas mais refinadas de tradução, fossem adicionados a ele sem que isso interferisse em sua interface quanto à maneira como esse componente se comporta em relação aos elementos externos.

Dessa forma, foi possível a adição de módulos de eliminação de ambigüidade léxica utilizando classificação bayesiana e um módulo com o analisador sintático usando o algoritmo de Early [RIBEIRO 2007].

### **3.2. Componente de Persistência**

Esse componente é responsável por gerenciar a maneira como é efetivada a persistência dos dados no Falibras. Embora o processo de tradução seja realizado inteiramente no componente tradutor, ele necessita de uma base de dados contendo as regras de tradução, onde essas regras são carregadas da fonte de persistência via o componente de persistência, que fica responsável por efetuar a devida conexão com a base de dados e recuperar os dados gravados.

Esse componente representa papel semelhante em relação ao componente GUI (explicado a seguir), visto que também é responsável por disponibilizar a correta animação dado a palavra em LIBRAS.

### **3.3. Componente de gerenciamento GUI**

O principal objetivo desse componente é prover a fonte de dados para a tradução e o destino após o processo de tradução pelo componente de tradução. Juntamente com esse componente está integrada a interface gráfica do sistema Falibras. A partir desse componente, juntamente com a interface gráfica é possível definir se a entrada de dados é proveniente de um arquivo texto, de uma entrada manual ou da captação de voz, dando, portanto, o devido tratamento e o encaminhamento para a tradução.

Como mencionado anteriormente, esse componente também é responsável por definir o devido tratamento a saída, resultando do processamento do componente tradutor. A partir dele, é possível definir se o texto traduzido é exibido ao usuário ou utilizada a animação correspondente em Libras.

### **3.4. Componente de gerência**

Componente responsável por interligar todos os componentes anteriores. Por exemplo, a partir dele o texto em português é requisitado ao componente GUI, encaminhado ao componente Tradutor para que seja efetuada a sua devida tradução, a

qual é retornada ao componente Principal e encaminhada a componente que contenha as animações que são, finalmente encaminhadas ao componente GUI para sua correta exibição ao usuário.

### 3.5. A Estruturação do Falibras-TS Baseada em Componentes de Software

O Falibras-TS [CORADINE 2006] utiliza técnicas baseadas no processamento de linguagem natural, com gramáticas, usadas para processar as expressões. No reconhecedor, utilizou-se inicialmente o PROLOG, com integração com JAVA através da API **InterProlog**.

O aperfeiçoamento, a partir da modularização do sistema, usou o paradigma de orientação a objetos, com a notação UML (*Unified Modeling Language*) [CALADO 2006], e foi programado na linguagem C++. Essa nova versão do Falibras-TS incorpora tratamento de ambigüidade e transferência sintática [RIBEIRO 2007]. Esses problemas em tradução automática são motivos de diversos trabalhos científicos expostos na literatura, como mostra [MANNING 1999], [MARTINS 2005], etc.

O diagrama de classes proposto por [CALADO 2006] é apresentado na figura 2. Nesse diagrama, tem-se livre a implementação de cada componente. Também foram apresentadas as relações de dependência de dados entre os componentes. Essa relação de dependência é representada pela seta tracejada. O componente de origem da seta é o componente que necessita dos dados. De uma forma geral, são citadas algumas das funções desses elementos:

