

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIA DA COMPUTAÇÃO – BACHARELADO

PROCESSAMENTO DE ROTAS POR UM SISTEMA
CONVERSOR TEXTO-FALA

ADRIANO FERNANDES STRINGARI

BLUMENAU
2010

2010/2-02

ADRIANO FERNANDES STRINGARI

PROCESSAMENTO DE ROTAS POR UM SISTEMA

CONVERSOR TEXTO-FALA

Trabalho de Conclusão de Curso submetido à Universidade Regional de Blumenau para a obtenção dos créditos na disciplina Trabalho de Conclusão de Curso II do curso de Ciência da Computação — Bacharelado.

Profa. Joyce Martins, Mestre – Orientadora

**BLUMENAU
2010**

2010/2-02

**PROCESSAMENTO DE ROTAS POR UM SISTEMA
CONVERSOR TEXTO-FALA**

Por

ADRIANO FERNANDES STRINGARI

Trabalho aprovado para obtenção dos créditos na disciplina de Trabalho de Conclusão de Curso II, pela banca examinadora formada por:

Presidente: _____
Profa. Joyce Martins, Mestre – Orientadora, FURB

Membro: _____
Prof. Roosevelt dos Santos Junior – FURB

Membro: _____
Prof. José Roque Voltolini da Silva – FURB

Blumenau, 13 de dezembro de 2010

Dedico este trabalho a todas as pessoas com
deficiência visual.

AGRADECIMENTOS

À minha família, que me formou como indivíduo e sempre me apoiou.

À minha namorada, Ângela, que me permitiu trabalhar no projeto durante todos os finais de semana, sacrificando o único tempo que posso ficar ao seu lado.

À minha orientadora, Joyce Martins, por ter me aceito como aluno orientando e me apoiado desde o início, antes mesmo da escolha do assunto proposto.

Às professoras Ana Andréia Karnopp Brüske e Valéria Contrucci de Oliveira Mailer, pela ajuda na língua alemã.

It is a mistake to think you can solve any major problems just with potatoes.

Douglas Noel Adams

RESUMO

Este trabalho apresenta a especificação e a implementação de um software para o processamento de rotas através de um sistema conversor texto-fala para a língua portuguesa, com a capacidade de síntese (pronúncia) de nomes de origem alemão-americano. O software proposto é de usabilidade direcionada a pessoas com deficiência visual. Para o processamento acústico foi utilizado o sintetizador MBROLA.

Palavras-chave: Sistema texto-fala. Síntese de texto. Análise linguística. Nomes alemão-americano. Google Maps. Google Directions API.

ABSTRACT

This work presents the specification and implementation of a software for the processing of routes through a text-to-speech system for the portuguese language, with the ability of synthesis (pronunciation) of German-American names. The proposed software has a usability directed to blind people. For acoustic processing it was used MBROLA synthesizer.

Key-words: Text-to-speech system. Text synthesizing. Linguistic analysis. German-American names. Google Maps. The Google Directions API.

LISTA DE ILUSTRAÇÕES

Quadro 1 – Fonemas da língua portuguesa.....	19
Quadro 2 – Dígrafos da língua portuguesa	21
Quadro 3 – Letras do alfabeto alemão transcritas para o português.....	22
Quadro 4 – Pronúncia dos grupos consonantais e das consoantes simples	24
Quadro 5 – Sobrenomes e seu significado quando derivados por características do terreno ...	27
Figura 1 – Obtendo uma rota no Google Maps	30
Quadro 6 – Formato padrão da URL utilizada pelo Google Directions API.....	30
Quadro 7 – Parâmetros da URL utilizada pelo Google Directions API.....	31
Quadro 8 – Exemplo de uma URL no formato utilizado pelo Google Directions API.....	31
Figura 2 – Tons de voz disponíveis no VoiceOver.....	33
Figura 3 – Diferentes tons de voz utilizados ao mesmo tempo no VoiceOver.....	34
Quadro 9 – Fonemas suportados pelo protótipo FurbTTS, extraídos do banco de fonemas BR3	38
Quadro 10 – Sequências consonantais e vogais que indicam que o nome é de origem alemã	39
Quadro 11 – Lista de encontros consonantais e vogais substituídos no software	40
Figura 4 – Diagrama de casos de uso	41
Quadro 12 – Detalhamento do caso de uso <code>Informa os endereços de origem e destino</code>	41
Quadro 13 – Detalhamento do caso de uso <code>Pesquisa rota</code>	42
Quadro 14 – Detalhamento do caso de uso <code>Informa texto</code>	42
Quadro 15 – Detalhamento do caso de uso <code>Solicita leitura</code>	43
Figura 5 – Diagrama de classes	44
Quadro 16 – Abreviaturas suportadas pelo software desenvolvido.....	46
Quadro 17 – Exemplos de tratamentos de abreviaturas.....	47
Quadro 18 – Regra de transcrição do fonema /d/	48
Quadro 19 – Definição da URL de requisição.....	48
Quadro 20 – Código de retorno apresentado no XML	49
Quadro 21 – Método <code>GetRouteXML</code>	50
Quadro 22 – Elemento <code>step</code> e seus campos no XML de retorno	51
Quadro 23 – Método <code>RemovePunctuation</code>	52
Quadro 24 – Método utilizado para identificar padrões em nomes.....	52

Quadro 25 – Método utilizado para alterar a escrita dos nomes.....	53
Figura 6 – Nomes alemão-americano alterados no método <code>Modify</code>	54
Quadro 26 – Método utilizado para processar letras e palavras	55
Figura 7 – Interface do software	56
Figura 8 – Resultado da pesquisa entre os pontos A e B.....	57
Figura 9 – Exemplo de mensagem informativa ao realizar uma pesquisa.....	58
Figura 10 – Mensagem para endereço de destino vazio ao realizar pesquisa de rota.....	58
Figura 11 – Leitor da tela.....	59
Figura 12 – Tela de atalhos.....	60
Figura 13 – Sobre o TTS Rota	61
Figura 14 – Mapa da rota entre os pontos A e B	62
Quadro 27 – XML de retorno do Google Directions API	69

LISTA DE SIGLAS

API – *Application Programming Interface*

ccTLD – *country code Top-Level Domain*

EBCT – Empresa Brasileira de Correios e Telégrafos

FAB – Força Aérea Brasileira

FURB – Universidade Regional de Blumenau

GPS – *Global Positioning System*

HTML – *HyperText Markup Language*

HTTP – *HyperText Transfer Protocol*

IDE – *Integrated Development Environment*

JSON – *JavaScript Object Notation*

MT – Mato Grosso

SENAC – Serviço Nacional de Aprendizagem Comercial

RF – Requisito Funcional

RNF – Requisito Não-Funcional

UC – *Use Case*

UML – *Unified Modeling Language*

URL – *Uniform Resource Locator*

XML – *eXtensible Markup Language*

SUMÁRIO

1 INTRODUÇÃO	13
1.1 OBJETIVOS DO TRABALHO	14
1.2 ESTRUTURA DO TRABALHO	14
2 FUNDAMENTAÇÃO TEÓRICA	16
2.1 SISTEMAS DE CONVERSÃO TEXTO-FALA	16
2.2 APARELHO FONADOR.....	17
2.3 GRAMÁTICA DA LÍNGUA PORTUGUESA.....	18
2.3.1 Fonemas	18
2.3.1.1 Encontros consonantais e dígrafos.....	20
2.3.2 Abreviaturas e siglas	21
2.4 GRAMÁTICA DA LÍNGUA ALEMÃ.....	22
2.4.1 Alfabeto	22
2.4.2 Fonemas	23
2.4.3 Nomes alemão-americano	26
2.4.3.1 Nomes: significado e origem	26
2.4.3.2 Sobrenomes: significado e origem.....	26
2.4.3.3 Americanização de nomes alemães	28
2.5 FURBTTS.....	28
2.6 GOOGLE MAPS	29
2.6.1 Google Directions API.....	30
2.7 TRABALHOS CORRELATOS	31
2.7.1 Sistema de conversão texto-fala para a língua portuguesa utilizando a abordagem de síntese por regras	32
2.7.2 Jaws for Windows	32
2.7.3 Voice Over	33
2.7.4 Navegadores GPS.....	34
3 DESENVOLVIMENTO DO SOFTWARE	36
3.1 REQUISITOS DO SOFTWARE A SER DESENVOLVIDO	36
3.2 ESPECIFICAÇÃO	37
3.2.1 Lista de fonemas.....	37
3.2.2 Lista de padrões de nomes alemão-americano.....	38

3.2.3 Lista de encontros consonantais e vogais.....	40
3.2.4 Diagrama de casos de uso	41
3.2.5 Diagrama de classes	43
3.3 IMPLEMENTAÇÃO.....	44
3.3.1 Técnicas e ferramentas utilizadas.....	44
3.3.2 Pré-processamento do texto	45
3.3.3 Encontros consonantais	47
3.3.4 Obtenção das rotas	48
3.3.5 Identificação dos nomes	51
3.3.6 Alteração dos nomes para o português.....	53
3.3.7 Interface do software.....	54
3.3.8 Operacionalidade da implementação.....	55
3.4 RESULTADOS E DISCUSSÃO.....	62
4 CONCLUSÕES	64
4.1 EXTENSÕES	65
REFERÊNCIAS BIBLIOGRÁFICAS.....	66
APÊNDICE A – Exemplo de XML retornado pelo Google Directions API.....	69

1 INTRODUÇÃO

Sabe-se que tarefas consideradas simples, como leitura e locomoção, tornam-se um tanto quanto complicadas para deficientes visuais ou com deficiência severa. Pessoas cegas podem ler e escrever utilizando o método Braille (INSTITUTO BENJAMIN CONSTANT, 2005). Porém, segundo Borges (2002, p. 2), raramente o método Braille é entendido por pessoas que enxergam, de tal forma que as pessoas com deficiência visual severa ficam isoladas num gueto cultural: um cego só escreve em Braille para outro cego ler.

Considerando estes usuários, o acesso a *sites* com notícias, por exemplo, é uma tarefa difícil sem a ajuda de um vidente, porém, não impossível. Mas, como é possível esta interação entre o usuário deficiente visual e a máquina, já que a principal forma de interação homem-máquina é a escrita? Hoje existem algumas soluções que possibilitam este tipo de atividade. Como exemplo, têm-se os sistemas de conversão texto-fala (*text-to-speech*), onde o software reproduz, da forma mais humana possível, o texto apresentado na tela. Segundo Oechsler (2009, p. 13), “alguns destes sistemas já podem ser encontrados gratuitamente na Internet, inclusive para o sistema operacional Windows”. Sistemas como estes podem e já são utilizados em vários aplicativos, como no caso dos celulares, onde são utilizados para a síntese de mensagens, e em navegadores *Global Positioning System* (GPS), para a síntese de rotas. Ainda, alguns sistemas operacionais mais atuais, como o Mac OS X v10.5 Leopard, também já vêm acompanhados de um sistema de conversão texto-fala, visando principalmente suas funcionalidades básicas e aplicativos nativos.

Diante do exposto, propõe-se disponibilizar um software para sintetizar rotas através de conversão texto-fala. Para tanto, foi aprimorado o protótipo descrito por Oechsler (2009), o FurbTTS, que é um sistema que efetua a conversão texto-fala a partir do processamento de texto com vocabulário irrestrito escrito em língua portuguesa. O software aqui proposto é uma extensão do FurbTTS e tem como entrada uma localização origem e uma localização destino e como saída a síntese das possíveis rotas definidas no Google Maps (GOOGLE, 2010a). O problema é que as rotas obtidas do Google Maps não estão em formato de texto padrão. Assim é necessário extrair informações do Google Directions API, que indicam as possíveis rotas entre a origem e o destino, obrigando então a manipulação para que estas rotas possam ser processadas pelo FurbTTS.

Faz-se necessário também efetuar um tratamento mais adequado do texto a ser sintetizado, conforme indica Oechsler (2009, p. 73). Segundo Franzen (2002, p. 13), a

conversão de texto para unidades linguísticas equivalentes aos sons não é direta e varia de um idioma para outro. E, se tratando de localizações, é fundamental que sejam implementadas regras para que o sistema de síntese possa reproduzi-las da forma mais clara possível.

1.1 OBJETIVOS DO TRABALHO

O objetivo deste trabalho é estender o FurbTTS para processar rotas obtidas através do Google Maps.

Os objetivos específicos do trabalho são:

- a) permitir que o usuário informe, em forma de texto, uma localização origem e uma localização destino (rua, bairro, cidade, estado, país);
- b) validar a entrada, sintetizando-a;
- c) obter, manipular e sintetizar a(s) rota(s) definida(s) no Google Directions *Application Programming Interface* (API);
- d) efetuar um tratamento mais adequado do texto de entrada, sintetizando corretamente os nomes de origem alemão-americano.

1.2 ESTRUTURA DO TRABALHO

Já foram apresentados a introdução e os objetivos do trabalho. No capítulo dois é descrita a fundamentação teórica. Nele é explicado o que são sistemas de conversão texto-fala, bem como apresentada uma breve introdução sobre o aparelho fonador. Ainda no capítulo dois são explanadas algumas características gramaticais da língua portuguesa e da língua alemã, focando principalmente em nomes de origem alemã (alemão-americano). Também são apresentados o protótipo FurbTTS, descrito por Oechsler (2009), e os principais conceitos da ferramenta Google Maps. Por fim, são descritos os trabalhos correlatos.

No capítulo três são descritos a especificação, a implementação e o funcionamento do sistema proposto. Este capítulo trata de cada etapa implementada, dos algoritmos e das técnicas utilizadas e dos resultados do processamento.

O capítulo quatro traz as conclusões provenientes do desenvolvimento deste trabalho,

bem como as possíveis extensões do mesmo.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo é exposta uma descrição dos assuntos necessários para o desenvolvimento do software proposto. Na primeira seção tem-se uma explicação sobre os sistemas de conversão texto-fala. Na segunda seção, o aparelho fonador e sua importância são brevemente explanados. A terceira trata sobre a gramática da língua portuguesa, com ênfase na fonética e nos encontros consonantais. Na quarta seção deste capítulo tem-se uma breve introdução à gramática da língua alemã e um estudo mais aprofundado dos nomes e sobrenomes de origem alemã. Na quinta seção é apresentado o protótipo descrito por Oechsler (2009), o FurbTTS. A sexta seção apresenta a ferramenta Google Maps (GOOGLE, 2010a), utilizada para a obtenção de rotas. Por fim, a sétima seção traz os trabalhos correlatos.

2.1 SISTEMAS DE CONVERSÃO TEXTO-FALA

Segundo Gomes (1998, p. 5), os primeiros sistemas para sintetizar a fala surgiram a partir de 1992. Ainda segundo Gomes (1998, p. 4), os sistemas de conversão texto-fala têm por função gerar um sinal de fala a partir de um texto genérico. Isso é, um sistema de conversão texto-fala recebe como entrada um texto e efetua a síntese do mesmo, tornando-se, assim, diferente de outras máquinas como gravadores ou equipamentos que produzem sons a partir da concatenação de palavras ou sentenças (DUTOIT, 1996 apud FRANZEN, 2002, p. 15).

As etapas do processo de conversão texto-fala incluem: pré-processamento, análise linguística, transcrição fonética, processamento prosódico e síntese.

Oechsler (2009, p. 20) afirma que “um sistema de conversão texto-fala deve primeiramente eliminar sentenças estranhas e formatar o texto de acordo com o vocabulário suportado”, efetuando o pré-processamento do texto de entrada. Abreviações, siglas e símbolos devem ser substituídos para suas formas extensas. Ainda, deve ser realizada uma análise sintática do texto, verificando se as palavras encontradas pertencem à linguagem que está sendo processada (OECHSLER, 2009, p. 21).

O analisador linguístico é responsável por definir as classes gramaticais das palavras encontradas no texto de entrada. É com o resultado desta classificação que o programa irá

definir a pronúncia de cada palavra (OECHSLER, 2009, p. 21).

A transcrição fonética consiste em fazer a transformação da sequência ortográfica em uma cadeia de símbolos que represente a sequência de sons que compõe cada uma das palavras do texto (SIMÕES, 1999, p. 52).

Segundo Oechsler (2009, p. 23), o processador prosódico é responsável por determinar as durações segmentais dos termos transcritos, definindo grupos interrogativos, imperativos e afirmativos. O processador determina características como entonação, duração do segmento e intensidade sonora.

2.2 APARELHO FONADOR

“A utilização da língua pelo indivíduo denomina-se fala. A fala nasce da inelutável necessidade humana de comunicação” (CEGALLA, 1985, p. 17). Utiliza-se a fala para comunicar informação de um locutor para um ou mais ouvintes. A mensagem é formulada para então ser transmitida através da voz.

Os sons da fala são produzidos pelo aparelho fonador (CEGALLA, 1985, p. 4). Ostermann Filho (2002, p. 14) complementa afirmando que “a fala pode ser dividida em segmentos de som, que compartilham algumas propriedades acústicas e articulatórias comuns umas com as outras, por um curto intervalo de tempo”.

O inglês faz um uso do sistema articulatório e exige um esforço muscular e uma movimentação de seus órgãos, especialmente da língua, significativamente diferentes, quando comparado à fonética do português. A articulação de muitos sons do inglês bem como de outras línguas de origem germânica, pode ser facilmente classificada como sendo de natureza difícil. Isto está provavelmente relacionado ao fato de que o inglês é rico na ocorrência de consoantes enquanto que o português é abundante na ocorrência de vogais e combinações de vogais (ditongos e tritongos). (SCHÜTZ, 2008).

Segundo Ostermann Filho (2002, p. 14), para cada som há um posicionamento dos articuladores do trato vocal (cordas vocais, língua, lábios, dentes, palato e maxilar), sendo que a produção da fala pode ser vista como uma operação de filtragem em que uma fonte de som excita o filtro do trato vocal.

O uso que o ser humano faz de seu aparelho articulatório para comunicar-se varia consideravelmente de idioma para idioma, o que explica o porquê de ser na pronúncia que a interferência entre duas línguas se torna mais evidente e é mais crítica. A interferência fonológica da língua materna na língua estrangeira que se aprende, na maioria dos casos permanece para sempre, mesmo com pessoas que já adquiriram pleno domínio sobre o vocabulário e a gramática da língua estrangeira. (SCHÜTZ, 2008).

É possível identificar na língua estrangeira sons quase idênticos aos da língua materna. Assim, estudantes de idiomas normalmente baseiam sua pronúncia num modelo acústico resultante de pares de sons semelhantes das duas línguas, em vez de baseá-la nos sons específicos da língua estrangeira como se fosse o aprendizado da língua materna (FLEGE, 1991 apud SCHÜTZ, 2008). Diante disso, percebe-se a importância de uma pronúncia clara e correta de palavras de origem estrangeira para deficientes visuais.

2.3 GRAMÁTICA DA LÍNGUA PORTUGUESA

Oechsler (2009, p. 17) afirma que os sistemas de conversão texto-fala devem compreender as normas da língua para terem um bom desempenho. Ainda, apesar de ser uma forma irrestrita de comunicação, a linguagem natural deve seguir formas e regras. Assim, segundo Cegalla (1985, p. 16), “A gramática [...] aponta normas para a correta utilização oral e escrita do idioma [...]” e é dividida em cinco partes distintas:

- a) fonética, que estuda os sons da fala (CEGALLA, 1985, p. 17);
- b) morfologia, que estuda as diversas classes de palavras isoladamente, analisando estrutura, formação, flexão e propriedade (CEGALLA, 1985, p. 17);
- c) sintaxe, que estuda a palavra com relação a outras, ou seja, estuda a estrutura da frase, quer completa ou quer incompleta (ALMEIDA, 1969, p. 24);
- d) semântica, que estuda o significado das palavras (CEGALLA, 1985, p. 17);
- e) estilística, que visa o lado estético da atividade linguística, em oposição ao aspecto intelectual, tratando, basicamente, dos processos expressivos próprios para suggestionar, despertar o sentimento estético e a emoção (CEGALLA, 1985, p. 17).

Segundo Oechsler (2009, p. 17), para um sistema conversor texto-fala a parte mais importante do estudo da gramática está na fonética. Na fonética são tratadas questões importantes como: fonemas, vogais, semivogais e consoantes, divisão silábica, encontros vocálicos e consonantais.

2.3.1 Fonemas

Segundo Cegalla (1985, p. 3), fonemas são sons elementares da fala que, articulados e

combinados, formam as sílabas, os vocabulários e as frases. Para isso seria ideal que cada fonema correspondesse a uma só letra e vice-versa. Porém isso não acontece, pois o sistema ortográfico da língua portuguesa não é rigorosamente fonético, mas ainda está preso à origem etimológica das palavras (CEGALLA, 1997, p. 21).

Cegalla (1985, p. 6) descreve que os fonemas da língua portuguesa são classificados em: vogais, fonemas ou sons laríngicos que chegam livremente ao exterior sem fazer ruído; semivogais, fonemas /i/ e /u/ átonos que se unem a uma vogal, formando uma só sílaba; e consoantes, ruídos originários da resistência que os órgãos bucais opõem à corrente de ar. “A vogal é o elemento básico, suficiente e indispensável para a formação da sílaba na língua portuguesa. Consoantes e semivogais são fonemas dependentes, só podendo formar sílabas com o concurso das vogais” (CEGALLA, 1985, p. 6).

