

FUNDAÇÃO DE ENSINO "EURÍPIDES SOARES DA ROCHA"
CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA - UNIVEM
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

GUSTAVO LUIZ DA COSTA ROSA

**ESTUDO DE FERRAMENTAS DE RECONHECIMENTO E COMANDOS POR
VOZ PARA PORTUGUÊS DO BRASIL**

MARÍLIA

2015

GUSTAVO LUIZ DA COSTA ROSA

ESTUDO DE FERRAMENTAS DE RECONHECIMENTO E COMANDOS POR
VOZ PARA PORTUGUÊS DO BRASIL

Trabalho de Conclusão do Curso
apresentado ao Curso de Bacharelado em
Ciência da Computação da Fundação de
Ensino "Eurípides Soares da Rocha"
mantenedora do Centro Universitário
Eurípedes de Marília - UNIVEM como
requisito parcial para obtenção do grau
de Bacharel em Ciência da Computação.

Orientador

Prof. Me. Ildeberto de Genova Bugatti

Coorientadores

Prof. Me. Edio Roberto Manfio

Prof. Dr. Pedro Henrique Bugatti

MARÍLIA

2015

ROSA, Gustavo Luiz da Costa

Estudo de ferramentas de reconhecimento e comandos por voz para português do Brasil / Gustavo Luiz da Costa Rosa; orientador: Prof. Me. Ildeberto de Genova Bugatti. Marília, SP: [s.n.], 2015.

41 folhas

Monografia (Bacharelado em Ciência da Computação): Centro Universitário Eurípides de Marília.

Dedico esse trabalho a todas as pessoas que se mantiveram próximas e auxiliaram de alguma maneira para a construção do projeto, em especial, meus pais e minha namorada Juliani.

AGRADECIMENTO

Agradeço primeiramente à Deus, por me dar forças para me mostrar o caminho das pedras que foi a graduação. Agradeço de todo meu coração aos meus pais: Maria Helena e Joaquim José, por sempre acreditar na minha formação de bacharelado e nos meus planos.

Agradeço ao meu irmão Artur, mesmo distante, me trouxe força de vontade e sua experiência me tranquilizou nesse processo de monografia.

Agradeço especialmente minha namorada Juliani, por estar presente nos quatro anos de faculdade, me trazendo uma alegria imensa ao se formar comigo.

Agradeço também aos meus orientadores Prof. Me. Ildeberto, seu filho o Prof. Dr. Pedro Henrique Bugatti e o Prof. Me. Edio Roberto Manfio por sempre me ajudarem em momentos tão marcantes como foi esse projeto, e pela experiência por ter feito esse projeto em parceria.

“A imaginação é mais importante que a ciência, porque a ciência é limitada, ao passo que a imaginação abrange o mundo inteiro.”

Albert Einstein

RESUMO

A necessidade de organizar as atividades diárias através da otimização do tempo disponível para efetuá-las, gera a necessidade de utilizar ferramentas automatizadas para auxiliar a execução de tarefas repetitivas. Esse fato motiva o desenvolvimento de equipamentos automatizados que auxiliem a execução das funções de forma mais eficiente e organizada. Para esse fim existe um grande gama de processos automatizados aplicados às mais diversas áreas do conhecimento. No entanto, a utilização efetiva dessa automação está diretamente relacionada à qualidade da interface disponibilizada para o usuário. Para gerar um contato mais amigável o projeto propõe a introdução comandos de voz nos processos de controle de equipamentos automatizados como por exemplo um veículo movimentando pelo eixo da horizontal e vertical.

Neste contexto, o projeto propõe o estudo de viabilidade para o reconhecimento de voz utilizando como referência informações contidas no Atlas Linguístico do Brasil (ALiB), que possibilitará gerar um banco de dados contendo conjuntos de comandos de voz, contemplando as características fonéticas regionais da língua portuguesa falada no Brasil e aplicar esses comandos em sistemas de controle de processos.

O banco de dados gerado será disponibilizado a todos os interessados da área que possam utilizá-los em uma diversidade de aplicações ou funcionalidades.

Palavras-reservadas: Reconhecimento de voz; Comandos de voz; Controle de processos; Microcontroladores; ALiB.

ABSTRACT

The need to organize daily activities by optimizing the time available to modify them, makes it necessary to use automated tools to assist the execution of repetitive tasks. This fact motivates the development of automated equipment to assist carrying out tasks in a more efficient and organized manner. For this purpose there is a wide range of automated processes applied to various areas of knowledge. However, the effective use of this automation is directly related to the quality of available user interface. To generate a more friendly contact design proposes to introduce voice commands the automated process control equipment such as a vehicle moving along the axis of horizontal and vertical.

In this context, the project proposes the feasibility study for speech recognition using as reference information in the Linguistic Atlas of Brazil (ALiB), which will enable to generate a database containing sets of voice commands, contemplating regional phonetic characteristics of the language Portuguese spoken in Brazil and apply these commands in process control systems.

The generated database will be made available to all stakeholders in the area that might use them in a variety of applications or features.

Reserved words: Speech Recognition; Voice commands; Process control; Microcontrollers; ALiB.

