

UNIVERSIDADE DE TAUBATÉ
Gilza Paim Mandelman

**TRANSCRIÇÃO EM TEMPO REAL DE
TEXTOS UTILIZANDO UM DICIONÁRIO
FONÉTICO**

Taubaté – SP
2011

UNIVERSIDADE DE TAUBATÉ
Gilza Paim Mandelman

TRANSCRIÇÃO EM TEMPO REAL DE TEXTOS
UTILIZANDO UM DICIONÁRIO FONÉTICO

Dissertação apresentada para obtenção do
Título de Mestre pelo Curso de Mestrado
Profissional em Engenharia Mecânica do
Departamento de Engenharia Mecânica da
Universidade de Taubaté - UNITAU.
Área de Concentração: Automação
Orientador: Prof. Dr. Marcio Abud Marcelino

Taubaté – SP
2011

Ficha catalográfica

Mandelman, Gilza Paim

Transcrição em tempo real de textos utilizando um dicionário fonético / Gilza Paim Mandelman. – 2011.

88f. : il.

Dissertação (mestrado) – Universidade de Taubaté, Departamento de Engenharia Mecânica, área de concentração: Automação, 2011.

Orientação: Prof. Dr. Marcio Abud Marcelino, Departamento de Engenharia Mecânica.

1. Acessibilidade. 2. “Brazilês”. 3. Reconhecimento automático de voz. I. Título.

GILZA PAIM MANDELMAN

**TRANSCRIÇÃO EM TEMPO REAL DE TEXTOS UTILIZANDO UM
DICIONÁRIO FONÉTICO**

Dissertação apresentada para obtenção do
Título de Mestre pelo Curso de Mestrado
Profissional em Engenharia Mecânica do
Departamento de Engenharia Mecânica da
Universidade de Taubaté- UNITAU.
Área de Concentração: Automação
Orientador: Prof. Dr. Márcio Abud Marcelino

Data: 03/09/2011

Resultado: **APROVADA**

BANCA EXAMINADORA

Prof. Dr. Marcio Abud Marcelino - Universidade de Taubaté e
FEG/UNESP – Universidade Estadual Paulista

Assinatura _____

Prof. Dr. João Bosco Gonçalves - Universidade de Taubaté

Assinatura _____

Prof. Dr. Francisco Antonio Lotufo – FEG/UNESP – Universidade
Estadual Paulista

Assinatura _____

Dedico este trabalho aos meus pais Gil Vianna Paim e Zayde Morenz Paim (*in memoriam*) que mostraram o caminho para chegar a este intento.

Ao meu marido Marcio, minhas
filhas Cláudia e Luciana.

AGRADECIMENTOS

Ao Professor Dr. Marcio Abud Marcelino, pela habilidade com que orientou este trabalho.

Aos colegas, professores e funcionários da UNITAU e do IFSP pelo apoio e acompanhamento durante todo o desenvolvimento do mestrado, especialmente ao colega prof. Mestre Renato Fernandez.

Aos gestores do Instituto Federal de Ciência e Tecnologia do Estado de São Paulo – IFSP, especialmente do Campus São Paulo, pela bolsa que forneceram.

A Cláudia Paim Mandelman pela revisão ortográfica (português e inglês).

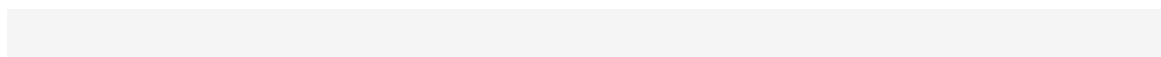
“O certo em literatura é escrever com o mínimo possível de literatura. (...) a mim me salvaram as crianças. De tanto escrever para elas, simplifiquei-me.”

Monteiro Lobato

RESUMO DO TRABALHO

Na busca de desenvolver uma técnica facilitadora do processo de reconhecimento automático da voz para transcrição em tempo real de textos utilizando um dicionário fonético, este trabalho adotou uma proposta cognominada *brazilês* somada a utilização de sílabas neste processo de transcrição buscando as possíveis melhorias para área de automação, especialmente em sistemas voltados à acessibilidade, ou mesmo no auxílio à interatividade. Verificou-se o grau de melhora com a utilização da técnica apresentada especialmente na resposta do processo interativo, na diminuição do número de rotinas programáveis, na própria interpretação das sílabas utilizando o português falado no Brasil, somado ao grau de facilidade que possibilitará nos processos de acessibilidade. Assim sendo, este trabalho possibilita adequação da língua portuguesa para uso em sistemas computacionais, utilizando a linguagem natural e apresentando uma proposta de rotina simplificada para ser utilizada em softwares de reconhecimento de voz, melhorando as rotinas atuais que usam desde redes neurais a outros métodos que produzam a interação esperada. Para comprovar as vantagens desta técnica houve estudo aprofundado da proposta *brazilês* e definiu-se proposições buscando a idéia básica de simplificação, estudando as formas de reconhecimento automático de voz (RAV), desenvolvendo também, um programa que apresenta a formação de sílabas da língua portuguesa e análise da grafia de fonemas nas duas codificações da língua escrita, o português e o *brazilês*.

PALAVRAS CHAVE: Interatividade, interação, acessibilidade, *brazilês*, reconhecimento automático de voz.



ABSTRACT

In the quest to develop a technique which facilitates the process of automatic speech recognition for transcription real-time text using a phonetic dictionary, this work adopts a proposal nicknamed *brazilês* plus the use of syllables in the transcription process seeking possible improvements to the automation especially in systems focused on accessibility, or even in helping to interactivity. There was the degree of improvement with the use of our technique especially in the response of the interactive process, decreasing the number of programmable routines, in their own interpretation of syllables using the portuguese spoken in Brazil and the degree of ease in the processes that enable accessibility. Thus, this work allows adjustment of the portuguese language for use in computer systems, using natural language and presenting a proposal for a simplified routine for use in voice recognition software, improving the current routines that use neural networks from the other methods that produce the proposed interaction. To show the advantages of this technique was in-depth study of the proposed *brazilês* and set up the basic idea of seeking proposals for simplification, studying of automatic voice recognition (AVR), also developing a program that displays the formation of syllables of the portuguese language and analysis of the spelling of phonemes in the two encodings of the written language, portuguese and *brazilês*.

KEYWORDS: Interactivity, interaction, accessibility, *brazilês*, automatic voice recognition.

SUMÁRIO

1. INTRODUÇÃO.....	11
1.1. OBJETIVO.....	12
1.2. FINALIDADE.....	13
1.3. PROPOSTA.....	14
1.4. MOTIVAÇÃO.....	15
1.5. DISPOSIÇÃO DO TEXTO.....	17
2. REVISÃO DA LITERATURA.....	19
2.1. ACESSIBILIDADE.....	19
2.2. COMUNICAÇÃO HUMANA.....	20
2.3. TÉCNICA DE RECONHECIMENTO DE VOZ.....	25
2.4. PROCESSAMENTO DA FALA.....	29
3. DESCRIÇÃO DO BRAZILÊS.....	32
4. COMBINAÇÕES DE SÍLABAS.....	35
4.1. DEFINIÇÃO DO PROGRAMA.....	35
4.2. COMBINAÇÕES NÃO UTILIZADAS.....	42
4.3. COMBINAÇÕES NÃO UTILIZADAS PELO <i>BRAZILÊS</i>	42
4.4. DIFERENÇAS ENTRE AS CODIFICAÇÕES.....	43
5. GRAFIA DE FONEMAS.....	45
6. RECONHECIMENTO AUTOMÁTICO DE VOZ (RAV).....	47
6.1. FUNCIONAMENTO DO RAV NOS SISTEMAS ATUAIS.....	47
6.2. PROPOSTA DE FUNCIONAMENTO DO RAV UTILIZANDO O <i>BRAZILÊS</i>	48
6.3. COMPARATIVO ENTRE A MONTAGEM POR PALAVRA X MONTAGEM SILÁBICA.....	48
6.4. ANÁLISE DOS RESULTADOS.....	50
7. CONCLUSÃO.....	51
REFERÊNCIAS BIBLIOGRÁFICAS.....	53
ANEXO 1.....	55
ANEXO 2.....	82

LISTA DE FIGURAS

Figura 1-Aquisição do sinal de voz.....	30
Figura 2-Processo silábico para a técnica apresentada.....	31
Figura 3-Diagrama de formação de sílabas.....	36
Figura 4-Diagrama de formação de sílabas de 2 letras.....	37
Figura 5-Diagrama de formação de sílabas de 3 letras.....	38
Figura 6-Diagrama de formação de sílabas de 4 letras.....	41
Figura 7-Diagrama de formação de sílabas de 5 letras.....	41
Figura 8 - Diagrama de funcionamento do RAV nos sistemas em operação.....	47
Figura 9-Diagrama de funcionamento do RAV utilizando o português.....	49

LISTA DE TABELAS

tabela1-Comparativo entre a grafia fonética da língua portuguesa e do português.....	46
tabela 2 - Comparativo entre montagem por palavra e a montagem silábica (português).....	49

1. INTRODUÇÃO

Devido ao grande número de símbolos gráficos usados pela língua portuguesa para representar os mesmos sons, maior é o número de comparações e, conseqüentemente, maior é o tempo necessário para o reconhecimento destes símbolos o que pode até inviabilizar comunicação homem-máquina em tempo real.

A transcrição de textos utilizando palavras, se modifica de acordo com a inclusão ou exclusão de cada uma das palavras, pois o tempo necessário até acessar o dicionário de palavras pode trazer grande alteração na informação total do texto, fato que adotando a técnica silábica, a sílaba como elemento de transcrição, traz maior rapidez e segurança em todo processo.

Os Processadores Digitais de Sinais (DSP) utilizados para a identificação da fala, são dispositivos que realizam operações lógicas e aritméticas em tempo real, porém não são eficientes em aplicações com um número excessivo de comparações [Mandelman et al., 2010].

No Brasil a quase totalidade dos equipamentos utiliza outra língua, não o português, para serem controlados, especialmente aqueles idiomas mais difundidos mundialmente, entre poucas, a língua inglesa é a mais utilizada. Mesmo assim, devido ao grande número de comparações, em tempo real, as tarefas identificadas são restritas à um número limitado de possibilidades. Observa-se, ainda, o sotaque de Portugal, na maioria dos equipamentos que hoje no Brasil utilizam o idioma português.

Versolato (2011) apresenta algumas histórias verídicas em seu artigo onde discute o futuro distante, casos de paraplégicos que voltaram a mexer as pernas após terapias experimentais, que trazem esperança, mas estão longe de virar realidade. Inclusive observa que para uma nova técnica ser considerada boa, é necessário que ela seja testada em um número expressivo de pacientes. A reabilitação, muitas vezes confundida com fisioterapia, tem como meta melhorar apenas a qualidade de vida.

O grande número de brasileiros que por algum motivo não possuem movimentos, totais ou parciais, dos membros superiores, torna-os excluídos digitais, poderiam ser beneficiados por alguma forma de reconhecimento oral de voz, em tempo

real, possibilitando a edição de texto e comandos rápidos por meio de simples computadores pessoais.

A técnica apresentada neste trabalho também tem este objetivo, ou seja, melhorar a qualidade de vida daqueles que têm dificuldade ou impossibilidade de teclar, utilizar as mãos, ou qualquer outro meio para interagir com equipamentos eletrônicos, especialmente computadores, no entanto possuem a voz como único meio.

Com um programa possível de ser desenvolvido, utilizando a técnica apresentada neste trabalho, o paraplégico ou deficiente físico dos membros superiores pode operar máquinas e se comunicar com diferentes equipamentos, com segurança e eficiência, através da voz e do português falado no Brasil.

Este programa pode utilizar rotinas mais simples que as encontradas nos sistemas atuais, necessitando a participação do locutor apenas para definir parâmetros de separação de sílabas e de palavras, como um “*gap*” ou espaço sonoro. Assim, o sistema proposto interpretará perfeitamente o comando sonoro e executará a tarefa esperada pelo usuário do processo, ou no mínimo proporcionará a transcrição de suas idéias, expressas pela voz, para um computador, podendo ser apresentado diretamente para o monitor de vídeo, com resposta temporal equivalente à resposta do cérebro humano.

A solução plena para o problema de falta de acessibilidade esta distante, mas esta técnica minimiza tal problema e soluciona uma gama enorme de necessidades, sendo que a principal é prover a comunicação homem/máquina com rapidez, eficiência e a um custo provavelmente ínfimo ao comparar com os processos atuais.

1.1. OBJETIVO

Analisar uma técnica facilitadora do processo de reconhecimento automático de voz para transcrição em tempo real de textos utilizando um dicionário fonético adotando-se como base de estudos uma proposta cognominada *brazilês* somada a utilização de sílabas neste processo de transcrição de textos falados utilizando português do Brasil, possibilitando aplicar em comando de máquinas na área industrial, auxiliar a interatividade possibilitando melhora da qualidade de vida dos cidadãos, especialmente em processos de acessibilidade.

1.2. FINALIDADE

Na área de automação, o comando de máquinas através de voz não é para o futuro, visto que hoje, grande parte destes equipamentos utiliza este método de comunicação.

De acordo com a ABNT NBR 9050, válida a partir de 30/06/2004, acessibilidade é a possibilidade e condição de alcance, percepção e entendimento para a utilização com segurança e autonomia de edificações, espaço, mobiliário, equipamento urbano e qualquer dispositivo de comando, acionamento, comutação ou comunicação. Exemplos: telefones, intercomunicadores, interruptores, torneiras, registros, válvulas, botoeiras, painéis de comando, considerando também como tecnologia assistida a um conjunto de técnicas, aparelhos, instrumentos, produtos e procedimentos que visam auxiliar a mobilidade, percepção e utilização do meio ambiente por pessoas, especialmente com deficiência.

Países mais desenvolvidos que o Brasil trabalham com processos para utilização de voz, e já possuem muitos trabalhos implementados neste sentido.

Na comunicação homem-máquina por meio de linguagem natural, o português falado no Brasil, devido ao grande número de símbolos usados pela língua portuguesa para representar os mesmos sons, a existência de duplo sentido das palavras gera indefinição, maior número de comparações e, conseqüentemente, maior tempo necessário para o reconhecimento dos mesmos, podendo eventualmente, até inviabilizar a comunicação homem-máquina ou produzir alto custo do produto final que utilize tal tipo de comunicação, necessitando inclusive sistemas de inteligência artificial para atender a compreensão de pequenas frases, pois podem produzir diferentes tipos de interpretação conforme o contexto.

Assim sendo, para atender o objetivo deste trabalho simplificando a comunicação homem-máquina adota-se o *brazilês*, apresentado por Marcelino (2008), no qual, cada som do português falado no Brasil será representado por um único símbolo gráfico, estudando o processo de tradução dos sons (fonemas) da língua, para símbolos gráficos únicos (padronização da representação gráfica do fonema), assim possibilita o

desenvolvimento de um dicionário fonema-símbolo gráfico, trazendo facilidades na interpretação pelo computador (dispositivo de reconhecimento de voz, ou interpretador audível), e apresenta a propositura de uma rotina simplificada para ser utilizada em softwares de reconhecimento de voz, apenas alterando a base de dados específica para estes fins.

A língua portuguesa possui uma quantidade muito maior de símbolos gráficos para representação de determinados fonemas que os encontrados no *brazilês*, por este motivo, comparando-se as duas codificações, verifica-se que para o primeiro existe a necessidade de fazer um número muito maior de comparações para se obter a tradução de textos escritos ou falados. Em aplicações em que tais comparações devam ser realizadas quase que instantaneamente ao cérebro humano, pode ser impossível com as tecnologias atuais.

Observa-se ainda que outros idiomas, como o inglês, possuem ortografia muito mais dissociada da fonética que o próprio português, mas por nacionalidade e facilidade para realização do presente trabalho, escolheu-se a língua portuguesa, mas esta idéia pode ser aproveitada por outros idiomas, sem, em nenhum momento, propor quaisquer mudanças dos mesmos nem a criação de uma nova língua, como já fora mencionado.

A história da língua, as dificuldades geradas por ela a seu reconhecimento eletrônico, as dificuldades políticas e sociais para criação e implementação de uma nova língua, problemas de duplo sentido de palavras e ou frases, controle de pensamento e diferenças de pronuncia, todas foram discutidas neste trabalho para melhor embasar as proposituras.

Não se pode deixar de lembrar que as linguagens naturais cada vez mais tornam necessárias para possibilitar a comunicação homem-máquina seja por acessibilidade ou mesmo por comodidade.

1.3. PROPOSTA

Este trabalho não tem a intenção de propor alterações na língua escrita no Brasil, muito menos a criação de uma nova língua, mas sim, propõe criar uma nova base de dados e conseqüentemente otimizar as rotinas utilizadas por computadores responsáveis pela

comunicação homem-máquina em tempo real, podendo atingir os mesmos resultados encontrados atualmente, porém com maior velocidade de processamento e menor possibilidade de erros.

Linguagens naturais cada vez mais se tornam necessárias para possibilitar a comunicação homem-máquina, seja por acessibilidade, por comodidade, ou para melhorar a interatividade diminuindo ao máximo a produção de erros produzidos pela dificuldade de interpretação de palavras, por dicção ou mesmo desconhecimento, ou seja, falta de determinadas palavras no dicionário interpretativo na relação de palavras que produzirão ações pré definidas.

Considerando reconhecimento de voz como sendo:

- Reconhecimento de palavras caracteriza-se por processar apenas um pequeno trecho de fala, de modo a identificar a ação que o sistema deva tomar. Exemplo: centrais telefônicas.

- Reconhecimento da linguagem ou fala natural, várias palavras que tenham sentido semântico. A fala é convertida em texto ou ação. Exemplo: processadores de texto, ditado de documentos, interpretação que gera ação, auxiliando a acessibilidade principalmente de deficientes físicos.

- Síntese de voz é o processo contrário ao do reconhecimento da fala. O sintetizador recebe um texto na forma digital e transforma-o em ondas sonoras, ou em outras palavras, fazendo uma leitura em voz alta. Exemplo: acessibilidade para deficientes visuais.