Segundo Oechsler (2009, p. 18), o português utiliza trinta e quatro fonemas, sendo treze vogais, dezenove consoantes e duas semivogais, que estão representados no Quadro 1.

	fonema ¹	características fonéticas	exemplos ²
vogais	/á/	aberta, frontal, oral, não arredondada	<i>átomo, arte</i>
	/â/	semiaberta, central, oral, não arredondada	<i>pano, ramo, lanho</i>
	/ã/	semiaberta, central, nasal, não arredondada	<i>antes, amplo, maçã, âmbito</i>
	/é/	semiaberta, frontal, oral, não arredondada	<i>métrica, peça</i>
	/ê/	semifechada, frontal, oral, não arredondada	<i>medo, pêssego</i>
	/ẽ/	semifechada, frontal, nasal, não arredondada	<i>sempre, êmbolo, centro, concêntrico, têm, também³</i>
	/ó/	semiaberta, posterior, oral, arredondada	<i>ótima, ova</i>
	/ô/	semifechada, posterior, oral, arredondada	<i>rolha, avô</i>
	/õ/	semifechada, posterior, nasal, arredondada	<i>ombro, ontem, cômputo, cônsul</i>
	/i/	fechada, frontal, oral, não arredondada	<i>item, silvícola</i>
	/ĩ/	fechada, frontal, nasal, não arredondada	<i>simples, símbolo, tinta, síncrono</i>
	/u/	fechada, posterior, oral, arredondada	<i>Uva, útero</i>
semivogais	/y/	oral, palatal, sonora	<i>uivo, mãe, área, têm, também³</i>
	/w/	oral, velar, sonora	<i>automático, móvel, pão, falam⁴</i>
¹ Foi utilizado um conjunto de grafemas adaptado à realidade brasileira, que não corresponde integralmente ao Alfabeto Fonético Internacional. ² Em ortografia oficial do português. ³ Os grafemas em negrito nas palavras <i>têm</i> e <i>também</i> representam o encontro vocálico da vogal /ẽ/ com a semivogal /y/. ⁴ Os grafemas em negrito na palavra <i>falam</i> representam o encontro vocálico da vogal /ã/ com a semivogal /w/.			

Fonte: adaptado de Manosso (2008 apud OECHSLER, 2009, p. 18).

Quadro 1 – Fonemas da língua portuguesa

	fonema ¹	características fonéticas	exemplos ²
consoantes	/m/	nasal, sonora, bilabial	Marca
	/n/	nasal, sonora, alveolar	Nervo
	/ɲ/	nasal, sonora, palatal	Arranhado
	/b/	oral, oclusiva, bilabial, sonora	Barco
	/p/	oral, oclusiva, bilabial, surda	Pato
	/d/	oral, oclusiva, linguodental, sonora	Data
	/t/	oral, oclusiva, linguodental, surda	Telha
	/g/	oral, oclusiva, velar, sonora	Gato
	/k/	oral, oclusiva, velar, surda	carro, quanto
	/v/	oral, fricativa, labiodental, sonora	Vento
	/f/	oral, fricativa, labiodental, surda	Farelo
	/z/	oral, fricativa, alveolar, sonora	zero, casa, exalar
	/s/	oral, fricativa, alveolar, surda	seta, cebola, espesso, excesso, açúcar, auxílio, asceta
	/j/	oral, fricativa, pós-alveolar, sonora	gelo, jarro
	/x/	oral, fricativa, pós-alveolar, surda	xarope, chuva
	/r/	oral, vibrante, sonora, uvular	rato, carroça
	/r/	oral, vibrante, sonora, alveolar	Varição
/ʎ/	oral, lateral aproximante, sonora, palatal	Cavalheiro	
/l/	oral, lateral aproximante, sonora, alveolar	Luz	
¹	Foi utilizado um conjunto de grafemas adaptado à realidade brasileira, que não corresponde integralmente ao Alfabeto Fonético Internacional.		
²	Em ortografia oficial do português.		
³	Os grafemas em negrito nas palavras <i>têm</i> e <i>também</i> representam o encontro vocálico da vogal /ê/ com a semivogal /y/.		
⁴	Os grafemas em negrito na palavra <i>falam</i> representam o encontro vocálico da vogal /ã/ com a semivogal /w/.		

Fonte: adaptado de Manosso (2008 apud OECHSLER, 2009, p. 18).

Quadro 1 – Fonemas da língua portuguesa (continuação)

2.3.1.1 Encontros consonantais e dígrafos

Encontro consonantal é a sequência de dois ou mais fonemas consonânticos num vocabulário, podendo ocorrer na mesma sílaba ou em sílabas diferentes. Quando na mesma sílaba, são encontros consonantais inseparáveis, mais frequentemente formados de consoante seguida da letra /f/ ou da letra /r/. Quando em sílabas diferentes, são encontros consonantais separáveis, ocorrendo sempre no interior das palavras e geralmente formados de duas consoantes (CEGALLA, 1985, p. 12).

Dígrafo é o grupo de duas letras representando um só fonema. Os dígrafos representam consoantes (/ch/: chapéu, cheio; /rr/: barro, erro) ou figuram vogais nasais (/am/: tampa; /un/:

mundo). Cegalla (1985, p. 13) descreve também que /am/ e /em/ no fim de palavras não são dígrafos, pois representam um ditongo nasal. No Quadro 2 são apresentados dígrafos que representam consoantes ou figuram vogais nasais.

	Dígrafo	exemplos ¹
dígrafos que representam consoantes	/ch/	<i>chapéu, cheio</i>
	/lh/	<i>Pilha, galho</i>
	/nh/	<i>banho, ganhar</i>
	/rr/	<i>barro, erro</i>
	/ss/	<i>asseio, passo</i>
	/gu/ (antes de /e/ ou /i/)	<i>guerra, seguinte</i>
	/qu/ (antes de /e/ ou /i/)	<i>leque, aquilo</i>
	/sc/ (antes de /e/ ou /i/)	<i>descer, piscina</i>
	/sç/ (antes de /a/ ou /o/)	<i>desça, cresço</i>
	/xc/ (antes de /e/ ou /i/)	<i>exceção, excitar</i>
dígrafos que figuram vogais nasais	/am/	<i>Tampa</i>
	/em/	<i>Tempo</i>
	/im/	<i>Limpo</i>
	/om/	<i>Ombro</i>
	/um/	<i>Jejum</i>
	/an/	<i>Santa</i>
	/en/	<i>Venda</i>
	/in/	<i>Linda</i>
	/on/	<i>Sonda</i>
	/un/	<i>Mundo</i>
¹ Em ortografia oficial do português.		

Fonte: Cegalla (1985, p. 12).

Quadro 2 – Dígrafos da língua portuguesa

2.3.2 Abreviaturas e siglas

Uma abreviatura é a representação de uma palavra ou expressão. Abreviaturas em geral terminam por consoante seguida de ponto final (CEGALLA, 1985, p. 72). São exemplos de abreviaturas: Av. (Avenida), ed. (Edição), Sra. (Senhora) e Dr. (Doutor).

“Sigla é a abreviatura formada com as letras iniciais das palavras de um nome ou título” (CEGALLA, 1985, p. 72), como as siglas: FAB para Força Aérea Brasileira, MT para Mato Grosso e SENAC para Serviço Nacional de Aprendizagem Comercial.

Por serem práticas e cômodas, e com o objetivo de poupar tempo e espaço, siglas e abreviaturas vêm se multiplicando cada vez mais nos dias de hoje. Algumas siglas até passam a funcionar como substantivos (CEGALLA, 1985, p. 72). Abreviaturas e siglas de uso mais

frequente podem ser encontradas em literaturas da língua de origem. Porém, para siglas específicas, tem-se uma maior dificuldade, pois são de mais fácil entendimento das áreas afins. Assim, para deficientes visuais, apenas a sigla, falada ou soletrada da forma literal, dificulta e torna o entendimento mais complexo.

2.4 GRAMÁTICA DA LÍNGUA ALEMÃ

No contexto deste trabalho o estudo da gramática da língua alemã compreende: o alfabeto e a respectiva pronúncia na língua portuguesa; os fonemas separados em vogais e consoantes, juntamente com a pronúncia de conjuntos de letras; e os nomes e sobrenomes de origem alemão-americano, significados e origem, bem como a americanização dos mesmos.

2.4.1 Alfabeto

Mais amplo que o alfabeto da língua portuguesa, o alfabeto alemão é constituído de vinte e seis letras, onde são acrescentados os chamados *Unlaute*¹ e a letra /β/, que corresponde a /ss/ e não existe como maiúscula. No Quadro 3 são apresentadas as letras e sua transcrição de maneira mais aproximada possível, usando letras e acentos do português onde possível (WELKER, 1992, p. 15).

letra	transcrição para a língua portuguesa	letra	transcrição para a língua portuguesa	letra	Transcrição para a língua portuguesa
A	a	J	Yót	S	És
Ä	é	K	KA	β	és-tsét
B	bê	L	El	T	Te
C	tsê	M	Émm	U	U
D	dê	N	Énn	Ü	Ü
E	ê	O	O	V	FAU
F	éf	Ö	Ö	W	Vê
G	guê	P	Pê	X	Iks
H	há	Q	Ku	Y	Üpsilonn
I	i	R	ÉR	Z	Tsét

Fonte: adaptado Welker (1992, p. 15).

Quadro 3 – Letras do alfabeto alemão transcritas para o português

¹ Segundo Welker (1992, p. 15), são as vogais /a/, /o/ e /u/ modificadas por metafonia, na escrita marcada por um trema.

2.4.2 Fonemas

Welker (1992, p. 16) afirma que em alemão é difícil, ou até mesmo impossível, descrever sons que não existem em português, sem recorrer a termos fonéticos. Na Alemanha, a diversidade de dialetos é tanta que para a maioria das pessoas é incompreensível. Segundo Welker (1992, p. 16), existem movimentos nas diversas regiões que são a favor do revigoramento dos dialetos.

Na Alemanha, existem diversos dialetos regionais tão diferentes uns dos outros que muitas vezes são ininteligíveis [incompreensíveis] entre si; ou seja, quando um habitante de determinada região fala seu dialeto, pessoas de outras regiões, até mesmo próximas – por exemplo, uma distância de cem quilômetros – compreendem-no com dificuldade, ou de maneira alguma. (WELKER, 1992, p. 16).

Porém, a maioria dos alemães não domina mais estes dialetos, falados hoje em dia, principalmente por pessoas idosas longe das grandes cidades, utilizados em peças por grupos de teatro ou por escritores de poemas e contos (WELKER, 1992, p. 16).

Assim como na língua portuguesa, os fonemas alemães são separados em vogais e consoantes. Os sons vocálicos da língua alemã são representados pelas seguintes letras: /a/, /ã/, /e/, /i/, /o/, /ö/, /u/, /ü/, /y/. De acordo com Welker (1992, p. 18), as vogais possuem duas características: quantidade (longas/breves) e qualidade (abertas/fechadas). Ainda, “as vogais são longas e abertas quando seguidas de duas consoantes, quer na fala, quer na escrita, portanto, para caso em que duas os mais consoantes representam um único som (*ch*, *ck*, *ng*, *sch* e todas as duplas consoantes: *ff*, *ll*, *mm*, etc.), e a letra *x*, que representa dois sons (*ks*)” (WELKER, 1992, p. 18). Para palavras monossilábicas e palavras compostas das quais elas fazem parte, a vogal também é breve.

Nos casos em que a consoante é única ou nenhuma, a vogal geralmente é longa. Quando reduplicada (/aa/, /oo/), seguida de um /h/ mudo ou no caso de /i/ seguida de /e/ (que não é pronunciado, a não ser em palavras de língua estrangeira, como *Familie*, *Immobilien*), a vogal é sempre longa (CAMPANA, 1987, p. 4). Também são longas as vogais que precedem duas consoantes das quais a segunda é /l/ ou /r/ (WELKER, 1992, p. 18).

Segundo Campana (1987, p. 6), além das consoantes simples, existem no alemão as consoantes dobradas ou compostas, definidas como grupos de consoantes que nunca se separam. Estas consoantes são: /ch/, /sch/, /st/, /sz/, /tz/, /ck/, /sp/, /ss/, /th/ e /chs/.

No Quadro 4 encontra-se a pronúncia destes grupos consonantais, onde está exposta e explicada simultaneamente com a pronúncia das consoantes simples, para maior facilidade de compreensão devido a grande quantidade de casos.

consoante / grupo consonantal	Pronúncia	exemplos ²
/b/	corresponde a /b/, mas no final de vocábulos ¹ e diante de /s/ e /t/, tem um som aproximado de /p/	<i>Weib</i> (váip), <i>erbt</i> (éspit), <i>erbst</i> (éspst)
/c/	pronuncia-se ts quando antes de /ä/, /e/ e /i/, mas antes de outras vogais e consoantes tem som de /k/. Atualmente, é substituída por /k/ e /z/	<i>Cäsar</i> , <i>Ceder</i> , <i>Citrome</i> , <i>Cypern</i> , <i>Catalini</i> , <i>Cato</i> , <i>Cortez</i> , <i>Caludius</i>
/ch/	corresponde a /h/ fortemente aspirado, sobretudo quando precedido das vogais /a/, /o/, /u/, /au/, circunstância em que equivale a /j/ espanhol como em <i>pájaro</i> , <i>ejército</i> , <i>mujer</i> . Tem, porém, som mais brando quase igual a /ch/, quando vem precedido de /a/, /i/, /ä/, /ö/, /ü/, /äu/	<i>Bach</i> , <i>Loch</i> , <i>Buch</i> , <i>Recht</i> , <i>Licht</i> , <i>Bächer</i> , <i>Bücher</i> , <i>Löcher</i> , <i>welche</i> , <i>Räuchern</i> , <i>heucheln</i>
/ck/	corresponde a /k/	<i>Glocke</i> , <i>Stock</i> , <i>Ecke</i>
/chs/	em final de sílabas corresponde a /ks/, mas nas palavras compostas conserva o som separado de /ch/, sendo que o /s/, que entra no segundo elemento do composto, também é pronunciado separadamente	<i>Wachs</i> , <i>Ochs</i> , <i>Fuchs</i> , <i>Achsel</i> , <i>sechs</i> , <i>Wachsam</i> , <i>nachsuchen</i>
/d/	em final de palavras e diante de /s/, pronuncia-se quase como /t/	<i>mild</i> (milt), <i>Gold</i> (golt), <i>beredsam</i> (berétsam)
/dt/	corresponde a /t/	<i>Stadt</i> (chtát), <i>beredt</i> (berét)
/g/	no início de palavras ou sílabas, é sempre duro como o /g/ de <i>gato</i> , mesmo antes de /e/ e de /i/, mas no final de palavras tem som aproximado de /k/	<i>Garten</i> , <i>Geld</i> , <i>Gift</i> , <i>Gold</i> , <i>Güte</i> , <i>Glas</i> , <i>Gnade</i> , <i>Dreissig</i> (dráicik)
/gn/	pronuncia-se separadamente e nunca como /nh/ ou /gn/ italiano	<i>Begegnen</i> (bêgêgnen), <i>Gnade</i> (g-náde)
/h/	é fortemente aspirado no início de palavras ou de sílabas. Entre vogais, a aspiração é leve, mal se ouvindo. Seguido de consoante e no final da palavra é mudo, servindo apenas para prolongar a vogal precedente	<i>Haus</i> , <i>Held</i> , <i>haben</i> , <i>hören</i> , <i>Ziehen</i> , <i>Schuhe</i> , <i>sehen</i> , <i>blühen</i> , <i>Ohr</i> , <i>Hahn</i> , <i>ihnen</i> , <i>ehren</i> , <i>Zähne</i> , <i>Uhr</i> , <i>führen</i>
/j/	corresponde a /i/	<i>Já</i> (ia), <i>Jahr</i> (iá:r)
/k/	corresponde a /c/ duro como na palavra <i>casa</i>	<i>Kahl</i> , <i>Knade</i> , <i>Kind</i> , <i>Käse</i>
/r/	tem som aproximado ao /r/ dos vocábulos <i>faro</i> , <i>caro</i> , <i>louro</i> . Em algumas regiões da Alemanha, porém, soa mais ou menos como o /r/ francês, ligeiramente arrastado	<i>Rad</i> , <i>Ruhm</i> , <i>Erde</i> , <i>Haar</i> , <i>Fahrrad</i> , <i>Rohr</i>
/s/	tem som de /z/ antes de vogal, entre duas vogais ou entre uma líquida (/l/, /m/, /n/, /r/), como nas palavras <i>casa</i> , <i>lousa</i> , <i>mesa</i> . Nos demais casos, ou seja, no final das palavras e precedida de uma consoante dura (/k/, /ck/, /p/, /t/, /ch/) tem som forte e sibilante de /s/ como nos vocábulos <i>sal</i> , <i>sol</i> , <i>sul</i>	<i>Sagen</i> , <i>Sahne</i> , <i>Sinn</i> , <i>Sorge</i> , <i>Sumpf</i> , <i>Elsa</i> , <i>emsig</i> , <i>Das</i> , <i>Haus</i> , <i>Maus</i> , <i>Lotse</i> , <i>Häcksel</i> , <i>knipsen</i> , <i>wachsen</i>
/sch/	corresponde a /ch/, como em <i>chá</i> , <i>China</i> e <i>Chile</i>	<i>Schön</i> , <i>schon</i> , <i>Schiff</i>
/sp/, /st/	corresponde a /ch/ no início de palavras ou sílabas do radical como nos vocábulos <i>chá</i> , <i>China</i> , <i>Chile</i> , <i>chusma</i> . Nos demais casos, a saber, quando no corpo da palavra, conserva o som originário	<i>Stock</i> (chtók), <i>sprechen</i> (chpréjen), <i>Spiel</i> (chpi:l), <i>Stroh</i> (chtrô:), <i>besprechen</i> (bêchpréjen)
¹	Palavra que faz parte de uma língua (FERREIRA, 1991).	
²	Em ortografia oficial do alemão.	

Fonte: adaptado de Campana (1987, p. 6).

Quadro 4 – Pronúncia dos grupos consonantais e das consoantes simples

consoante / grupo consonantal	Pronúncia	exemplos ²
/ss/	tem exatamente o som de /ss/, como nas palavras <i>massa</i> , <i>passar</i> , <i>sussurro</i> , porém só se emprega entre duas vogais, em que a primeira é acentuada e breve	<i>Gasse</i> , <i>besser</i> , <i>müssen</i> , <i>lassen</i>
/sz/	corresponde a /s/ áspero depois de uma vogal longa, que na escrita manual é grafado /ß/; corresponde a /b/ no final da palavra ou da sílaba; e a /c/ antes da desinência /t/	<i>Hass</i> (<i>Hasz</i>), <i>müssen</i> (<i>müßten</i>) <i>gewusst</i> (<i>gewußt</i>), <i>Fluss</i> (<i>Fluß</i>), <i>Gruss</i> (<i>Gruß</i>), <i>Schoss</i> (<i>Shoß</i>), <i>Fuss</i> (<i>Fuß</i>)
/qu/	corresponde a /qv/	<i>Qual</i> (<i>qvál</i>), <i>Quelle</i> (<i>qvéle</i>)
/ph/	corresponde a /f/	<i>Prophet</i> (<i>profét</i>), <i>Photograph</i> (<i>fotográf</i>), <i>Philosophie</i> (<i>filosofi:</i>), <i>Sofá</i>
/pf/	pronuncia-se ligeiramente reunindo numa só letra, devendo a letra /p/ ser pronunciada muito rapidamente	<i>Pferd</i> (<i>pfér</i>), <i>Pfeil</i> (<i>pfáil</i>), <i>Apfel</i> (<i>ápfel</i>), <i>Pfund</i> (<i>pfünt</i>)
/th/	corresponde a /t/, sendo pronunciado simplesmente como /t/ português e nunca como /th/ inglês. Ademais, nos nomes puramente germânicos é oscilante o uso do /h/	<i>Theodor</i> , <i>Theater</i> , <i>Apotheke</i> , <i>Thomas</i> , <i>Themse</i> ; <i>Mat(h)ilde</i> , <i>Walt(h)er</i>
/ti/	corresponde a /tsi/	<i>Patient</i> (<i>patsient</i>)
/tz/	corresponde /ts/, pronunciando-se ambas as consoantes separadamente, devendo-se ouvir o som individual de cada uma. Frequentemente, contudo, encontram-se palavras escritas apenas com /z/, desacompanhadas do /t/. A pronúncia, entretanto, será a mesma de /ts/	<i>Tatze</i> (<i>tádetse</i>), <i>Blitz</i> (<i>blíts</i>), <i>Mütze</i> (<i>mutse</i> , com pronúncia de “u” francês), <i>stizen</i> (<i>sítsen</i>); <i>Herz</i> (<i>hérts</i>), <i>Tanz</i> (<i>tánts</i>), <i>Sturz</i> (<i>chtúrts</i>)
/tsch/	corresponde a /tch/, mais precisamente ao som do /ch/ inglês	<i>Deutsch</i> (<i>dóitich</i>), <i>Deutschland</i> (<i>dóitichland</i>); (<i>child</i> , <i>chesse</i> , <i>chess</i>)
/v/	corresponde a /f/ nas palavras genuinamente alemãs, mas nas palavras de origem estrangeira conserva o som próprio de /v/	<i>Vater</i> (<i>fáter</i>), <i>Vetter</i> (<i>féter</i>), <i>verlieren</i> (<i>ferlí:ren</i>), <i>viel</i> (<i>fi:l</i>), <i>Eva</i> (<i>éva</i>), <i>Violine</i> , <i>Venedig</i> , <i>Klavier</i>
/w/	corresponde a /v/ português. O /w/ alemão não tem o som do /w/ inglês, ou seja, de /u/. Assim, Walter pronunciar-se-á <i>Valter</i> e não <i>Uóltár</i> como o inglês	<i>Wald</i> (<i>váld</i>), <i>Wasser</i> (<i>vásser</i>), <i>Weg</i> (<i>vék</i>), <i>Wille</i> (<i>vile</i>), <i>Woche</i> , <i>Wunder</i> , <i>Wurzel</i> (<i>vóje</i> , <i>vúnder</i> , <i>vúrtsel</i>)
/y/	corresponde a /i/. Pouquíssimo empregada em alemão, esta semi-vogal só se encontra em palavras de origem estrangeira	<i>Zylinder</i> , <i>typisch</i> , <i>Ägypten</i>
/z/	soa como /ts/ português, nunca, porém, como o /z/ português. Tem sempre, portanto, som áspero	<i>Zahn</i> (<i>tsán</i>), <i>Zeit</i> (<i>tsáit</i>), <i>Zeitung</i> (<i>tssáitunk</i>), <i>Zug</i> (<i>tsúk</i>), <i>Zweck</i> (<i>tsvék</i>)

¹ Palavra que faz parte de uma língua (FERREIRA, 1991).