LISTA DE ILUSTRAÇÕES

FIGURA 1. ETAPAS DO PROJETO	16
FIGURA 2. VETOR "TRÁS" PRONUNCIADO POR UMA MULHER ANTES DO TRATAMENTO	19
FIGURA 3. EXEMPLO DE MODELO ARFF	22
FIGURA 4. RANKING LINGUAGEM DE PROGRAMAÇÃO DE 2015 EM RELAÇÃO AO RANKING DE 2014. [7].....	37
FIGURA 5. FLUXOGRAMA DA IMPLEMENTAÇÃO	38
FIGURA 6. COMANDO PARA A EXECUÇÃO DO ALGORITMO.....	38
FIGURA 7. RETORNO API DO WEKA.....	39
FIGURA 8. RESULTADOS COM VALIDAÇÃO PELO APLICATIVO DO WEKA	39

LISTA DE TABELAS

TABELA 1: DISTRIBUIÇÃO DE VARIÂNCIAS - FONTE: CARDOSO 2014[5]	17
TABELA 2. COMPARAÇÃO DO VETOR DIREITA	21
TABELA 3. DISTRIBUIÇÃO DOS ALGORITMOS EM CLASSES.....	23
TABELA 4. IDENTIFICAÇÃO DO SEXO - ALGORITMO CEPSTRO	25
TABELA 5. MATRIZ DE CONFUSÃO - MELHOR ALGORITMO CEPSTRO NA IDENTIFICAÇÃO DO SEXO	26
TABELA 6. MATRIZ DE CONFUSÃO - PIOR ALGORITMO CEPSTRO NA IDENTIFICAÇÃO DO SEXO	26
TABELA 7. IDENTIFICAÇÃO DO SEXO - ALGORITMO ESPECTRO	27
TABELA 8. MATRIZ DE CONFUSÃO - MELHOR ALGORITMO ESPECTRO NA IDENTIFICAÇÃO DO SEXO	28
TABELA 9. MATRIZ DE CONFUSÃO - PIOR ALGORITMO ESPECTRO NA IDENTIFICAÇÃO DO SEXO.....	28
TABELA 10. IDENTIFICAÇÃO DA DIREÇÃO - ALGORITMO CEPSTRO	29
TABELA 11. MATRIZ DE CONFUSÃO - MELHOR ALGORITMO CEPSTRO NA IDENTIFICAÇÃO DA DIREÇÃO	30
TABELA 12. MATRIZ DE CONFUSÃO - PIOR ALGORITMO CEPSTRO NA IDENTIFICAÇÃO DA DIREÇÃO	30
TABELA 13. IDENTIFICAÇÃO DA DIREÇÃO - ALGORITMO ESPECTRO	31
TABELA 14. MATRIZ DE CONFUSÃO - MELHOR ALGORITMO ESPECTRO NA IDENTIFICAÇÃO DA DIREÇÃO.....	32
TABELA 15. MATRIZ DE CONFUSÃO - PIOR ALGORITMO ESPECTRO NA IDENTIFICAÇÃO DA DIREÇÃO.....	32
TABELA 16. IDENTIFICAÇÃO DA DIREÇÃO E DO SEXO - ALGORITMO CEPSTRO.....	33
TABELA 17. MATRIZ DE CONFUSÃO - MELHOR ALGORITMO CEPSTRO NA IDENTIFICAÇÃO DA DIREÇÃO E SEXO	34
TABELA 18. MATRIZ DE CONFUSÃO - PIOR ALGORITMO CEPSTRO NA IDENTIFICAÇÃO DA DIREÇÃO E SEXO	34
TABELA 19. IDENTIFICAÇÃO DO DIREÇÃO E SEXO - ALGORITMO ESPECTRO	35
TABELA 20. MATRIZ DE CONFUSÃO - MELHOR ALGORITMO ESPECTRO NA IDENTIFICAÇÃO DA DIREÇÃO E DO SEXO	36

LISTA DE ABREVIATURAS E SIGLAS

ALiB – Atlas Linguístico do Brasil

SUMÁRIO

1. INTRODUÇÃO	13
1.1. OBJETIVO	13
1.2. Metodologia	14
1.2.1. O que é Reconhecimento de Voz.....	15
1.2.2. O que é Reconhecimento de Voz para Português do Brasil?.....	15
1.2.3. O que é o ALiB?.....	15
1.2.4. Quais são os comandos pré-definidos?.....	15
2. FERRAMENTAS UTILIZADAS	15
2.1. Utilização do ALiB - Atlas Linguístico do Brasil	16
2.2. Extração de Áudio	18
2.3. Sound Forge	19
2.4. Audacity.....	20
2.4.1. Sample Data Export.....	20
2.4.2. Espectro da Frequência.....	20
2.5. Excel	21
2.6. Weka	21
3. RESULTADOS OBTIDOS NO SOFTWARE WEKA	24
3.1. Identificação do Sexo	24
3.1.1. Algoritmo Cepstro	25
3.1.2. Algoritmo Espectro.....	26
3.1.3. Comparação dos valores.....	28
3.2. Identificação da direção	28
3.2.1. Algoritmo Cepstro	29
3.2.2. Algoritmo Espectro.....	31
3.2.3. Comparação dos valores.....	32
3.3. A identificação do vetor e do sexo	32
3.3.1. Algoritmo Cepstro	33
3.3.2. Algoritmo Espectro.....	35
3.3.3. Comparação dos valores.....	37
3.4. ESTUDO E PROPOSTA DE IMPLEMENTAÇÃO	37
4. CONCLUSÕES.....	40
5. REFERÊNCIAS BIBLIOGRÁFICAS.....	41