- Sistemas de autenticação com objetivo de identificar quem está falando. Exemplo: sistemas de segurança.

A técnica que será apresentada tem como objetivo trabalhar com a linguagem natural, ou seja, possibilitar que o computador reconheça a fala natural e escreva a palavra ou frase na tela do vídeo ou desenvolva uma ação, atendendo ao solicitado.

1.4. MOTIVAÇÃO

Uma das questões atuais de acessibilidade é a criação de sistemas capazes de reconhecer e eventualmente executar alguma ação a partir da voz humana. Entre suas

inúmeras aplicações, pode-se citar a interação de deficientes visuais com o computador e a operação de equipamentos quando um trabalhador não se encontra com as mãos disponíveis.

Embora já existam inúmeras pesquisas e produtos nessa área, poucos deles são compatíveis com o idioma português do Brasil.

No Brasil, o conceito de acessibilidade é definido como condição para utilização, com segurança e autonomia, total ou assistida, dos espaços, mobiliários e equipamentos urbanos, das edificações, dos serviços de transporte e dos dispositivos, sistemas e meios de comunicação e informação, por pessoa portadora de deficiência ou com mobilidade reduzida. (Decreto federal 5296 de 02-12-2004 que regulamenta as Leis nºs 10.048, de 8 de novembro de 2000, que dá prioridade de atendimento às pessoas definidas neste Decreto, e 10.098, de 19 de dezembro de 2000, que estabelece normas gerais e critérios básicos para a promoção da acessibilidade das pessoas portadoras de deficiência ou com mobilidade reduzida, e dá outras providências.).

Estima-se que no país 14% da população tenha alguma deficiência, isso significa mais de 20.000.000 (vinte milhões) de cidadãos brasileiros.

O Jornal Folha de São Paulo informou, em 10 de outubro de 2010 através de seu site, que o Google tem investido pesado em comandos de voz no seu sistema operacional móvel, o *Android* e que em agosto, a empresa introduziu as *Voice Actions* (ações de voz), que permitem, por exemplo, enviar mensagens de texto, iniciar a reprodução de uma música, ligar para um contato, enviar um e-mail e obter orientações de direção por meio de comandos de voz.

O site observou, ainda, em relação ao português/brasileiro só funcionar na versão mais recente do *Android -a 2.2*, que não está disponível em nenhum celular no Brasil, e que as *Voice Actions* só entendem inglês, por enquanto.

Até hoje, aliás, a oferta de sistemas de reconhecimento de fala em português para celulares e computadores é quase nula.

Inclusive o próprio sistema operacional *Windows* da *Microsoft* apresenta uma observação indicando que “o reconhecimento de fala está disponível somente em inglês, francês, espanhol, alemão, japonês, chinês simplificado e chinês tradicional”.

Felix Ximenes, diretor de comunicação do Google Brasil, afirmou à Folha (Jornal Folha de São Paulo, 2010) que não há previsão de transpor a tecnologia de reconhecimento de fala da empresa para a língua portuguesa, mas que essa adaptação está no horizonte do Google.

O principal desafio, segundo Ximenes, é tornar o sistema capaz de compreender com igual precisão os diversos sotaques brasileiros, o português de Portugal e o de outros países lusófonos, além da fala de pessoas que não tenham o português como língua primária.

O funcionamento de reconhecimento automático de voz, que em sua maioria são baseados em reconhecimento estatístico de padrões, onde após o treinamento dos modelos, o sistema realiza a busca pela sequência de palavras mais provável, ou seja, que melhor representa o sinal de entrada, faz-se uso do Teorema de Bayes, onde se busca maximizar uma probabilidade condicional e é composto por blocos, entre outros, o *front end*, o modelo acústico, neste inclui as cadeias ocultas de Markov, a modelagem da língua e o decodificador [TEVAH, 2006].

A técnica que neste trabalho aproveita o *brazilês*, pode inclusive ser utilizada para outras línguas, com objetivo de proporcionar a comunicação homem máquina, possibilitando ganhos de velocidade e facilidade na implementação, pois os processos aqui apresentados podem ser aproveitados para outros idiomas que possuam as características parecidas.

1.5. DISPOSIÇÃO DO TEXTO

No capítulo 2, deste trabalho, apresenta-se a revisão da literatura especificando a acessibilidade, a comunicação humana e a técnica de reconhecimento de voz.

No capítulo 3, apresenta a proposta do *brazilês*.

O capítulo 4 apresenta as combinações de palavras e sílabas obtidas através de programa de computador, separando aquelas sílabas que não existem no *brazilês* com o objetivo de quantificar e comprovar a vantagem para utilização da técnica apresentada.

O capítulo 5 trata do comparativo da representações gráficas de fonemas da língua portuguesa e do *brazilês*.

O atual funcionamento do reconhecimento automático de voz é comparado ao funcionamento do reconhecimento de voz utilizando a técnica apresentada neste trabalho no capítulo 6.

A conclusão, no capítulo 7, trata dos benefícios que a técnica traz e os processos para sua implementação.

2. REVISÃO DA LITERATURA

2.1. ACESSIBILIDADE

De acordo com a Norma Brasileira 9050 [ABNT, 2004], tem-se:

acessibilidade: Possibilidade e condição de alcance, percepção e entendimento para a utilização com segurança e autonomia de edificações, espaço, mobiliário, equipamento urbano e elementos.

acessível: Espaço, edificação, mobiliário, equipamento urbano ou elemento que possa ser alcançado, acionado, utilizado e vivenciado por qualquer pessoa, inclusive aquelas com mobilidade reduzida. O termo acessível implica tanto acessibilidade física como de comunicação.

deficiência: Redução, limitação ou inexistência das condições de percepção das características do ambiente ou de mobilidade e de utilização de edificações, espaço, mobiliário, equipamento urbano e elementos, em caráter temporário ou permanente.

elemento: Qualquer dispositivo de comando, acionamento, comutação ou comunicação. São exemplos de elementos: telefones, intercomunicadores, interruptores, torneiras, registros, válvulas, botoeiras, painéis de comando, entre outros.

tecnologia assistida: Conjunto de técnicas, aparelhos, instrumentos, produtos e procedimentos que visam auxiliar a mobilidade, percepção e utilização do meio ambiente e dos elementos por pessoas com deficiência.

A Anatel (Agência Nacional de Telecomunicações) adaptou seu portal conforme as recomendações de acessibilidade do governo brasileiro, onde programas avaliadores de acessibilidade detectam o código HTML de uma página *web* e fazem uma análise do seu conteúdo; a acessibilidade das páginas é avaliada segundo três níveis de prioridade:

- Nível de acessibilidade de prioridade 1 - são as exigências básicas de acessibilidade, condições que, se não forem cumpridas, poderão fazer com que grupos de usuários sejam impedidos de acessar as informações do documento;
- Nível de acessibilidade de prioridade 2 - equivale às normas e às recomendações de acessibilidade que garantem o acesso às informações do documento. Se não forem cumpridas, grupos de usuários terão dificuldades para navegar e acessar os conteúdos;

- Nível de acessibilidade de prioridade 3 - normas e recomendações de acessibilidade que, sendo implementadas, facilitarão o acesso aos documentos publicados na *internet*. Se essas regras não forem cumpridas, grupos de usuários poderão enfrentar dificuldades para acessar as informações dos documentos publicados.

Quanto acessibilidade pode-se dizer também que deficientes físicos superam várias dificuldades diariamente, por necessidades especiais, portanto facilmente podem se habituar a fornecer sons com as devidas separações silábicas.

2.2. COMUNICAÇÃO HUMANA

Por Monteiro (2000) o processo de comunicação humana iniciou a partir de uma mudança radical no tipo de mensagem a ser transmitida. A língua falada foi substituída por outro tipo de linguagem, para que ficasse registrada à disposição daqueles que necessitassem ler posteriormente o que havia sido dito. Esta nova técnica deveria ter, pelo menos, três características: durabilidade, profundidade e clareza. A escrita fonética foi se desenvolvendo muito rapidamente, mais precisamente na Grécia.

No século XV os indivíduos já se preocupavam em preparar e reproduzir os livros através da técnica de copiar, a mão, os livros já existentes. Poucos tinham acesso a estas reproduções.

Com o advento da impressão milhares de livros poderiam ser reproduzidos. A descoberta da impressão conforme se conhece hoje se deu graças à invenção de Gutemberg, que conseguiu, depois de muitas experiências, uma impressão nítida e perfeita.

A escrita foi, sem dúvida, uma das tecnologias de comunicação mais importantes para o progresso da humanidade. Sem ela, provavelmente, a história das grandes civilizações do mundo estaria perdida.

Há aproximadamente 3.000 anos, os escribas eram considerados uma classe das mais poderosas, unicamente porque dominavam a técnica da escrita. Na Grécia Antiga, só eram considerados cidadãos e aptos a votar, aqueles que soubessem ler e escrever. Esses dois exemplos apontam indícios da estreita relação entre o uso das novas tecnologias, na educação e a participação dos cidadãos.

Em 1870, Thomas Edison consegue gravar e conservar a voz humana em um fonógrafo. Esse instrumento conseguia levar às casas um novo som, capaz, de se tornar uma alternativa de comunicação. Em 1906, Reginald A. Fessenden constrói uma aparelhagem capaz de irradiar sinais. Pessoas falavam através de um transmissor e as vozes eram recebidas em um aparelho receptor, dando origem à radiotelefonía. No mesmo ano, foram descobertas substâncias capazes de melhorar a transmissão e com um preço bastante acessível. Entre 1923 e 1926, muitos países dão início às transmissões no rádio.

Hugo (1995) descreve a interatividade como um conceito voltado para a comunicação, o qual esteve sempre presente na vida do homem a partir dos relacionamentos estabelecidos com os diversos ambientes em que ele está inserido. A interatividade se diversifica em níveis e patamares diferentes de acordo com os múltiplos ambientes e as formas de se relacionar com eles. Assim, pode se identificar diversas formas de interação como homem-máquina, homem-homem, homem-técnica e outras. A interação homem-técnica é uma atividade que aconteceu em todas as etapas da civilização. No contexto atual, esta interação não acontece apenas direcionada a objetos (homem-computador), mas ocorre principalmente orientada para a criação de processos baseados na gestão das informações. Isto se deve à evolução da tecnologia que aliada aos modernos processos de comunicação possibilitou a disseminação de um manancial de informações capaz de proporcionar facilidades na obtenção de conhecimentos.

A facilidade de acesso a este acervo recebeu um significativo impulso procedente do avanço de recursos como a multimídia e a *web*.

Apresentando a diferença entre interação, interatividade e interface, Hugo (1995) define: interação como a ação entre entes (inter - ação = ação entre) o que define uma relação entre dois agentes. Assim, interagir é agir mutuamente. Porém, muito do que se tem classificado como interativo é na verdade apenas reativo. A interatividade pode ser definida como uma atividade mútua e simultânea da parte de dois agentes, normalmente trabalhando em direção a um mesmo objetivo podendo provocar mudanças comportamentais entre eles. Com relação a este conceito, pode-se acrescentar a característica da bidirecionalidade do processo, onde o fluxo se dá nos dois sentidos e os agentes (emissor e receptor) dialogam entre si durante a construção da mensagem.

Um sistema pode ser considerado interativo quando apresenta determinadas características como permitir que os participantes atuem quando bem entenderem, que eles possam obter respostas que não estejam disponíveis em um determinado momento, e que não sejam direcionados, pois a inexistência de um padrão pré-determinado possibilita maior liberdade.

A partir daí, Dias (2003) lembra que uma interface homem-máquina, designa o conjunto de aparelhos materiais que permite a comunicação entre um sistema de informações e seu usuário humano.

Com o desenvolvimento das interfaces, a relação homem-máquina vem se transformando e agregando cada vez mais elementos gráficos e sonoros.

Para Lévy, appud Dias (2003), "a interface contribui para definir o modo de captura da informação oferecido aos autores da comunicação. Ela abre, fecha e orienta os domínios de significação, de utilizações possíveis de uma mídia".

Assim sendo, Dias conclui que o computador, agente fundamental nas relações sociais contemporâneas têm permitido o estabelecimento de interações múltiplas identificadas pelos relacionamentos homem-máquina e máquina-máquina. Mas, o processo de interação usuário-computador é construído principalmente pela utilização das interfaces apresentadas nos softwares. E para que esta interação aconteça da melhor maneira possível é necessário que a construção destas interfaces sejam aprimoradas, trabalhadas de forma a proporcionar maior poder de diálogo, de respostas e alterações nos comportamentos dos agentes envolvidos.

No que tange o diálogo homem-máquina, Rover (2001) apresenta que a língua, seja escrita ou falada, é o meio de comunicação. Até os anos 70, tinha-se um "mosteiro" de especialistas em informática para manipular o computador; homens que dominavam a utilização de linguagens especializadas/específicas no uso do computador. Na década de 80, com o advento e expansão de microcomputadores e suas redes, o usuário final foi colocado frente-a-frente com a máquina e seus dados. Desta forma, linguagens mais naturais, bem como meios de acesso mais naturais eram prementes.

Em adição ao exposto, lembra-se que a informação e o conhecimento se apresentam como grandes riquezas atualmente. A informação e o conhecimento têm como suporte (veículo) básico nada mais que a linguagem natural, visto que grande parte do

conhecimento da humanidade encontra-se armazenado (geralmente escrito) nesta forma. Daí advém à necessidade de compreender a linguagem, seja de forma escrita ou oral.

Rover (2001) ainda lembra que Pierrel e Anick entendem que com o crescimento de tamanho e importância das bases de dados, os computadores necessitam ter meios de interpretar a linguagem natural. Desta forma um usuário pode mais facilmente encontrar algum argumento de pesquisa na base de dados, sem se preocupar com sua especificação exata - seja em termos de comandos de busca, seja em palavras a pesquisar - permitindo-lhe realizar suas pesquisas através do vocabulário que lhe é conhecido, com o computador responsável por oferecer-lhe sinônimos e ajuda direcionada ao argumento.

Fiorin (2009), define linguagem como a capacidade de os seres humanos comunicarem-se por meio de um sistema de signos. Essa faculdade corporifica-se em línguas, sistemas de signos utilizados por diferentes comunidades linguísticas.

De acordo com Saussure, appud Fiorin (2009), “a língua não se confunde com a linguagem; é somente uma parte determinada, essencial dela, indubitavelmente. A língua é, ao mesmo tempo, um produto social da faculdade da linguagem e um conjunto de convenções necessárias, adotadas pelo corpo social para permitir o exercício dessa faculdade nos indivíduos”.

Ainda para Saussure "a língua é um princípio de classificação". O discurso é a atividade verbal social. A chamada hipótese Sapir-Whorf mostra que a língua modela a representação do mundo de cada falante.

A língua não é uma nomenclatura, que se apõe a uma realidade pré-categorizada, ela é que classifica a realidade.

Quanto ao duplo sentido das palavras e ou frases Fiorin (2009) lembra que deslocam-se os sentidos das palavras, jogando-se com eles, inclusive, reproduz um poema de Bráulio Tavares:

"Eu quero é orgia!

A safadeza!

A indecência!

Deixo pros padres

E pros militares

A continência. "

O poema joga com os dois sentidos da palavra *continência*: "castidade" e "saudação militar", para se posicionar contra os comportamentos contidos, as hierarquias, as normas de decência, preconizando um mundo de liberdade. Esses deslocamentos operam em todos os níveis e dimensões da língua.

Para confirmar o poder da língua, ou mesmo, a busca pelo controle do pensamento através da língua, especialmente no que se refere na dimensão política, na obra de George Orwell, intitulada "1984", onde os detentores do poder criaram a Novilíngua com o objetivo maior de transformar a literatura anterior, escrita em Anticlíngua (língua utilizada até então), em obras contraditórias do que eram. Propõe que "todo o mecanismo do pensamento será diferente. Com efeito, não haverá pensamento, como hoje entendemos."

A forma proposta pela Novilíngua é reduzir a língua à expressão mais simples possível, destruindo palavras. "Não apenas os sinônimos; os antônimos também. Afinal de contas, que justificativa existe para a existência de uma palavra que é apenas o contrário da outra? ... No fim, todo o conceito de bondade e maldade será descrito por seis palavras – ou melhor, uma única." A busca é controlar totalmente os indivíduos pois nem todas as realidades poderão ser pensadas, pela redução do número de palavras e a impossibilidade de aumentar este número.

Haveria muitos crimes e erros que estariam além da capacidade do homem de cometê-los, simplesmente pelo fato de que eles não tinham nomes e, portanto, eram inimagináveis. E se esperava que, com o passar do tempo, as características que distinguiam a Novilíngua se tornassem cada vez mais pronunciadas; o número de palavras diminuiria, seus significados se tornariam cada vez mais restritos e a possibilidade de utilizar palavras de maneira imprópria se tornaria cada vez menor.

Assim pode-se concluir que a língua constrói o real e torna-o dizível, portanto possui dimensão política.

No que tange a diferença de pronúncia, Fiorin (2009) encontra na Bíblia Sagrada (Juizes 12:6) um episódio. A palavra hebraica *shibolet*, "espiga", tinha uma variação dialetal *sibolet*. A tribo de Galaad estava em guerra com a de Efraim. Para identificar os efraimitas, pedia-se que a pessoa pronunciasse essa palavra. Quem dissesse

sibolet era morto. Foram eliminados 42.000 efraimitas. Até hoje o termo *shibolet* nomeia uma maneira de pronunciar uma palavra, que identifica a origem de quem a diz. A fonologia torna-se letal. A diferença linguística é o lugar onde reside o ódio ao outro, é o lugar da discriminação, do preconceito. Certas pronúncias são estigmatizadas, determinadas variedades são consideradas inaceitáveis. Tudo isso serve para classificar, para selecionar, para excluir, para condenar.