² Em ortografia oficial do alemão.

Fonte: adaptado de Campana (1987, p. 6).

Quadro 4 – Pronúncia dos grupos consonantais e das consoantes simples (continuação)

2.4.3 Nomes alemão-americano

Jones (2006, p. 2) explica que o termo “nome alemão-americano” é definido como qualquer nome derivado da língua alemã ou seus dialetos, mesmo se mudanças na pronúncia e na soletração o tornaram irreconhecíveis. Segundo Jones (2006, p. 2), o termo “nome alemão” é um tanto quanto mais difícil. Pode-se citar como exemplo as pessoas que na Holanda² falavam os idiomas *Low Franconian*, *Low Saxon* e *Frisian*. Jones (2006, p. 2) afirma que os dois primeiros, *Low Franconian* e *Low Saxon*, são dialetos alemães, enquanto *Frisian* era uma língua independente, também falada na Alemanha. Como hoje o holandês é uma língua nativa, alguns nomes são considerados holandeses, mesmo por séculos utilizados em famílias alemãs.

2.4.3.1 Nomes: significado e origem

Muitas pessoas com nomes em alemão conhecem o significado da palavra correspondente em alemão apenas através do dicionário. Mas sendo um nome, pode-se ter um significado completamente diferente. Infelizmente, a onomástica (ciência dos nomes) não é uma ciência exata, como é provado quando os peritos discordam a respeito do significado de nome (JONES, 2006, p. 1). Ainda, apenas os pais de uma criança sabem o que eles acreditam ser o significado do nome e, muitas vezes, eles não sabem. Jones (2006, p. 3) também afirma que nem sempre é possível determinar de que língua o nome é derivado. O nome *Horn* por exemplo, pode ser alemão, inglês, holandês ou escandinavo.

2.4.3.2 Sobrenomes: significado e origem

Devido ao grande aumento da população após as migrações, tornou-se necessário distinguir entre os vários indivíduos da comunidade que compartilhavam de um nome em comum. Segundo Jones (2006, p. 24), esta distinção podia ser feita por referência à filiação da

² A Holanda originalmente era do Sagrado Império Romano. Segundo Sainty (1992, p. 15), o Império Alemão foi fundado por Carlos Magno, cuja coroação no dia de Natal de 800 deu a aprovação papal para a unificação sob o seu domínio a França, a maior parte da atual Alemanha, Holanda, Bélgica e Luxemburgo, parte da Suíça moderna e norte da Itália.

peessoa, sua residência, uma característica topográfica do terreno ou perto de sua residência, sua profissão, seu empregador, sua aparência ou até mesmo seu comportamento. Esta distinção, na maioria das vezes, era acrescentada ao nome, formando assim seu significado, como *Friedemann* (homen de paz) e *Hering* (vendedor de arenques).

Assim como os filhos assumem os sobrenomes dos pais, escravos quando libertados assumiam nome e sobrenome de seus antigos donos. Famílias nobres habitualmente assumiam o nome de seu castelo. Jones (2006, p. 24) descreve que os donos do castelo *Wolkenstein* eram chamados de *von Wolkenstein* ou os *Wolkensteiners*. No entanto, quando eles vendiam o castelo e se mudavam para outro lugar, deixavam para trás seu sobrenome antigo e tomavam o nome de seu novo lugar.

Muitas das famílias eram designadas frequentemente por características do terreno ou local onde viviam, como montes, montanhas, vales, campos e florestas. Segundo Jones (2006, p. 26), se *Johann* morava perto de um monte, ele poderia ser chamado de *Johann Buehl* (monte). Jones (2006, p. 26) afirma que mais numerosos são os nomes derivados de *berg* e *berger*, o nome mais comum para montanha, como encontrado nos nomes *Leimberg*, *Isenberg*, *Rautenberg*, *Litzenberger* e *Krohberger*. No Quadro 5 são descritos alguns sobrenomes e seu significado quando derivados por características do terreno. Como exemplos, procurou-se utilizar nomes de ruas e bairros na cidade de Blumenau.

derivado	Significado	Exemplos
<i>bach</i>	maior que um riacho/ribeiro	<i>Bachmann</i> ¹ , <i>Weissbach</i> ² (riacho branco)
<i>berg, berger</i>	nome mais comum para montanha	<i>Achterberg</i> ² , <i>Krohberger</i> ² , <i>Oberberger</i> ¹ , <i>Uterberg</i> ¹
<i>burg</i>	Castelo	<i>Altenburg</i> ² (castelo antigo), <i>Hoburg</i> ¹ (castelo alto)
<i>furt, fort</i>	riacho que pode ser cruzado a pé	<i>Badenfurt</i> ² , <i>Frankfurt</i> ¹
<i>hoff</i>	pátio, fazenda	<i>Althoff</i> ² (fazenda antiga), <i>Hoffmann</i> ²
<i>horn</i>	pico da montanha	<i>Berghorn</i> ¹ , <i>Horn</i> ² , <i>Matterhorn</i> ¹
<i>meyer, mayer</i>	nome mais comum para designar um fazendeiro (tão comum quanto <i>mann</i>)	<i>Germeyer</i> ¹ (fazendeiro do pântano), <i>Kallemeyer</i> ¹ , <i>Meyer</i> ²
<i>stein</i>	cúpula rochosa, rochedo, pedra	<i>Morgenstein</i> ¹ (pedra de campo), <i>Stein</i> ² , <i>Steinberg</i> ¹ (montanha de pedra)
<i>tal, thal</i>	Vales	<i>Rosenthal</i> ¹ , <i>Thalman</i> ¹
¹ Obtido de Jones (2006). ² Ruas da cidade de Blumenau (SECRETARIA MUNICIPAL DE PLANEJAMENTO URBANO, 2010).		

Fonte: adaptado de Jones (2006, p. 67-354).

Quadro 5 – Sobrenomes e seu significado quando derivados por características do terreno

Muitas vezes as pessoas ganhavam seus sobrenomes decorrentes de seu apelido ou características físicas, como a cor ou estilo do cabelo. Jones (2006, p. 42) afirma que no

alemão podem ser encontrados os nomes *Schwarz* (moreno), *Braun* (cabelos castanhos), *Roth* (cabelos ruivos), *Weiss* (cabelos loiros) e *Krause* ou *Kraus* (cabelos crespos). A estatura física e a idade também originaram sobrenomes, como *Kurtz* (baixo), *Lang* (alto), *Gross* (grande), *Klein* (pequeno), *Alt* (velho), *Jung* (novo), *Juengling* (jovem) e *Greis* (barba grisalha) (JONES, 2006, p. 42). Ainda segundo Jones (2006, p. 43), sobrenomes poderiam resultar de comportamentos pessoais, assim como um homem poderia ser sério (*Ernst*) ou jovial (*Froehlich*), cortês (*Huebsch*) ou bruto (*Rauh, Grob*).

2.4.3.3 Americanização de nomes alemães

Quando os imigrantes embarcaram em seus navios em *Rotterdam*, os capitães ingleses tiveram dificuldades em escrever a lista de passageiros do navio. Sem nenhum conhecimento no alemão e não familiarizados com os dialetos alemães, os escrivães escreveram os nomes como eles os ouviam, ou até na forma de nomes em inglês, dos quais eram mais parecidos com os sons. Desta forma, *Theiss* e *Weiss* se tornaram *Dice* e *Wise*, enquanto *Albrecht* e *Leitner* tornaram-se *Albright* e *Lightner* (JONES, 2006, p. 57). Os imigrantes também tentavam soletrar os seus nomes, mas com os sons das letras alemãs. Assim, se um homem chamado *Diehl* soletrou o seu nome como /day/, /ee/, /ay/, /há/, /ell/, o escrivão pode ter entendido como *Deahl* (JONES, 2006, p. 57).

A razão pela qual muitos alemães assinavam seus nomes com um X não era porque eles eram analfabetos, mas sim porque eles só sabiam o alemão e as autoridades inglesas não sabiam ler. Em alguns casos, o escritor da lista de passageiros do navio desistiu e pediu o significado dos nomes, assim nomes como *Zimmermann* foi traduzido para a língua inglesa como *Carpenter* (carpinteiro). Alguns imigrantes preferiram até alterar a pronúncia de seus nomes, preservando assim a correta pronúncia, que antes era impronunciável na língua inglesa (JONES, 2006, p. 58).

2.5 FURBTTS

Oechsler (2009) descreve a especificação e a implementação de um protótipo de sistema texto-fala para a língua portuguesa, que realiza o pré-processamento léxico e sintático

do texto, o tratamento de abreviaturas e a síntese de palavras, números cardinais e siglas, na ordem que são apresentadas no texto, com entonação prosódica.

A operacionalidade do FurbTSS é bastante simplificada, mas o protótipo ainda não está preparado para ser operado por pessoas com deficiência visual severa. A partir de uma entrada de texto em português, o protótipo processa as palavras, classificando-as e separando-as silabicamente. Em seguida deve ser feita a transcrição completa das palavras processadas para que o identificador fonético seja acionado. Por último, o texto de entrada é sintetizado utilizando o sintetizador MBROLA, bem como é gerado um arquivo *wave* para ser executado.

Observa-se que, embora apresente limitações, Oechsler (2009, p. 58-59) afirma que a “maioria dos textos escritos em português inseridos no protótipo foram transcritos e também sintetizados sem problemas [...] com desempenho satisfatório e principalmente inteligível”. Entre as limitações descritas está o processamento de nomes próprios, que podem apresentar sequências inválidas de letras se comparado com as normas da língua portuguesa, gerando erros irrecuperáveis durante o processamento linguístico ou durante a síntese.

Além do sistema de conversão texto-fala, o protótipo disponibiliza uma biblioteca que efetua a conversão texto-fala a partir do processamento de texto com vocabulário irrestrito escrito em língua portuguesa.

2.6 GOOGLE MAPS

O Google Maps é uma ferramenta de pesquisa e visualização, criada pela empresa Google (GOOGLE, 2010a), que por imagens via satélite possibilita ao usuário a consulta de rotas e mapas, tendo como entrada localizações (rua, bairro, cidade, estado, país), nome de empresas ou mapas criados por usuários, como mostra a Figura 1.

Após realizada uma consulta, dependendo da forma como o usuário pretende percorrer o caminho (de carro, de transporte público ou a pé), diferentes trajetos são sugeridos na página, como também informações sobre a distância e o tempo necessário para percorrê-los. As rotas obtidas podem ser enviadas por *email*, impressas ou disponibilizadas em outras páginas através de um *link*.

Outros recursos como a possibilidade de empresas preencherem dados cadastrais, como logotipos e fotos, entre outras informações, e também a criação de mapas, que podem ser posteriormente disponibilizados para a visualização, são apresentados pela ferramenta.

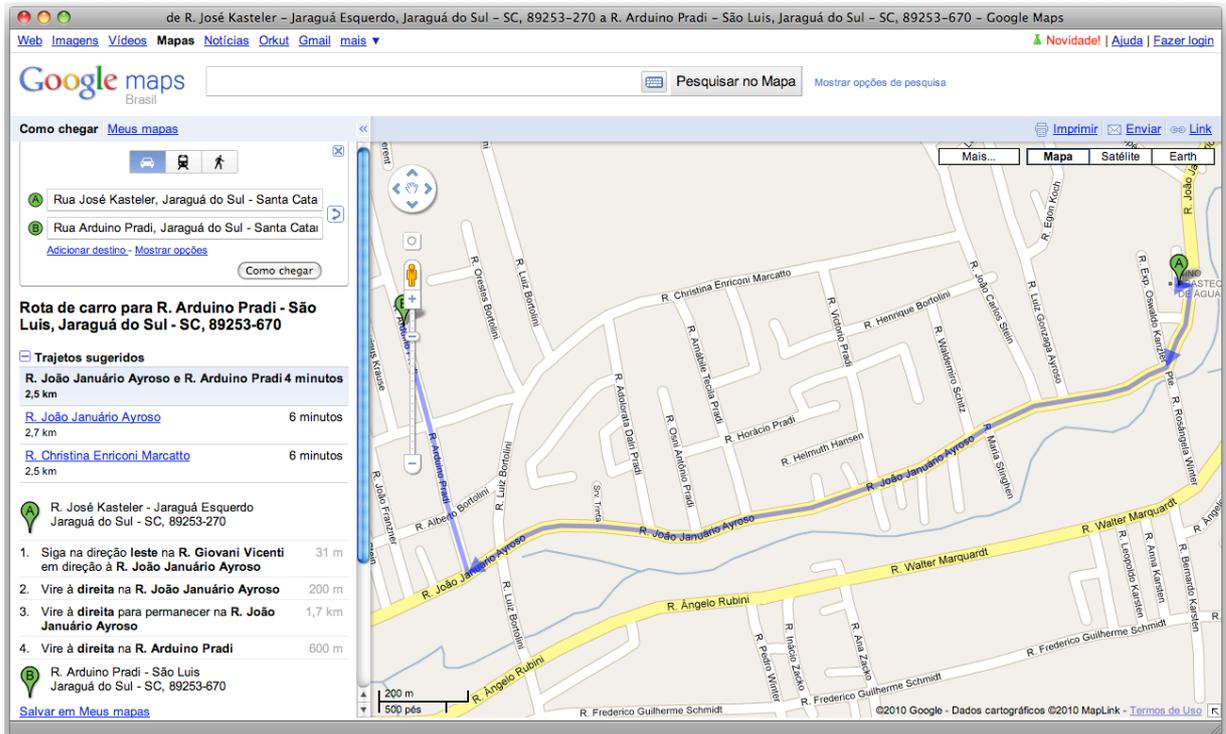


Figura 1 – Obtendo uma rota no Google Maps

2.6.1 Google Directions API

O Google Directions API (GOOGLE, 2010b) é um serviço destinado a calcular direções entre locais usando requisições de *HyperText Transfer Protocol* (HTTP). O uso da API é limitado a 2.500 solicitações de direções por dia, sendo que usuários com conta Google Maps Premier possuem 100.000 solicitações diárias.

Para solicitações na API deve ser utilizado um formato padrão de *Uniform Resource Locator* (URL) (Quadro 6), contendo a forma de saída, apresentada no quadro como `output`, e uma sequência de parâmetros, apresentada no quadro como `parameters`.

```
http://maps.googleapis.com/maps/api/directions/output?parameters
```

Fonte: Google (2010b).

Quadro 6 – Formato padrão da URL utilizada pelo Google Directions API

O `output` na URL define a forma de retorno de como serão apresentados os dados, que pode ser em *eXtensible Markup Language* (XML) ou *JavaScript Object Notation* (JSON). Os parâmetros são opcionais ou obrigatórios. O Quadro 7 apresenta os parâmetros, sua obrigatoriedade e funcionalidade.

parâmetro	obrigatório	Descrição
Origin	sim	Endereço de origem ou latitude/longitude
destination	sim	Endereço de destino ou latitude/longitude
Mode	não	modo de transporte utilizado para calcular as direções, sendo <code>driving</code> o valor padrão, que pode ser alterado para <code>walking</code> ou <code>bicycling</code>
waypoints	não	local por onde se deseja passar, podendo ser endereço ou latitude/longitude
alternatives	não	caso seja <code>true</code> , pode retornar mais de um rota como sugestão
Avoid	não	local por onde não se deseja passar, podendo ser evitadas pontes e estradas de alta velocidade (<code>tolls</code> ou <code>highways</code>)
Units	não	sistema de unidade usado para retornar o resultado, podendo ser <code>metric</code> ou <code>imperial</code>
region	não	região onde deve ser efetuada a pesquisa, devendo ser enviado o <code>country code Top-Level Domain (ccTLD)</code>
language	não	língua para retornar o resultado, devendo ser utilizado o código da língua desejada
sensor	sim	indicação se o dispositivo utilizado na pesquisa possui sensor de localização

Fonte: Google (2010b).

Quadro 7 – Parâmetros da URL utilizada pelo Google Directions API

No Quadro 8 tem-se um exemplo de uma URL já com o formato de retorno e alguns parâmetros no padrão utilizado pela API.

```
http://maps.google.com.br/maps/api/directions/json?origin=Rua José Kasteler, Jaraguá Esquerdo, Jaraguá do Sul - Santa Catarina&destination=Rua Arduino Pradi, São Luis, Jaraguá do Sul - Santa Catarina&language=pt-BR&sensor=false
```

Fonte: Google (2010b).

Quadro 8 – Exemplo de uma URL no formato utilizado pelo Google Directions API

2.7 TRABALHOS CORRELATOS

Pode-se encontrar exemplos de sistema de conversão texto-fala derivados de trabalhos acadêmicos, na maioria dissertações de mestrado e trabalhos de conclusão de curso. Um deles é o sistema de conversão texto-fala de Gomes (1998). Existem também ferramentas e dispositivos comerciais, como Jaws for Windows (FREEDOM SCIENTIFIC, 2009), VoiceOver (APPLE INCORPORATION, 2010) e navegadores GPS (MORIMOTO, 2009).

2.7.1 Sistema de conversão texto-fala para a língua portuguesa utilizando a abordagem de síntese por regras

O trabalho apresentado em Gomes (1998) utiliza algumas regras elaboradas pelo próprio autor para efetuar a conversão do texto em fala. Oferece uma solução completa de síntese de fala para textos em português do Brasil, e traz uma descrição genérica dos sistemas de conversão texto-fala, incluindo um breve histórico e comentários sobre a estrutura e operação do sistema.

O trabalho foi dividido em seis etapas definidas, as quais são: pré-processador, classificador gramatical, divisor silábico, transcritor ortográfico-fonético, processador prosódico e processamento de sinais de fala. Para o processamento de sinais de fala é implementado o sintetizador por formantes de Klatt, modelo utilizado para a construção de aplicações de síntese de fala.

2.7.2 Jaws for Windows

Jaws for Windows é o leitor de tela para os sistemas operacionais Windows 95, 98, ME, NT, 2000, XP, Vista e 7 (FREEDOM SCIENTIFIC, 2009). Possui suporte para a maioria das aplicações no sistema operacional e outras aplicações mais populares como Microsoft Office, Internet Explorer, Firefox, Thunderbird e Adobe Acrobat Reader. Além dos programas básicos, programas não inclusos também podem ser adicionados através de *scripts*, que são disponibilizados pela Freedom Scientific, ou em grupos de suporte como o JFWlite e o JFWlist, mantidos por usuários e interessados na ferramenta.

O sistema é comercializado e possui sintetizadores de voz multilíngue (alemão, espanhol, finlandês, francês, inglês, italiano, português). Basicamente funciona sintetizando todos os textos encontrados na tela ou transferindo os mesmos para *displays* de caracteres em Braille, dispositivos que podem ser utilizados em conjunto à ferramenta e à máquina.

Também possibilita que o usuário trabalhe com diferentes vozes para diferentes aplicações. Utilizando teclas de atalho é possível que o usuário pare, avance e volte linhas de um determinado texto, e também aumente e diminua a velocidade da leitura (WEBAIM, 2010).