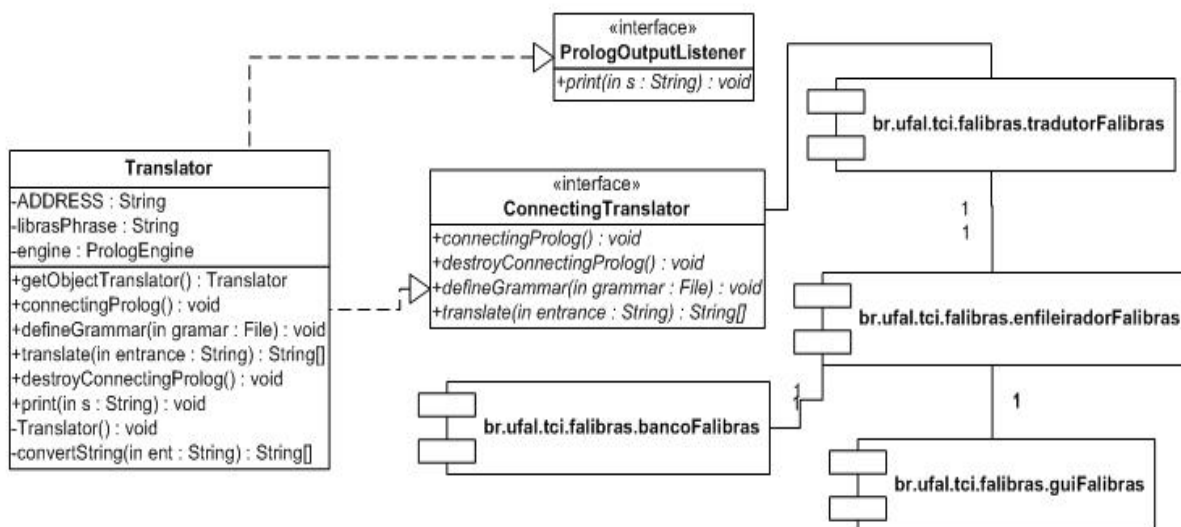


Figura 2 – diagrama de classes.

- **Componente Tradutor:** Onde se processa um período em português e a construção de um período em Libras de forma conveniente. Períodos resultantes serão encaminhados ao componente **Enfileirador**;
- **Componente Enfileirador:** É o responsável por reter animações prontas para exibição na tela. Recebe um período do **Tradutor** e, fazendo uso do banco, busca as animações, ou utiliza a dactilologia;

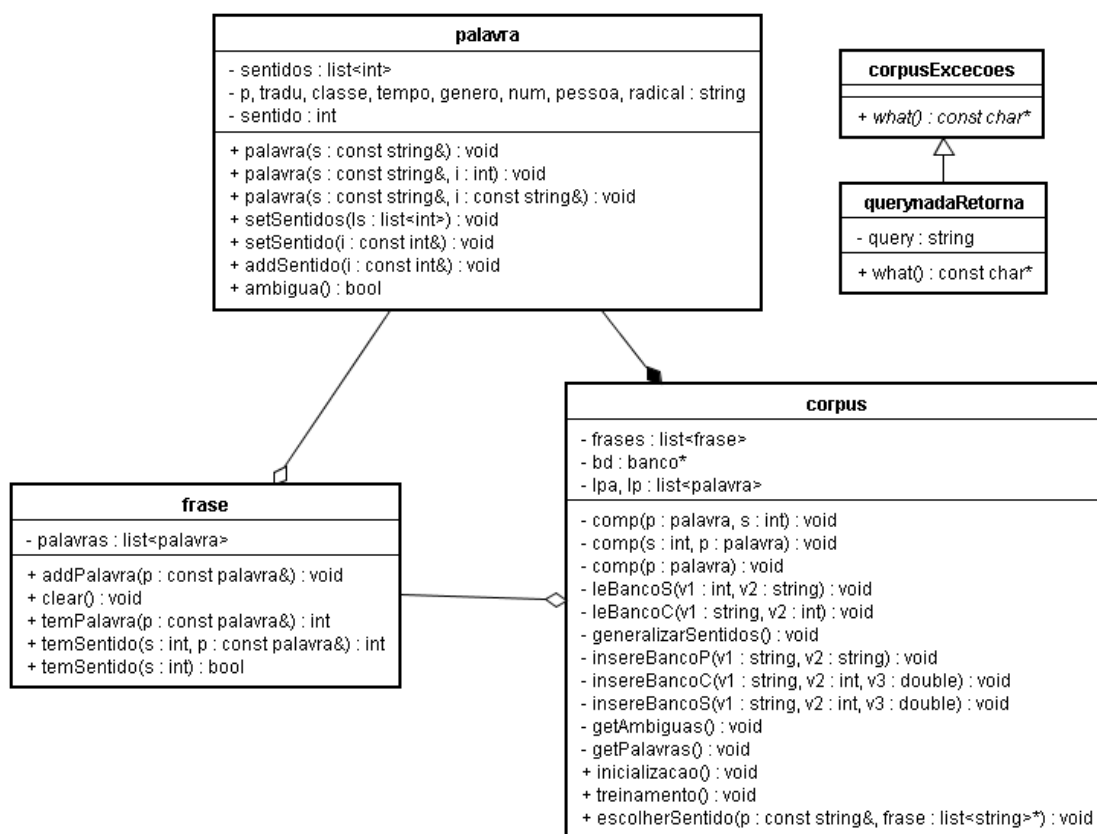
- **Componente GUI (Graphical User Interface):** É uma interface gráfica com usuário que recebe a entrada em português e exibe a saída com animações em Libras.