As regras do "bom" uso da língua são relações de poder. Elas obrigam a recalcar, a renegar uma língua primeira (por exemplo, os descendentes de alemães ou italianos no Brasil deviam eliminar seu sotaque ou certos decalques de sua língua primeira) ou uma variedade primeira da língua (as variedades populares ou regionais do português), que são objeto de gozações, reprimendas ou punições. Essa sanção a línguas ou variedades pode produzir uma resignação, ou seja, a aceitação de uma "inferioridade", ou uma revolta ativa, isto é, a reafirmação com orgulho de uma determinada origem ou de um dado falar.

Lembra Barthes, apud Fiorin (2009), que o "objeto em que se inscreve o poder é a linguagem ou, para ser mais preciso, sua expressão obrigatória: a língua". Explica ainda, com base nas idéias de Foucault, que o poder é múltiplo, é onipresente, atravessa toda a História.

Como pode se ver, são múltiplas as maneiras pela qual o poder se inscreve na linguagem. Sua natureza é intrinsecamente política, porque ela sujeita os que a falam a sua ordem. Os silenciamentos operados pelo discurso manifestam uma relação de poder.

Os discursos que circulam no espaço social são submetidos à ordem do poder, não são todos equivalentes. A língua não é um instrumento neutro de comunicação, mas é atravessada pela política, pelo poder, pelos poderes.

A literatura, pelos deslocamentos que produz, é uma forma de trapacear a língua, desvelando os poderes nela inscritos. Com o desenvolvimento das interfaces, a relação homem-máquina vem se transformando e agregando cada vez mais elementos gráficos e sonoros, minimizando as trapaças da língua e democratizando os poderes.

2.3. TÉCNICA DE RECONHECIMENTO DE VOZ

Muller (1996) lembra que um dos primeiros sistemas desenvolvidos dentro dessa categoria foi o concebido por McClelland e Kawamoto. Este sistema consiste numa

rede *perceptron* (rede de neurônios dispostos em camadas) de duas camadas, para a qual é ensinado um padrão de entrada que contém a montagem sintática das palavras, as quais têm em si características semânticas codificadas à mão. Um determinado conjunto dessas características (sentença) é associado a um padrão de saída, o qual é um código representando o contexto semântico.

O processamento de sentenças: reconhecimento e classificação, foi concebido para demonstrar a possibilidade de se produzir regras semânticas automaticamente através das RNAs (redes neurais artificiais), baseando-se apenas na sintaxe e no contexto semântico de sentenças. No ano de 1990 houveram diversas publicações sobre o reconhecimento de sentenças.

Outro algoritmo de aprendizado, utilizado nos sistemas de aprendizagem para formação das frases são as redes *backpropagation*, algoritmo baseado na retropropagação de erros para realizar os ajustes dos pesos das camadas intermediárias, sendo que a saída da primeira é a entrada da segunda rede. A função da primeira rede é aprender a formação das frases, para que na sua saída resulte um tipo padrão de sentença. Desta forma tem-se um sistema que permite a associação das questões aos contextos semânticos das sentenças.

O resultado deste processamento é um sistema ágil de consulta a determinado banco de conhecimento. Muller (1996) ainda lembra que outro sistema de Jain e Waibel (1990), que possui um encadeamento de três redes em *backpropagation* que permitem a distinção de três níveis para o reconhecimento de uma sentença: de frase, de oração (sentença de mais de uma frase) e de estrutura de oração e relacionamentos entre as frases da oração.

O objetivo deste sistema é permitir a identificação das relações semânticas dos componentes de uma sentença.

Não esquecendo que Kohonen em 1990 apresentou um sistema simples para reconhecimento das características semânticas de palavras de frases compostas apenas por três elementos. Utilizando um mapa de características auto-organizáveis, e um sistema de pré-processamento de frases de três elementos, foi possível a distinção entre as características semânticas, sem a necessidade da realização de uma atribuição manual de códigos, permitindo um treinamento automático na relação frase-reconhecimento.

Silva (2010) apresenta em seu trabalho que o reconhecimento automático de voz (RAV) consiste no processo onde o sinal de voz (analógico) é convertido em sua representação textual e tem sido estudado desde os anos 50, sendo que atualmente baseia-se nos modelos ocultos de Markov e uma das principais limitações para o português brasileiro é a disponibilidade de corpora de grande porte. (o termo corpus – plural corpora - significa grande conjunto de dados, sendo corpus de texto o conjunto estruturado de sentenças e corpus de voz o conjunto de arquivos de áudio com suas respectivas transcrições).

O modelo oculto de Markov é um modelo estatístico usado em aplicações de reconhecimento de padrões, o modelo encontra qual a palavra tem maior probabilidade de ocorrer depois da identificação da ocorrência de uma determinada palavra.

Martins (1997) destacou que a teoria de Modelos Ocultos de Markov (*Hidden Markov Models* também conhecido como HMM) foi introduzida por Baum no final da década de 60 e foi usada pela primeira vez para reconhecimento de fala por Bakes e Jelinek no início dos anos 70.

O RAV tem os seguintes fatores que influenciam o desempenho dos reconhecedores de voz:

- Dependência de Locutor

Sistema Dependente – treinado por um locutor específico e será reconhecido apenas por ele, tem boa taxa de acerto.

Sistema Independente – é capaz de reconhecer a fala de qualquer locutor, até mesmo os que não participaram do treino. Porém, quando de seu treino deve ter grande variedade de locutores no corpus do treino.

- Tipo de Fala

Palavras isoladas – a taxa de reconhecimento, acertos, possui tendência a apresentar bom resultado;

Palavras contínuas – por ocorrer poucas pausas apresenta difícil implementação, não tem controle de onde começa e termina a palavra.

- Tamanho do Vocabulário

Quanto maior o número de palavras que o RAV é apto a reconhecer, maiores são as chances de equívocos por parte do decodificador que fará o reconhecimento, pois

existe maior quantidade de palavras ambíguas e homófonas (que possuem a mesma pronúncia).

- Tamanho do Corpus para Treino

O sistema RAV é baseado em modelos estatísticos com etapas de treinos e testes, sendo assim, o tamanho do corpus utilizado para estes treinos e testes é de extrema importância para se obter dados confiáveis, portanto quanto mais dados houver para treino dos modelos acústicos e de linguagem melhor adaptados os mesmos estarão.

Os corpora existentes no português brasileiro são considerados de pequeno porte (poucas horas) pois sistemas de grande porte necessitam de centenas de horas para treinar seus modelos acústicos e textos de milhões de linhas para os modelos de linguagem.

Observa-se ainda que muitos problemas afetam a precisão do reconhecimento, dentre eles destacam-se:

- A mesma palavra pronunciada várias vezes, pode apresentar diferentes formas de onda;

- Uma mesma palavra pode ser pronunciada de várias formas (sotaque);

- Rapidez da fala de alguns locutores;

- Vocabulário extenso com palavras homófonas (caso do português brasileiro), e neste item o *brazilês* será de grande valia;

- Dificuldade de segmentação da fala (reconhecer onde termina uma palavra ou sílaba e começa outra);

- Variações como timbre, ritmo e intensidade da fala;

- Dificuldade de conseguir arquivos de boa qualidade (com pouco ruído) para treino dos modelos acústicos

Quanto ao dicionário fonético a conversão de uma seqüência de caracteres em seqüência de fonemas é importante pré-requisito para o reconhecimento e/ou síntese de voz e basicamente o dicionário consiste de um conjunto de palavras com suas respectivas transcrições fonéticas.

O *brazilês* pode ser considerado o próprio dicionário fonético com seus respectivos significados, com possibilidade de maior rapidez através da transcrição fonética por sílabas. (detalhes apresentados no próximo capítulo com a explicação da rotina

possível através da técnica de reconhecimento automático de voz português brasileiro utilizando o *brazilês*).

O funcionamento do reconhecimento automático de voz (RAV), em sua maioria, é baseado em reconhecimento estatístico de padrões, onde após o treinamento dos modelos, o sistema realiza a busca pela sequência de palavras mais provável, ou seja, que melhor representa o sinal de entrada. Basicamente, faz-se uso da regra de Bayes, onde se busca uma sequência de palavras que maximiza uma probabilidade condicional e é composto por blocos.

Os blocos mais relevantes são:

- Front-End, responsável pela extração de parâmetros do sinal de voz.
- Modelo Acústico, busca modelar, a partir dos parâmetros o sinal acústico, neste inclui as cadeias ocultas de Markov.
- Modelo de Linguagem, tenta, a partir de textos da língua, obter as possíveis seqüências de palavras a serem reconhecidas.
- Decodificador, realiza junto com os blocos anteriores a transcrição do sinal de voz.

2.4. PROCESSAMENTO DA FALA

Por PETRY (2002), verifica-se que a aquisição do sinal de voz consiste na conversão das ondas sonoras em sinais elétricos através de um transdutor (microfone), filtragem desse sinal elétrico e conversão analógico-digital (A/D) desse sinal. Esse processo é ilustrado na figura 1.

O sinal de voz geralmente possui componentes de frequência superiores a 5 KHz com amplitudes baixas. Assim, frequência de amostragem superiores a 11 KHz preservam grande parte da informação contida no sinal, mas em geral um espectro de até 8 KHz é suficiente para a maioria das aplicações com voz, o que justifica a frequência de amostragem em torno de 8 KHz.

O desempenho de todo sistema de processamento de fala está diretamente relacionado à qualidade da informação que é fornecida, a construção de métodos para obtenção de parâmetros representativos normalmente utiliza uma modelagem matemática para a produção do sinal de voz.

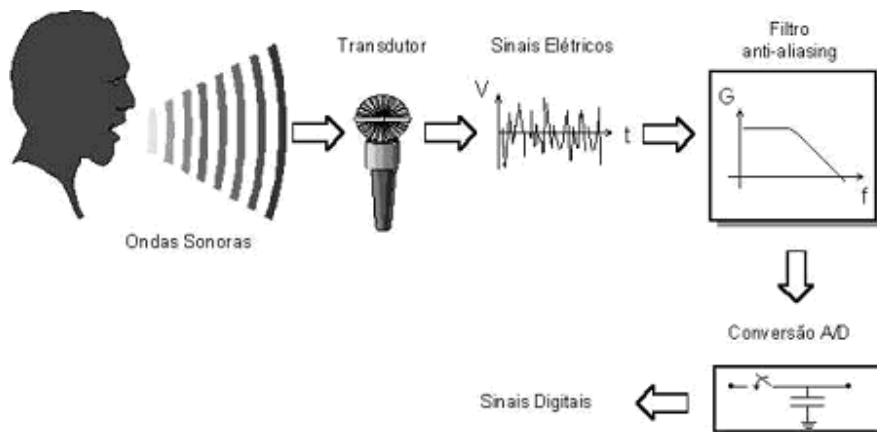


Figura 1-Aquisição do sinal de voz

Martins (1997), comparando os desempenhos de reconhecedores de palavras isoladas, usando-se Modelos Ocultos de Markov, redes *Multilayer Perceptron* e sistemas híbridos combinando essas duas técnicas, a literatura é unânime em afirmar que nenhum é perfeito .

Comparando os RAV atualmente em uso, com a proposta deste trabalho utilizando o processamento da fala tem-se:

- RAV atualmente em uso - espera montagem da possível palavra - exposição da palavra no monitor. Por exemplo na montagem da palavra “atenção” existe a busca no banco de dados as seguintes palavras: atensão, atenssão, atenção, assim como, para a palavra “professores” existe a busca das palavras: profesoeres, professores, profeçores.

- RAV utilizando a proposta deste trabalho - sílaba pronunciada - exposição instantânea no monitor (locutor falar silabicamente). Pode ser verificado na figura 2 o

processo de falar silabicamente, com tempos específicos entre sílabas e tempo diferenciado entre palavras, e em tempo real estas sílabas são reconhecidas e apresentadas no monitor, sem quaisquer alterações posteriores, nem a necessidade de promover varias comparações .



Figura-2 Processo silábico para técnica apresentada

3. DESCRIÇÃO DO BRAZILÊS

Marcelino (2008) propôs o *brazilês*, uma língua escrita que:

- tem ligação direta com o som falado, dispensando o conhecimento prévio das palavras;

- valoriza o conhecimento útil evitando discriminações devido à grafia equivocada;

- reduz o número de horas-aula para o aprendizado, restando mais tempo para as idéias e os pensamentos do que é ou está escrito.

O *brazilês* procura representar os sons da língua portuguesa falada no Brasil de forma coerente e lógica, já que um mesmo fonema é representado por um único símbolo gráfico, o que leva a uma simplificação da codificação da língua, facilitando assim, o reconhecimento eletrônico de sílabas.

A proposta inicial do *brazilês* foi a simplificação do português para o reconhecimento de voz em tempo real, mas o autor preocupado em justificar uma possível proposta de codificação para a língua escrita, procurou polemizar a idéia, desviando o foco original e deixando algumas sugestões não implementadas. Em função disto, este trabalho com o único objetivo de RAV em tempo real, manteve os princípios que originaram o *brazilês*, tomou como definições as principais sugestões deixadas em aberto na proposta e outras que vieram simplificar ainda mais a codificação. Como por exemplo a substituição de AM, EM, IM, OM, UM respectivamente por AN, EN, IN, ON, UM, o uso do N no final de palavras (SON e não mais SOM) e a utilização de acentos somente para a diferenciação dos fonemas, independentemente das regras de acentuação do português.

Princípios originais do *brazilês*:

- No *brazilês* cada fonema tem uma única grafia;

- O alfabeto possui 22 letras (A,B,C,D,E,F,G,H,I,J,L,M,N,O,P, R,S,T,U,V,X,Z) as letras Ç, O, K,W,Y foram excluídas;

- Os dígrafos CH, SS, (SC, SÇ, XC, com som de S), são substituídos respectivamente por X, S, S;

- O GUE, GUI, QUA, QUE, QUI, QUO, são substituídos respectivamente por GE, GI, CUA, CE, CI, CUO;

- O fonema /s/ é representado somente pela letra S. Exemplos: TESTO (texto), ESCLAMASÃO (exclamação), POSÍVEL (possível), FLORESER (florescer), DESA (desça), COBISA (cobiça), etc. A letra S também continua representando o plural de palavras;

- O fonema /c/ é representado exclusivamente pela letra C, levando a eliminação da letra Q. Exemplos: CAPELA, CE (que), ACILO (aquilo), CONBUSTÍVEL (combustível), COSIENTE (quociente), CUBA, etc;

- O fonema /g/ é representado unicamente pela letra G. Exemplos: GATO, FOGETE (fogete), GERRA (guerra), GITARRA (guitarra), AGUA, etc;

- O fonema /j/ é representado pela letra J. Exemplos: JANELA, JOGO, GARAJEM (garagem), VIAJAR, JENTE (gente), JIGANTE (gigante), etc;

- O fonema /x/ é representado somente pela letra X, levando à extinção do CH. Exemplos: XÁ (chá), XÁCARA (chácara), XÍCARA, XEFE (chefe), CAXORRO (cachorro), XUPA (chupa), etc;

- O fonema /z/ é representado unicamente pela letra Z. Exemplos: CAZA (casa), EZENPLO (exemplo), BRAZIL (Brasil), GAZODUTO (gasoduto), ESPOZISÃO (exposição), etc;

- A letra H permanece nos dígrafos NH e LH, mas desaparece nas demais palavras. Exemplos: GALINHA, FOLHA, COLHEITA, AVER (haver), OMEN (homem), INO (hino), etc;

- A letra N passa a ser usada também antes de P e B (não mais a letra M) Exemplos: CANPO (campo), BONBA (bomba), etc.

Como exemplo que este processo de simplificação não foi uma inovação, pode-se verificar a redução de caracteres do alfabeto chinês pela revolução comunista de Mao Tsé-Tung, que de acordo com Chu (1977), teve sua alteração através da seleção dos caracteres mais habitualmente usados, a fim de formar uma lista das palavras básicas. Esses caracteres foram ensinados nas escolas elementares e nos cursos de alfabetização. O Comitê de Reforma da Língua da China Comunista publicou em 1952 uma lista de palavras comuns contendo 1010 caracteres na primeira classe com referência à frequência de utilização, e 490 caracteres na segunda classe, totalizando 1500. Além disso, há uma lista suplementar de 500 caracteres na terceira ordem de frequência. Calcula-se que, tendo

aprendido os 1 500 caracteres básicos, uma pessoa esteja capacitada a ler cerca de noventa e cinco por cento dos “textos de leitura popular”, mesmo dentro dos 52 dialetos presentes no território Chinês.

A proposta do *brazilês* é uma codificação que baseia-se na língua escrita como ferramenta (meio) e não como objetivo (fim) e facilita o comando oral de computadores e robôs, em tempo real, podendo, por exemplo, ser a solução para a síndrome dos movimentos repetitivos, que atinge as pessoas que trabalham intensamente com computadores e a inclusão digital de pessoas tetraplégicas ou que possuam problema motor em membro superior.

Este trabalho manteve os princípios que originaram o *brazilês*, teve a oportunidade de analisar e aperfeiçoar as idéias apresentadas, especialmente no que tange às sílabas, aos fonemas e na própria grafia apresentada, sempre com o objetivo de simplificar ainda mais a codificação.

Exemplificando o uso do *brazilês* em uma frase, e apresentado utilizando separação silábica apresentado neste trabalho, tem-se:

“O Bra zil po su i no vo pro se so de co mu ni ca são u zan do a te ci ni ca bra zi lês !”

4. COMBINAÇÕES DE SÍLABAS

Após a definição de algumas possibilidades que ficaram abertas na proposta original do *brazilês*, este trabalho buscou comprovar as reais reduções nas cargas computacionais nos reconhecimento de voz via *brazilês*. Isso se deu principalmente porque os programas comerciais existentes são fechados, em geral em inglês, impedindo uma comparação prática das possíveis simplificações.

Para obter as combinações de palavras e sílabas realizou-se um programa de computador, o qual gerou tabelas com as combinações esperadas dando a possibilidade de quantificá-las, subtraindo destas tabelas as combinações não utilizadas pela língua portuguesa, assim como, aquelas que não existem no *brazilês*, pode-se comprovar percentualmente a vantagem na utilização da técnica apresentada.