2.7.3 Voice Over

Segundo Apple Incorporation (2010), o VoiceOver é uma solução já presente em cada Mac para facilitar o uso dos computadores por pessoas com deficiência visual. A solução possui alta qualidade de voz, grande velocidade de fala e suporte a monitores Braille.

Possui a capacidade de pulmonar e de análise da frase para decifrar seu contexto, conseguindo pronunciar frases de acordo com o local e com base nos conceitos introduzidos nas sentenças anteriores. O sintetizador insere a respiração baseado em diferentes fatores como: cabimento, estrutura do texto que está sendo lido, tempo desde a última respiração e do tempo restante para terminar o texto (APPLE INCORPORATION, 2010).

Uma ampla variedade de tons de voz em inglês está disponível, conforme mostra a Figura 2. A principal voz utilizada pelo VoiceOver é *Alex*. Também é possível adicionar vozes em alemão, chinês, coreano, dinamarquês, espanhol, finlandês, francês, holandês, italiano, japonês, norueguês, português, russo e sueco, adquiridas de empresas.

Male
Alex
Bruce
Fred
Junior
Ralph
Female
Agnes
Kathy
Princess
Vicki
Victoria
Novelty
Albert
Bad News
Bahh
Bells
Boing
Bubbles
Cellos
Deranged
Good News
Hysterical
Pipe Organ
Trinoids
Whisper
Zarvox

Figura 2 – Tons de voz disponíveis no VoiceOver

Diferentes tons de voz e pronúncia também podem ser utilizados ao mesmo tempo, permitindo configurações para determinados eventos no sistema, conforme mostra a Figura 3.

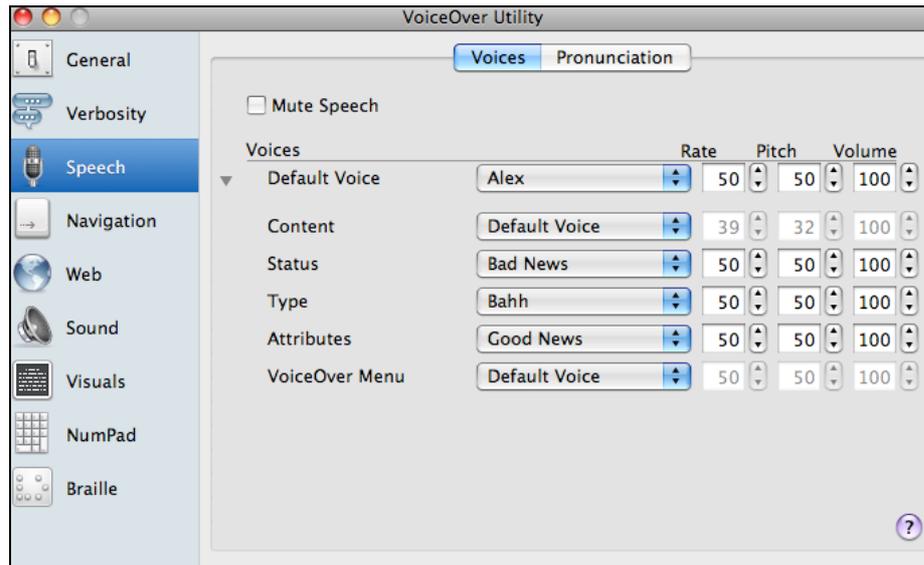


Figura 3 – Diferentes tons de voz utilizados ao mesmo tempo no VoiceOver

Teclas de atalho, navegação no sistema, sons de alerta, leitura de textos e *sites* também podem ser customizados e gravados em um *pen drive*. Quando inserido em outro Mac, o VoiceOver detecta a presença e busca as informações do *pen drive* enquanto ligado à máquina. Segundo Apple Incorporation (2010), recursos como estes são excelentes para deficientes visuais que compartilham computadores em bibliotecas e laboratórios, não limitando os mesmos a utilizarem apenas seus computadores pessoais.

2.7.4 Navegadores GPS

Segundo Morimoto (2009), o GPS é um sistema de cálculo de posicionamento a partir de sinais enviados por satélites. Os primeiros receptores GPS eram muito mais simples que os atuais, pois forneciam apenas a latitude e a longitude. O usuário precisava calcular a localização no mapa (MORIMOTO, 2009). Morimoto (2009) afirma que “os modelos atuais combinam as coordenadas de localização com mapas digitais em 3D e um software que calcula a posição no mapa, oferecendo uma orientação por voz”.

“O GPS basicamente funciona com a informação obtida através de satélites da órbita terrestre. O receptor estima a distância considerando o tempo de demora para receber o sinal proveniente do satélite, e assim é possível encontrar a localização” (TAKANO, 2009, p. 15). Basicamente, os produtos que fazem uso do sistema GPS capturam sinais de alta frequência vindo de satélites estacionados ao redor da terra e fazem um cálculo matemático para obter uma localização. Para que o serviço funcione de forma satisfatória é necessário que haja pelo

menos três ou mais deles visíveis no instante do cálculo. Inicialmente existiam vinte e quatro satélites, mas em setembro de 2008, o número aumentou para 32. Os satélites orbitam o planeta em uma trajetória geoestacionária, a 20.2 km de altitude, de forma que pelo menos quatro deles sejam visíveis a partir de qualquer ponto do planeta (MORIMOTO, 2009).

3 DESENVOLVIMENTO DO SOFTWARE

Neste capítulo são apresentados o desenvolvimento do software proposto e sua utilização, sendo que nas seções seguintes tem-se:

- a) a especificação dos Requisitos Funcionais (RF) e dos Requisitos Não-Funcionais (RNF);
- b) a lista de fonemas suportados;
- c) a lista de padrões de nomes alemão-americano suportados;
- d) a lista de encontros consonantais e vogais substituídos;
- e) os diagramas de casos de uso e de classes;
- f) a implementação do pré-processamento do texto, de encontros consonantais, da obtenção de rotas, da identificação e alteração de nomes alemão-americano e da interface do software;
- g) a operacionalidade do software desenvolvido.

Também são detalhadas técnicas e ferramentas utilizadas na implementação, assim como as dificuldades encontradas e os resultados obtidos.

3.1 REQUISITOS DO SOFTWARE A SER DESENVOLVIDO

O software para obtenção das rotas através de um sistema conversor texto-fala deve:

- a) disponibilizar uma interface para entrada em forma de texto de uma localização origem e uma localização destino (rua, cidade, estado, país) (RF);
- b) obter a rota correspondente à localização de origem e destino através do Google Directions API, efetuando um tratamento mais adequado para a mesma ser apresentada e sintetizada (RF);
- c) efetuar o pré-processamento, análise linguística, transcrição fonética e processamento prosódico tanto das localizações de origem/destino quanto da(s) possível(eis) rota(s) (RF);
- d) identificar nomes de origem alemão-americano (RF);
- e) efetuar a síntese da fala da entrada e da saída, procurando sintetizar nomes alemão-americano com sua correta pronúncia (RF);

- f) ser implementado na linguagem de programação *Object Pascal* (RNF);
- g) ser desenvolvido no ambiente de programação Delphi (RNF);
- h) utilizar o sintetizador MBROLA (DUTOIT et al., 2005) para síntese da fala (RNF);
- i) utilizar o protótipo FurbTTS (OECHSLER, 2009) para o processamento de textos em português e acionamento do identificador fonético (RNF);
- j) funcionar no sistema operacional Windows XP ou superior (RNF).

3.2 ESPECIFICAÇÃO

Nesta seção são apresentadas a relação dos fonemas e a lista de padrões de nomes alemão-americano suportados pelo software e a lista de encontros consonantais e vogais substituídos no software. Por fim, são descritos os diagramas de casos de uso e de classes, seguindo a notação *Unified Modeling Language* (UML), os quais foram confeccionados utilizando as ferramentas OmniGraffle (THE OMNI GROUP, 2010) e Netbeans (ORACLE CORPORATION, 2010).

3.2.1 Lista de fonemas

Oechsler (2009, p. 27) afirma que “em um sistema texto-fala a especificação dos fonemas suportados implica diretamente no resultado final do programa”. Os fonemas suportados no software desenvolvido são os mesmos do protótipo FurbTTS (OECHSLER, 2009), onde são baseados em um banco de difones³ encontrado e disponibilizado pelo sintetizador MBROLA⁴ (DUTOIT et al., 2005).

No protótipo Oechsler (2009, p. 27) descreve que a partir do banco de fonemas denominado BR3⁵ (COSTA, 2005), foi realizado um mapeamento de regras gramaticais e de

³ Os difones “são pequenas sequências de áudio que mostram a transcrição da metade de um fonema para a metade do outro” (MACHADO, 2007), desta forma concatenando-os linearmente.

⁴ O sintetizador é um projeto iniciado em 1995 pelo *TCTS Lab* da *Faculté Polytechnique de Mons* (Bélgica). Segundo Oechsler (2009, p. 27), foi incorporado ao protótipo FurbTTS por disponibilizar gratuitamente um banco de fonemas no português brasileiro (BR3).

⁵ No software desenvolvido optou-se em utilizar o banco de fonemas no português brasileiro BR3 para síntese de nomes em alemão, pois a troca pelo banco para o DE8 (fonemas no alemão) resultaria também na troca da voz.

regras baseadas nas limitações do sintetizador para cada fonema encontrado no banco. O Quadro 9 apresenta a lista de fonemas suportados pelo protótipo FurbTTS (OECHSLER, 2009), sendo que para cada fonema tem-se um exemplo de uma palavra que o contenha.

fonema	exemplo	fonema	exemplo	fonema	exemplo	Fonema	Exemplo
/b/	<i>barco</i>	/s/	<i>sala</i>	/r/	<i>puro</i>	/i/	<i>Pico</i>
/k/	<i>com</i>	/s2/	<i>casca</i>	/r2/	<i>arpa</i>	/im/	<i>Brinco</i>
/d/	<i>doce</i>	/x/	<i>chave</i>	/rr/	<i>torre</i>	/o/	<i>Tolo</i>
/g/	<i>grande</i>	/z/	<i>asa</i>	/a/	<i>vale</i>	/oo/	<i>Bola</i>
/p/	<i>pai</i>	/m/	<i>mesmo</i>	/@/	<i>tamanho</i>	/om/	<i>Ombro</i>
/t/	<i>taco</i>	/n/	<i>nunca</i>	/AM/	<i>campanha</i>	/u/	<i>Duro</i>
/f/	<i>fácil</i>	/nh/	<i>galinha</i>	/e/	<i>pêra</i>	/um/	<i>Algum</i>
/v/	<i>vinho</i>	/l/	<i>lanche</i>	/ee/	<i>quero</i>	/y/	<i>Mais</i>
/j/	<i>jato</i>	/h/	<i>alho</i>	/em/	<i>quente</i>	/w/	<i>Mau</i>

Fonte: Oechsler (2009, p. 28).

Quadro 9 – Fonemas suportados pelo protótipo FurbTTS, extraídos do banco de fonemas BR3

Fonemas correspondentes às letras /b/, /d/, /p/, /t/, /f/ e /v/, junto aos encontros consonantais /nh/ e /lh/ não necessitam de nenhum tratamento em sua transcrição⁶. Porém, algumas regras foram realizadas para uma correta transcrição dos demais fonemas. Destas regras, algumas estão diretamente ligadas à gramática da língua portuguesa e outras estão ligadas às limitações do sintetizador MBROLA (OECHSLER, 2009, p. 27). Assim, certos fonemas existentes na língua alemã, como /kv/, /ks/, /j/ e outros, também tiveram de ser alterados para uma correta transcrição de nomes em alemão.

3.2.2 Lista de padrões de nomes alemão-americano

Stemmer (1999) afirma que a grafia alemã divergia muito de um registro para outro, dependendo ainda de como as pessoas eram conhecidas pelos seus familiares e amigos. Jones (2006, p. 24) complementa que, no alemão, a distinção dos sobrenomes podia ser feita por referência à filiação da pessoa, sua residência, uma característica topográfica do terreno, sua profissão, seu empregador, sua aparência, ou até mesmo seu comportamento, como explicado anteriormente na seção 2.4.3.2. A maioria dos nomes alemão-americano segue este padrão, tendo assim uma formação parecida, como no caso dos nomes terminados em *mann*, que significa homem, ou em *berg*, que significa montanha.

⁶ A transcrição ortográfico-fonética faz a transformação da sequência ortográfica em uma cadeia de símbolos que representa a sequência de sons que compõe as palavras do texto (SIMÕES, 1999, p. 52).

No Quadro 10 é apresentada a lista de padrões suportados pelo software desenvolvido, sendo que para cada padrão, sendo ele encontro consonantal ou vogal tem-se: a posição no nome onde pode ser encontrado, o encontro consonantal ou vogal e também exemplos retirados de literaturas e listagem de ruas da cidade de Blumenau.

	padrões	Exemplos
nomes que iniciam com	<i>kn</i>	<i>Knaesel², Knie¹, Knopf¹, Knoth²</i>
	<i>pf</i>	<i>Pfaffendorf², Pfarr¹, Pfiffer², Pfund¹</i>
	<i>str</i>	<i>Straube², Strasburg¹, Strehlau², Straus¹</i>
	<i>hain</i>	<i>Hainle¹, Hains¹</i>
	<i>hein</i>	<i>Hein², Heinecken¹, Heinrich², Heins¹, Heinz²</i>
	<i>neu</i>	<i>Neufeld¹, Neudorff¹</i>
nomes que iniciam ou terminam com	<i>stein</i>	<i>Derstein¹, Steinbach², Steinborn¹, Steinnour¹, Wetzstein²</i>
	<i>bach</i>	<i>Bachmann¹, Bachstein¹, Miesbach¹, Steinbach², Weissbach²</i>
nomes que terminam com	<i>mann</i>	<i>Foremann¹, Hermann², Kuhlmann², Tallmann², Zimmermann²</i>
	<i>burg</i>	<i>Brondenburg¹, Hamburg², Wendeburg²</i>
	<i>berg</i>	<i>Achterberg², Rautenberg², Rosenberg¹, Wenberg¹</i>
	<i>berger</i>	<i>Derennerger¹, Hiderberger¹, Litzenberger², Krohberger²</i>
	<i>buehl</i>	<i>Ansbuehl¹, Zumbuehl¹</i>
	<i>buehler</i>	<i>Brechtbuehler¹, Geissbuehler¹, Zauberbuehler¹</i>
	<i>kofler</i>	<i>Underkofler¹, Unterkofler¹</i>
	<i>horn</i>	<i>Anhorn¹, Berghorn¹, Horn²</i>
	<i>wald</i>	<i>Answald¹, Ewald², Hauswald¹, Zumwald¹</i>
	<i>lich</i>	<i>Friedrich², Hadlich²</i>
	<i>thal</i>	<i>Erthal², Hachthal¹</i>
	<i>nthal</i>	<i>Blumenthal¹, Lowenthal¹, Lichtenthal¹</i>
	<i>nz</i>	<i>Franz², Leverenz¹</i>
	<i>ntz</i>	<i>Hintz¹, Schwantz²</i>
	<i>eiss</i>	<i>Theiss², Weiss¹</i>
	nomes que possuem	<i>sch</i>
<i>feld</i>		<i>Feldmann², Hasselfelde², Kreutzfeld², Neufeld¹, Weitenfeld¹</i>
<i>felder</i>		<i>Felder¹, Ferlderstein¹</i>
<i>bau</i>		<i>Baumann¹, Baumbach¹, Baumbusch¹, Baumgarten²</i>
<i>bauer</i>		<i>Bauer², Bierbauer¹, Eichbauer¹, Gebauer³</i>
<i>hof</i>		<i>Hofbauer¹, Hofer¹, Hofnar¹</i>
<i>hoff</i>		<i>Althoff¹, Buchhoff¹, Herkenhoff², Hoffmann²</i>
<i>hoffer</i>		<i>Bakhoffer¹, Hofferth¹</i>
<i>mueller</i>		<i>Bodenmueller², Giegmuehler¹, Heinmueller¹, Mueller²</i>
<i>meyer</i>		<i>Burkmeyer¹, Heinmeyer¹, Nemeyer¹, Stratemeyer¹</i>
<i>schmidt</i>		<i>Linderschmidt¹, Schmidt², Schmidts¹, Waldschmidt³</i>
<i>ü</i>		<i>Jürgen³, Lübke², Müller³, Rüdiger²</i>
<i>ä</i>		<i>Bärbel³, Käthe³</i>
<i>ö</i>		<i>Jörg³, Köhler³</i>
¹ Obtido de Jones (2006). ² Ruas da cidade de Blumenau (SECRETARIA MUNICIPAL DE PLANEJAMENTO URBANO, 2010). ³ IUPUI University Library (2009).		

Quadro 10 – Sequências consonantais e vogais que indicam que o nome é de origem alemã

	padrões	Exemplos
nomes que possuem	<i>tt</i>	<i>Hergott¹, Hettich¹, Hoette², Schaette², Schmitt²</i>
	<i>mm</i>	<i>Emmendoerfer², Nimmersatt¹, Nimmervoll¹, Zimmer²</i>
	<i>tz</i>	<i>Fritz², Heinritz¹, Hoehenholtz¹, Schwantz², Metzger³</i>
¹ Obtido de Jones (2006). ² Ruas da cidade de Blumenau (SECRETARIA MUNICIPAL DE PLANEJAMENTO URBANO, 2010). ³ IUPUI University Library (2009).		

Quadro 10 – Sequências consonantais e vogais que indicam que o nome é de origem alemã (continuação)

3.2.3 Lista de encontros consonantais e vogais

Segundo Campana (1987, p. 12), no alemão não existe acento gráfico ou ortográfico, como os acentos agudo, grave ou circunflexo existentes no português. O que existe é o acento prosódico, ou seja, que indica qual é a sílaba tônica que deve ser pronunciada com mais intensidade do que as demais na mesma palavra. Tem-se também, da língua alemã para a língua portuguesa, diferenças na pronúncia dos sons de vogais e encontros consonantais.

Assim, sons existentes na língua alemã tiveram de ser adaptados para sons do português. No Quadro 11 são apresentados encontros consonantais e vogais que tiveram suas alterações no software desenvolvido.

existente	substituído	existente	substituído	existente	substituído	existente	Substituído
/ã/	/é/	/ö/	/e/	/y/	/ü/	/st/	/xt/
/ei/	/ai/	/ü/	/i/	/ll/	/l/	/s/	/z/
/ie/	/ii/	/h/	/r/	/tt/	/t/	/tz/	/ts/
/ey/	/ai/	/ann/	/ã/	/chs/	/ks/	/z/	/ts/
/ay/	/ai/	/g/	/gu/	/sch/	/x/	/x/	/ks/
/äu/	/oi/	/j/	/i/	/ch/	/h/	-	-
/eu/	/oi/	/v/	/f/	/sp/	/xp/	-	-

Quadro 11 – Lista de encontros consonantais e vogais substituídos no software

Sendo assim, nomes como *Müller* e *Hermann* são tratados para que sua pronúncia se torne o mais parecido possível com o alemão, alterando sua escrita para *Miler* e *Rermã*.

3.2.4 Diagrama de casos de uso

O software desenvolvido possui quatro casos de uso: Informa os endereços de origem e destino, Pesquisa rota, Informa texto e Solicita leitura, como pode ser observado na Figura 4. Há uma sequência obrigatória para os casos de uso: primeiro deve-se informar os endereços de origem e destino, para então solicitar a leitura da rota obtida.

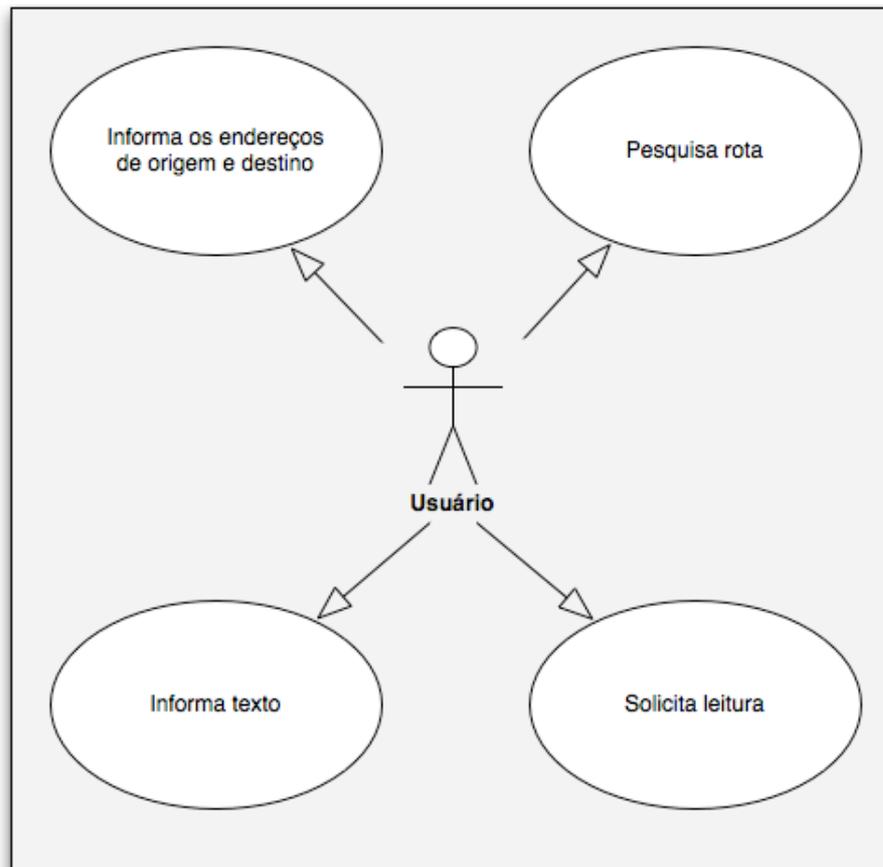


Figura 4 – Diagrama de casos de uso

O caso de uso Informa os endereços de origem e destino, detalhado no Quadro 12, é a ação inicial do usuário no software. Para que a busca da rota seja realizada com mais rapidez e exatidão deve-se inserir os endereços descrevendo nome da rua, do bairro, da cidade e do estado, devidamente separados, como por exemplo, Rua José Kasteler, Jaraguá Esquerdo, Jaraguá do Sul - Santa Catarina.