1. INTRODUÇÃO

A tecnologia de semicondutores aplicada a sistemas computacionais em conjunto com ferramentas de software evolui de forma muito rápida. Atualmente é possível utilizar essa tecnologia em uma diversidade de aplicações. Dentre as áreas de pesquisa computacional a interface Homem-Máquina está revolucionando o acesso a sistemas computacionais tornando cada vez mais cômoda e acessível essa comunicação. O recurso de comando de voz possibilita a construção de técnicas de interface contendo alto grau de naturalidade e proximidade do homem.

Atualmente há várias tecnologias disponíveis no mercado para reconhecimento de voz aplicada a sistemas de controle de processos e automação. Entre estas aplicações, as mais comuns estão sendo utilizadas na indústria automobilística; possibilitando a execução de várias funções substituindo os sistemas de controles tradicionais por comandos de voz. Há também comandos de voz para controlar funções em aparelhos portáteis como computadores, smartphones, celulares dentre muitos outros.

No entanto, de forma geral, poucas dessas aplicações utilizam técnicas de reconhecimento de voz aplicada para Língua Portuguesa falada no Brasil, gerando problemas quando da utilização desses comandos de voz: veículos comercializados na região sul do país não devem levar em consideração apenas a variante linguística local para o reconhecimento e o mesmo deve ocorrer no Nordeste. Os sistemas de reconhecimento de voz utilizados pelas indústrias estão geralmente associados a sistemas fonéticos da língua inglesa gerando falhas e problemas quando aplicadas a outro idioma, entre eles o Português do Brasil. Isto se deve ao fato de que no Brasil existem variantes típicas de cada região.

1.1. OBJETIVO

Durante o projeto foram estudados e implementados algoritmos de reconhecimento de voz existentes, para verificar a eficiência dos mesmos e gerar resultados para subsidiar o desenvolvimento de ferramentas e técnicas de reconhecimento de voz aplicadas ao controle de processos e interfaces naturais. Os estudos iniciais abrangeram um conjunto mínimo de fonemas, definidos como necessários, para gerar o controle de uma diversidade de

equipamentos. Foram estudadas e utilizadas um conjunto de técnicas de reconhecimento de voz aplicadas ao conjunto mínimo de fonemas escolhidos. Os resultados obtidos com essas técnicas foram comparados e contribuíram para determinar as técnicas mais adequadas para realizar o reconhecimento de voz aplicada ao controle de processos.

O reconhecimento dos fonemas estudados além de contemplar variantes regionais da língua portuguesa falada no Brasil deve também considerar outros aspectos relevantes para o reconhecimento de voz, tais como: sexo, voz masculina, voz feminina, faixa etária (crianças, adolescentes, adultos e, idosos), entre outros.

A análise eficiente de todos esses fatores exige conhecimentos multidisciplinares envolvendo diversas áreas do conhecimento, tais como: Processamento de Sinais, Ciência da Computação, Reconhecimento de Padrões, Inteligência Artificial, Neurofisiologia, Teoria das Comunicações e Linguística. Como consequência demandam profissionais de várias áreas, tais como: Linguística, Mecatrônica e Computação.

Além dos parâmetros já listados, os sistemas de reconhecimento da voz, em geral, devem ser aptos a funcionar em condições desfavoráveis, que envolvem a existência de ruídos nos ambientes onde os sistemas de reconhecimento de voz podem ser utilizados. Esse fato, exige o estudo de técnicas extras para conseguir eficiência e robustez do sistema. Essa multidisciplinaridade motivou o desenvolvimento e a implementação do projeto que caracteriza um grupo de pesquisas apto a ser formalizado junto à instituição ou órgãos de fomento e pesquisa.

A fase de estudos e definição das ferramentas mais adequadas para o desenvolvimento efetivo do projeto foi realizada a contento. As ferramentas utilizadas estão descritas em conjunto com suas funcionalidades. Foi gerado um conjunto de fonemas que foram capturados em laboratório gerando um conjunto de amostras satisfatório para aplicar as ferramentas estudadas gerando resultados de eficiência e robustez que foram compilados e comprados. Gerando subsídios para trabalhos futuros.

1.2. Metodologia

O nosso projeto, está separados em quatro grandes questionamentos, que são: O que é reconhecimento de Voz? O que é reconhecimento de Voz para o português do Brasil? O que é o ALiB? Quais são os comandos pré-definidos?

Para as respostas desses questionamentos esse capítulo foi separado em tópicos.

1.2.1. O que é Reconhecimento de Voz

O reconhecimento de voz veio para criar uma interface homogênea e didática na relação homem/máquina. Tornando possível essa ligação simples usando apenas a voz para realizar suas tarefas.

Com o conceito de *hands free* (mãos livres), é possível realizar 2 coisas ao mesmo tempo, como por exemplo, em uma fábrica, um funcionário pode dar um comando uma máquina apenas com sua voz, enquanto pode carregar uma caixa em suas mãos.