### 3.6. Exemplo de Novos Aportes devido a Estrutura em Componentes de Software

Utilizar a linguagem Prolog para criação do tradutor apresentou pontos positivos e negativos. Um ponto positivo é a facilidade de criação da gramática, porém por ser uma linguagem interpretada, apresenta problemas relativos ao desempenho, importante numa tradução automática.

Uma alternativa é a utilização da linguagem de programação C++, muito conhecida, pelo seu bom desempenho, na criação do tradutor. Nesse sentido, a construção de novos aportes do sistema já está sendo feita com essa linguagem, como por exemplo o componente de eliminação de ambigüidade.

O componente de eliminação de ambigüidade do sistema Falibras é formado por três classes que abstraem os conceitos de palavra, frase e corpus, que processam o algoritmo para eliminar ambigüidade utilizando classificação bayesiana [RIBEIRO 2007]. Na figura 3 tem-se o seu diagrama de classes.



**Figura 3. Diagrama de classes do componente de eliminação de ambigüidade.**

A classe **palavra** possui como atributos uma lista de todos os sentidos possíveis (representados por inteiros, que são a chave primária da tabela de sentidos do banco), um inteiro que referencia o sentido da palavra e uma seqüência de *strings* que contêm a palavra e informação léxica da mesma. A classe também fornece uma seqüência de métodos triviais.

A classe **frase** é trivial. Ela guarda uma lista de **palavra**, fornecendo algumas operações, tais como:

- **temPalavra**: Retorna um inteiro resultante da contagem das ocorrências da palavra **p** na frase.
- **temSentido**: A sobrecarga do método com dois parâmetros conta o número de ocorrências da palavra **p** com sentido **s** na frase.

A classe **corpus** guarda toda a complexidade do componente. Possui uma lista de **frase** (que representa o *corpus*), um ponteiro para conexão com o banco de dados e duas listas de **palavra** que guardam as palavras que ocorrem no *corpus*: **lpa** possui todas as palavras ambíguas e **lp** possui todas as palavras, seja ela ambígua ou não. Segue uma breve descrição dos métodos:

- **comp**: Possui três sobrecargas. A sobrecarga (*palavra p, int s*) computa  $C(v_j, s_k)$ , sendo  $v_j$  a palavra  $p$  e  $s_k$  o sentido  $s$ . A sobrecarga (*int s, palavra p*) computa  $C(s_k)$  e a sobrecarga (*palavra p*) computa  $C(w)$ .
- **inicialização**: Método que lê as frases no banco de dados, alimentando a lista de frases e desencadeando todo o processo de inicialização dos parâmetros.
- **generalizarSentidos**: Método chamado após a execução de **inicializacao**, que opera sobre a lista de frases, descobrindo quantos sentidos podem existir para cada palavra, guardando estes sentidos no banco de dados. Também inicializa **lpa** e **lp** através de **getAmbiguas** e **getPalavras**.
- **treinamento**: Método que executa o algoritmo de treinamento. As probabilidades  $Pr(v_j / s_k)$  e  $Pr(s_k)$  são guardadas nas tabelas **pr\_cond** e **pr\_sent**, respectivamente.
- **escolherSentido**: Método que recebe uma *string p* e uma lista de *string frase*, retornando uma **palavra** já classificada com o sentido escolhido pelo algoritmo.

Já que o sentido de cada palavra da frase pode ser dado pela frase em que ela está inserida, a análise léxica se simplifica a um simples percurso na frase.

#### 4. Conclusões

Este trabalho descreve o processo de aperfeiçoamento do Falibras-TS, como tradutor automático, a partir da reestruturação baseada em componentes de software. Verifica-se a grande vantagem dessa implementação no aporte de novas funcionais do Tradutor Falibras.