Etapas para chegar ao objetivo:

- Definir o programa, plataforma e metodologia de apresentação dos resultados;
- Criar o diagrama de blocos do referido programa;
- Desenvolver o programa;
- Ensaiai o programa buscando todas as combinações;
- Separar as combinações não utilizadas pela língua portuguesa;
- Separar as combinações não utilizadas pelo *brazilês*;
- Apresentar os resultados chegando aos percentuais como forma de medir a vantagem da técnica apresentada.

4.1. DEFINIÇÃO DO PROGRAMA

Foi definido um programa em linguagem C++ com o objetivo de combinar letras, formando sílabas de 2, 3, 4 e 5 letras existentes na língua portuguesa (na língua portuguesa não existem sílabas com mais de 5 letras) e apresentando em telas os resultados.

Isto se deu principalmente porque os programas comerciais existentes de reconhecimento de voz são fechados, em geral no idioma inglês, impedindo uma comparação prática das possíveis simplificações.

A listagem do programa fonte em C++ está apresentado no ANEXO 1, e a seguir apresenta-se as combinações procuradas e respectivas restrições.

A figura 3 apresenta o diagrama geral do programa, composto pelas rotinas de formação de sílabas de 2, 3, 4 e 5 letras.

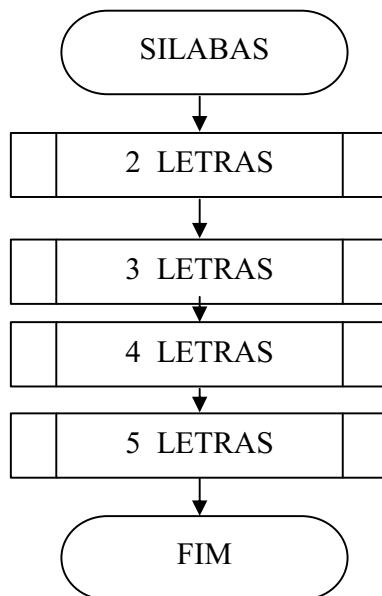


Figura 3-Diagrama de formação de sílabas

Sílabas de 2 letras - Para a formação de sílabas de 2 letras criar as seguintes combinações:

- Entre todas as letras do alfabeto (A,B,C,D,E,.....,Z) e todas as vogais (A,E,I,O,U), não permitir combinações entre duas letras iguais;
- Entre a consoante Ç e as vogais (A,O,U);
- Entre todas as vogais e as seguintes consoantes (B,D,H,L,M,N,R,S,X,Z).

A figura 4 apresenta a rotina de formação de sílabas de 2 letras.

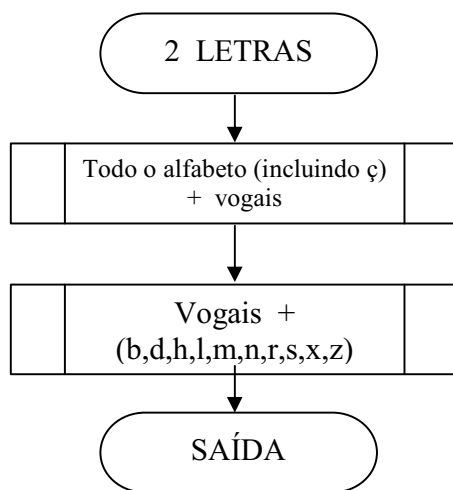


Figura 4-Diagrama de formação de sílabas de 2 letras

Sílabas de 3 letras - Para a formação de sílabas de 3 letras criar as seguintes combinações:

- Entre as letras SS e todas as vogais;
- Entre as letras RR e todas as vogais;
- Entre todas as consoantes, todas as vogais e a letra L;
- Entre todas as consoantes, todas as vogais e a letra M;
- Entre todas as consoantes, todas as vogais e a letra N;
- Entre todas as consoantes, todas as vogais e a letra R;
- Entre todas as consoantes, todas as vogais e a letra S;
- Entre todas as consoantes, todas as vogais e a letra X;
- Entre todas as consoantes, todas as vogais e a letra Z;
- Entre as consoantes (B,C,D,F,G,P,R,T,V), mais a letra R, mais todas as vogais;
- Entre as consoantes (B,C,D,F,G,P,T,V), mais a letra L, mais todas as vogais;
- Entre as letras NH e todas as vogais;

- Entre as letras LH e todas as vogais;
- Entre as letras GU e as vogais (A,E,I,O);
- Entre as letras QU e as vogais (A,E,I,O);
- Entre todas as consoantes e as vogais AO.

A figura 5 apresenta a rotina de formação de sílabas de 3 letras.

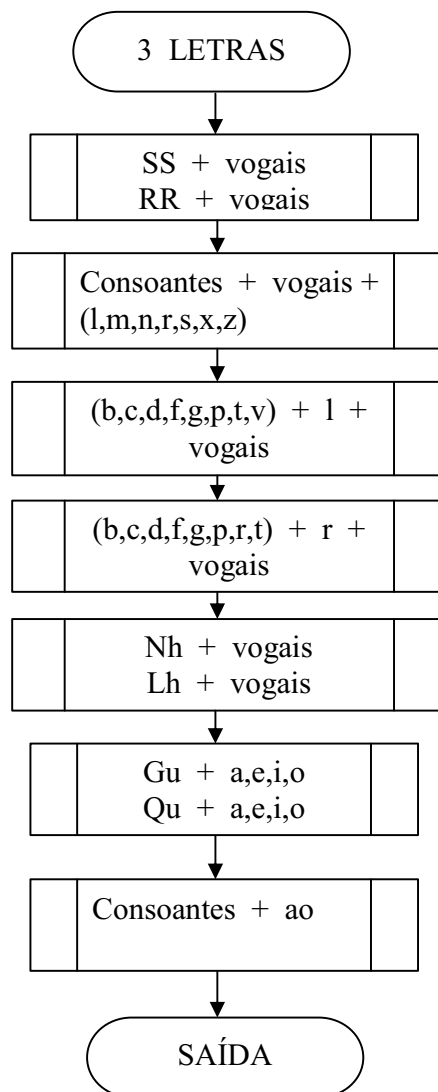


Figura 5-Diagrama de formação de sílabas de 3 letras

Sílabas de 4 letras - Para a formação de sílabas de 4 letras criar as seguintes combinações:

- Entre as seguintes consoantes (B,C,D,F,G,P,R,T,V), mais a letra R, mais todas as vogais, mais a letra L;

- Entre as seguintes consoantes (B,C,D,F,G,P,R,T,V), mais a letra R, mais todas as vogais, mais a letra M;

- Entre as seguintes consoantes (B,C,D,F,G,P,R,T,V), mais a letra R, mais todas as vogais, mais a letra N;

- Entre as seguintes consoantes (B,C,D,F,G,P,R,T,V), mais a letra R, mais todas as vogais, mais a letra R;

- Entre as seguintes consoantes (B,C,D,F,G,P,R,T,V), mais a letra R, mais todas as vogais, mais a letra S;

- Entre as seguintes consoantes (B,C,D,F,G,P,R,T,V), mais a letra R, mais todas as vogais, mais a letra U;

- Entre as seguintes consoantes (B,C,D,F,G,P,R,T,V), mais a letra R, mais todas as vogais, mais a letra Z;

- Entre as seguintes consoantes (B,C,D,F,G,P,T,V), mais a letra L, mais todas as vogais, mais a letra M;

- Entre as seguintes consoantes (B,C,D,F,G,P,T,V), mais a letra L, mais todas as vogais, mais a letra N;

- Entre as seguintes consoantes (B,C,D,F,G,P,T,V), mais a letra L, mais todas as vogais, mais a letra R;

- Entre as seguintes consoantes (B,C,D,F,G,P,T,V), mais a letra L, mais todas as vogais, mais a letra S;

- Entre as seguintes consoantes (B,C,D,F,G,P,T,V), mais a letra L, mais todas as vogais, mais a letra U;

- Entre as seguintes consoantes (B,C,D,F,G,P,T,V), mais a letra L, mais todas as vogais, mais a letra Z;

- Entre as consoantes NH, mais todas as vogais, mais a letra L;

- Entre as consoantes NH, mais todas as vogais, mais a letra M;

- Entre as consoantes NH, mais todas as vogais, mais a letra N;

- Entre as consoantes NH, mais todas as vogais, mais a letra R;
- Entre as consoantes NH, mais todas as vogais, mais a letra S;
- Entre as consoantes NH, mais todas as vogais, mais a letra U;
- Entre as consoantes LH, mais todas as vogais, mais a letra M;
- Entre as consoantes LH, mais todas as vogais, mais a letra N;
- Entre as consoantes LH, mais todas as vogais, mais a letra R;
- Entre as consoantes LH, mais todas as vogais, mais a letra S;
- Entre as letras GU, mais as seguintes vogais (A,E,I,O), mais a letra L;
- Entre as letras GU, mais as seguintes vogais (A,E,I,O), mais a letra M;
- Entre as letras GU, mais as seguintes vogais (A,E,I,O), mais a letra N;
- Entre as letras GU, mais as seguintes vogais (A,E,I,O), mais a letra R;
- Entre as letras GU, mais as seguintes vogais (A,E,I,O), mais a letra S;
- Entre as letras QU, mais as seguintes vogais (A,E,I,O), mais a letra L;
- Entre as letras QU, mais as seguintes vogais (A,E,I,O), mais a letra M;
- Entre as letras QU, mais as seguintes vogais (A,E,I,O), mais a letra N;
- Entre as letras QU, mais as seguintes vogais (A,E,I,O), mais a letra R;
- Entre as letras QU, mais as seguintes vogais (A,E,I,O), mais a letra S;
- Entre todas as consoantes, mais as vogais AO, mais a letra S;
- Entre todas as consoantes, mais as vogais AE, mais a letra S.

A figura 6 apresenta o diagrama de formação de sílabas de 4 letras.

Sílabas de 5 letras - Para a criação de sílabas de 5 letras criar as seguintes combinações:

- Entre as consoantes (B,C,D,F,G,P,R,T,V), mais a letra R, mais vogais, mais a letra N, mais a letra S;
- Entre as consoantes (B,C,D,F,G,P,R,T,V), mais a letra L, mais vogais, mais a letra N, mais a letra S.

A figura 7 apresenta o diagrama de formação de sílabas de 5 letras.

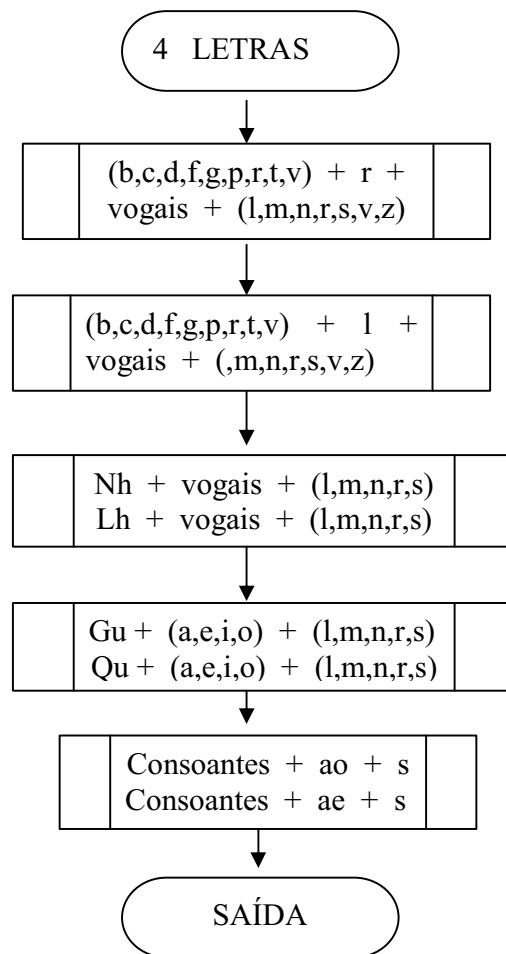


Figura 6-Diagrama de formação de sílabas de 4 letras

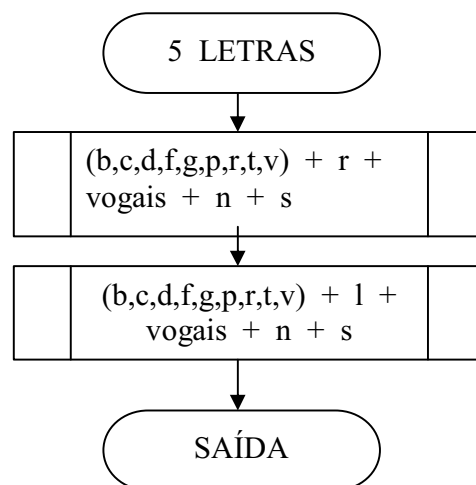


Figura 7-Diagrama de formação de sílabas de 5 letras

O programa fonte desenvolvido em linguagem C++ está apresentado no ANEXO 1.

Os resultados obtidos pela execução do programa estão apresentados no ANEXO 2.

4.2. COMBINAÇÕES NÃO UTILIZADAS

As combinações a seguir, foram apresentadas pelo programa mas não existem no português e no *brazilês*, e portanto não participaram do cálculo de totalização das sílabas:

Com 2 letras: qa, qe, qi, qo, qu

Com 3 letras: qal, qel, qil, qol, qul, qam, qem, qim, qom, qum, qan, qen, qin, qon, qun, qar, qer, qir, qor, qur, qas, qes, qis qos, qus, qax, qex, qix, qox, qux, qaz, qez, qiz, qoz, quz.

4.3. COMBINAÇÕES NÃO UTILIZADAS PELO *BRAZILÊS*

Combinações de letras existentes no português e não existentes no *brazilês*:

Com 2 letras: ha, he, hi, ho, hu, ka, ke, ki, ko, ku, wa, we, wi, wo, wu, ya, ye, yi, yo, yu, ça, ço, çu.

Com 3 letras: ssa, sse, ssi, sso, ssu, hál, hel, hil, hol, hul, kal, kel, kil, kol, kul, wal, wel, wil, wol, wul, yal, yel, yil, yol, yul, hán, hen, hin, hon, hun, kan, ken, kin, kon, kun, wan, wen, win, won, wun, yan, yen, yin, yon, yun.

Com 4 letras: bram, brem, brim, brom, brum, cram, crem, crim, crom, crum, dram, drem, drim, drom, drum, fram, frem, frim, from, frum, gram, grem, grim, grom, grum, pram, prem, prim, prom, prum, rram, rrem, rrim, rrom, rrum, tram, trem, trim, trom, trum, vram, vrem, vrim, vrom, vrum, braz, brez, briz, broz, bruz, craz, crez, criz, croz, cruz, draz, drez, driz, droz, druz, fraz, frez, friz, froz, fruz, graz, grez, griz, groz, gruz, praz, prez, priz, proz, pruz, rraz, rrez, rriz, rroz, rruz, traz, trez, triz, troz, truz, vraz, vrez, vriz, vroz, vruz, blam, blem, blim, blom, blum, clam, clem, clim, clom, clum, dlam, dlem, dlim, dlom, dlum, flam, flem, flim, flom, flum, glam, glem, glim, glom, glum, plam, plem, plim, plom, plum, tlam, tlem, tlim, tlom, tlum, vlam, vlem, vlim, vlom, vlum, blaz, blez, bliz, bloz,

bluz, claz, clez, cliz, cloz, cluz, dlaz, dlez, dliz, dloz, dluz, flaz, flez, fliz, floz, fluz, glaz, glez, gliz, gloz, gluz, plaz, plez, pliz, ploz, pluz, tlaz, tlez, tliz, toz, tluz, vlaz, vlez, vliz, vloz, vluz, nham, nhem, nhim, nhom, nhum, lham, lhem, lhim, lhom, lhum, guel, guil, guol, guam, guem, guim, guom, guen, guin, guon, guer, guir, guor, gues, guis, guos, qual, quel, quil, quol, quam, quem, quim, quom, quan, quen, quin, quon, quar, quer, quir, quor, quas, ques, quis, quos, haos, kaos, waos, yaos, haes, kaes, waes, yaes.

4.4. DIFERENÇAS ENTRE AS CODIFICAÇÕES

Através do levantamento quantitativo, que possibilitou a métrica a seguir apresentada, pode-se verificar:

- Quantidade de possíveis sílabas no português:

Com 1 letra (vogais):	5		
Com 2 letras:	123		
Com 3 letras:	798		
Com 4 letras:	700		
Com 5 letras:	85	TOTAL	: 1711

- Quantidade de possíveis sílabas no *brazilês*:

Com 1 letra (vogais):	5		
Com 2 letras:	80		
Com 3 letras:	482		
Com 4 letras:	471		
Com 5 letras:	85	TOTAL	: 1143

Portanto o *brazilês* possui 568 sílabas a menos que a língua portuguesa, assim sendo, mais de 33% de redução, o que em qualquer sistema de automação é uma redução expressiva.

Este número de sílabas apresentado é importante para a montagem da rotina silábica, possível com o *brazilês*.

Observa-se, portanto, que no reconhecimento de texto falado em português há necessidade de se esperar a montagem da possível palavra para que a codificação tenha sucesso, enquanto para o *brazilês* a cada sílaba pronunciada provocará a sua exposição no

monitor, contudo numa dicção normal este processo apresentará dificuldades, mas com o objetivo de atender um deficiente físico, e possibilitar a ele que suas informações sejam transcritas com velocidade superior e com menor possibilidade de erros, causados pela interpretação errônea de palavras, basta o locutor acostumar falar silabicamente.

Muitos destes locutores conseguem pintar com a boca, tocar violão ou outros instrumentos com os membros inferiores e superam varias dificuldades, portanto em pouco tempo se habituarão a fornecer sons com as devidas separações silábicas, estando aptos a utilizar a técnica de transcrição neste trabalho proposta.