1.2.2. O que é Reconhecimento de Voz para Português do Brasil?

O reconhecimento de voz para o Português do Brasil esta atrasada em relação a outros idiomas como, por exemplo, ao inglês esse atraso deve a dois pontos: baixo estímulo financeiro de empresas pesquisadoras e sua ampla diversidade de dialetos presentes no país devido sua ampla extensão territorial.

1.2.3. O que é o ALiB?

O ALiB é um Atlas Linguístico que contém dados sobre quase todos os dialetos do território brasileiro. Foi desenvolvido em parceria com várias Universidades do país, entre elas a Universidade Estadual de Londrina (UEL). Com mais de 30 anos de existência, sua edição oficial aconteceu em outubro do ano passado. Mais posteriormente, falarei mais sobre o Atlas.

1.2.4. Quais são os comandos pré-definidos?

Para tornar um projeto com mais ênfase e um melhor desenvolvimento, foram escolhidos cinco comandos pré-definidos que são: “frente”, “trás”, “direita”, “esquerda” e “para”. Com esses comandos, é possível a locomoção de qualquer veículo em um eixo da horizontal e na vertical por exemplo.

2. FERRAMENTAS UTILIZADAS

Para o desenvolvimento desse projeto, foi utilizado os conceitos do ALiB, tecnologias de extração e edição de áudio junto de ferramentas para o reconhecimento de imagens. Nessa fase além do ALiB, foram utilizadas as seguintes ferramentas: Sound Forge, Audacity, Excel e Weka.

A figura 1 mostra o fluxograma do desenvolvimento do projeto e cada tópico será discutido nos demais capítulos.

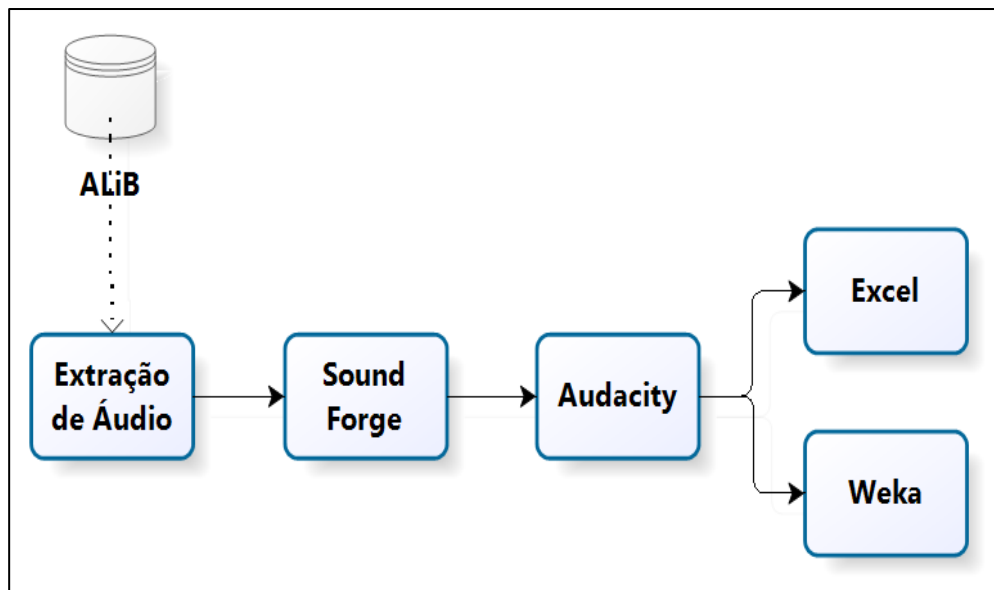


Figura 1. Etapas do projeto

2.1. Utilização do ALiB - Atlas Linguístico do Brasil

Para iniciar os estudos, foi necessário contar com o conceito do ALiB, que oferece um estudo bastante completo de variantes da Língua Portuguesa do Brasil. Paralelamente a isso, foram selecionados cinco comandos que representam os vetores de direção/posicionamento ‘direita’, ‘esquerda’, ‘frente’, ‘trás’ e ‘para’. A razão da escolha por esses comandos/vetores – que chamaremos apenas de vetores a partir de agora – é simples: com eles é possível movimentar qualquer veículo sobre rodas, para ficar apenas em um exemplo.

O atlas é indispensável para o desenvolvimento desse projeto, pois ele contém uma quantidade enorme de informações sobre nossa língua, incluindo fonemas de todas as regiões do Brasil. Os seis objetivos do Atlas Linguístico do Brasil (ALiB) [6] seguem descritos:

1. *Descrever a realidade linguística do Brasil, no que tange à língua portuguesa, com enfoque prioritário na identificação das diferenças diatópicas (fônicas, morfossintáticas e léxico-semânticas) consideradas na perspectiva da Geolinguística;*
2. *Oferecer aos estudiosos da língua portuguesa (linguistas, lexicólogos, etimólogos, filólogos, etc.), aos pesquisadores de áreas afins (história, antropologia, sociologia, etc.) e aos pedagogos (gramáticos, autores de livros-texto, professores) subsídios para o aprimoramento do ensino/aprendizagem e para uma melhor interpretação do caráter multidialetal do Brasil;*