UC01 – Informa os endereços de origem e destino	
Pré-condições	Não existe.
Cenário principal	01) O usuário informa, em texto, os endereços de origem e destino, nos campos Endereço de origem (A) e Endereço de destino (B).
Pós-condições	Os endereços de origem e destino devem ter sido digitados.

Quadro 12 – Detalhamento do caso de uso Informa os endereços de origem e destino

O caso de uso Pesquisa rota (Quadro 13) apresenta a pesquisa da rota baseada nos

endereços de origem e destino. O computador utilizado para rodar o software deve estar conectado à Internet e os endereços de origem e destino devem estar digitados em seus respectivos campos. Caso não existam retornos válidos para os endereços, as respectivas mensagens são apresentadas em tela e sintetizadas.

UC02 – Pesquisa rota	
Pré-condições	O computador deve estar conectado à Internet. Os endereços de origem e destino devem estar digitados.
Cenário principal	01) O usuário clica no botão <i>Pesquisar</i> . 02) O software monta a URL com os campos <i>Endereço de origem (A)</i> , <i>Endereço de destino (B)</i> . 03) O software envia a URL para o Google Maps API. 04) O software recebe a rota em XML e a apresenta em forma de texto corrido no campo <i>Rota entre os pontos A e B</i> .
Exceção 01	No passo 01, caso não exista conexão com a Internet, uma mensagem de erro é apresentada em tela e sintetizada.
Exceção 02	No passo 01, caso o campo <i>Endereço de origem (A)</i> esteja vazio, uma mensagem de erro é apresentada em tela e sintetizada.
Exceção 03	No passo 04, caso a pesquisa não encontre uma rota entre os endereços, no XML de retorno é apresentado o motivo, que pode ser: <ul style="list-style-type: none"> – a rota não foi encontrada; – as coordenadas geográficas para os endereços não foram encontradas; – a pesquisa é inválida; – a quantidade limite de 8 pontos foi ultrapassada; – a quantidade de pesquisas realizadas por hora foi ultrapassada; – a pesquisa não pode ser realizada devido a um erro no servidor. Então, uma mensagem de erro é apresentada em tela e sintetizada.
Pós-condições	A rota entre os pontos A e B deve ter sido apresentada em tela.

Quadro 13 – Detalhamento do caso de uso *Pesquisa rota*

Quando o usuário desejar sintetizar qualquer outro texto, o mesmo deve ser informado no campo *Rota entre os pontos A e B*. O texto informado deve ser escrito em português. Caso seja inserido um texto em outra língua, é possível que ocorram problemas. Contudo, alguns termos aportuguesados, que fazem parte do hábito linguístico brasileiro, podem ser escritos, como por exemplo, as palavras *software*, *web* ou *site*. O detalhamento deste caso de uso é apresentado no Quadro 14.

UC03 – Informa texto	
Pré-condições	Não existe.
Cenário principal	01) O usuário informa um texto no campo <i>Rota entre os pontos A e B</i> .
Pós-condições	Um texto de entrada deve ter sido digitado.

Quadro 14 – Detalhamento do caso de uso *Informa texto*

O caso de uso *Solicita leitura*, apresentado no Quadro 15, representa a síntese da rota obtida através dos endereços de origem e destino, ou o texto informado no campo *Rota entre os pontos A e B*.

UC04 – Solicita leitura	
Pré-condições	O campo <i>Rota entre os pontos A e B</i> deve estar preenchido.
Cenário principal	01) O usuário clica no botão <i>Sintetizar</i> ou utiliza as teclas de atalhos <i>ctrl+s</i> . 02) Caso o usuário tenha clicado no botão <i>Sintetizar</i> , o software sintetiza o texto descrito no campo <i>Rota entre os pontos A e B</i> . Sendo utilizadas as teclas de atalho, o software sintetiza a descrição do campo e o respectivo valor onde o foco do <i>mouse</i> está.
Exceção 01	No passo 02, caso o texto viole uma regra sintática da gramática definida, como por exemplo, abrir parênteses e não fechar, uma mensagem de erro é apresentada em tela e sintetizada. Caso o texto possua algum problema na separação silábica ou tendo sido transcrito com problema (uma sequência impronunciável de fonemas), uma mensagem de erro é apresentada em tela e sintetizada.
Pós-condições	O texto do campo <i>Rota entre os pontos A e B</i> ou do campo focado deve ter sido sintetizado.

Quadro 15 – Detalhamento do caso de uso *Solicita leitura*

3.2.5 Diagrama de classes

A Figura 5 apresenta o diagrama de classes, fornecendo uma visão de como está estruturado o software desenvolvido. No diagrama não está detalhado o funcionamento do protótipo FurbTTS. A utilização de rotinas do protótipo FurbTTS é detalhada, na descrição das demais classes, conforme necessidade.

A classe principal é denominada *UTCCDesenv*. Esta classe é responsável pela interação com o usuário, acionando assim todas as demais classes, inclusive a classe *TMotor*, existente no FurbTTS. Segundo Oechsler (2009, p. 35), a classe *TMotor* é a única que deve ser implementada pelos projetos de interface de interação com usuário.

A classe *UGoogleMaps* é a responsável pela comunicação com o Google Maps API. A classe monta uma URL com os endereços de origem e destino, a forma de transporte e os demais parâmetros obrigatórios pela API. Através da URL é realizada uma chamada e tratado o XML de retorno, sendo ele a rota ou mensagens de erro.

A classe responsável por identificar nomes de origem alemão-americano é a *UIdentifyGermanName*. Após identificados os nomes, a classe também modifica os mesmos para que tenham uma pronúncia mais parecida possível com a língua alemã.

A classe *UUtils* fica encarregada de tratamentos gerais, como verificar a existência de conexão com a Internet, remoção de textos de uma *Hypertext Markup Language* (HTML) e controles de interface do software.

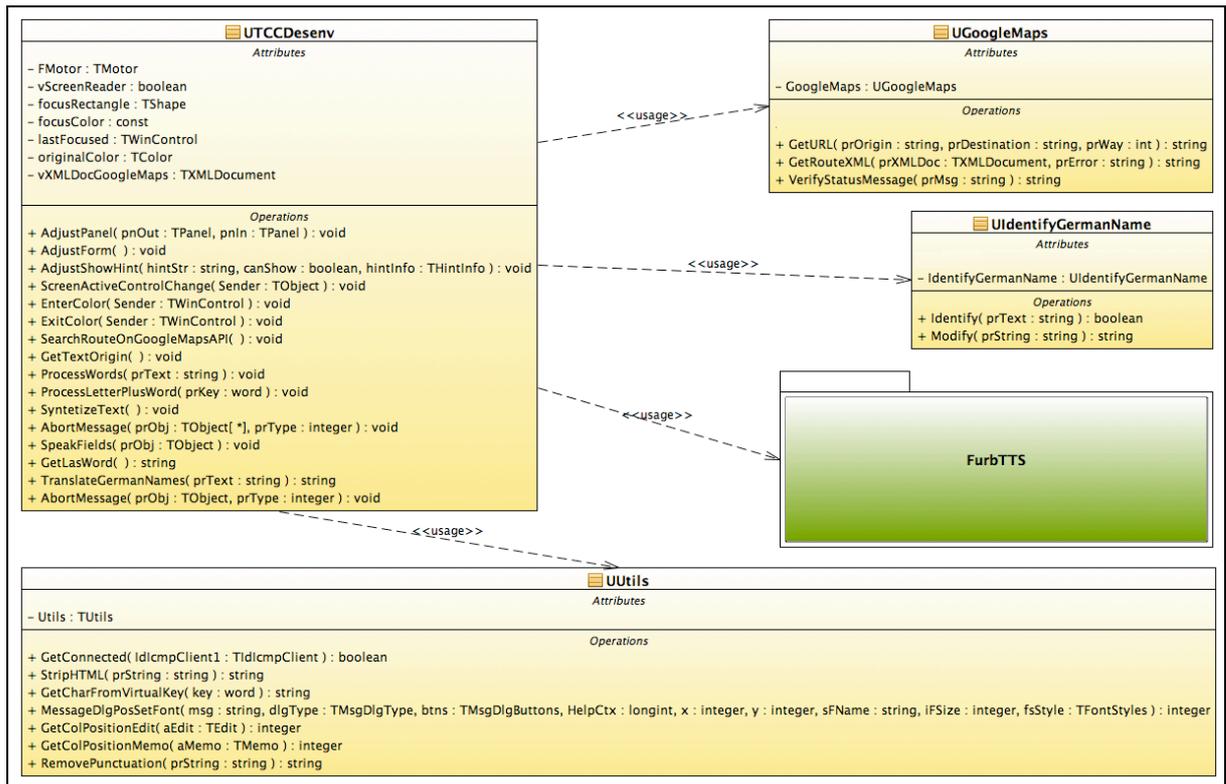


Figura 5 – Diagrama de classes

3.3 IMPLEMENTAÇÃO

Esta seção apresenta informações sobre as ferramentas utilizadas para o desenvolvimento do software proposto, juntamente com uma descrição de sua operacionalidade e trechos do código fonte para um melhor entendimento.

3.3.1 Técnicas e ferramentas utilizadas

As técnicas e ferramentas utilizadas para o desenvolvimento do software proposto foram as seguintes:

- a) a linguagem de programação utilizada para realizar a implementação foi a *Object Pascal*, que é uma linguagem derivada do Pascal;
- b) a *Integrated Development Environment (IDE)* utilizada para a codificação do software desenvolvido foi o Delphi 7;

- c) a API utilizada para obter as rotas foi a Google Directions API, um serviço que calcula as direções entre os locais usando solicitações HTTP. A API faz parte do Google Maps API *web services*, onde são usadas solicitações HTTP para URLs específicas, passando então parâmetros de URL como argumentos para os serviços;
- d) o sintetizador MBROLA Tools 3.5 que, segundo Oechsler (2009, p. 40), é baseado na concatenação de difones e utiliza como entrada uma lista de fonemas (em forma textual), juntamente com alguma informação prosódica (duração do fonema, descrição linear de entonação) para produzir amostras em 16 bits na frequência do banco de difones utilizado. Oechsler (2009, p. 40) também descreve que o protótipo não aceita texto bruto como entrada, não se tratando então de um sistema *Text-To-Speech* (texto-fala);
- e) o protótipo FurbTTS, que, além do sistema de conversão texto-fala, disponibiliza um pacote (biblioteca) que efetua a conversão texto-fala a partir do processamento de texto com vocabulário irrestrito escrito em língua portuguesa.

3.3.2 Pré-processamento do texto

O principal objetivo do pré-processamento do texto é evitar que erros léxicos e sintáticos ocorram. Assim, na etapa de pré-processamento da biblioteca FurbTTS são identificadas abreviaturas e expandidas em suas formas por extenso. Também são identificados caracteres que devem ser ignorados pelo analisador léxico e substituídos por palavras quando necessário, além de eliminados os conjuntos de caracteres que podem causar problemas (OECHSLER, 2009, p. 41).

“As abreviaturas que o protótipo FurbTTS consegue identificar são limitadas e foram obtidas de uma gramática da língua portuguesa.” (OECHSLER, 2009, p. 41). Sendo assim, devido a grande quantidade de abreviaturas retornadas nas pesquisas das rotas, foi necessário acrescentar ao conjunto de abreviaturas tratadas pelo FurbTTS abreviaturas de vias e lugares públicos, obtidas de Cegalla (1985, p. 72) e de Scarton e Smith (2010), que são abreviaturas da Empresa Brasileira de Correios e Telégrafos (EBCT). Estas abreviaturas devem ser obrigatoriamente seguidas de ponto final. No Quadro 16 é apresentada a lista das abreviaturas suportadas pelo software desenvolvido, junto com as abreviaturas já existentes no FurbTTS.

abreviatura	forma extensa	Abreviatura	forma extensa	abreviatura	forma extensa
apto. ¹	Apartamento	estr. ²	Estrada	pgto. ¹	Pagamento
al. ²	Alameda	ex. ¹	Exemplo	pq. ²	Parque
alm. ¹	Almirante	fat. ¹	Fatura	pr. ²	Pastor
apart. ¹	Apartamento	fig. ¹	Figura	pres. ¹	Presidente
arr. ¹	Arroba	fr. ²	Frei	proc. ¹	Processo
av. ¹	Avenida	fut. ¹	Futuro	prof. ¹	Professor
b. ²	Beco	gal. ²	Galeria	profa. ²	Professora
BC. ²	Beco	gen. ¹	General	prq. ²	Parque
bel. ¹	Bacharel	gov. ²	Governador	pst. ²	Pastor
cal. ²	Calçada	hab. ¹	Habitante	pte. ²	Ponte
calç. ²	Calçada	ind. ¹	Índice	r. ²	Rua
cap. ¹	Capital	inf. ¹	Informação	rdv. ²	Rodoviária
cel. ²	Coronel	jd. ²	Jardim	rel. ¹	Relatório
cfm. ¹	Conforme	jorn. ²	Jornalista	remte. ¹	Remetente
cg. ¹	Centígrama	kg ¹	Quilograma	rod. ²	Rodovia
cia. ¹	Companhia	km ¹	Quilômetro	rtn. ²	Retorno
cl. ¹	Classe	l. ²	Largo	sarg. ²	Sargento
cm ¹	Centímetro	lat. ¹	Latitude	sr. ¹	Senhor
cmt. ¹	Comandante	lg. ²	Largo	sra. ¹	Senhora
cop. ¹	Copiado	ltda. ¹	Limitada	srta. ¹	Senhorita
cv. ¹	Cavalo	m ²	Metros	t. ²	Travessa
d. ²	Distrito	maj. ²	Major	tel. ¹	Telefone
dep. ²	Deputado	mal. ²	Marechal	ten. ²	Tenente
div. ¹	Divisão	mar. ²	Marechal	tit. ¹	Titulo
doc. ¹	Documento	mq. ²	Marques	trv. ²	Trevo
dr. ²	Doutor	mun. ²	Município	tv. ²	Travessa
dt. ¹	Data	ns. ²	Nossa Senhora	v. ²	Via
dz. ¹	Dúzia	nsr. ²	Nosso Senhor	vcto. ¹	Vencimento
ed. ¹	Edição	p. ²	Praça	vd. ²	Viaduto
eng. ²	Engenheiro	pc. ²	Praça	ver. ²	Vereador
est. ²	Estrada	pda. ²	Parada		

¹ Oeschler (2009).

² Abreviaturas retornadas pelo Google (2010b).

Quadro 16 – Abreviaturas suportadas pelo software desenvolvido

Conjuntos de caracteres que podem causar erros no analisador sintático, como por exemplo, ponto seguido de travessão, onde o travessão, que não tem função fonética, é eliminado do texto, também recebem tratamento no pré-processamento do texto. No Quadro 17 é mostrado o código dos tratamentos adicionados na etapa de pré-processamento do FurbTTS.

```

// Abreviaturas da Empresa Brasileira de Correios e Telégrafos (EBCT).
if pos(' dt.', LowerCase(Result)) > 0 then
  Result := StringReplace(Result, ' Dt.', 'distrito', [rfReplaceAll,
    rfIgnoreCase]);

if pos(' lg.', LowerCase(Result)) > 0 then
  Result := StringReplace(Result, ' Lg.', 'largo', [rfReplaceAll, rfIgnoreCase]);

if pos(' maj.', LowerCase(Result)) > 0 then
  Result := StringReplace(Result, ' Maj.', 'major', [rfReplaceAll,
    rfIgnoreCase]);

if pos(' eng.', LowerCase(Result)) > 0 then
  Result := StringReplace(Result, ' Eng.', 'engenheiro', [rfReplaceAll,
    rfIgnoreCase]);

if pos(' pç.', LowerCase(Result)) > 0 then
  Result := StringReplace(Result, ' Pç.', 'praça', [rfReplaceAll, rfIgnoreCase]);

if pos(' pda.', LowerCase(Result)) > 0 then
  Result := StringReplace(Result, ' Pda.', 'parada', [rfReplaceAll,
    rfIgnoreCase]);

if pos(' prq.', LowerCase(Result)) > 0 then
  Result := StringReplace(Result, ' Prq.', 'parque', [rfReplaceAll,
    rfIgnoreCase]);

if pos(' profa.', LowerCase(Result)) > 0 then
  Result := StringReplace(Result, ' Profa.', 'professora', [rfReplaceAll,
    rfIgnoreCase]);

if pos(' dr.', LowerCase(Result)) > 0 then
  Result := StringReplace(Result, ' Dr.', 'doutor', [rfReplaceAll,
    rfIgnoreCase]);

if pos(' ns.', LowerCase(Result)) > 0 then
  Result := StringReplace(Result, ' Ns.', 'nossa senhora', [rfReplaceAll,
    rfIgnoreCase]);

```

Quadro 17 – Exemplos de tratamentos de abreviaturas

3.3.3 Encontros consonantais

Oechsler (2009, p. 46) descreve que, “por limitações do sintetizador alguns encontros consonantais são impronunciáveis e ocasionam erros irrecuperáveis durante a síntese, como por exemplo nas palavras *gnomo*, *apto*, *digno*, *mnemônico* e *ritmo*”. Estes encontros de difícil pronúncia são destruídos pela intercalação de vogais no protótipo FurbTTS. Nestes encontros consonantais, cuja transcrição poderia ser direta, faz-se necessário um tratamento especial no identificador fonético, como mostra o tratamento do fonema /d/, no Quadro 18, onde é adicionado o fonema /i/.

```

if letra = 'd' then
begin
  vTranscricaoPalavra.Add(MontaFonemaComTempo('d')+' 50 90');
  // inserir i mudo após o d caso existir uma consoante
  if (pos(vPalavra[posletra + 1], CONSOANTES_semRL) > 0) then
    vTranscricaoPalavra.Add(MontaFonemaComTempo('imudo'));
  inc(posLetra);
end;

```

Quadro 18 – Regra de transcrição do fonema /d/

Com a inclusão de nomes de origem alemão-americano, a quantidade de encontros consonantais problemáticos aumentou, tornando-se necessária a implementação de novos tratamentos para encontros existentes na língua alemã, também resolvidos com a intercalação de vogais.

3.3.4 Obtenção das rotas

As rotas são retornadas pela Google Directions API em formato XML, onde suas solicitações são realizadas através de requisições HTTP. O método `GetURL` existente na classe `UGoogleMaps` é o responsável por montar a URL de requisição para o cálculo da rota, conforme parâmetros definidos/preenchidos pelo usuário na interface do software ou obrigatórios no funcionamento da API. Os parâmetros `origin`, `destination` e `sensor` têm obrigatoriedade na URL, já os parâmetros `language` e `mode` foram definidos como obrigatórios no software proposto. Sendo assim, `origin`, `destination` e `mode` são definidos pelo usuário, enquanto `sensor` e `language` são passados fixos na URL, conforme apresentado no Quadro 19, onde é exposto o código que monta a URL.

```

01 case prWay of
02   0: sWay := 'walking';
03   1: sWay := 'driving';
04 end;
05
06 Result :=
07   AnsiToUtf8('http://maps.google.com.br/maps/api/directions/xml?origin=' +
08             prOrigin + '&destination=' + prDestination + '&mode=' + sWay +
09             '&language=pt-BR&sensor=false');

```

Quadro 19 – Definição da URL de requisição

Ao calcular as rotas, deve-se especificar o modo de transporte a ser utilizado. Como padrão, caso não utilize o parâmetro `mode` (linha 08), as rotas são calculadas como `driving` (de carro). A Google Directions API suporta três modos de transporte, sendo que apenas dois são utilizados no software (linha 01 a 04):

- a) `driving`: ruas e rodovias utilizadas por veículos;
- b) `walking`: percursos para pedestres ou calçadas, quando existentes;

- c) `bicycling`: percursos para ciclistas ou ciclovias, atualmente disponível apenas nos Estados Unidos da América.