5. GRAFIA DE FONEMAS

O comparativo das representações gráficas de fonemas da língua portuguesa e do *brazilês*, objetiva demonstrar que o *brazilês* possui menor número de representações gráficas dos fonemas, simplificando rotinas computadorizadas que utilizam estas representações. A tabela 1 apresenta um comparativo entre a representação gráfica dos fonemas na língua portuguesa e no *brazilês*

Da quantidade de grafias dos fonemas na língua portuguesa e da quantidade de grafia dos fonemas no *brazilês* obteve-se os seguintes números:

nº de fonemas (sons) na língua portuguesa -----	31
nº de grafias dos fonemas na língua portuguesa -----	54 (100%)
nº de grafias dos fonemas no <i>brazilês</i> -----	31 (-42,6%)

Comprovando-se uma diminuição até 42,6 % no tempo do processo complexo de entendimento, tendo em vista o menor número de grafias dos fonemas no *brazilês* se comparado à língua portuguesa.

tabela1-Comparativo entre a grafia fonética da língua portuguesa e do brasilês

FONEMAS	REPRESENTAÇÃO	
	LÍNGUA PORTUGUESA	BRAZILÊS
/a/	A	A
/e/	E	E
/é/	E É	É
/i/	I	I
/o/	O	O
/ó/	O, Ó	Ó
/u/	U	U
/ã/	Ã, AM, AN	AN
/~e/	EM, EN	EN
/~i/	IM, IN	IN
/õ/	Õ, OM, ON	ON
/~u/	UM, UN	UN
/p/	P	P
/b/	B	B
/m/	M	M
/f/	F	F
/v/	V	V
/t/	T	T
/d/	D	D
/n/	N	N
/nh/	NH	NH
/l/	L	L
/lh/	LH	LH
/r/	R	R
/rr/	RR, R	RR
/z/	Z, S, X	Z
/s/	S, SS, Ç, C, SC, SÇ, X, XC	S
/j/	J, G	J
/x/	X, CH	X
/g/	G, GU	G
/q/	C, QU	C

6. RECONHECIMENTO AUTOMÁTICO DE VOZ (RAV)

6.1. FUNCIONAMENTO DO RAV NOS SISTEMAS ATUAIS

O funcionamento do reconhecimento automático de voz em tempo real nos sistemas atualmente em operação está apresentado na figura 8 com respectivos detalhamentos, em português ou outro idioma qualquer.

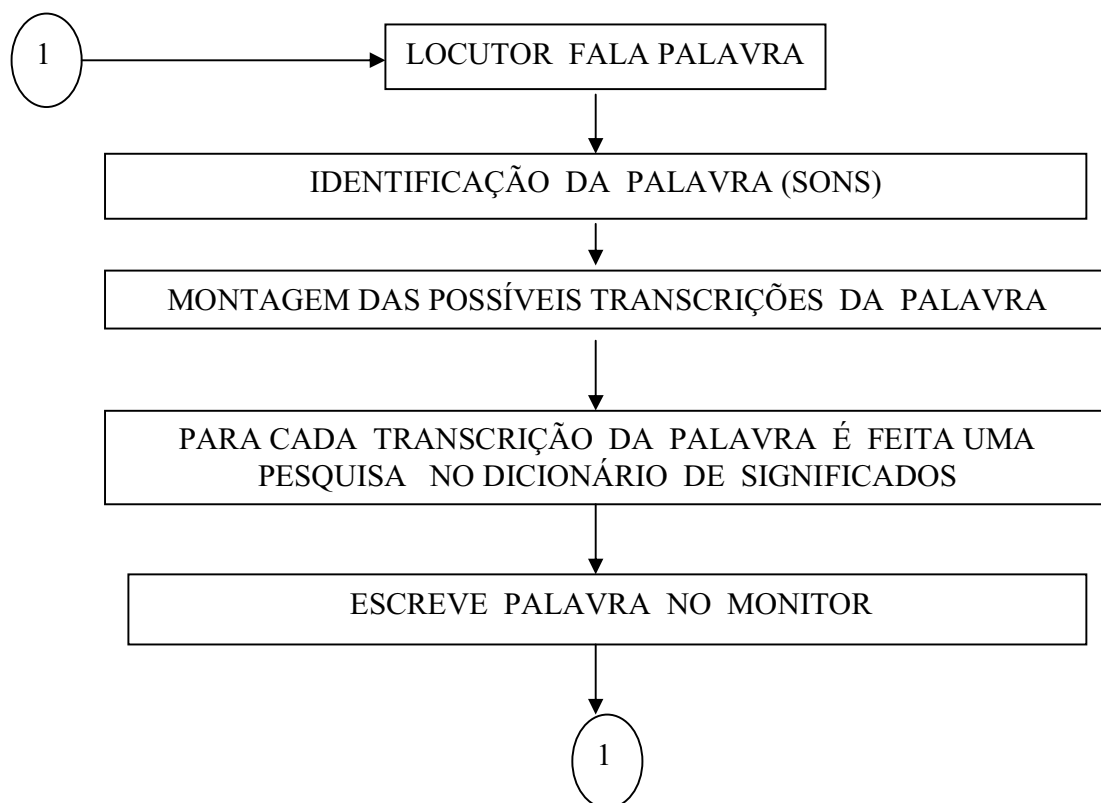


Figura 8 - Diagrama de funcionamento do RAV nos sistemas em operação

Através de bancos de dados sonoro, utilizando modelos ocultos de Markov e redes neurais, é gerada a identificação da palavra (som) falada;

È feita a montagem das possíveis transcrições das palavra (muitas vezes mais que 3, dependendo dos fonemas que compõem a palavra);

Para cada transcrição da palavra é feita uma pesquisa no dicionário de significados (se para uma mesma palavra houver mais de uma forma de transcrição, haverá esta mesma quantidade de pesquisas no dicionário de significados).

6.2. PROPOSTA DE FUNCIONAMENTO DO RAV UTILIZANDO O *BRAZILÊS*

Através de banco de dados sonoro silábico, Markov e redes neurais, é gerada a identificação da sílaba (som), dirigindo a formação da palavra;

Há a montagem da transcrição de cada sílaba, onde cada fonema terá somente uma grafia, propiciando o processo de transcrição da palavra;

A sílaba é transcrita (cada sílaba terá uma única transcrição).

O funcionamento do reconhecimento automático de voz nos sistemas utilizando a proposta *brazilês* que trata este trabalho está apresentado na figura 9.

6.3. COMPARATIVO ENTRE A MONTAGEM POR PALAVRA X MONTAGEM SILÁBICA

Somado aos benefícios apresentados pelo funcionamento do RAV utilizando o *brazilês*, tem-se ainda a rapidez que os sistemas computacionais vão adquirir com a possibilidade de montagem das palavras através da interpretação silábica.

A tabela 2 apresenta o comparativo entre as duas situações, uma atualmente em operação e outra com a implementação da proposta *brazilês*.

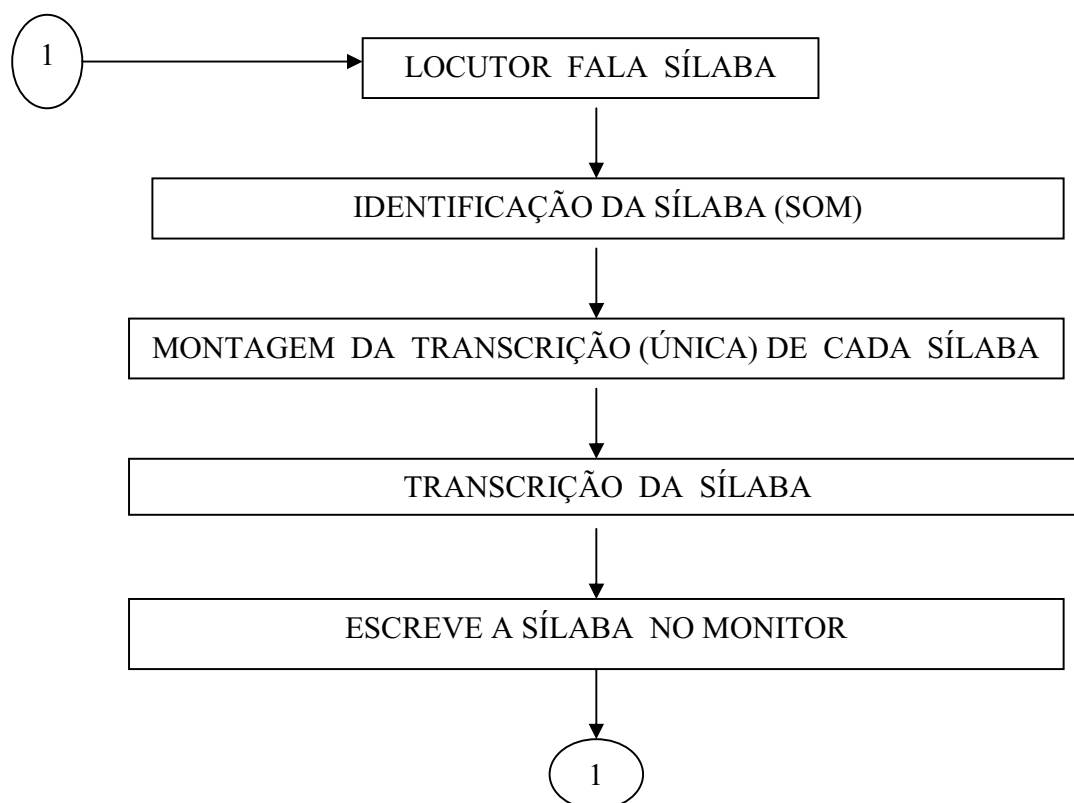


Figura 9-Diagrama de funcionamento do RAV utilizando o dicionário fonético

tabela 2 - Comparativo entre montagem por palavra e a montagem silábica (*brazilês*)

ATUALMENTE	COM O <i>BRAZILÊS</i>
Banco de dados sonoro por palavra (o dicionário (ufpadic.3.0 possui mais de 65.000 palavras)	Banco de dados sonoro por sílabas (1.143 sílabas <i>brazilês</i>)
Busca pela palavra (exigindo sua montagem)	Busca pela sílaba (dirigindo a definição da palavra)
Para cada transcrição da palavra é feita uma pesquisa no dicionário de significados	Há somente uma transcrição da sílaba

6.4. ANÁLISE DOS RESULTADOS

Este trabalho teve como objetivo apresentar a técnica e discuti-la à exaustão, portanto não teve como princípio desenvolver um banco de dados contendo o som das sílabas e produzir a interpretação destas através de programa específico, tendo em vista que estes programas possuem tecnologias de processamento digital de sinais e são proprietárias, exemplo: iListen, Via Voice, Naturally Speaking, entre outros.

Porém em caráter de estudos, alterando apenas o banco de dados já existente com a proposta *brazilês*, o processo em funcionamento ficará mais veloz e com maior eficiência, sem diminuir as qualidades dos sistemas hoje em funcionamento.

Esta velocidade é verificada através da substituição do banco de dados composto por mais de 65.000 palavras, necessitando número expressivo de comparações, por um banco silábico composto por 1143 sílabas no caso do *brazilês*. Considerando apenas a quantidade de elementos dos dois bancos de dados, um de palavras outro de sílabas, sem considerar o tempo de processamento necessário para a interpretação das palavras que é muito maior que a interpretação de sílabas, pode-se representar numericamente o tempo gasto pelo processador, somado a plataforma operacional e o número de acesso ao respectivo banco de dados.

Assim sendo, supondo que o tempo de uma intervenção ao processador seja de 10 a 100 μ s, o tempo de processamento para executar da transcrição em tempo real, ou seja, desde o momento da emissão do som até o aparecimento do resultado no monitor de um computador, levará para um banco de dados com 65.000 itens, no caso o reconhecimento de palavras em português, de 650.000 μ s à 6.500.000 μ s, convertendo para segundos tem-se: de 0,65 segundos por palavra à 6,5 segundos por palavra, enquanto para o mesmo processo utilizando o banco de dados silábico proposto pela técnica neste trabalho apresentada, através do *brazilês*, com o banco de dados de 1143 sílabas, seriam gastos, para o mesmo processador utilizado, de 11430 μ s à 114300 μ s, convertendo para segundos tem-se: de 0,01 segundos por sílaba à 0,11 segundos por sílaba.

Comparando vê-se que para as palavras em português seriam transcritas, desde a emissão do som até a apresentação no monitor do computador, de 0,65 a 6,5 seg/palavra enquanto para o *brazilês* de 0,01 a 0,11 seg/sílaba, ou seja, com o dicionário fonético aqui tratado, uma palavra de 3 sílabas levaria de 0,03 a 0,33 s.

7. CONCLUSÃO

A automação pode se agregar a outros processos com objetivo de minimizar o problema da acessibilidade, buscando facilitar a comunicação homem-máquina.

Neste trabalho utilizou-se a proposta do *brazilês*, agregada a algumas sugestões tanto de grafia como de acentuação de fonemas, chegando-se a uma codificação simplificada para uso em reconhecimento eletrônico de sílabas, permitindo a transcrição fonética de textos automaticamente, facilitando o comando oral de computadores e robôs em tempo real com maior coerência e lógica, já que o mesmo fonema é representado por um único símbolo.

Constatou-se que a transcrição fonética no português utiliza um número de comparações muito grande, já que o mesmo símbolo ortográfico pode representar sons diferentes e, diversos símbolos podem representar o mesmo som. Devido à imprevisibilidade da codificação do português a comunicação homem-máquina, em tempo real, verificou-se que atualmente só é viável nas tarefas específicas em que o vocabulário é bem reduzido.

Somado ao apresentado, é possível promover a transcrição em tempo real de textos falados, que posteriormente poderiam ser convertidos, novamente para o português usando-se um dicionário eletrônico *brazilês*-português.

Os softwares de reconhecimento de voz atuais usam sistemas de modelos estatísticos poderosos e complicados. Eles usam funções de probabilidade e matemática para determinar o resultado mais provável através de modelos como o Oculto de Markov e as redes neurais, com resultados satisfatórios, principalmente em tarefas e comandos restritos, visto que o tempo de processamento é barreira para aplicações *on-line*.

O uso do *brazilês* abre a possibilidade de criação de um banco de dados silábico, em que cada fonema terá somente uma representação gráfica, propiciando a concatenação das sílabas para formação das palavras, diminuindo sensivelmente o tamanho do banco de dados, o número de comparações e por consequência dando maior rapidez ao processo de interação homem-máquina.

Assim softwares como o iListen, Via Voice, Naturally Speaking, entre outros, poderão utilizar esta nova rotina, aqui proposta, com o objetivo de aumentar suas respectivas velocidades de resposta. Pois a diminuição de uma ou mais pesquisas no

processo de interpretação de cada fonema, no software para reconhecimento de voz, aumenta sensivelmente a velocidade de processamento.

O objetivo desta técnica é agilizar o processo hoje utilizado à interação homem-máquina melhorando a acessibilidade usando o português falado no Brasil.

A técnica, portanto, possibilita trabalhar com a linguagem natural, ou seja, que o computador reconheça a fala natural e escreva a sílaba, a palavra, a frase no monitor de vídeo ou desenvolva uma ação, atendendo ao solicitado.

A proposta de que trata este trabalho traz melhorias irrefutáveis no processo automático de interpretação e tradução de textos escritos e falados, inclusive para sistemas embarcados muito utilizados na área industrial e especialmente em processos de acessibilidade, através da execução de rotinas com menor número de comparações, facilitando a comunicação homem-máquina em tempo real, atingindo os resultados encontrados atualmente, porém com maior velocidade de processamento e menor possibilidade de erros.

Enfim, comparando-se a situação dos sistemas de reconhecimento automático de voz em funcionamento atualmente e utilizando a proposta aqui tratada, encontra-se os seguintes resultados positivos:

- palavras x sílabas,
- acesso a 65.000 palavras X 1.143 sílabas (98,25% menor),
- várias formas de transcrição da palavra X única forma de transcrição da sílaba,
- 54 grafias de fonemas X 31 grafias de fonemas (mais de 42% menor).

Pode-se verificar a diminuição do tempo gasto utilizando o *brazilês* como técnica facilitadora do processo de reconhecimento automático de voz auxiliando a acessibilidade através da voz humana com português falado no Brasil.

Assim sendo, pode-se desenvolver programas para computador, através da técnica aqui apresentada, para transcrição em tempo real de textos falados em português do Brasil possibilitando ao usuário interagir mais facilmente com os computadores e proporcionando a inclusão digital a deficientes físicos, muitos dos quais pintam com a boca ou com os pés, tocam violão com os pés, mas que para utilizar o computador precisam de alguém para auxiliá-lo ou adquirir equipamentos com recursos de *software e hardware*, com custos muitas vezes proibitivos.

REFERÊNCIAS BIBLIOGRÁFICAS

ABNT NBR 9050. Válida a partir de 30.06.2004, Acessibilidade a edificações, mobiliário, espaços e equipamentos urbanos.

ACESSIBILIDADE ANATEL

<http://www.anatel.gov.br/Portal/exibirPortalNivelDois.do?codItemCanal=1019&nomeVisao=Cidad%E3o&nomeCanal=Informa%E7%F5es%20e%20consultas&nomeItemCanal=Acessibilidade>. Página acessada em abril de 2011

CHU, YU KUANG. *Interação entre linguagem e pensamento em chinês, ideograma* (1977), Cultrix São Paulo, acessado em junho de 2011, pela página:

<http://chinaimperial.blogspot.com/2008/04/interao-entre-linguagem-e-pensamento-em.html>

DIAS, R de F. *Ser ou não ser interativo*, 2003.

www.saladeaulainterativa.pro.br/texto_0007.htm

FIORIN, JOSÉ LUIZ. *Língua, discurso e política*. Alea Estudos Neolatinos vol.11 no.1 Rio de Janeiro Jan./Junho 2009

http://www.scielo.br/scielo.php?pid=S1517-106X2009000100012&script=sci_arttext

HUGO, MARCEL. *Uma interface de reconhecimento de voz para o sistema de gerenciamento de central de informação de fretes*. (1995) Dissertação submetida A Universidade Federal de Santa Catarina para a obtenção do Grau de Mestre em Engenharia. Florianópolis, Setembro de 1995. http://www.eps.ufsc.br/disserta/hugo/cap_2/cp2_hug.htm

JORNAL FOLHA DE SÃO PAULO, em 10 de outubro de 2010, <http://www1.folha.uol.com.br/tec/811967-reconhecimento-de-fala-tem-inumeras-aplicacoes.shtml>. Página acessada em abril de 2011

LÉVY, P. *As tecnologias da inteligência: o futuro do pensamento na era da informática*. São Paulo: 34, 2000.