3. *Estabelecer isoglossas com vistas a traçar a divisão dialetal do Brasil, tornando evidentes as diferenças regionais através de resultados cartografados em mapas linguísticos e realizar estudos interpretativos de fenômenos considerados;*
4. *Examinar os dados coletados na perspectiva de sua interface com outros ramos do conhecimento - história, sociologia, antropologia, etc. - de modo a poder contribuir para fundamentar e definir posições teóricas sobre a natureza da implantação e desenvolvimento da língua portuguesa no Brasil;*
5. *Oferecer aos interessados nos estudos linguísticos um considerável volume de dados que permita aos lexicógrafos aprimorarem os dicionários, ampliando o campo de informações; aos gramáticos atualizarem as informações com base na realidade documentada pela pesquisa empírica; aos autores de livros didáticos adequarem a sua produção à realidade cultural de cada região; aos professores aprofundar o conhecimento da realidade linguística, refletindo sobre as variantes de que se reveste a língua portuguesa no Brasil e, conseqüentemente, encontrando meios de, sem desprestigiar os seus dialetos de origem, levar os estudantes ao domínio de uma variante tida como culta;*
6. *Contribuir para o entendimento da língua portuguesa no Brasil como instrumento social de comunicação diversificado, possuidor de várias normas de uso, mas dotado de uma unidade sistêmica.*

A Tabela 1, definida por *Cardoso 2014[4]*, ilustra bem a distribuição das diferentes variantes da língua Portuguesa falada no Brasil.

Região	Direita	Esquerda	Trás	Frente	Para
Centro Oeste de São Paulo e Norte do Paraná	direita 'a'	esquerda 'a'	trás 'a'	frente 'a'	para 'a'
	direita 'c'				
Cidade de São Paulo		esquerda 'c'			
Santa Catarina e Rio Grande do Sul	direita 'b'	esquerda 'b'	trás 'b'	frente 'b'	para 'b'
Cidade do Rio de Janeiro e parte do Estado do Rio de Janeiro		esquerda 'd'	trás 'c'		
		esquerda 'e'			

Tabela 1: Distribuição de variâncias - Fonte: Cardoso 2014[5]

Nota-se, que na região Centro-Oeste do estado de São Paulo, possui duas variações para a palavra *Direita*, e uma única variação para os demais comandos (*Esquerda*, *Trás*, *Frente* e *Para*).

Já para a cidade de São Paulo, possui uma variação apenas na palavra *Esquerda*.

Com os estados de Santa Catarina e Rio Grande do Sul, possui uma variação para os comandos de 'direita', 'esquerda', 'trás', 'frente' e 'para'.

Para parte do estado do Rio de Janeiro e a sua capital, possui duas variações para a palavra 'esquerda', e uma para a palavra 'trás'.

As demais regiões não citadas acima, as demais variações mostradas, já contemplam esses cenários.

2.2. Extração de Áudio

Para iniciar os estudos e a implementação deste projeto, foi necessário contar com alguns dados do ALiB, que oferece um estudo bastante completo de possíveis variantes da Língua Portuguesa do Brasil. Paralelamente a isso, foram selecionados cinco comandos que representam os vetores de direção/posicionamento 'direita', 'esquerda', 'frente', 'trás' e 'para'. A razão da escolha por esses comandos/vetores – que chamaremos apenas de vetores a partir de agora – é simples: com eles é possível movimentar qualquer veículo sobre rodas, para ficar apenas em um exemplo.

Os cinco vetores foram capturados em peças de áudio individuais e considerando as principais variantes do Português do Brasil relativas à região Centro-Oeste Paulista.

Foi realizada a aquisição dos cinco comandos de voz a partir de 66 informantes na Fundação de Ensino Eurípides Soares da Rocha de Marília envolvendo alunos dos cursos de Bacharelado em Ciência da Computação, Bacharelado em Ciências Contábeis, Bacharelado em Sistema de Informação, Bacharelado em Direito, Bacharelado em Engenharia de Produção e também de alguns funcionários da fundação. Os métodos para aquisição dessas peças de áudio contaram com critérios bastante cuidadosos: estúdio de gravação livre de ruídos, microfone profissional de boa qualidade, sistema 'anti-puf', software de gravação específico entre outros.

Ao final, para o processamento, estiveram disponíveis 345 peças de áudio representando os cinco vetores mencionados há pouco.

Com os vetores disponíveis, o próximo passo foi encontrar uma maneira de comparar cada vetor e descobrir as similaridades entre as diferentes variantes em relação ao mesmo comando e entre sexos diferentes.

2.3. Sound Forge

Para o tratamento prévio dos áudios, foi usado o programa Sound Forge, que apresenta facilidades para manipulação de arquivos de áudio. No projeto, foram usadas algumas regras para padronizá-lo os arquivos de áudio que seguem relacionados:

- Vetores de áudio de tamanho 800 ms;
- Nessa primeira etapa do projeto, foi desconsiderado no vetor “trás”, o som da letra “a”, transformado a palavra “atrás” em “trás”;
- Cortar barulhos (ruídos) antes do pronunciamento do vetor de áudio;
- Ignorar vetores contendo ruídos (risadas) durante a gravação do fonema.