Várias sugestões de rotas podem ser retornadas caso o parâmetro `alternatives` na URL de envio estiver como `true`, caso contrário, apenas uma rota é retornada. No software proposto, este parâmetro não é enviado.

Após realizada a requisição das rotas, o XML retornado é tratado nos métodos `GetRouteXML` e `VerifyStatusMessage`. A estrutura do esquema XML é mostrada no Apêndice A, onde é apresentado o exemplo de um XML de retorno completo.

No primeiro passo, utilizando o método `VerifyStatusMessage`, é realizada a verificação do elemento `status`, que contém o `status` da requisição e pode ser um resultado válido ou inválido da rota pesquisada. Quando válido, o elemento possui o valor `OK` (linha 03) (Quadro 20). Caso contrário, no elemento serão retornados os códigos `NOT_FOUND`, `ZERO_RESULTS`, `MAX_WAYPOINTS_EXCEEDED`, `INVALID_REQUEST`, `OVER_QUERY_LIMIT`, `REQUEST_DENIED` ou `UNKNOWN_ERROR`, também tratados pelo software desenvolvido (GOOGLE, 2010b).

```

01 <?xml version="1.0" encoding="UTF-8"?>
02 <DirectionsResponse>
03 <status>OK</status>
04 <route>
05 <summary>BR-470</summary>
06 <leg>

```

Quadro 20 – Código de retorno apresentado no XML

O Quadro 21 apresenta o código do método `GetRouteXML`, responsável por juntar todas as informações do XML retornado uma `string`. No XML retornado podem existir zero ou mais rotas, definida(s) no(s) elemento(s) `route`. No protótipo, apenas uma é utilizada, caso exista (linhas 05 e 06 – Quadro 21). Assim que encontrado o elemento `route`, tem-se em seguida o elemento `leg`, podendo existir várias vezes se utilizado o parâmetro `waypoints`, que não é tratado pelo software. Cada elemento `leg` consiste em uma série de `steps` e outros campos, como `start_address`, `end_address`, `distance` e `duration`, apresentados em tela e posteriormente sintetizados pelo software (linhas 22 a 36). Os campos `distance` e `duration` são os valores totais de distância e duração entre os endereços de origem e destino dentro de cada elemento `leg`. Caso a distância e/ou duração seja desconhecida, não são apresentados valores nos campos. Latitude e longitude também são apresentadas nos campos `start_location` e `end_location` (linhas 13 a 20), onde ambos são armazenados e posteriormente utilizados para carregar o mapa em tela. No software desenvolvido, ambos os campos não possuem utilidade visual, sendo assim, não são tratados no método `GetRouteXML`.

```

01  nodeStatus := prXMLDoc.DocumentElement.ChildNodes.FindNode('status');
02  prError := VerifyStatusMessage(nodeStatus.Text);
03  // treat google maps api response
04
05  nodeRoute := prXMLDoc.DocumentElement.ChildNodes.FindNode('route');
06  if not (nodeRoute = nil) then
07  begin
08
09      nodeLeg := nodeRoute.ChildNodes.FindNode('leg');
10      if not (nodeLeg = nil) then
11      begin
12
13          nodeSLocation := nodeLeg.ChildNodes.FindNode('start_location');
14          if not (nodeSLocation = nil) then
15              pS := nodeSLocation.ChildNodes['lat'].Text + ',' +
16                  nodeSLocation.ChildNodes['lng'].Text;
17          nodeELocation := nodeLeg.ChildNodes.FindNode('end_location');
18          if not (nodeELocation = nil) then
19              pE := nodeELocation.ChildNodes['lat'].Text + ',' +
20                  nodeELocation.ChildNodes['lng'].Text;
21
22          sReturn := sReturn + 'Origem: ' +
23                      nodeLeg.ChildNodes['start_address'].Text + #13#10;
24          sReturn := sReturn + 'Destino: ' +
25                      nodeLeg.ChildNodes['end_address'].Text + #13#10;
26
27          nodeDistance := nodeLeg.ChildNodes.FindNode('distance');
28          if not (nodeDistance = nil) then
29              sReturn := sReturn + 'Distância: ' +
30                          AnsiUpperCase(nodeDistance.ChildNodes['text'].Text) + ' ' + #13#10;
31
32
33          nodeDuration := nodeLeg.ChildNodes.FindNode('duration');
34          if not (nodeDuration = nil) then
35              sReturn := sReturn + 'Duração: ' +
36                          nodeDuration.ChildNodes['text'].Text + ' ' + #13#10;
37
38          nodeStep := nodeLeg.ChildNodes.FindNode('step');
39          if not (nodeStep = nil) then
40          begin
41              nodeStep.ChildNodes.First;
42              while nodeStep <> nil do
43              begin
44                  nodeStepDuration := nodeStep.ChildNodes.FindNode('duration');
45                  if not (nodeStepDuration = nil) then
46                      sStepDuration := nodeStepDuration.ChildNodes['text'].Text;
47
48                  nodeStepDistance := nodeStep.ChildNodes.FindNode('distance');
49                  if not (nodeStepDistance = nil) then
50                      sStepDistance := nodeStepDistance.ChildNodes['text'].Text;
51
52                  sStepInstructions :=
53                      nodeStep.ChildNodes['html_instructions'].Text;
54                  if sStepInstructions <> '' then
55                  begin
56                      (* tag html_instructions response modified by google 11/09/2010
57                         iDistance := Pos('- siga',sStepInstructions);
58                         sStepInstructions := Copy(sStepInstructions, 1, iDistance); *)
59                      sStepInstructions := sStepInstructions + ' - siga ' +
60                          AnsiUpperCase(sStepDistance) + ' - '; // + dintance to go ahead
61                      sReturn := sReturn + sStepInstructions + #13#10;
62                  end;
63                  nodeStep := nodeStep.NextSibling;
64              end;
65          end;
66      end;
67  end;

```

Cada elemento no *array* de `steps` define um único passo da direção/rota calculada. No Quadro 21, entre as linhas 38 e 66, é possível verificar como são tratadas informações de cada passo, onde são concatenados ao campo `html_instructions`, os campos de duração e distância. Este campo possui informações dos passos a serem seguidos para se chegar ao destino. O mesmo pode ser visto no Quadro 22, onde tem-se o exemplo de um elemento `step` e seus campos.

```

<step>
  <travel_mode>DRIVING</travel_mode>
  <start_location>
    <lat>-26.8237400</lat>
    <lng>-49.2710800</lng>
  </start_location>
  <end_location>
    <lat>-26.8251100</lat>
    <lng>-49.2715400</lng>
  </end_location>
  <polyline>
    <points>j_vbDfgvkHpGzA</points>
    <levels>BB</levels>
  </polyline>
  <duration>
    <value>14</value>
    <text>1 min</text>
  </duration>
  <html_instructions>Siga na direção <b>sul</b> na <b>R. Mal. Deodoro da Fonseca</b> em direção à <b>R. Honduras</b>
  </html_instructions>
  <distance>
    <value>159</value>
    <text>0,2 km</text>
  </distance>
</step>

```

Quadro 22 – Elemento `step` e seus campos no XML de retorno

3.3.5 Identificação dos nomes

Na classe `UIdentifyGermanName` são identificados os nomes de origem alemão-americano. Os nomes só são identificados pelo método `Identifying` caso os mesmos iniciem com letra maiúscula. Assim, para cada palavra iniciada em letra maiúscula nos campos Endereço de origem (A), Endereço de destino (B) e Rota entre os pontos A e B, quando sintetizadas, são removidos os sinais de pontuação `, ; : . ? ! ... () " [] *`, utilizando o método `RemovePunctuation` da classe `UUtils`. O Quadro 23 apresenta o método `RemovePunctuation` utilizado antes da identificação dos nomes.

```

function TUtils.RemovePunctuation(prString: string): string;
begin
    { CEGALLA, pontuacao p. 62-70: , ; : . ? ! ... ( ) " [ ] * }

    prString := StringReplace(prString, ', ', ' ', [rfReplaceAll]);
    prString := StringReplace(prString, '; ', ' ', [rfReplaceAll]);
    prString := StringReplace(prString, ': ', ' ', [rfReplaceAll]);
    prString := StringReplace(prString, '. ', ' ', [rfReplaceAll]);
    prString := StringReplace(prString, '? ', ' ', [rfReplaceAll]);
    prString := StringReplace(prString, '! ', ' ', [rfReplaceAll]);
    prString := StringReplace(prString, '... ', ' ', [rfReplaceAll]);
    prString := StringReplace(prString, '( ', ' ', [rfReplaceAll]);
    prString := StringReplace(prString, ') ', ' ', [rfReplaceAll]);
    prString := StringReplace(prString, '" ', ' ', [rfReplaceAll]);
    prString := StringReplace(prString, '[' ', ' ', [rfReplaceAll]);
    prString := StringReplace(prString, ']', ' ', [rfReplaceAll]);
    prString := StringReplace(prString, '* ', ' ', [rfReplaceAll]);

    Result := prString;
end;

```

Quadro 23 – Método RemovePunctuation

O método `Identifying` utiliza de padrões para identificar nomes de origem alemão-americano, já descritos no Quadro 10, que são basicamente formados por sequências consonantais e vogais. O Quadro 24 apresenta alguns exemplos destes padrões implementados no software desenvolvido.

```

... if Copy(LowerCase(prText), iLength - 3, iLength) = 'bach' then
    Result := True
else if Copy(LowerCase(prText), iLength - 3, iLength) = 'mann' then
    Result := True
else if Copy(LowerCase(prText), iLength - 3, iLength) = 'burg' then
    Result := True
else if Copy(LowerCase(prText), iLength - 4, iLength) = 'buehl' then
    Result := True
else if Copy(LowerCase(prText), iLength - 6, iLength) = 'buehler' then
    Result := True
else if Copy(LowerCase(prText), iLength - 4, iLength) = 'hofel' then
    Result := True
else if Copy(LowerCase(prText), iLength - 5, iLength) = 'kofler' then
    Result := True
else if Copy(LowerCase(prText), iLength - 5, iLength) = 'kaufer' then
    Result := True
else if Copy(LowerCase(prText), iLength - 3, iLength) = 'horn' then
    Result := True
else if Copy(LowerCase(prText), iLength - 3, iLength) = 'wald' then
    Result := True
else if Copy(LowerCase(prText), iLength - 3, iLength) = 'berg' then
    Result := True

```

Quadro 24 – Método utilizado para identificar padrões em nomes

Alguns destes padrões não foram utilizados, como o ditongo /ei/. Nomes com /ei/ são, na maioria deles de origem alemã, como *Reimann*, *Reimers* e *Klein*, porém nem todos. Desta forma, nomes existentes no português, como Wanderlei e Arlei seriam definidos como de origem alemão-americano e teriam sua pronúncia alterada. Sendo assim, grande parte dos padrões foi retirada de literaturas, buscando contemplar a maior quantidade possível de nomes identificados nas ruas, principalmente nomes de origem alemão-americano existentes na

cidade de Blumenau, colonizada inicialmente por alemães.

No *site* da prefeitura municipal de Blumenau, foi disponibilizada pela Secretaria Municipal de Planejamento Urbano (2010) uma listagem das ruas do município, onde muitos dos padrões definidos no software podem ser encontrados.

3.3.6 Alteração dos nomes para o português

Quando identificados os nomes de origem alemão-americano, sua escrita é alterada pelo método `Modify`, existente na classe `UIIdentifyGermanName`. As alterações são necessárias para que seja possível reproduzir nomes na língua alemã, mantendo assim o banco de fonemas no português brasileiro BR3, utilizado no FurbTTS. Com a escrita “incorreta” dos nomes, consegue-se uma pronúncia similar à alemã, de modo que, nomes antes não entendidos, possam ser identificados com mais facilidade quando sintetizados. O Quadro 25 apresenta algumas alterações realizadas na escrita dos nomes.

```

01 prString := StringReplace(prString, 'ä', 'é', [rfReplaceAll, rfIgnoreCase]);
02 prString := StringReplace(prString, 'ei', 'ai', [rfReplaceAll, rfIgnoreCase]);
03 prString := StringReplace(prString, 'ie', 'ii', [rfReplaceAll, rfIgnoreCase]);
04 prString := StringReplace(prString, 'ey', 'ai', [rfReplaceAll, rfIgnoreCase]);
05 prString := StringReplace(prString, 'ay', 'ai', [rfReplaceAll, rfIgnoreCase]);
06 prString := StringReplace(prString, 'äu', 'ói', [rfReplaceAll, rfIgnoreCase]);
07 prString := StringReplace(prString, 'eu', 'ói', [rfReplaceAll, rfIgnoreCase]);
08 prString := StringReplace(prString, 'ö', 'e', [rfReplaceAll, rfIgnoreCase]);
09 prString := StringReplace(prString, 'ü', 'i', [rfReplaceAll, rfIgnoreCase]);
10
11 prString := StringReplace(prString, 'll', 'l', [rfReplaceAll, rfIgnoreCase]);
12 // Mueller, Müller, Rismiller
13 prString := StringReplace(prString, 'tt', 't', [rfReplaceAll, rfIgnoreCase]);
14 // Strattmann, Stuttts

```

Quadro 25 – Método utilizado para alterar a escrita dos nomes

No método, vogais com *Unlaut*, como a vogal /ü/, que não possuem som equivalente na língua portuguesa, são modificadas (linhas 01, 06, 08 e 09). Além das consoantes simples, existem ainda no alemão consoantes dobradas, como os exemplos /ll/, /mm/, /nn/, /pp/ e /tt/, encontradas em muitos dos nomes. Estas são então, substituídas por consoantes simples no software desenvolvido, conforme as linhas 11 e 13 do Quadro 25.

Na Figura 6 pode-se verificar, através da ferramenta Notepad++ (HO, 2010), um comparativo da rota entre os endereços Timbó - SC, Brasil e Furb - Campus IV - Complexo de Computação e Informática, antes (à esquerda) e depois (à direita) de alterada pelo método `Modify`.

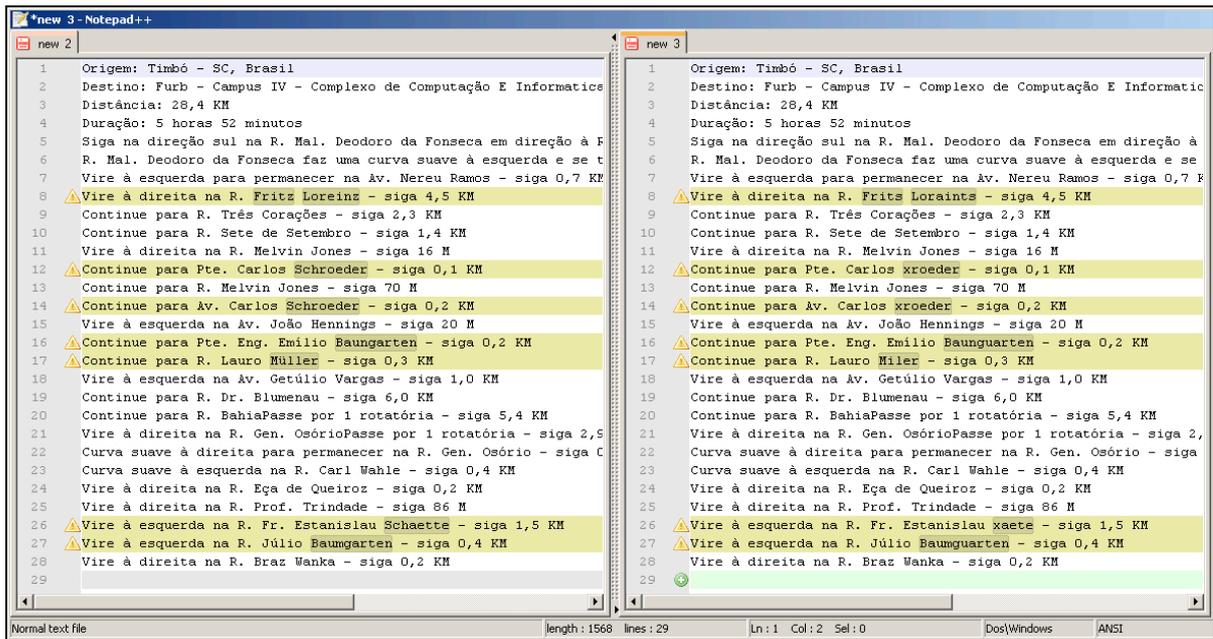


Figura 6 – Nomes alemão-americano alterados no método Modify

3.3.7 Interface do software

No desenvolvimento da interface, uma ampla quantidade de tratamentos foi realizada para que uma usabilidade simplificada e rápida fosse possível para usuários com deficiência visual. A maioria dos tratamentos foi necessária para o desenvolvimento da parte visual. Cores, botões, navegação e mensagens apresentadas em tela receberam uma atenção especial.

Para a parte não visual, os métodos `ProcessLetterPlusWord` e `GetLastWord`, existentes na classe `UTCCDesenv`, foram desenvolvidos para que fosse possível a síntese de letras, seguidas da síntese da palavra. No Quadro 26 é apresentado o método `ProcessLetterPlusWord`, responsável por sintetizar as letras e as palavras, quando digitadas pelo usuário em tempo de execução. Se a tecla espaço for pressionada, o método `GetLastWord` é acionado (linhas 03 a 07), retornando e sintetizando a última palavra escrita pelo usuário. Caso contrário, as letras são sintetizadas (linhas 23 a 39). No mesmo quadro, a linha 21 mostra o método `MBR_SetDurationRatio(0.8)`, utilizado para alterar a velocidade da fala para a síntese de letras.

```

01  if vScreenReader then
02  begin
03      if prKey = VK_SPACE then // space bar
04          begin
05              sWord := GetLastWord();
06              ProcessWords(sWord);
07          end
08      else if prKey = VK_BACK then // backspace key
09          begin
10              end
11          else
12              begin
13                  if Utils.GetCharFromVirtualKey(prKey) <> '' then
14                      begin
15                          if Utils.GetCharFromVirtualKey(prKey) <> '' then
16                              sChar := Utils.GetCharFromVirtualKey(prKey)[1];
17                          end
18                          else
19                              Exit;
20
21                          MBR_SetDurationRatio(0.8);
22
23                          if sChar in ['a'..'z'] + ['A'..'Z'] + ['0'..'9'] then
24                              begin
25                                  if (sChar = 'i') or (sChar = 'v') or (sChar = 'x')
26                                      or (sChar = '0') then
27                                      begin
28                                          if sChar = 'i' then
29                                              ProcessWords(AnsiUpperCase(sChar) + '-i');
30                                          if sChar = 'v' then
31                                              ProcessWords(AnsiUpperCase(sChar) + '-v');
32                                          if sChar = 'x' then
33                                              ProcessWords(AnsiUpperCase(sChar) + '-x');
34                                          if sChar = '0' then
35                                              ProcessWords('zero');
36                                          end
37                                      else
38                                          ProcessWords(AnsiUpperCase(sChar));
39                                  end;
40                              end;
41                          end;

```

Quadro 26 – Método utilizado para processar letras e palavras

3.3.8 Operacionalidade da implementação

A operacionalidade do software desenvolvido é bastante simplificada. O software foi desenvolvido para ser operado por pessoas com deficiência visual leve, moderada, severa, profunda (que compõem o grupo de visão subnormal ou baixa visão) e ausência total da resposta visual (cegueira). Foi atribuído o nome TTS Rota para o software.

Para usuários com deficiência visual leve, moderada, severa e profunda, foram utilizados em todo o software, padrões de fontes de tamanho superior aos normalmente utilizados, o que não impedirá de serem utilizados junto à parte sonora. Sempre que alterado o foco, uma borda na cor azul é adicionada ao redor do componente responsável pelo mesmo,

destacando e facilitando a localização em tela. Quando o componente possuir o fundo na cor branca, seu fundo é alterado, distinguindo-o dos demais. A navegação na tela é bastante simplificada e pode ser feita sem o uso do *mouse*, apenas utilizando as teclas do teclado: *tab*, para se dirigir ao próximo campo, e *shift+tab*, para o campo anterior.

Como estudo de caso é realizada a obtenção das rotas entre os endereços Rua José Kasteler, Jaraguá Esquerdo, Jaraguá do Sul - Santa Catarina e Rua Arduino Pradi, Jaraguá Esquerdo, Jaraguá do Sul - Santa Catarina.

Para se utilizar o software desenvolvido, deve ser instalado o sintetizador MBROLA. Ao executar o software é apresentada a tela da Figura 7, em conjunto à síntese do texto *abrindo TTS Rota*. O software sempre é aberto em tela cheia. Uma barra de tarefas pode ser vista logo à direita, contendo os botões *Pesquisar* (Figura 7(5)), *Sintetizar* (Figura 7(6)), *Atalhos* (Figura 7(7)), *Fechar* (Figura 7(8)), *Sobre* (Figura 7(9)) e *Mapa* (Figura 7(10)) e também um campo chamado *Leitor da tela* (Figura 7(11)), utilizado apenas para identificar se a função de leitura da tela está ativada (na cor verde), ou desativada (na cor vermelha).