MANDELMAN, GILZA PAIM. MARCELINO, MÁRCIO ABUD. MANDELMAN MARCIO. *Comunicação homem-máquina por meio de linguagem natural falada no Brasil*. Revista Sinergia volume 11 n. 2 IFSP (julho – dezembro de 2010)

MARCELINO, MÁRCIO ABUD. *Brazilês uma proposta de codificação para a língua escrita no Brasil*. Editora da UNESP, Taubaté: 2008. 152p.

MARTINS, JOSÉ ANTÔNIO. *Avaliação de diferentes técnicas para reconhecimento de fala*. Trabalho de doutorado em engenharia elétrica UNICAMP – dezembro de 1997

MONTEIRO, CLAUDIA GUERRA. *O papel educativo dos meios de comunicação* http://www.ipv.pt/forumedia/3/3_fi3.htm Revista do curso de comunicação social da esev Forum media n. 3 (2000)

MÜLLER, DANIEL NEHME. *Reconhecimento Semântico Através de Redes Neurais Artificiais* _Porto Alegre: CPGCC da UFRGS, 1996. Dissertação (mestrado)_ Universidade Federal do Rio Grande do Sul. Curso de Pós-Graduação em Ciência da Computação, Porto Alegre, BRRS, 1996

ORWELL, GEORGE. "1984", trad. Wilson Velloso. 17ª edição -Cia Edit Nacional - (páginas 51 a 53)

PETRY, A. *Reconhecimento Automático de Locutor Utilizando medidas de invariantes dinâmicas não-lineares*. Tese de doutorado. UFRS,2002.

ROVER, J. A.. *Representação da realidade*. In: *Informática no Direito: inteligência artificial. Introdução aos sistemas especialistas legais*. Curitiba, 2001. <http://www.infojur.ufsc.br/aires/arquivos/tese%20representacao%20realidade.pdf>

SILVA, CARLOS PATRICK ALVES DA . *Um Software de Reconhecimento de Voz para Português Brasileiro*. UFPA/ITEC/PPGEE. (Dissertação de Mestrado). Belém, PA. (2010)

TEVAH, RAFAEL TERUSZKIN, *Implementação de um Sistema de Reconhecimento de Fala Contínua com Amplo Vocabulário para o Português Brasileiro* [Rio de Janeiro] 2006 XI, (COPPE/UFRJ, M.Sc., Engenharia Elétrica, 2006) Dissertação - Universidade Federal do Rio de Janeiro, COPPE

VIOLARO, FÁBIO. *Minicurso: processamento digital de sinais de fala*. XV Simpósio Brasileiro de Telecomunicações pela UNICAMP. Recife, 8 a 11 de setembro de 1997

ANEXO 1 – Apresenta a listagem do programa fonte em C++ elaborado para criação das combinações de letras para formação de sílabas na língua portuguesa

```
// PROGRAMA EM C++ PARA IMPRESSÃO DAS POSSÍVEIS COMBINAÇÕES
// DE LETRAS PARA FORMAÇÃO DE SÍLABAS NA LINGUA PORTUGUESA
//-----
```

```
#include <vcl.h>
#include <stdio.h>
#include <conio.h>
#pragma hdrstop
```

```
//-----
```

```
#pragma argsused
int wi, wj, wk, wl;
char vogais[5] = {'A','E','I','O','U'};
char consoantes[21] =
{'B','C','D','F','G','H','J','K','L','M','N','P','Q','R','S','T','V','W','X','Y','Z'};
char si3l[4];
char car;
char car1;
char car2;
char si3letra[130][4];
char si3lf[130][4];
char si3mf[130][4];
char si3nf[130][4];
char si3rf[130][4];
char si3sf[130][4];
char si3xf[130][4];
char si3zf[130][4];
```

```
char cons9[9] = {'B','C','D','F','G','P','R','T','V'};
char cons8[8] = {'B','C','D','F','G','P','T','V'};
char clv[4];
char cletrav[130][4];
char crv3f[130][4];
char clv3f[130][4];
```

```
char clvl1[5];
char cletrav1[130][5];
char crvl[130][5];
char crvm[130][5];
```



```
char crvn[130][5];
char crvr[130][5];
char crvs[130][5];
char crvu[130][5];
char crvz[130][5];
```

```
char clvl2[5];
char cletrav2[130][5];
char clvm[130][5];
char clvn[130][5];
char clvr[130][5];
char clvs[130][5];
char clvu[130][5];
char clvz[130][5];
```

```
char nh1[5] = "NH";
char si3fnh1[99][5];
char nhfl[99][5];
char nhfm[99][5];
char nhfn[99][5];
char nhfr[99][5];
char nhfs[99][5];
char nhfu[99][5];
```

```
char lh1[5] = "LH";
char si3flh1[99][5];
char lhfm[99][5];
char lhfn[99][5];
char lhfr[99][5];
char lhfs[99][5];
```

```
char vogais4[4] = {'A','E','I','O'};
char gu1[5] = "GU";
char si3fgu1[99][5];
char gufl[99][5];
char gufm[99][5];
char gufn[99][5];
char gufr[99][5];
char gufs[99][5];
```

```
char qu1[5] = "QU";
char si3fqu1[99][5];
char qufl[99][5];
char qufm[99][5];
char qufn[99][5];
char qufr[99][5];
```

```

char qufs[99][5];

char co1[5];
char si3fco1[99][5];
char cofs[99][5];

char ca1[5];
char si3fca1[99][5];
char cafs[99][5];

// função para combinações entre todas as consoantes, todas as vogais e uma letra
consvogaisletra(char car)
{
    wl=0;
    for (wi=0;wi<21;wi++)
    {
        for (wj=0;wj<5;wj++)
        {
            wk=0;
            si3l[wk] = consoantes[wi];
            si3l[wk+1] = vogais[wj];
            si3l[wk+2] = car;
            si3l[wk+3] = '\0';
            strcpy(si3letra[wl] , si3l);
            wl++;
        }
    }
}

conletravogais9(char car)
{
    wl=0;
    for (wi=0;wi<9;wi++)
    {
        for (wj=0;wj<5;wj++)
        {
            wk=0;
            clv[wk] = cons9[wi];
            clv[wk+1] = car;
            clv[wk+2] = vogais[wj];
            clv[wk+3] = '\0';
            strcpy(cletrav[wl] , clv);
            wl++;
        }
    }
}

```

```

consletravogais8(char car)
{
wl=0;
for (wi=0;wi<9;wi++)
{
for (wj=0;wj<5;wj++)
{
wk=0;
clv[wk] = cons8[wi];
clv[wk+1] = car;
clv[wk+2] = vogais[wj];
clv[wk+3] = '\0';
strcpy(cletrav[w1] , clv);
wl++;
}
}
}

consletravogaisletra1(char car1, char car2)
{
wl=0;
for (wi=0;wi<9;wi++)
{
for (wj=0;wj<5;wj++)
{
if (vogais[wj] != car2)
{
wk=0;
clv11[wk] = cons9[wi];
clv11[wk+1] = car1;
clv11[wk+2] = vogais[wj];
clv11[wk+3] = car2;
clv11[wk+4] = '\0';
strcpy(cletrav1[w1] , clv11);
wl++;
}
}
}
}

consletravogaisletra2(char car1, char car2)
{
wl=0;
for (wi=0;wi<8;wi++)
{
for (wj=0;wj<5;wj++)
{

```

```

        if (vogais[wj] != car2)
        {
            wk=0;
            clv12[wk] = cons8[wi];
            clv12[wk+1] = car1;
            clv12[wk+2] = vogais[wj];
            clv12[wk+3] = car2;
            clv12[wk+4] = '\0';
            strcpy(cletrav2[w1] , clv12);
            w1++;
        }
    }
}

```

```

nhvogaisletra(char car)
{
    w1=0;
    for (wj=0;wj<5;wj++)
    {
        if (vogais[wj] != car)
        {
            wk=0;
            nh1[wk+2] = vogais[wj];
            nh1[wk+3] = car;
            nh1[wk+4] = '\0';
            strcpy(si3fnh1[w1] , nh1);
            w1++;
        }
    }
}

```

```

lhvogaisletra(char car)
{
    w1=0;
    for (wj=0;wj<5;wj++)
    {
        if (vogais[wj] != car)
        {
            wk=0;
            lh1[wk+2] = vogais[wj];
            lh1[wk+3] = car;
            lh1[wk+4] = '\0';
            strcpy(si3flh1[w1] , lh1);
            w1++;
        }
    }
}

```

```
}

```

```
guvogaisletra(char car)
{
    wl=0;
    for (wj=0;wj<4;wj++)
    {
        wk=0;
        gu1[wk+2] = vogais4[wj];
        gu1[wk+3] = car;
        gu1[wk+4] = '\0';
        strcpy(si3fgu1[wl] , gu1);
        wl++;
    }
}

```

```
quvogaisletra(char car)
{
    wl=0;
    for (wj=0;wj<4;wj++)
    {
        wk=0;
        qu1[wk+2] = vogais4[wj];
        qu1[wk+3] = car;
        qu1[wk+4] = '\0';
        strcpy(si3fqu1[wl] , qu1);
        wl++;
    }
}

```

```
consvogaisaoletra(char car)
{
    wl=0;
    for (wi=0;wi<21;wi++)
    {
        wk=0;
        col[wk] = consoantes[wi];
        col[wk+1] = 'A';
        col[wk+2] = 'O';
        col[wk+3] = car;
        col[wk+4] = '\0';
        strcpy(si3fcol[wl] , col);
        wl++;
    }
}

```

```
consvogaisaeletra(char car)

```

```

{
wl=0;
for (wi=0;wi<21;wi++)
    {
    wk=0;
    ca1[wk] = consoantes[wi];
    ca1[wk+1] = 'A';
    ca1[wk+2] = 'E';
    ca1[wk+3] = car;
    ca1[wk+4] = '\0';
    strcpy(si3fca1[wl] , ca1);
    wl++;
    }
}

int main(int argc, char* argv[])
{
int l;

char letras[26] =
{'A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z'};
char si2[3];
char si2f[130][3];

char cedilha = 'ç';
char vogaisaou[3]= {'A','O','U'};
char si2fced[4][3];

char consoant[10] = {'B','D','H','L','M','N','R','S','X','Z'};
char si2fbdh[99][3];

char ss[4] = "SS";
char si3fss[99][4];

char rr[4] = "RR";
char si3fir[99][4];

char nh[4] = "NH";
char si3fnh[99][4];

char lh[4] = "LH";
char si3flh[99][4];

char vogaisaeio[4]= {'A','E','I','O'};
char gu[4] = "GU";
char si3fgu[99][4];

```

```

char qu[4] = "QU";
char si3fqu[99][4];

char vogaisao[2]= {'A','O'};
char consvogao[3];
char consvogaisao[99][3];

// limpar as matrizes de sílabas de duas, tres e quatro letras
for (wi=0;wi<130;wi++)
    {
    strcpy(si2f[wi]," ");
    strcpy(si3letra[wi]," ");
    strcpy(si3lf[wi]," ");
    strcpy(cletrav[wi]," ");
    strcpy(crv3f[wi]," ");
    strcpy(clv3f[wi]," ");
    strcpy(cletrav1[wi]," ");
    strcpy(crv1[wi]," ");
    strcpy(crvm[wi]," ");
    strcpy(crvn[wi]," ");
    strcpy(crvr[wi]," ");
    strcpy(crvs[wi]," ");
    strcpy(crvu[wi]," ");
    strcpy(crvz[wi]," ");
    strcpy(cletrav2[wi]," ");
    strcpy(clvm[wi]," ");
    strcpy(clvn[wi]," ");
    strcpy(clvr[wi]," ");
    strcpy(clvs[wi]," ");
    strcpy(clvu[wi]," ");
    strcpy(clvz[wi]," ");
    }

for (wi=0;wi<4;wi++)
    strcpy(si2fced[wi]," ");

for (wi=0;wi<99;wi++)
    {
    strcpy(si2fbdh[wi]," ");
    strcpy(si3fss[wi]," ");
    strcpy(si3frr[wi]," ");
    strcpy(si3fnh[wi]," ");
    strcpy(si3flh[wi]," ");
    strcpy(si3fgu[wi]," ");
    strcpy(si3fqu[wi]," ");
    strcpy(consvogaisao[wi]," ");
    strcpy(si3fnh1[wi]," ");
    }

```

```

    strcpy(si3flh1[wi], " ");
    strcpy(si3fgu1[wi], " ");
    strcpy(si3fqu1[wi], " ");
    strcpy(si3fco1[wi], " ");
    strcpy(si3fca1[wi], " ");
}
// criar as combinações

wl=0;
for (wi=0;wi<26;wi++)
{
    for (wj=0;wj<5;wj++)
    {
        if (letras[wi] != vogais[wj]) // verificação de
            // igualdade
        {
            wk=0;
            si2[wk] = letras[wi];
            si2[wk+1] = vogais[wj];
            si2[wk+2] = '\0';
            strcpy(si2f[wl] , si2);
            wl++;
        }
    }
}
printf("\n\nAPRESENTACAO DAS COMBINACOES DAS SILABAS DE DUAS
LETRAS\n\n");
wi=0;
while(1)
{
    if (strcmp(si2f[wi], " ")==0)
        break;
    printf("%s ", si2f[wi]);
    wi++;
}

wl=0;
for (wj=0;wj<3;wj++)
{
    wk=0;
    si2[wk] = cedilha;
    si2[wk+1] = vogaisaou[wj];
    si2[wk+2] = '\0';
    strcpy(si2fced[wl] , si2);
    wl++;
}

```



```

printf("\n\n\n");
wi=0;
while(1)
{
    if (strcmp(si2fced[wi]," ")==0)
        break;
    printf("%s ",si2fced[wi]);
    wi++;
}

wl=0;
for (wi=0;wi<10;wi++)
{
    for (wj=0;wj<5;wj++)
    {
        wk=0;
        si2[wk] = consoant[wi];
        si2[wk+1] = vogais[wj];
        si2[wk+2] = '\0';
        strcpy(si2fbdh[wl] , si2);
        wl++;
    }
}
printf("\n\n\n");
wi=0;
while(1)
{
    if (strcmp(si2fbdh[wi]," ")==0)
        break;
    printf("%s ",si2fbdh[wi]);
    wi++;
}
printf("\n\n\nAPRESENTACAO DAS COMBINACOES DAS SILABAS DE TRES
LETRAS\n\n");

wl=0;
for (wj=0;wj<5;wj++)
{
    wk=0;
    ss[wk+2] = vogais[wj];
    ss[wk+3] = '\0';
    strcpy(si3fss[wl] , ss);
    wl++;
}
wi=0;
while(1)
{

```

```

    if (strcmp(si3fss[wi], " ")==0)
        break;
    printf("%s ", si3fss[wi]);
    wi++;
}

wl=0;
for (wj=0;wj<5;wj++)
    {
    wk=0;
    rr[wk+2] = vogais[wj];
    rr[wk+3] = '\0';
    strcpy(si3frr[wl] , rr);
    wl++;
    }
printf("\n\n\n");
wi=0;
while(1)
{
    if (strcmp(si3frr[wi], " ")==0)
        break;
    printf("%s ", si3frr[wi]);
    wi++;
}

car='L';
consvogaisletra(car);
for (l=0;l<130;l++)
    strcpy(si3lf[l] , si3letra[l]);

printf("\n\n\n");
wi=0;
while(1)
{
    if (strcmp(si3lf[wi], " ")==0)
        break;
    printf("%s ", si3lf[wi]);
    wi++;
}

car='M';
consvogaisletra(car);
for (l=0;l<130;l++)
    strcpy(si3mf[l] , si3letra[l]);

printf("\n\n\n");
wi=0;

```

```

while(1)
{
    if (strcmp(si3mf[wi], " ")==0)
        break;
    printf("%s ", si3mf[wi]);
    wi++;
}

```

```

car='N';
consvogaisletra(car);
for (l=0;l<130;l++)
    strcpy(si3nf[l], si3letra[l]);

```

```

printf("\n\n\n");
wi=0;
while(1)
{
    if (strcmp(si3nf[wi], " ")==0)
        break;
    printf("%s ", si3nf[wi]);
    wi++;
}

```

```

car='R';
consvogaisletra(car);
for (l=0;l<130;l++)
    strcpy(si3rf[l], si3letra[l]);

```

```

printf("\n\n\n");
wi=0;
while(1)
{
    if (strcmp(si3rf[wi], " ")==0)
        break;
    printf("%s ", si3rf[wi]);
    wi++;
}

```

```

car='S';
consvogaisletra(car);
for (l=0;l<130;l++)
    strcpy(si3sf[l], si3letra[l]);

```

```

printf("\n\n\n");
wi=0;
while(1)
{

```

```

        if (strcmp(si3sf[wi], " ")==0)
            break;
        printf("%s ", si3sf[wi]);
        wi++;
    }

    car='X';
    consvogaisletra(car);
    for (l=0;l<130;l++)
        strcpy(si3xf[l], si3letra[l]);

    printf("\n\n\n");
    wi=0;
    while(1)
    {
        if (strcmp(si3xf[wi], " ")==0)
            break;
        printf("%s ", si3xf[wi]);
        wi++;
    }

    car='Z';
    consvogaisletra(car);
    for (l=0;l<130;l++)
        strcpy(si3zf[l], si3letra[l]);

    printf("\n\n\n");
    wi=0;
    while(1)
    {
        if (strcmp(si3zf[wi], " ")==0)
            break;
        printf("%s ", si3zf[wi]);
        wi++;
    }

    car='R';
    consletravogais9(car);
    for (l=0;l<130;l++)
        strcpy(crv3f[l], cletrav[l]);

    printf("\n\n\n");
    wi=0;
    while(1)
    {
        if (strcmp(crv3f[wi], " ")==0)
            break;