A Figura 2 mostra um vetor “trás” antes do tratamento da gravação. Foram realizados os seguintes tratamentos nos vetores:

- a) Região de ruídos antes do início da pronuncia do vetor foi descartada;
- b) Região onde é pronunciado o “a” do vetor “atrás”, foi descartada;
- c) Região onde é pronunciado o “trá” foi considerada o início do vetor;
- d) Região onde é pronunciado o “s”, foi considerada o final do vetor;
- e) Região de ruídos após a pronuncia do vetor foi descartada.

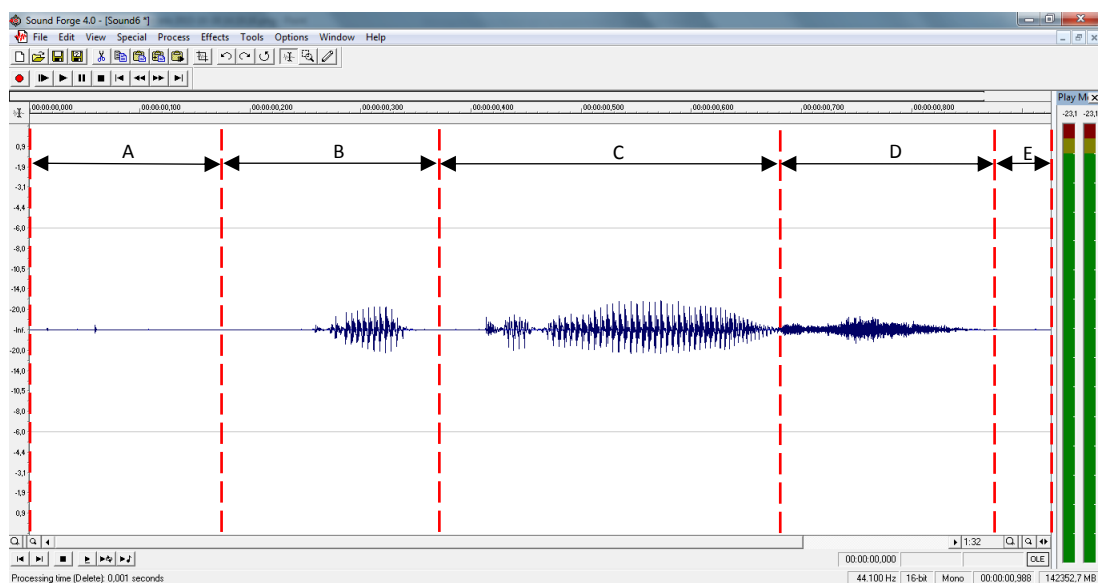


Figura 2. Vetor "trás" pronunciado por uma mulher antes do tratamento

2.4. Audacity

Com o uso do software Audacity, foi possível o manuseio dos áudios e exportar os dados das vozes em forma de gráficos. Foram realizados 2 tipos de transformações dos dados das vozes para números: Sample Data Export e através do desenho do Espectro de Frequência dos vetores.

2.4.1. Sample Data Export

Para início dos estudos a primeira forma de extração que foi testada para o uso foi a Sample Data Export. Esse método é capaz de transformar o arquivo áudio em escala linear de frequências, gerando milhares de linhas com as informações de frequência.

Para exemplificar, seguem informações referentes ao fonema “Direita”. Um arquivo de áudio para o comando “Direita”, contém mais de 18 mil linhas de informações, sendo que o arquivo possuía 800ms de duração. Para dificultar ainda mais a comparação entre os comandos, as quantidades de linhas de informações são variáveis tornando a comparação a olho nu inviável.

Para realizar uma comparação mais assertiva, foi usado o gerenciador de planilhas da Microsoft, o Office Excel 2013 que está descrito de forma detalhada nos tópicos abaixo.

2.4.2. Espectro da Frequência

Com o objetivo de possuir mais de uma forma de estudo e de comparação, foi gerado o espectro da frequência do áudio para possibilitar a comparação entre os métodos utilizados.

Com a metodologia utilizada para transformar as informações pertinentes ao áudio em dados gráficos, foi possível desenhar o espectro da Frequência e determinar e escolher a quantidade de linhas que será reduzida no arquivo e também definir o algoritmo utilizado para essa redução.

Inicialmente foi utilizado o algoritmo de redução com 256, 512 e 1024 linhas. As análises dos testes realizados determinaram que o fator de redução de 256 linhas apresentou melhores resultados.

Os testes realizados com os diferentes tamanhos dos algoritmos de extração foram executados da seguinte forma: Foram extraídos os vetores de áudio através do Audacity, gerado o arquivo para a comparação no software Weka, onde foram realizadas as análises dos resultados. Como a extração de 256 linhas apresentou melhores resultados, essa informação

foi utilizada na análise, estudos e posteriormente na implementação do projeto. O software Weka será descrito e abordado de forma mais detalhada posteriormente.

2.5. Excel

Com o intuito de encontrar algum padrão para os arquivos gerados pelo Sample Data Export, foi usado o gerenciador de tabelas da Microsoft Office Excel 2013 na tentativa de encontrar algum padrão nos arquivos de texto. A Tabela 2 contém um trecho da comparação do vetor “direita”.