#### 5. Referências

- CALADO, I. A. A. R.; CORADINE, L. C. (2006): **Aperfeiçoamento do Tradutor Falibras com a Estruturação Baseada em Componentes de Software**. In: Relatório Final, projeto PIBIC (Iniciação Científica), IC/UFAL. Maceió, AL, julho/2006.
- CORADINE, L. C.; ALBUQUERQUE, F. C.; BRITO, P. H. S.; SILVA, R. L.; SILVA, T. F. L. (2002): **Sistema Falibras: Interpretação Animada, em LIBRAS, de Palavras e Expressões em Português**. In: III Congresso Ibero-Americano de Informática na Educação Especial – CIIEE 2002 - Demonstração. Fortaleza, CE, de 20 a 23 de Agosto de 2002. Anais do III Congresso Ibero-Americano de Informática na Educação Especial – CIIEE 2002, ago./2002.

CORADINE, L. C.; ALBUQUERQUE, F. C.; SILVA, A. F.; MADEIRO, J. M.; PEREIRA, M. C.; BRITO, P. H. S. (2004): **Interpretação de Pequenas Frases com Análise Léxica no Sistema Falibras**. In: III Fórum de Informática aplicada a Pessoas Portadoras de Necessidades Especiais (IV Congresso Brasileiro de Computação). Itajaí, SC, de 08 a 12 de Outubro de 2004. Anais do IV Congresso Brasileiro de Computação (CBcomp 2004), pp.678-682, out./2004. ISSN 1677-2822.

CORADINE, L. C.; TAVARES, O. L.; ALBUQUERQUE, F. C.; BREDAS, W.L.; SILVA, A.F.; SILVA, C.B.; RIBEIRO, M.N. (2006): **Interfaces Tradutoras do Sistema Falibras**. In: IV Congresso Ibero-Americano Sobre Tecnologias de Apoio a Portadores de Deficiência (Iberdiscap 2006). Vitória, ES, de 20 a 22 de fevereiro de 2006. Anais do IV Congresso Ibero-Americano Sobre Tecnologias de Apoio a Portadores de Deficiência (Iberdiscap 2006), pp., fev./2006. ISBN 84-96023-45-1.

GILL, N. S.(2006): **Importance of Software Component Characterization For Better Software Reusability**. ACM SIGSOFT Software Engineering Notes. Janeiro/2006.

LIRA, G. A. (2003): **O Impacto da Tecnologia na Educação e Inclusão Social da Pessoa Portadora de Deficiência Auditiva: Tradutor Digital Português X Língua Brasileira de Sinais - Tlibras**. BOLETIM TÉCNICO DO SENAC. Vol. 29, Nº 3, pp. 42-51, set./dez. 2003.

MANNING, C. D. and SCHÜTZE, H. (1999). **Foundations of Statistical Natural Language Processing**. The MIT Press, Cambridge, 1999.

MARTINS, R. T. ; PELIZZONI, J. M. ; HASEGAWA, R. (2005). **Para um Sistema de Tradução Semi-Automática Português-Libras**. In: III Workshop em Tecnologia da Informação e da Linguagem Humana – TIL 2005 (XXV Congresso da Sociedade Brasileira de Computação). São Leopoldo, RS, de 25 à 29 de julho de 2005. Anais do XXV Congresso da Sociedade Brasileira de Computação – SBC 2005, jul./2005.

PINHEIRO, E. (2002): **Um Ambiente de Apoio à Tradução de Texto**. Dissertação de Mestrado em Informática do PPGI/UFES. Vitória, ES, mar./2002.

RIBEIRO, M. N. (2007): **Tradução Automática Utilizando Arquitetura de Transferência**. In: Trabalho de Final de Curso, IC/UFAL. Maceió, AL, jan./2007.

SOMMERVILLE, I. (2003): **Engenharia de Software**. Makron Books, 6ª ed., Brasil, 2003.

TAVARES, O. L.; CORADINE, L. C.; BREDAS, W. L. (2005): **Falibras-MT – Autoria de tradutores automáticos de textos do português para Libras, na forma gestual animada: Uma abordagem com memória de tradução**. In: III Workshop em Tecnologia da Informação e da Linguagem Humana – TIL 2005 (XXV Congresso da Sociedade Brasileira de Computação). São Leopoldo, RS, de 25 à 29 de julho de 2005. Anais do XXV Congresso da Sociedade Brasileira de Computação – SBC 2005, pp. 2099-2107, jul./2005.

VITHARANA, P.(2003): **Risks and Challenges of Component-Based Software Development**. Communications of the ACM. Vol 46 nº8. Agosto/2003.