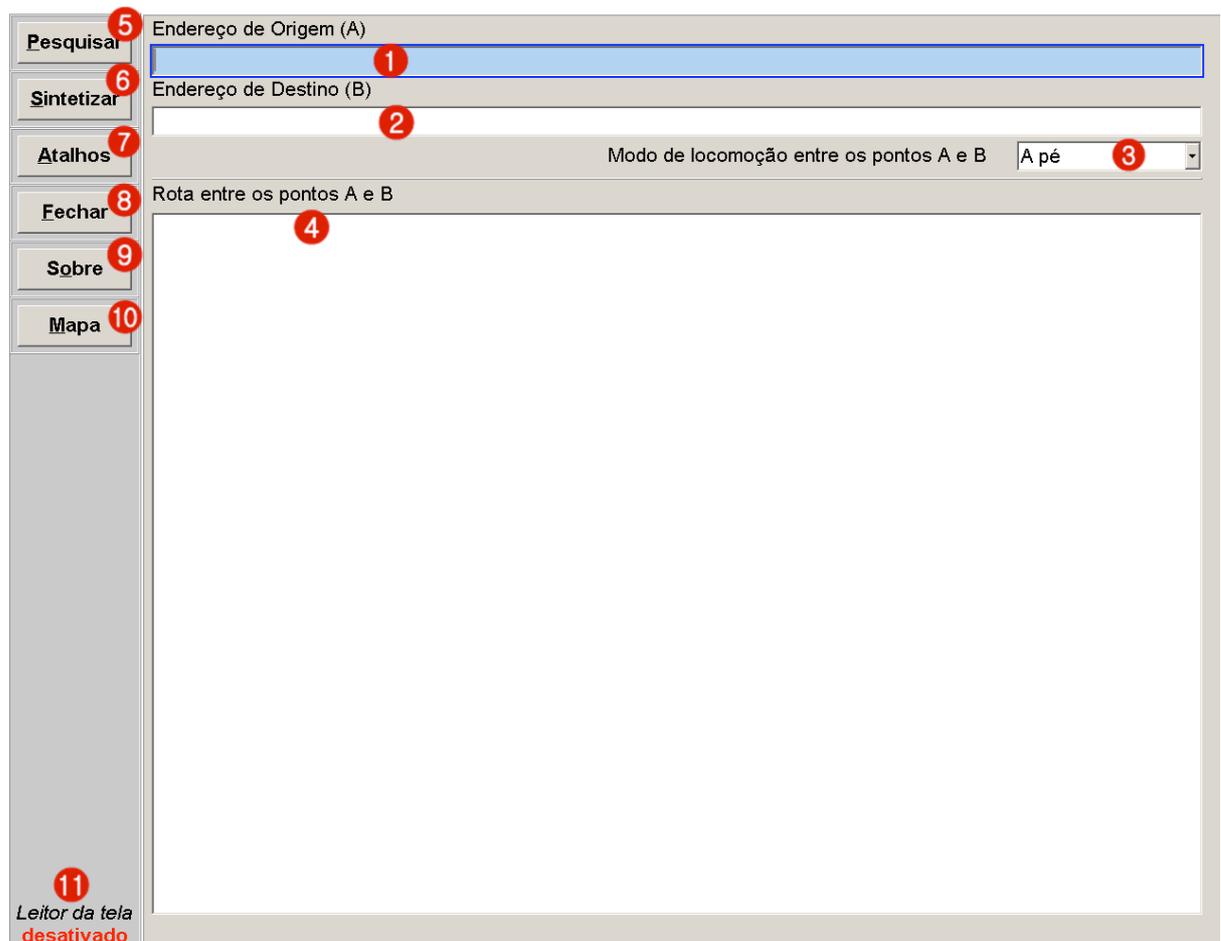


Figura 7 – Interface do software

Para iniciar a pesquisa das rotas, o usuário precisa possuir conexão com a Internet e

digitar ou colar, nos campos de Endereço de Origem (A) (Figura 7(1)) e Endereço de Destino (B) (Figura 7(2)) os respectivos endereços, onde os mesmos devem estar escritos em português, podendo haver nomes de origem alemão-americano. Sempre que realizada uma pesquisa, é verificada a conexão com a Internet, sendo que, quando não existir, uma mensagem é apresentada e sintetizada. O campo Modo de locomoção entre os pontos A e B (Figura 7(3)) já vem marcado, por padrão, com a opção *A pé*, podendo também ser alterado. Com isso, o usuário pode realizar a pesquisa das rotas através do botão *Pesquisar* (Figura 7(5)), onde também é realizada a síntese do texto *pesquisando a rota*. Caso encontrada a rota entre os endereços de origem e destino, o texto *rota encontrada* é sintetizado. O resultado da pesquisa é apresentado na Figura 8.

Pesquisar	Endereço de Origem (A) Rua José Kasteler, Jaraguá Esquerdo, Jaraguá do Sul - Santa Catarina
Sintetizar	Endereço de Destino (B) Rua Arduino Pradi, São Luis, Jaraguá do Sul - Santa Catarina
Atalhos	Modo de locomoção entre os pontos A e B <input type="text" value="A pé"/>
Fechar	Rota entre os pontos A e B
Sobre	Origem: R. José Kasteler - Jaraguá Esquerdo, Jaraguá do Sul - SC, 89253-270, Brasil Destino: R. Arduino Pradi - São Luis, Jaraguá do Sul - SC, 89253-670, Brasil Distância: 2,5 KM Duração: 30 minutos Siga na direção leste na R. Giovanni Vicenti em direção à R. João Januário Ayroso - siga 31 M Vire à direita na R. João Januário Ayroso - siga 0,2 KM Vire à direita para permanecer na R. João Januário Ayroso - siga 1,7 KM Vire à direita na R. Arduino Pradi - siga 0,6 KM
Mapa	

Leitor da tela desativado

Figura 8 – Resultado da pesquisa entre os pontos A e B

A rota pesquisa é apresentada no campo *Rota entre os pontos A e B* (Figura 7(4)), já destacado dentre os demais, podendo então ser sintetizada através do botão *Sintetizar* (Figura 7(6)). Caso o usuário deseje, pode digitar ou colar, no campo *Rota entre os pontos A e B* (Figura 7(4)), textos escritos em português, permitindo também nomes de origem alemão-americano, para serem sintetizados. Sempre que algum texto for sintetizado no campo, a velocidade da fala é diminuída, procurando melhorar o entendimento

do texto. Para cada tipo de síntese existente no software, são utilizadas diferentes velocidades, alteradas para uma melhora na usabilidade do mesmo. Assim, campos apenas para localização e informação, são sintetizados mais rapidamente que campos textuais.

Caso algum erro ocorra ao realizar a pesquisa, mensagens são apresentadas em tela e sintetizadas, como a mensagem apresentada na Figura 9, onde as coordenadas geográficas para os endereços não puderam ser encontradas.



Figura 9 – Exemplo de mensagem informativa ao realizar uma pesquisa

Se os endereços de origem ou de destino não estiverem preenchidos ao realizar a pesquisa, mensagens informativas são apresentadas em tela e sintetizadas para o usuário, conforme mostra a Figura 10.



Figura 10 – Mensagem para endereço de destino vazio ao realizar pesquisa de rota

Assim que a tecla OK for pressionada pelo usuário, o foco é direcionado para o campo Endereço de Destino (B) (Figura 7(2)), para então ser solicitada uma nova pesquisa.

Caso o leitor da tela (Figura 7(11)) estiver ativado, sempre que o usuário navegar entre os campos ou realizar alguma ação existente no software, os mesmos são sintetizados, permitindo assim a localização em tela, facilitando a utilização para usuários com ausência total de visão. Como exemplo, na Figura 11 é feita a navegação entre os campos Endereço de Origem (A) (Figura 7(1)) e Endereço de Destino (B) (Figura 7(2)). Ao entrar no primeiro campo é sintetizado: *endereço de origem, Rua José Kasteler, Jaraguá Esquerdo, Jaraguá do Sul – Santa Catarina*. Quando alterado o foco para o segundo campo, a síntese é: *endereço de destino, campo vazio*.

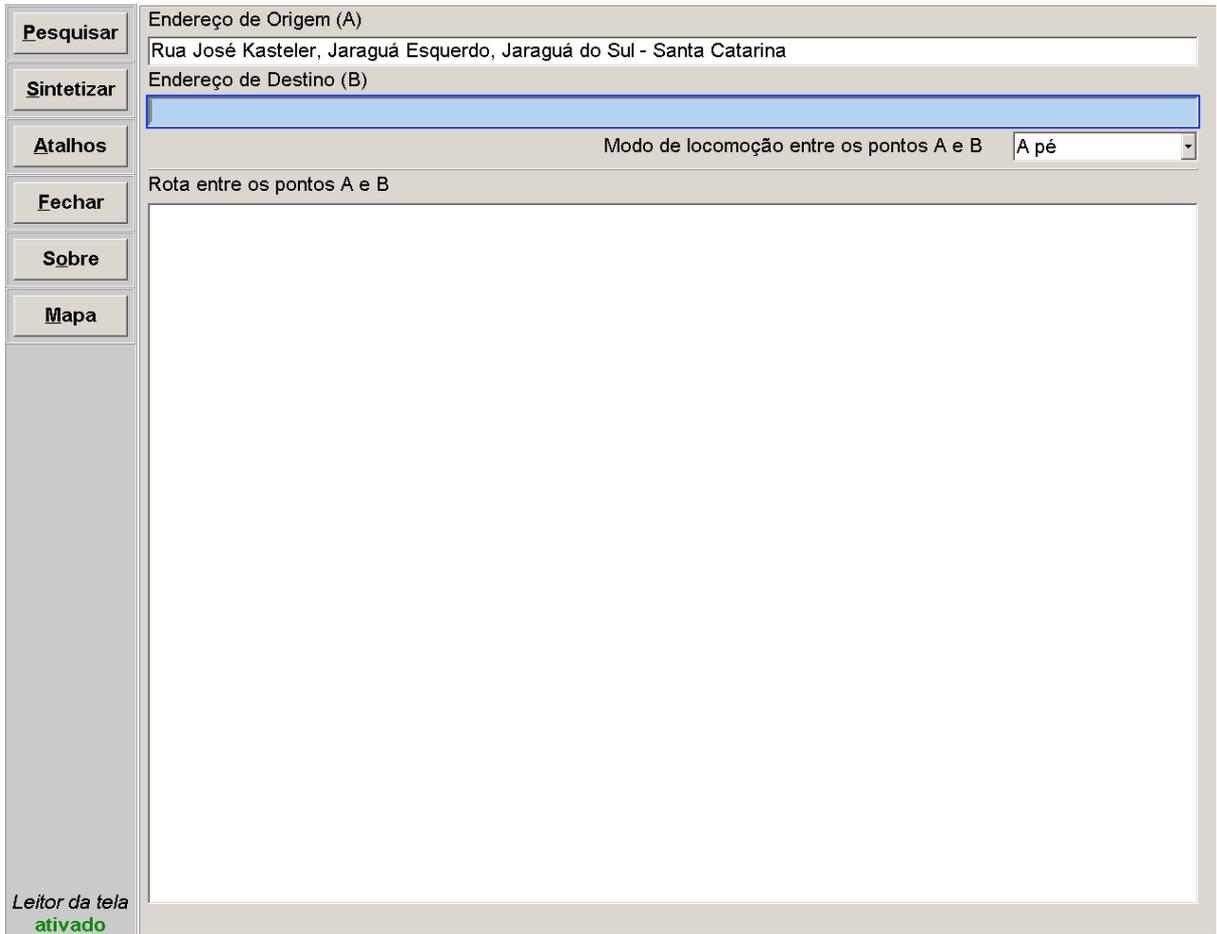


Figura 11 – Leitor da tela

Também, quando ativado o leitor da tela, para os campos *Endereço de Origem (A)* (Figura 7(1)), *Endereço de Destino (B)* (Figura 7(2)) e *Rota entre os campos A e B* (Figura 7(4)), sempre que digitada uma letra ou número, os mesmos são sintetizados, e ao final de uma palavra, assim que pressionada a tecla espaço (identificador do fim da palavra), a palavra é sintetizada. Assim, por exemplo, para a palavra *rua*, ao pressionar a tecla *r*, a síntese é realizada. Continuando a palavra, são pressionadas e sintetizadas as teclas *u* e *a*, e assim que pressionada a tecla de espaço, segue a síntese da palavra completa: *rua*.

A Figura 12 apresenta a tela com as teclas de atalhos, acionada através do botão *Atalhos* (Figura 7(6)). Quando aberta a tela, o texto *abrindo atalhos* é sintetizado. Os atalhos foram criados com o intuito de agilizar o processo para o usuário. As teclas de atalhos existentes no software são:

- a) ativar/desativar leitor da tela (teclas `ctrl+l`): quando realizada a navegação entre os campos em tela, caso ativado o leitor da tela, os mesmos são sintetizados;
- b) parar fala (teclas `ctrl+p`): utilizada em textos muito grandes, quanto torna-se necessário que a fala seja parada durante a síntese;
- c) sintetizar texto (teclas `ctrl+s`): utilizada para sintetizar o texto onde o campo

- focado possuir o fundo na cor branca;
- d) selecionar todo o texto (teclas `ctrl+t`): seleciona todo o texto e o sintetiza;
 - e) tela de atalho (teclas `ctrl+a`): apresenta uma janela com as teclas de atalhos existentes;
 - f) pesquisar rota (teclas `ctrl+p`): realiza a pesquisa da rota entre os endereços de origem e destino, quando existentes;
 - g) navegar entre os campos (tecla `tab`): navegação entre os campos sem utilização do *mouse*.

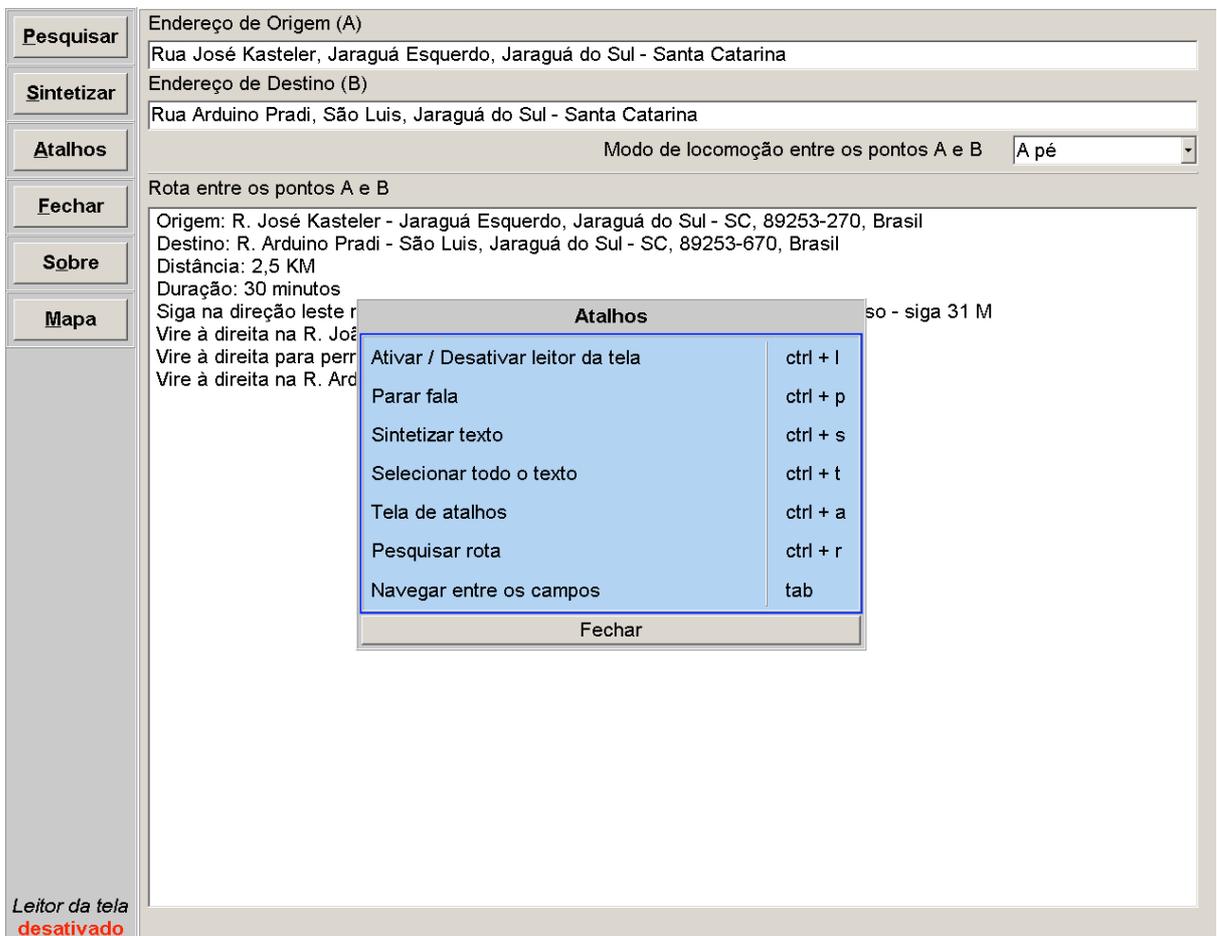


Figura 12 – Tela de atalhos

O botão `Fechar` (Figura 7(8)) é utilizado para fechar o software TTS Rota. Assim que acionado o botão, a síntese do texto *fechando TTS Rota* é realizada e, quando concluída, o software é fechado.

Na Figura 13 é apresentada a tela `Sobre` o TTS Rota, onde o foco está no campo `Fechar`. A tela contém informações sobre o software desenvolvido e contato. A tela é acionada através do botão `Sobre` (Figura 7(9)) na barra de ferramentas. Assim que aberta a tela, o texto *abrindo sobre o TTS Rota* é sintetizado. Caso o leitor da tela esteja ativado, o texto apresentado em tela também é sintetizado, sem a necessidade de utilizar as teclas

ctrl+s.

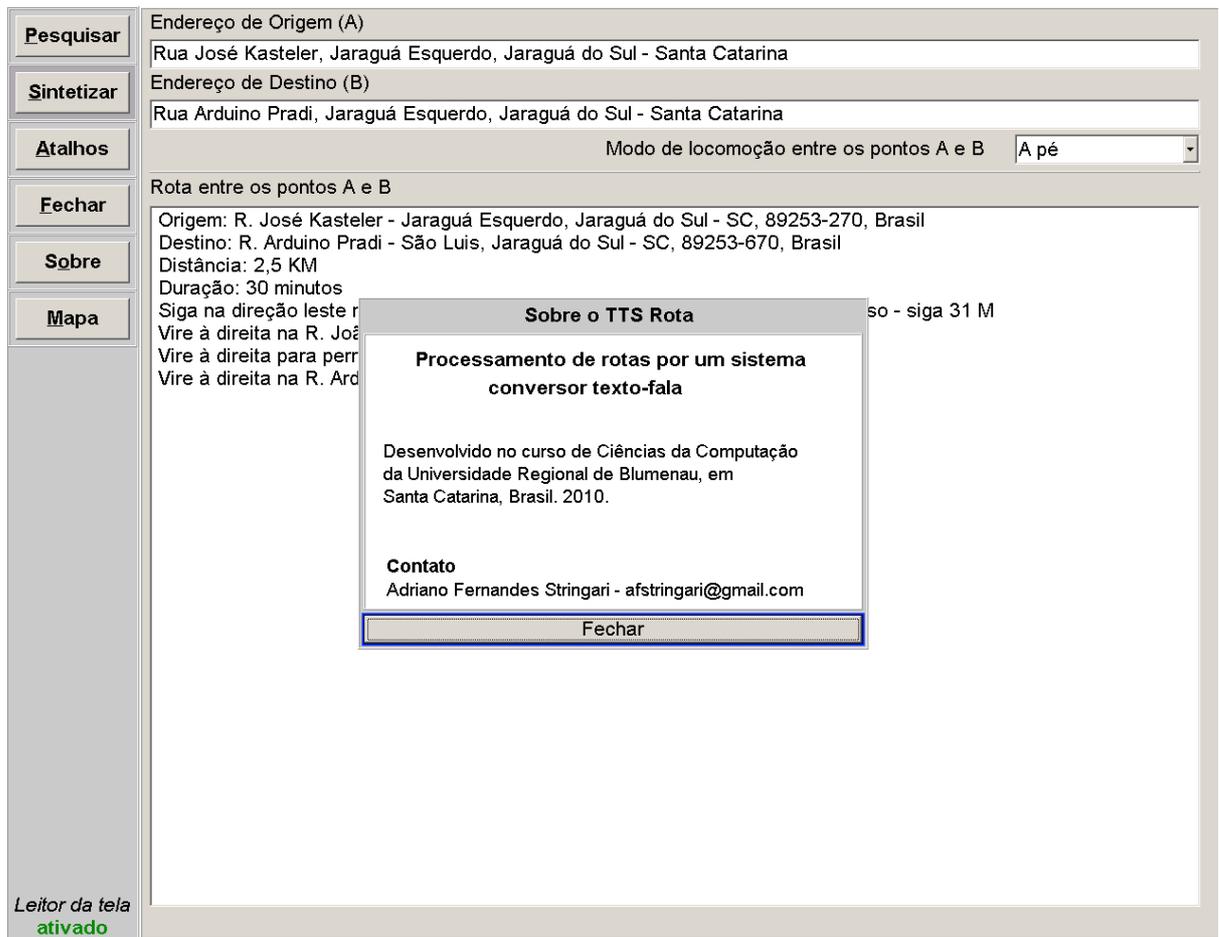


Figura 13 – Sobre o TTS Rota

O mapa da rota também é apresentado pelo software, onde o mesmo pode ser visualizado pelo botão `Mapa` (Figura 7(10)). A apresentação do mapa no software tornou-se necessária devido a uma obrigatoriedade nos termos de uso da Google Directions API. A Figura 14 apresenta o mapa da rota entre os endereços de origem e destino, pesquisado através da latitude e longitude dos endereços informados. O mapa só é visualizado caso uma pesquisa já tenha sido realizada, sendo utilizado apenas para visualização, não podendo ser editado pelo usuário no software.