Identificador	Direita A	Avaliação A	Direita B	Avaliação B	Direita E	Avaliação E
1	- 0,00040	Não escuto	0,00012	Não escuto	0	Não escuto
2	- 0,00003	Não escuto	0	Não escuto	0	Não escuto
3	- 0,00018	Não escuto	- 0,00034	Não escuto	0	Não escuto
4	- 0,00018	Não escuto	0,00018	Não escuto	0	Não escuto
5	- 0,00003	Não escuto	0,00006	Não escuto	0	Não escuto
6	- 0,00012	Não escuto	0	Não escuto	0	Não escuto
7	- 0,00012	Não escuto	- 0,00003	Não escuto	0	Não escuto
8	- 0,00027	Não escuto	- 0,00012	Não escuto	0	Não escuto
9	- 0,00015	Não escuto	0,00031	Não escuto	0	Não escuto

Tabela 2. Comparação do vetor Direita

A Tabela 2, a coluna “Identificador” é responsável pelo ID da linha do vetor de áudio. As colunas Direita A, Direita B e, Direita E, mostram as frequências que são atribuídas no instante do Identificador. A coluna Avaliação E, contém o resultado da avaliação da audição das frequências pelo ouvido humano.

Com as tabelas montadas, todos os tratamentos e avaliações realizadas não apresentou a existência de um padrão entre os fonemas existentes nos arquivos. Dessa forma, o método foi descartado para o projeto.

2.6. Weka

Após tentativas frustradas da busca pela similaridade das informações a “olho nu” e pelo software Excel, o professor Dr. Pedro Henrique Bugatti, nos apresentou o WEKA 3.6, um software *open source* que tem como objetivo agregar algoritmos provenientes de diferentes abordagens/paradigmas na subárea da inteligência artificial dedicada ao estudo da aprendizagem por parte de máquinas.

Com a colaboração do professor Dr. Pedro, foi possível testar diversos algoritmos que poderiam auxiliar na busca por similaridade. Definimos três objetivos diferentes para identificar quais os melhores algoritmos para cada situação, entre elas: *verificar qual algoritmo*

é o mais indicado para distinguir a direção falada pelo locutor, independente do sexo e da variância da voz do locutor; verificar qual algoritmo é o mais indicado para distinguir o sexo do locutor e por fim, verificar qual algoritmo é o mais indicado para distinguir a direção informada e qual o sexo do locutor. Vale destacar que a avaliação foi feita com dois tipos de extrações, uma com o algoritmo de Espectro com tamanho de 512 e outro com o algoritmo de Cepstro também de tamanho de 512.

Porém, antes de obter estes resultados, foi necessário entregar ao sistema, um conjunto de amostras/exemplos para que os diversos algoritmos fossem aplicados em amostras reais. Para isso, montamos o conjunto com todas os 30 vetores que o professor Ms. Edio Roberto Manfio disponibilizou. Para preparar o conjunto, foi necessário aprender a criar o mesmo conforme o modelo *arff* do software pede:

- Declarar o nome do projeto;
- Declarar quais atributos serão considerados;
- Declarar o conjunto de amostras.

Como exemplo, a Figura 3, mostra um exemplo de uma estrutura de dados válida para o arquivo *arff*.

Onde:

- Na linha 1, o comando *@relation* declara o nome do projeto, como no exemplo o nome é “*verificacaovetor*”;
- Nas linhas 3, 4 e 5, os comandos *@attribute* declara variáveis que são usadas para o reconhecimento, onde o atributo “*classe*”, o dado em que quer ser reconhecida, é separado em um vetor com os valores atribuídos a cada conjunto de informação;
- Já na linha 7, é colocado o comando “*@data*”, para informar que nas demais linhas do arquivo, são conjuntos de informações;
- E por fim, nas linhas 8 e 9, são as informações atribuídas a cada variável declaradas acima separadas por virgula.

```
1 @relation verificacaovetor
2
3 @attribute 'frequencial' real
4 @attribute 'amplitude1' real
5 @attribute 'classe' {para,direita,esquerda,frente,tras}
6
7 @data
8 43.066406,86.132813,direita
9 86.132813,172.265625,esquerda
```

Figura 3. Exemplo de modelo *arff*

Após a elaboração do arquivo *arff*, é necessário abrir a aplicação, selecionar a opção **Explorer** do botão à direita. Com o WEKA Explorer iniciado, é necessário selecionar a opção **Open File** e informar o arquivo ***.arff** que contém o conjunto que gostaríamos de analisá-lo e clicar em **Abrir**.

Após a abertura do arquivo, é necessário informar quais os algoritmos que gostaríamos de testar. Para isso, é necessário clicar na guia superior **Classify**, clicar em **Choose** e selecionar o algoritmo que será aplicado sobre o conjunto selecionado anteriormente. Depois de selecionado, é necessário informar o tipo de validação que deve ser realizada e por fim, clicar em **Start**. Nas avaliações realizadas, foram utilizados 18 algoritmos, entre eles 12 algoritmos diferentes de árvore, o NaiveBayes e o algoritmo Lazy IBk, com K valendo de 1 a 5. Além disto, selecionados a opção de Cross-Validation (Validação Cruzada) com o valor de 3 *folde* (pastas) para que o conjunto possa ser testado de acordo com o número de amostras disponibilizadas.