```

```

    printf("%s ", crv3f[wi]);
    wi++;
}

car='L';
consletravogais8(car);
for (l=0;l<130;l++)
    strcpy(clv3f[l] , cletrav[l]);

printf("\n\n\n");
wi=0;
while(1)
{
    if (strcmp(clv3f[wi], " ")==0)
        break;
    printf("%s ", clv3f[wi]);
    wi++;
}

wl=0;
for (wj=0;wj<5;wj++)
{
    wk=0;
    nh[wk+2] = vogais[wj];
    nh[wk+3] = '\0';
    strcpy(si3fnh[wl] , nh);
    wl++;
}
printf("\n\n\n");
wi=0;
while(1)
{
    if (strcmp(si3fnh[wi], " ")==0)
        break;
    printf("%s ", si3fnh[wi]);
    wi++;
}

//printf("\n\n\n");
wl=0;
for (wj=0;wj<5;wj++)
{
    wk=0;
    lh[wk+2] = vogais[wj];
    lh[wk+3] = '\0';
    strcpy(si3flh[wl] , lh);
    wl++;
}

```

```

    }
printf("\n\n\n");
wi=0;
while(1)
{
    if (strcmp(si3flh[wi], " ")==0)
        break;
    printf("%s ", si3flh[wi]);
    wi++;
}

printf("\n\n\n");
wl=0;
for (wj=0;wj<4;wj++)
{
    wk=0;
    gu[wk+2] = vogaisaeio[wj];
    gu[wk+3] = '\0';
    strcpy(si3fgu[wl] , gu);
    wl++;
}
wi=0;
while(1)
{
    if (strcmp(si3fgu[wi], " ")==0)
        break;
    printf("%s ", si3fgu[wi]);
    wi++;
}

printf("\n\n\n");
wl=0;
for (wj=0;wj<4;wj++)
{
    wk=0;
    qu[wk+2] = vogaisaeio[wj];
    qu[wk+3] = '\0';
    strcpy(si3fqu[wl] , qu);
    wl++;
}
wi=0;
while(1)
{
    if (strcmp(si3fqu[wi], " ")==0)
        break;
    printf("%s ", si3fqu[wi]);
    wi++;
}

```

```

}

wl=0;
for (wi=0;wi<21;wi++)
{
    for (wj=0;wj<2;wj++)
    {
        wk=0;
        consvogao[wk] = consoantes[wi];
        consvogao[wk+1] = vogaisao[wj];
        si3l[wk+3] = '\0';
        strcpy(consvogaisao[wl] , consvogao);
        wl++;
    }
}

printf("\n\n\n");
wi=0;
while(1)
{
    if (strcmp(consvogaisao[wi]," ")==0)
        break;
    printf("%s ",consvogaisao[wi]);
    wi++;
}

printf("\n\nAPRESENTACAO DAS COMBINACOES DAS SILABAS DE QUATRO
LETRAS");

car1='R';
car2='L';
consletravogaisletral(car1, car2);
for (l=0;l<130;l++)
    strcpy(crvl[l] , cletrav1[l]);

printf("\n\n\n");
wi=0;
while(1)
{
    if (strcmp(crvl[wi]," ")==0)
        break;
    printf("%s ",crvl[wi]);
    wi++;
}

car1='R';
car2='M';

```

```

consletravogaisletra1(car1, car2);
for (l=0;l<130;l++)
    strcpy(crvm[l] , cletrav1[l]);

printf("\n\n\n");
wi=0;
while(1)
{
    if (strcmp(crvm[wi], " ")==0)
        break;
    printf("%s ",crvm[wi]);
    wi++;
}

car1='R';
car2='N';
consletravogaisletra1(car1, car2);
for (l=0;l<130;l++)
    strcpy(crvn[l] , cletrav1[l]);

printf("\n\n\n");
wi=0;
while(1)
{
    if (strcmp(crvn[wi], " ")==0)
        break;
    printf("%s ",crvn[wi]);
    wi++;
}

car1='R';
car2='R';
consletravogaisletra1(car1, car2);
for (l=0;l<130;l++)
    strcpy(crvr[l] , cletrav1[l]);

printf("\n\n\n");
wi=0;
while(1)
{
    if (strcmp(crvr[wi], " ")==0)
        break;
    printf("%s ",crvr[wi]);
    wi++;
}

car1='R';

```



```

car2='S';
consletravogaisletra1(car1, car2);
for (l=0;l<130;l++)
    strcpy(crvs[l] , cletrav1[l]);

printf("\n\n\n");
wi=0;
while(1)
{
    if (strcmp(crvs[wi], " ")==0)
        break;
    printf("%s ",crvs[wi]);
    wi++;
}

car1='R';
car2='U';
consletravogaisletra1(car1, car2);
for (l=0;l<130;l++)
    strcpy(crvu[l] , cletrav1[l]);

printf("\n\n\n");
wi=0;
while(1)
{
    if (strcmp(crvu[wi], " ")==0)
        break;
    printf("%s ",crvu[wi]);
    wi++;
}

car1='R';
car2='Z';
consletravogaisletra1(car1, car2);
for (l=0;l<130;l++)
    strcpy(crvz[l] , cletrav1[l]);

printf("\n\n\n");
wi=0;
while(1)
{
    if (strcmp(crvz[wi], " ")==0)
        break;
    printf("%s ",crvz[wi]);
    wi++;
}

```

```

car1='L';
car2='M';
consletravogaisletra2(car1, car2);
for (l=0;l<130;l++)
    strcpy(clvm[l] , cletrav2[l]);

printf("\n\n\n");
wi=0;
while(1)
{
    if (strcmp(clvm[wi], " ")==0)
        break;
    printf("%s ",clvm[wi]);
    wi++;
}

car1='L';
car2='R';
consletravogaisletra2(car1, car2);
for (l=0;l<130;l++)
    strcpy(clvr[l] , cletrav2[l]);

printf("\n\n\n");
wi=0;
while(1)
{
    if (strcmp(clvr[wi], " ")==0)
        break;
    printf("%s ",clvr[wi]);
    wi++;
}

car1='L';
car2='S';
consletravogaisletra2(car1, car2);
for (l=0;l<130;l++)
    strcpy(clvs[l] , cletrav2[l]);

printf("\n\n\n");
wi=0;
while(1)
{
    if (strcmp(clvs[wi], " ")==0)
        break;
    printf("%s ",clvs[wi]);
    wi++;
}

```

```

car1='L';
car2='U';
consletravogaisletra2(car1, car2);
for (l=0;l<130;l++)
    strcpy(clvu[l] , cletrav2[l]);

```

```

printf("\n\n\n");
wi=0;
while(1)
{
    if (strcmp(clvu[wi], " ")==0)
        break;
    printf("%s ",clvu[wi]);
    wi++;
}

```

```

car1='L';
car2='Z';
consletravogaisletra2(car1, car2);
for (l=0;l<130;l++)
    strcpy(clvz[l] , cletrav2[l]);

```

```

printf("\n\n\n");
wi=0;
while(1)
{
    if (strcmp(clvz[wi], " ")==0)
        break;
    printf("%s ",clvz[wi]);
    wi++;
}

```

```

car='L';
nhvogaisletra(car);
for (l=0;l<99;l++)
    strcpy(nhfl[l] , si3fnh1[l]);
printf("\n\n\n");
wi=0;
while(1)
{
    if (strcmp(nhfl[wi], " ")==0)
        break;
    printf("%s ",nhfl[wi]);
    wi++;
}

```

```

car='M';
nhvogaisletra(car);
for (l=0;l<99;l++)
    strcpy(nhfm[l] , si3fnh1[l]);
printf("\n\n\n");
wi=0;
while(1)
{
    if (strcmp(nhfm[wi], " ")==0)
        break;
    printf("%s ",nhfm[wi]);
    wi++;
}

```

```

car='N';
nhvogaisletra(car);
for (l=0;l<99;l++)
    strcpy(nhfn[l] , si3fnh1[l]);
printf("\n\n\n");
wi=0;
while(1)
{
    if (strcmp(nhfn[wi], " ")==0)
        break;
    printf("%s ",nhfn[wi]);
    wi++;
}

```

```

car='R';
nhvogaisletra(car);
for (l=0;l<99;l++)
    strcpy(nhfr[l] , si3fnh1[l]);
printf("\n\n\n");
wi=0;
while(1)
{
    if (strcmp(nhfr[wi], " ")==0)
        break;
    printf("%s ",nhfr[wi]);
    wi++;
}

```

```

car='S';
nhvogaisletra(car);
for (l=0;l<99;l++)
    strcpy(nhfs[l] , si3fnh1[l]);
printf("\n\n\n");

```

```

wi=0;
while(1)
{
    if (strcmp(nhfs[wi], " ")==0)
        break;
    printf("%s ",nhfs[wi]);
    wi++;
}

```

```

car='U';
nhvogaisletra(car);
for (l=0;l<99;l++)
    strcpy(nhfu[l] , si3fnh1[l]);
printf("\n\n\n");
wi=0;
while(1)
{
    if (strcmp(nhfu[wi], " ")==0)
        break;
    printf("%s ",nhfu[wi]);
    wi++;
}

```

```

car='M';
lhvogaisletra(car);
for (l=0;l<99;l++)
    strcpy(lhfm[l] , si3flh1[l]);
printf("\n\n\n");
wi=0;
while(1)
{
    if (strcmp(lhfm[wi], " ")==0)
        break;
    printf("%s ",lhfm[wi]);
    wi++;
}

```

```

car='N';
lhvogaisletra(car);
for (l=0;l<99;l++)
    strcpy(lhfn[l] , si3flh1[l]);
printf("\n\n\n");
wi=0;
while(1)
{
    if (strcmp(lhfn[wi], " ")==0)
        break;
}

```

```

    printf("%s ",lhfn[wi]);
    wi++;
}

car='R';
lhvogaisletra(car);
for (l=0;l<99;l++)
    strcpy(lhfr[l] , si3flh1[l]);
printf("\n\n\n");
wi=0;
while(1)
{
    if (strcmp(lhfr[wi]," ")==0)
        break;
    printf("%s ",lhfr[wi]);
    wi++;
}

car='S';
lhvogaisletra(car);
for (l=0;l<99;l++)
    strcpy(lhfs[l] , si3flh1[l]);
printf("\n\n\n");
wi=0;
while(1)
{
    if (strcmp(lhfs[wi]," ")==0)
        break;
    printf("%s ",lhfs[wi]);
    wi++;
}

car='L';
guvogaisletra(car);
for (l=0;l<99;l++)
    strcpy(gufl[l] , si3fgu1[l]);
printf("\n\n\n");
wi=0;
while(1)
{
    if (strcmp(gufl[wi]," ")==0)
        break;
    printf("%s ",gufl[wi]);
    wi++;
}

car='M';

```

```

guvogaisletra(car);
for (l=0;l<99;l++)
    strcpy(gufm[l] , si3fgu1[l]);
printf("\n\n\n");
wi=0;
while(1)
{
    if (strcmp(gufm[wi]," ")==0)
        break;
    printf("%s ",gufm[wi]);
    wi++;
}

```

```

car='N';
guvogaisletra(car);
for (l=0;l<99;l++)
    strcpy(gufn[l] , si3fgu1[l]);
printf("\n\n\n");
wi=0;
while(1)
{
    if (strcmp(gufn[wi]," ")==0)
        break;
    printf("%s ",gufn[wi]);
    wi++;
}

```

```

car='R';
guvogaisletra(car);
for (l=0;l<99;l++)
    strcpy(gufr[l] , si3fgu1[l]);
printf("\n\n\n");
wi=0;
while(1)
{
    if (strcmp(gufr[wi]," ")==0)
        break;
    printf("%s ",gufr[wi]);
    wi++;
}

```

```

car='S';
guvogaisletra(car);
for (l=0;l<99;l++)
    strcpy(gufs[l] , si3fgu1[l]);
printf("\n\n\n");
wi=0;

```

```

while(1)
{
    if (strcmp(gufs[wi], " ")==0)
        break;
    printf("%s ",gufs[wi]);
    wi++;
}

```

```

car='L';
quvogaisletra(car);
for (l=0;l<99;l++)
    strcpy(qufl[l] , si3fqu1[l]);
printf("\n\n\n");
wi=0;
while(1)
{
    if (strcmp(qufl[wi], " ")==0)
        break;
    printf("%s ",qufl[wi]);
    wi++;
}

```

```

car='M';
quvogaisletra(car);
for (l=0;l<99;l++)
    strcpy(qufm[l] , si3fqu1[l]);
printf("\n\n\n");
wi=0;
while(1)
{
    if (strcmp(qufm[wi], " ")==0)
        break;
    printf("%s ",qufm[wi]);
    wi++;
}

```

```

car='N';
quvogaisletra(car);
for (l=0;l<99;l++)
    strcpy(qufn[l] , si3fqu1[l]);
printf("\n\n\n");
wi=0;
while(1)
{
    if (strcmp(qufn[wi], " ")==0)
        break;
    printf("%s ",qufn[wi]);
}

```



```

        wi++;
    }

    car='R';
    quvogaisletra(car);
    for (l=0;l<99;l++)
        strcpy(qufr[l] , si3fqu1[l]);
    printf("\n\n\n");
    wi=0;
    while(1)
    {
        if (strcmp(qufr[wi], " ")==0)
            break;
        printf("%s ",qufr[wi]);
        wi++;
    }

    car='S';
    quvogaisletra(car);
    for (l=0;l<99;l++)
        strcpy(qufs[l] , si3fqu1[l]);
    printf("\n\n\n");
    wi=0;
    while(1)
    {
        if (strcmp(qufs[wi], " ")==0)
            break;
        printf("%s ",qufs[wi]);
        wi++;
    }

    car='S';
    consvogaisaoletra(car);
    for (l=0;l<99;l++)
        strcpy(cofs[l] , si3fco1[l]);
    printf("\n\n\n");
    wi=0;
    while(1)
    {
        if (strcmp(cofs[wi], " ")==0)
            break;
        printf("%s ",cofs[wi]);
        wi++;
    }

    car='S';
    consvogaisaeletra(car);

```

```
for (l=0;l<99;l++)
    strcpy(cafs[l] , si3fca1[l]);
printf("\n\n\n");
wi=0;
while(1)
{
    if (strcmp(cafs[wi], " ")==0)
        break;
    printf("%s ",cafs[wi]);
    wi++;
}

getch();

    return 0;
}
//-----
```

ANEXO 2 – Apresenta os resultados obtidos pela execução do programa apresentado no ANEXO 1.

A figura A.2.1. apresenta combinações de letras gerando sílabas de 2 e 3 letras.

```

C:\Arquivos de programas\Borland\CBUILDER6\Projects\Project2.exe
APRESENTACAO DAS COMBINACOES DAS SILABAS DE DUAS LETRAS
AE AI AO AU BA BE BI BO BU CA CE CI CO CU DA DE DI DO DU EA
EI EO EU FA FE FI FO FU GA GE GI GO GU HA HE HI HO HU IA IE
IO IU JA JE JI JO JU KA KE KI KO KU LA LE LI LO LU MA ME MI
MO MU NA NE NI NO NU OA OE OI OU PA PE PI PO PU QA QE QI QO
QU RA RE RI RO RU SA SE SI SO SU TA TE TI TO TU UA UE UI UO
ZA ZE ZI ZO ZU

BA BO BU
BA BE BI BO BU DA DE DI DO DU HA HE HI HO HU LA LE LI LO LU
MA ME MI MO MU NA NE NI NO NU RA RE RI RO RU SA SE SI SO SU
XA XE XI XO XU

APRESENTACAO DAS COMBINACOES DAS SILABAS DE TRES LETRAS
SSA SSE SSI SSO SSU
RRA RRE RRI RRO RRU
BAL BEL BIL BOL BUL CAL CEL CIL COL CUL DAL DEL DIL DOL DUL FAL
FEL FIL FOL FUL GAL GEL GIL GOL GUL HAL HEL HIL HOL HUL JAL JEL
JIL JOL JUL KAL KEL KIL KOL KUL LAL LEL LIL LOL LUL MAL MEL MIL
MOL MUL NAL NEL NIL NOL NUL PAL PEL PIL POL PUL QAL QEL QIL QOL
QUL RAL REL RIL ROL RUL SAL SEL SIL SOL SUL TAL TEL TIL TOL TUL
UAL UEL UIL UOL UUL WAL WEL WIL WOL WUL XAL XEL XIL XOL XUL YAL
YEL YIL YOL YUL ZAL ZEL ZIL ZOL ZUL
BAM BEM BIM BOM BUM CAM CEM CIM COM CUM DAM DEM DIM DOM DUM FAM
FEM FIM FOM FUM GAM GEM GIM GOM GUM HAM HEM HIM HOM HUM JAM JEM
JIM JOM JUM KAM KEM KIM KOM KUM LAM LEM LIM LOM LUM MAM MEM MIM
MOM MUM NAM NEM NIM NOM NUM PAM PEM PIM POM PUM QAM QEM QIM QOM
QUM RAM REM RIM ROM RUM SAM SEM SIM SOM SUM TAM TEM TIM TOM TUM
UAM UEM UIM UOM UUM WAM WEM WIM WOM WUM XAM XEM XIM XOM XUM YAM
YEM YIM YOM YUM ZAM ZEM ZIM ZOM ZUM
BAN BEN BIN BON BUN CAN CEN CIN CON CUN DAN DEN DIN DON DUN FAN
FEN FIN FON FUN GAN GEN GIN GON GUN HAN HEN HIN HON HUN JAN JEN
JIN JON JUN KAN KEN KIN KON KUN LAN LEN LIN LON LUN MAN MEN MIN
MON MUN NAN NEN NIN NON NUN PAN PEN PIN PON PUN QAN QEN QIN QON
QUN RAN REN RIN RON RUN SAN SEN SIN SON SUN TAN TEN TIN TON TUN
UAN UEN UIN UON UUN WAN WEN WIN WON WUN XAN XEN XIN XON XUN YAN
YEN YIN YON YUN ZAN ZEN ZIN ZON ZUN
BAR BER BIR BOR BUR CAR CER CIR COR CUR DAR DER DIR DOR DUR FAR
FER FIR FOR FUR GAR GER GIR GOR GUR HAR HER HIR HOR HUR JAR JER
JIR JOR JUR KAR KER KIR KOR KUR LAR LER LIR LOR LUR MAR MER MIR
MOR MUR NAR NER NIR NOR NUR PAR PER PIR POR PUR QAR QER QIR QOR

```

Figura A.2.1.-Apresenta sílabas de duas letras e de três letras.