Pesquisar
Sintetizar
Atalhos
Fechar
Sobre
Mapa

Web Imagens Vídeos Mapas Notícias Orkut Gmail mais ▾

Google maps Brasil

from: -26.5034600,-49.0931400 to: -26.5042800,-49.1107200

Encontre empresas, endereços e locais de seu interesse.

Como chegar Meus mapas

Carro Ônibus Pessoa

Verificar destino

Versão beta da rota a pé.
 Seja cuidadoso – Este trajeto pode não ter calçadas ou caminhos de pedestres.

Rota a pé para R. Arduino Pradi

Trajetos sugeridos

R. João Januário Ayroso e R. Arduino Pradi	30 minutos	2,5 km
R. Christina Enriconi Marcatto	31 minutos	2,5 km
R. João Januário Ayroso	33 minutos	2,7 km

R. Arduino Pradi

- Siga na direção **leste** na **R. Giovanni Vicenti** em direção à **R. João Januário Ayroso** 31 m
- Vire à **direita** na **R. João Januário Ayroso** 200 m
- Vire à **direita** para permanecer na **R. João Januário Ayroso** 1,7 km
- Vire à **direita** na **R. Arduino Pradi** 600 m

R. Arduino Pradi

Arduino Kit Iniciante Links patrocinados
 Aprenda eletrônica de um modo fácil
 Experimentos práticos para montar!
kitiniciante.makexora.net

500 m
 1000 pés

©2010 Google - Dados cartográficos ©2010 MapLink - [Termos de Uso](#)

Leitor da tela desativado

Fechar

Figura 14 – Mapa da rota entre os pontos A e B

3.4 RESULTADOS E DISCUSSÃO

O presente trabalho apresentou o desenvolvimento de um software para o processamento de rotas para um sistema conversor texto-fala. Nele as rotas foram obtidas através do Google Directions API e tratadas para que pudessem ser sintetizadas pela biblioteca FurbTTS. Tornou-se importante efetuar um tratamento no texto de entrada, já que o FurbTTS não processa corretamente nomes próprios, o que pode ocasionar uma síntese não adequada das rotas. Porém, várias razões tornam a correta pronúncia dos nomes difícil, pois estes podem ser de origem etimológica muito diversa. Ainda, o número de nomes diferentes é muito amplo, não existe uma lista de nomes única e existem muitas pronúncias que contrariam padrões fonológicos, como pronúncias alternativas para determinadas sequências de grafemas.

Um dos problemas encontrados foi a identificação da origem dos nomes através de sua

escrita. Alguns testes foram realizados com a ferramenta LangID (OLIVATO, 2009), uma ferramenta *web* que permite a identificação de qual linguagem o texto pertence. Em contato com o desenvolvedor da ferramenta, o mesmo descreveu que o sistema não consegue reconhecer nomes corretamente e que, quando utilizada para o reconhecimento de linguagem apenas em palavras, a ferramenta geralmente falha completamente. Sendo assim, optou-se por restringir os nomes que seriam identificados no software proposto, sendo eles, nomes em português e alemão-americano, mais comuns na cidade de Blumenau, colonizada no início por alemães.

Após a definição da origem dos nomes nos quais o software estaria trabalhando, outro problema encontrado foi a necessidade de se continuar utilizando o banco de fonemas português BR3, utilizado junto ao sintetizador MBROLA, implementado na biblioteca FurbTTS. Nomes de ruas, avenidas, logradouros, bairros, largos e praças teriam de ser sintetizados em alemão, enquanto que os demais passos da rota em português. Com isso, a troca do banco de fonemas BR3 para o DE8, no decorrer da síntese, dificultaria o entendimento do texto para deficientes visuais.

Assim, o problema foi apresentado à professora Valéria Contrucci de Oliveira Mailer, do curso de Licenciatura em Alemão da Universidade Regional de Blumenau (FURB), onde a mesma o apresentou aos seus alunos. Uma possível solução foi lançada: uma alteração na escrita dos nomes para que os mesmos possuíssem uma correta pronúncia. Com a ajuda da professora Ana Andréia Karnopp Brüske, de Joinville, foram levantadas as principais alterações necessárias para a pronúncia na região norte de Santa Catarina. Embora nem todas as regras da gramática alemã puderam ser implementadas, uma grande quantidade de nomes existentes na listagem de ruas da cidade de Blumenau, antes não entendidos na síntese da rota, puderam ser compreendidos.

4 CONCLUSÕES

O trabalho aqui apresentado refere-se ao desenvolvimento de um software para o processamento de rotas para um sistema conversor texto-fala. O software disponibiliza uma interface mais acessível para deficientes visuais, possuindo uma entrada em forma de texto de uma localização origem e uma localização destino (rua, cidade, estado, país); obtém a rota correspondente à localização de origem e destino através do Google Directions API; efetua um tratamento mais adequado para que a rota possa ser sintetizada pelo FurbTTS; efetua o pré-processamento, a análise linguística, a transcrição fonética e o processamento prosódico tanto das localizações de origem/destino quanto da(s) possível(is) rota(s); identifica nomes de origem alemão-americano e efetua a síntese da fala da entrada e da saída, procurando sintetizar nomes alemão-americano com sua pronúncia correta.

Haja vista que a correta pronúncia dos nomes é um dos grandes desafios para sistemas conversores texto-fala, os resultados finais foram satisfatórios, pois foi possível realizar a síntese de rotas onde nomes de origem alemão-americano, antes não sintetizados, ou sintetizados com pronúncia incorreta, puderam ser sintetizados de forma a serem compreendidos.

A maior dificuldade dentro do trabalho foi pesquisar regras gramaticais da língua alemã e, principalmente, identificar nomes de origem alemã, pois em muitos casos não existem normas ou regras para sua identificação. Desta forma, tornou-se necessário um estudo da língua alemã, sua gramática e seus nomes. Muitas das regras para identificação dos nomes, definidas e implementadas no trabalho, foram geradas a partir de uma análise a dicionários de nomes alemão-americano.

Quanto às tecnologias utilizadas, o ambiente de programação Delphi mostrou-se adequado, sendo que apenas a utilização de seus componentes nativos foi suficiente para a construção do software. Já a Google Directions API, devido a uma obrigatoriedade encontrada nos termos de utilização da mesma, exigiu a implementação do mapa das rotas obtidas, mesmo não havendo a necessidade de apresentação no software.

Espera-se que o software seja operado por pessoas com deficiência visual leve, moderada, severa, profunda (que compõem o grupo de visão subnormal ou baixa visão) e também para usuários com ausência total da resposta visual (cegueira), onde os mesmos possam ter conhecimento de locais por onde desejem passar.

A síntese de nomes continua sendo uma limitação no trabalho. Para nomes não

identificados como de origem alemã, e que possuem uma sequência inválida de letras se comparados com as normas da língua portuguesa, podem ocorrer erros durante o processamento linguístico ou durante a síntese, não permitindo que o software funcione corretamente. Também podem existir nomes onde não ocorram erros, mas que sua pronúncia não seja compatível com a língua a qual pertence.

Apesar das limitações do software, conseguiu-se uma síntese muito parecida para nomes de origem alemã. Assim, para deficientes visuais, torna-se perceptível a existência de nomes de ruas durante a síntese, possibilitando a utilização da ferramenta para a obtenção de rotas na cidade de Blumenau.

4.1 EXTENSÕES

Como sugestão para trabalhos futuros, alguns pontos podem ser melhorados e incrementados, sendo eles:

- a) efetuar um tratamento mais adequado no texto de entrada, para que quando encontradas palavras em outros idiomas ou que contenham erros ortográficos e gramaticais, seja possível continuar o processamento das demais palavras;
- b) desenvolver um módulo para permitir que o software receba comandos de voz, facilitando assim sua usabilidade;
- c) permitir a identificação de nomes em outras línguas, possibilitando a pronúncia correta dos mesmos;
- d) reescrever o software para que possa ser utilizado em plataformas móveis;
- e) permitir que alterações na escrita dos nomes alemão-americano possam ser externamente parametrizadas;
- f) permitir que padrões utilizados para a identificação de nomes alemão-americano possam ser externamente parametrizadas;
- g) possibilitar que para cada trajeto que compõe uma rota, a síntese possa ser pausada, repetida ou pulada.

REFERÊNCIAS BIBLIOGRÁFICAS

ALMEIDA, Napoleão M. **Gramática metódica da língua portuguesa**. 22. ed. São Paulo: Saraiva, 1969.

APPLE INCORPORATION. **VoiceOver**. [S.l.], 2010. Disponível em: <<http://www.apple.com/br/accessibility/voiceover/>>. Acesso em: 23 out. 2010.

BORGES, José A. **DOSVOX**: uma nova realidade educacional para deficientes visuais. [Rio de Janeiro], [2002]. Disponível em: <<http://intervox.nce.ufrj.br/dosvox/textos.htm>>. Acesso em: 25 out. 2010.

CAMPANA, Milton. **Alemão para brasileiros**. 2. ed. São Paulo: Pioneira, 1987.

CEGALLA, Domingos P. **Novíssima gramática da língua portuguesa**. 26. ed. São Paulo: Companhia Editora Nacional, 1985.

_____. **Novíssima gramática da língua portuguesa**. 40. ed. São Paulo: Companhia Editora Nacional, 1997.

COSTA, Denis R. **BR3 database**: brazilian portuguese male. Release 2.021. [S.l.], 2005. Disponível em: <<http://tcts.fpms.ac.be/synthesis>>. Acesso em: 02 nov. 2010.

DUTOIT, Thierry D. et al. **The MBROLA project**: towards a freely available multilingual speech synthesizer. [Bélgica], 2005. Disponível em: <http://tcts.fpms.ac.be/synthesis/mbrola/mbrola_entrypage.html>. Acesso em: 02 abr. 2010.

FERREIRA, Aurélio B. H. **Pequeno dicionário brasileiro da língua portuguesa**. 11. ed. Rio de Janeiro: Civilização Brasileira, 1991.

FRANZEN, Evandro. **Estudo e implementação da programação genética para síntese de fala**. 2002. 113 f. Dissertação (Mestrado em Ciência da Computação) – Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre.

FREEDOM SCIENTIFIC. **Screen reading software documentation**. [S.l.], out. 2009. Disponível em: <<http://www.freedomscientific.com/documentation/screen-readers.asp>>. Acesso em: 26 mar. 2010.

GOMES, Leandro C. T. **Sistema de conversão texto-fala para a língua portuguesa utilizando a abordagem de síntese por regras**. 1998. 107 f. Dissertação (Mestrado em Engenharia Elétrica) – Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas.

GOOGLE. **Google Maps Brasil**. [S.l.], 2010a. Disponível em: <<http://maps.google.com.br>>. Acesso em: 26 mar. 2010.

_____. **The Google Directions API**. [S.l.], 2010b. Disponível em: <<http://code.google.com/apis/maps/documentation/directions>>. Acesso em: 16 out. 2010.

HO, Don. **Notepad++**. [S.l.], 2010. Disponível em: <<http://notepad-plus-plus.org/>>. Acesso em: 1 out. 2010.

INSTITUTO BENJAMIN CONSTANT. **Seja bem vindo ao nosso portal braille...** [Rio de Janeiro], 2005. Disponível em: <<http://www.ibr.gov.br/index.php?catid=67&itemid=397>>. Acesso em: 19 out. 2010.

IUPUI UNIVERSITY LIBRARY. **German-americans and their contributions to the american mainstream culture: german names and words**. [S.l.], 2009. Disponível em: <<http://liberalarts.iupui.edu/maxkade/nameword/nameword.html>>. Acesso em: 2 nov. 2010.

JONES, George F. **German-american names**. 3. ed. Baltimore: Genealogical Publishing Company, 2006.

MACHADO, Lucas. Síntese de voz para produção de livros falados e inclusão social para deficientes visuais. In: FEIRA DE INICIAÇÃO CIENTÍFICA, 16., 2007, Porto Alegre. **Anais eletrônicos...** Porto Alegre: UFRGS, 2007. Não paginado. Disponível em: <<http://ufrgsweb.ufrgs.br/node/247?page=1>>. Acesso em: 04 set. 2010.

MORIMOTO, Carlos E. **Uma introdução ao GPS**. [S.l.], jan. 2009. Disponível em: <<http://www.guiadohardware.net/artigos/gps/>>. Acesso em: 04 abr. 2010.

OECHSLER, Thiago M. **Processamento de texto escrito em linguagem natural para um sistema conversor texto-fala**. 2009. 73 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

OLIVATO, Andrea. **LangId: identify your language**. [S.l.], 2009. Disponível em: <<http://langid.net/identify-language-from-api.html>>. Acesso em: 05 set. 2010.

ORACLE CORPORATION. **NetBeans IDE**. [S.l.], 2010. Disponível em: <<http://netbeans.org/>>. Acesso em: 30 out. 2010.

OSTERMANN FILHO, Paulo E. **Desenvolvimento de regras de pronúncia para síntese de fala em língua portuguesa**. 2002. 129 f. Dissertação (Mestrado em Ciência da Computação) – Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre.

SAINTY, Guy S. **The holy roman empire**. [S.l.], [1992]. Disponível em: <<http://www.chivalricorders.org/nobility/holyroman/>>. Acesso em: 25 out. 2010.

SCARTON, Gilberto; SMITH, Marisa M. **Manual de redação da PUCRS**. [Porto Alegre], [2010?]. Disponível em: <<http://www.pucrs.br/manualred/abreviaturas.php>>. Acesso em: 12 set. 2010.

SCHÜTZ, Ricardo. **A importância da pronúncia**. [S.l.], abr. 2008. Disponível em: <<http://www.sk.com.br/sk-pron.html>>. Acesso em: 20 out. 2010.

SIMÕES, Flávio O. **Implementação de um sistema de conversão texto-fala para o português do Brasil**. 1999. 185 f. Dissertação (Mestrado em Engenharia Elétrica) – Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas.

SECRETARIA MUNICIPAL DE PLANEJAMENTO URBANO. **Blumenau**: listagem de ruas. [Blumenau], 2010. Disponível em: <<http://www.blumenau.sc.gov.br/gxpsites/hgxpp001.aspx?1,13,319,O,P,0,MNU;E;95;8;MNU;,>>>. Acesso em: 05 out. 2010.

STEMMER, Gaspar H. **Um pouco sobre a grafia dos nomes nas colônias alemãs do RS**. [S.l.], [1999?]. Disponível em: <http://buratto.org/gens/gn_nomesalemaes.html>. Acesso em: 5 nov. 2010.

TAKANO, Rafael H. **Motor de jogos 3D para o Iphone OS**. 2009. 50 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

THE OMNI GROUP. **OmniGraffle**. [S.l.], 2010. Disponível em: <<http://www.omnigroup.com/applications/OmniGraffle>>. Acesso em: 30 out. 2010.

WEBAIM. **Using JAWS to evaluate web accessibility**. [S.l.], abr. 2010. Disponível em: <<http://webaim.org/articles/jaws/>>. Acesso em: 4 abr. 2010.

WELKER, Herbert A. **Gramática alemã**. Brasília: EdUNB, 1992.

APÊNDICE A – Exemplo de XML retornado pelo Google Directions API

No Quadro 27 é apresentado o XML retornado da Google Directions API entre os endereços Rua José Kasteler, Jaraguá Esquerdo, Jaraguá do Sul - Santa Catarina e Rua Arduino Pradi, Jaraguá Esquerdo, Jaraguá do Sul - Santa Catarina.

```
<?xml version="1.0" encoding="UTF-8"?>
<DirectionsResponse>
  <status>OK</status>
  <route>
    <summary>R. João Januário Ayroso e R. Arduino Pradi</summary>
    <leg>
      <step>
        <travel_mode>DRIVING</travel_mode>
        <start_location>
          <lat>-26.5034600</lat>
          <lng>-49.0931400</lng>
        </start_location>
        <end_location>
          <lat>-26.5034600</lat>
          <lng>-49.0928300</lng>
        </end_location>
        <polyline>
          <points>rmw`DbosjH?}@</points>
          <levels>BB</levels>
        </polyline>
        <duration>
          <value>4</value>
          <text>1 min</text>
        </duration>
        <html_instructions>Siga na direção <b>leste</b> na <b>R.
          Giovanni Vicenti</b> em direção à <b>R. João
          Januário Ayroso</b></html_instructions>
        <distance>
          <value>31</value>
          <text>31 m</text>
        </distance>
      </step>
      <step>
        <travel_mode>DRIVING</travel_mode>
        <start_location>
          <lat>-26.5034600</lat>
          <lng>-49.0928300</lng>
        </start_location>
        <end_location>
          <lat>-26.5051200</lat>
          <lng>-49.0933900</lng>
        </end_location>
        <polyline>
          <points>rmw`DdmsjH~CX|@f@1B1@</points>
          <levels>B??B</levels>
        </polyline>
        <duration>
          <value>43</value>
          <text>1 min</text>
        </duration>
        <html_instructions>Vire à <b>direita</b> na <b>R. João
          Januário Ayroso</b></html_instructions>
      </step>
    </leg>
  </route>

```

Quadro 27 – XML de retorno do Google Directions API

```

<distance>
  <value>195</value>
  <text>0,2 km</text>
</distance>
</step>
<step>
<travel_mode>DRIVING</travel_mode>
<start_location>
  <lat>-26.5051200</lat>
  <lng>-49.0933900</lng>
</start_location>
<end_location>
  <lat>-26.5093300</lat>
  <lng>-49.1092200</lng>
</end_location>
<polyline>
  <points>~ww`DtpsjH^rAr@|AX|An@hIvArEfDhH~CrNh@~GhAbF?|BStAUxECzDLrAd@`
  BpClF</points>
  <levels>B?????@?????@?B</levels>
</polyline>
<duration>
  <value>152</value>
  <text>3 minutos</text>
</duration>
<html_instructions>Vire à <b>direita</b> para permanecer na
  <b>R. João Januário
  Ayroso</b></html_instructions>
<distance>
  <value>1686</value>
  <text>1,7 km</text>
</distance>
</step>
<step>
<travel_mode>DRIVING</travel_mode>
<start_location>
  <lat>-26.5093300</lat>
  <lng>-49.1092200</lng>
</start_location>
<end_location>
  <lat>-26.5042800</lat>
  <lng>-49.1107200</lng>
</end_location>
<polyline>
  <points>hrx`DrsvjHyG`B{Fx@{MnC</points>
  <levels>B??B</levels>
</polyline>
<duration>
  <value>99</value>
  <text>2 minutos</text>
</duration>
<html_instructions>Vire à <b>direita</b> na <b>R. Arduino
  Pradi</b></html_instructions>
<distance>
  <value>581</value>
  <text>0,6 km</text>
</distance>
</step>
<duration>
  <value>298</value>
  <text>5 minutos</text>
</duration>
<distance>
  <value>2493</value>
  <text>2,5 km</text>
</distance>
<start_location>

```

Quadro 27 – XML de retorno do Google Directions API (continuação)

```

    <lat>-26.5034600</lat>
    <lng>-49.0931400</lng>
  </start_location>
</end_location>
    <lat>-26.5042800</lat>
    <lng>-49.1107200</lng>
</end_location>
<start_address>R. José Kasteler - Jaraguá Esquerdo, Jaraguá do Sul - SC, 89253-
    270, Brasil</start_address>
<end_address>R. Arduino Pradi - São Luis, Jaraguá do Sul - SC, 89253-670,
    Brasil</end_address>
</leg>
<copyrights>Dados cartográficos ©2010 MapLink</copyrights>
<overview_polyline>
  <points>rmw`DbosjH?}@~CX|@f@lBl@^rAr@|AX|An@hIvArEfDhH~CrNh@~GhAbF?|BStAUxECz
    DLrAd@`BpClFyG`B{Fx@{MnC</points>
  <levels>B????@?@????@????@?@??B</levels>
</overview_polyline>
</route>
</DirectionsResponse>

```

Quadro 27 – XML de retorno do Google Directions API (continuação)