A figura A.2.2. apresenta combinações de letras gerando sílabas de 3 letras.

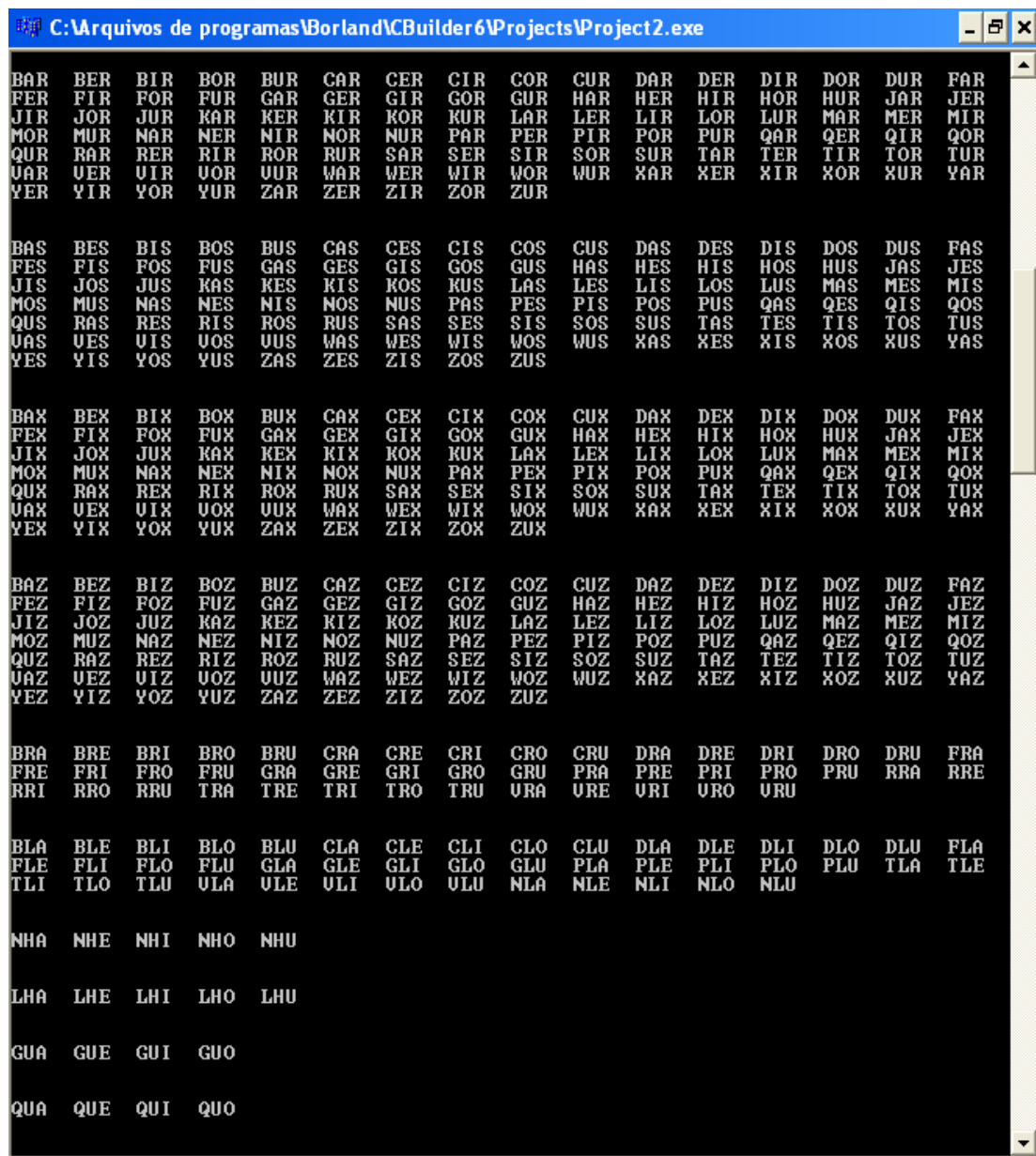


Figura A.2.2.-Apresenta sílabas de três letras.

A figura A.2.3. apresenta combinações de letras gerando sílabas de quatro letras.

```

C:\Arquivos de programas\Borland\CBUILDER6\Projects\Project2.exe
BA BO CA CO DA DO FA FO GA GO HA HO JA JO KA KO LA LO MA MO
NA NO PA PO QA QO RA RO SA SO TA TO UA UO WA WO XA XO YA YO
ZA ZO
APRESENTAÇÃO DAS COMBINAÇÕES DAS SÍLABAS DE QUATRO LETRAS
BRAL BREL BRIL BROL BRUL CRAL CREL CRIL CROL CRUL DRAL DREL DRIL DR
OL DRUL FRAL FREL FRIL FROL FRUL GRAL GREL GRIL GROL GRUL PRAL PREL
PRIL PROL PRUL RRAL RREL RRIL RROL RRUL TRAL TREL TRIL TROL TRUL
URAL UREL URIL UROL URUL
BRAM BREM BRIM BROM BRUM CRAM CREM CRIM CROM CRUM DRAM DREM DRIM DR
OM DRUM FRAM FREM FRIM FROM FRUM GRAM GREM GRIM GROM GRUM PRAM PREM
PRIM PROM PRUM RRAM RREM RRIM RROM RRUM TRAM TREM TRIM TROM TRUM
URAM UREM URIM UROM URUM
BRAN BREN BRIN BRON BRUN CRAN CREN CRIN CRON CRUN DRAN DREN DRIN DR
ON DRUN FRAN FREN FRIN FROM FRUN GRAN GREN GRIN GROM GRUN PRAN PREN
PRIN PRON PRUN RRAN RREN RRIN RRON RRUN TRAN TREN TRIN TRON TRUN
URAN UREN URIN URON URUN
BRAR BRER BRIR BROR BRUR CRAR CRRER CRIR CROR CRUR DRAR DRER DRIR DR
OR DRUR FRAR FRER FRIR FROR FRUR GRAR GRER GRIR GROR GRUR PRAR PRER
PRIR PROR PRUR RRAR RRRER RRIR RROR RRUR TRAR TRER TRIR TROR TRUR
URAR URER URIR UROUR URUR
BRAS BRES BRIS BROS BRUS CRAS CRES CRIS CROS CRUS DRAS DRES DRIS DR
OS DRUS FRAS FRES FRIS FROS FRUS GRAS GRES GRIS GROS GRUS PRAS PRES
PRIS PROS PRUS RRAS RRES RRIS RROS RRUS TRAS TRES TRIS TROS TRUS
URAS URES URIS UROS URUS
BRAU BREU BRIU BROU CRAU CREU CRIU CROU DRAU DREU DRIU DROU FRAU FR
EU FRIU FROU GRAU GREU GRIU GROU PRAU PREU PRIU PROU RRAU RREU RRIU
RROU TRAU TREU TRIU TROU URAU UREU URIU UROU TRES TRIS TROS TRUS
URAS URES URIS UROS URUS
BRAZ BREZ BRIZ BROZ BRUZ CRAZ CREZ CRIZ CROZ CRUZ DRAZ DREZ DRIZ DR
OZ DRUZ FRAZ FREZ FRIZ FROZ FRUZ GRAZ GREZ GRIZ GROZ GRUZ PRAZ PREZ
PRIZ PROZ PRUZ RRAZ RREZ RRIZ RROZ RRUZ TRAZ TREZ TRIZ TROZ TRUZ
URAZ UREZ URIZ UROZ URUZ
BLAM BLEM BLIM BLOM BLUM CLAM CLEM CLIM CLOM CLUM DLAM DLEM DLIM DL
OM DLUM FLAM FLEM FLIM FLOM FLUM GLAM GLEM GLIM GLOM GLUM PLAM PLEM
PLIM PLOM PLUM TLAM TLEM TLIM TLOM TLUM ULAM ULEM ULIM ULOM ULUM
BLAR BLER BLIR BLOR BLUR CLAR CLER CLIR CLOR CLUR DLAR DLER DLIR DL
OR DLUR FLAR FLER FLIR FLOR FLUR GLAR GLER GLIR GLOR GLUR PLAR PLER
PLIR PLOR PLUR TLAR TLER TLIR TLOR TLUR ULAR ULER ULIR ULOR ULUR

```

Figura A.2.3.-Apresenta sílabas de quatro letras

A figura A.2.4. continua apresentando combinações de letras gerando sílabas de 4 letras.

```

C:\Arquivos de programas\Borland\CBUILDER6\Projects\Project2.exe
BLAS BLES BLIS BLOS BLUS CLAS CLES CLIS CLOS CLUS DLAS DLES DLIS DL
OS DLUS FLAS FLES FLIS FLOS FLUS GLAS GLES GLIS GLOS GLUS PLAS PLES
  PLIS PLOS PLUS TLAS TLES TLIS TLOS TLUS ULAS ULES ULIS ULOS ULUS

BLAU BLEU BLIU BLOU CLAU CLEU CLIU CLOU DLAU DLEU DLIU DLOU FLAU FL
EU FLIU FLOU GLAU GLEU GLIU GLOU PLAU PLEU PLIU PLOU TLAU TLEU TLIU
  TLOU ULAU ULEU ULIU ULOU TLIS TLOS TLUS ULAS ULES ULIS ULOS ULUS

BLAZ BLEZ BLIZ BLOZ BLUZ CLAZ CLEZ CLIZ CLOZ CLUZ DLAZ DLEZ DLIZ DL
OZ DLUZ FLAZ FLEZ FLIZ FLOZ FLUZ GLAZ GLEZ GLIZ GLOZ GLUZ PLAZ PLEZ
  PLIZ PLOZ PLUZ TLAZ TLEZ TLIZ TLOZ TLUZ ULAZ ULEZ ULIZ ULOZ ULUZ

NHAL NHEL NHIL NHOL NHUL

NHAM NHEM NHIM NHOM NHUM

NHAN NHEN NHIN NHON NHUN

NHAR NHER NHIR NHOR NHUR

NHAS NHES NHIS NHOS NHUS

NHAU NHEU NHIU NHOU NHUS

LHAM LHEM LHIM LHOM LHUM

LHAN LHEN LHIN LHON LHUN

LHAR LHER LHIR LHOR LHUR

LHAS LHES LHIS LHOS LHUS

GUAL GUEL GUIL GUOL

GUAM GUEM GUIM GUOM

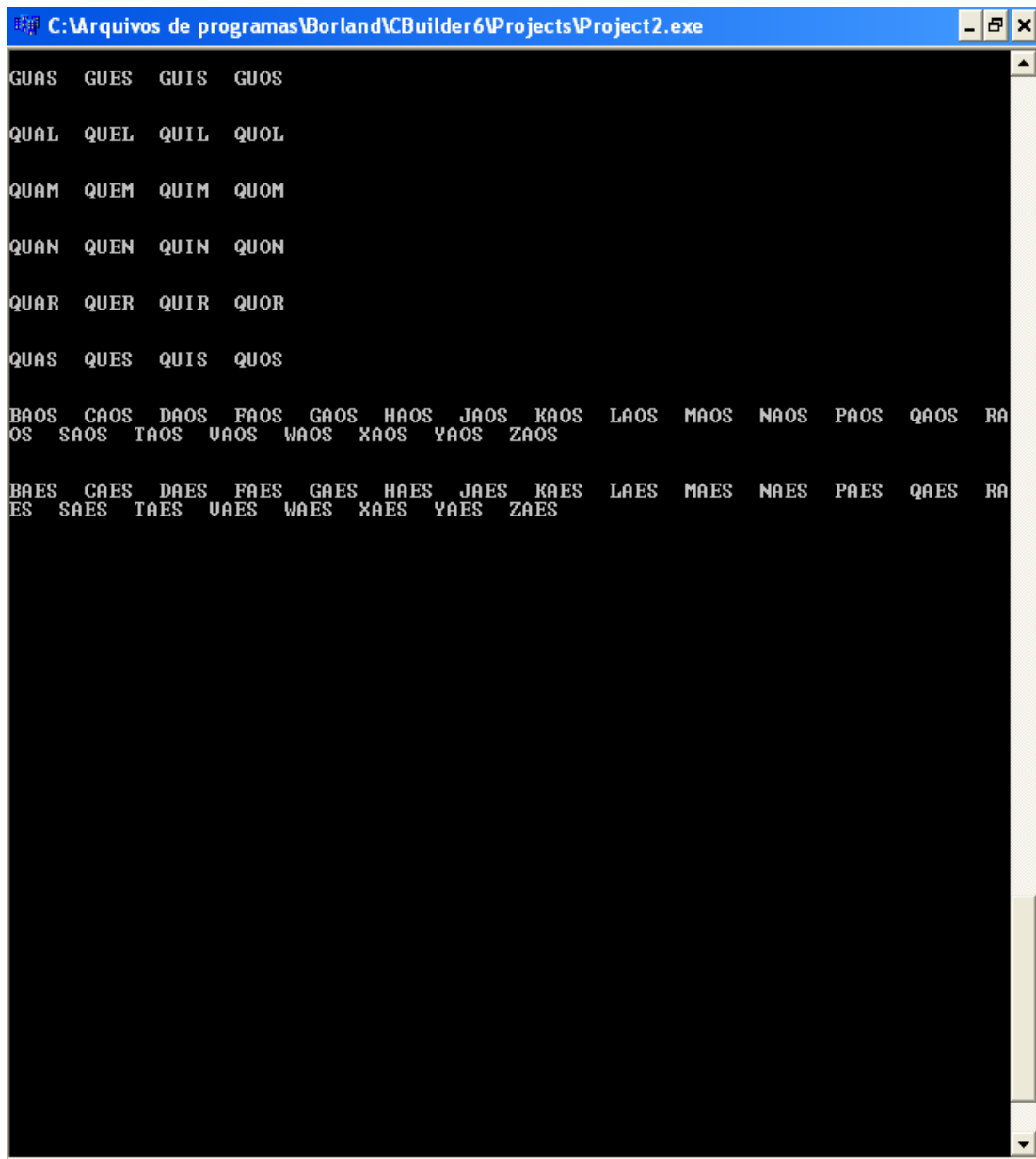
GUAN GUEN GUIN GUON

GUAR GUER GUIR GUOR

```

Figura A.2.4. -Continua apresentando sílabas de quatro letras

A figura A.2.5. continua apresentando combinações de letras gerando sílabas de 4 letras.



```
C:\Arquivos de programas\Borland\CBuilder6\Projects\Project2.exe
GUAS  GUES  GUIS  GUOS
QUAL  QUEL  QUIL  QUOL
QUAM  QUEM  QUIM  QUOM
QUAN  QUEN  QUIN  QUON
QUAR  QUER  QUIR  QUOR
QUAS  QUES  QUIS  QUOS
BAOS  CAOS  DAOS  FAOS  GAOS  HAOS  JAOS  KAOS  LAOS  MAOS  NAOS  PAOS  QAOS  RA
OS  SAOS  TAOS  UAOS  WAOS  XAOS  YAOS  ZAOS
BAES  CAES  DAES  FAES  GAES  HAES  JAES  KAES  LAES  MAES  NAES  PAES  QAES  RA
ES  SAES  TAES  UAES  WAES  XAES  YAES  ZAES
```

Figura A.2.5.-Continua apresentando sílabas de quatro letras

A figura A.2.6. apresenta combinações de letras gerando sílabas de 5 letras elaborada manualmente pela facilidade do processo.

brans, brens, brins, brons, bruns, crans, crens, crins, crons, cruns, drans, drens, drins, drons, druns, frans, frens, frins, frons, fruns, grans, grens, grins, grons, gruns, prans, prens, prins, prons, pruns, rrans, rrens, rrins, rrons, rruns, trans, trens, trins, trons, truns, vrans, vrens, vrins, vrons, vruns, blans, blens, blins, blons, bluns, clans, clens, clins, clons, cluns, dlans, dlens, dlins, dlons, dluns, flans, flens, flins, flons, fluns, glans, glens, glins, glons, gluns, plans, plans, prins, plons, pluns, tlans, tlens, tlins, tlons, tluns, vlans, vlens, vlins, vlons, vluns.

Figura A.2.6.-Apresenta silabas de cinco letras

A figura A.2.7. apresenta as combinações de letras elaboradas manualmente pela facilidade do processo e devidamente contabilizadas nas totalizações apresentadas neste trabalho.

ça, ço, çu, blan, blen, blin, blon, blun, clan, clen, clin, clon, clun, dlan, dlen, dlin, dlon, dlun, flan, flen, flin, flon, flun, glan, glen, glin, glon, glun, plan, plen, plin, plon, plun, tlan, tlen, tlin, tlon, tlun, vlan, vlen, vlin, vlon, vlun, a, e, i, o, u

Figura A.2 1-Apresenta combinação de letras elaboradas manualmente

Autorizo cópia total ou parcial desta obra, apenas para fins de estudo e pesquisa, sendo expressamente vedado qualquer tipo de reprodução para fins comerciais sem prévia autorização específica do autor.

Gilza Paim Mandelman

Taubaté, setembro de 2011