Onde os algoritmos utilizados estão distribuídos nas classes mostradas na tabela 3.

Classe	Algoritmo
Trees	BFTree
	DecisionStump
	FT
	J48
	J48graft
	LADTree
	LMT
	NBTree
	RandomForest
	RandomTree
	REPTree
	SimpleCart
Bayes	NaiveBayes
Lazy	IBk -1
	IBk -2
	IBk -3
	IBk -4
	IBk -5

Tabela 3. Distribuição dos algoritmos em Classes

Para verificar as informações do WEKA sobre a aplicação do algoritmo informado no conjunto, basta clicar sobre o nome do algoritmo na Result List (lista de resultados), localizada abaixo do botão **start**.

Lembrando que como foi citado nos sub-tópico do Audacity, foram usados dois algoritmos do Audacity (Cepstro e o Espectro) com os dezoito algoritmos do Weka, totalizando 36 diferentes análises que foram comparadas gerando os resultados descritos no item 4.

3. RESULTADOS OBTIDOS NO SOFTWARE WEKA

Com a análise dos dados obtidos pelo processamento dos arquivos do Weka, foi possível a análise dos dados de acordo com dois critérios de avaliação:

- Eficácia: Que será a porcentagem de acertos do algoritmo, pois o projeto necessita de certa precisão não interpretando de forma errônea os comandos;
- Eficiência: Que será o tempo de resposta do algoritmo, devido a necessidade do reconhecimento no menor tempo de processamento do algoritmo.

Para a execução dos algoritmos, a arquitetura utilizada para a realização dos testes foi:

- Sistema Operacional: Windows 7 Professional
- Memória RAM: 8GB
- Processador: Core i7 2670QM 2.2 GHz (até 3.1 GHz)

Com esses passos definidos, o projeto foi separado em três tipos de reconhecimentos: Reconhecimento do sexo do falante; Reconhecimento da Direção falada; e por fim o reconhecimento do sexo do falante e da direção falada.

Para facilitá-la a separação, as análises serão divididas nos tópicos de algoritmos utilizados no Audacity (Cepstro e o Espectro).

3.1. Identificação do Sexo

Foi decidido executar os algoritmos para a identificação do sexo de uma pessoa que gera um comando de voz, deve ser identificada para tornar mais real a interação humano-computador. Assim, o sistema de reconhecimento que identifica o gênero da pessoa que gera o comando, pode gerar uma resposta adequada. Gerando uma adequação ao gênero do interlocutor em seu turno, como por exemplo: “Bom dia senhor!” ou “Bom dia senhora!”, não causando constrangimento ao ouvinte devido ao erro de seu gênero.

3.1.1. Algoritmo Cepstro

Com o algoritmo Cepstro na identificação do Sexo, a tabela de assertividade é amostrada na Tabela 4.

Onde é possível identificar:

- A soma de amostras é de 345 (somatória da coluna assertividades com a coluna de erros);
- Os algoritmos da classe Lazy e com o algoritmo RandomTree da camada Trees possuíram o menor tempo de execução, todos com a execução instantânea;
- O algoritmo NBTree da classe Trees, possuiu o maior tempo de execução, com 26,76 segundos;
- Os melhores algoritmos em termos de assertividades são: o RandomForest da classe Trees e os algoritmos IBk-3 e o IBK-5 ambos da classe Lazy com 96,23%;
- O pior algoritmo em termos de assertividade, é o algoritmo RandomTree da classe Trees, com 26,96% de erros;
- Portanto os melhores algoritmos são os da classe Lazy, IBk-3 e o IBk-5;
- O pior algoritmo, considerando o tempo de execução, é o algoritmo NBTree da classe Trees.

Tipo	Algoritmo	Tempo (Segundos)	% Assertividade	Assertividade	% Erro	Erro
Trees	BFTree	1,04	84,64%	292	15,36%	53
Trees	DecisionStump	0,06	74,78%	258	25,22%	87
Trees	FT	0,49	95,36%	329	4,64%	16
Trees	J48	0,15	82,61%	285	17,39%	60
Trees	J48graft	0,28	84,06%	290	15,94%	55
Trees	LADTree	1,59	88,99%	307	11,01%	38
Trees	LMT	5,22	94,78%	327	5,22%	18
Trees	NBTree	26,76	94,20%	325	5,80%	20
Trees	RandomForest	0,39	96,23%	332	3,77%	13
Trees	RandomTree	0	73,04%	252	26,96%	93
Trees	REPTree	0,20	86,96%	300	13,04%	45
Trees	SimpleCart	0,91	84,93%	293	15,07%	52
Bayes	NaiveBayes	0,08	93,04%	321	6,96%	24
Lazy	IBk -1	0	92,46%	319	7,54%	26
Lazy	IBk -2	0	90,14%	311	9,86%	34
Lazy	IBk -3	0	96,23%	332	3,77%	13
Lazy	IBk -4	0	95,36%	329	4,64%	16
Lazy	IBk -5	0	96,23%	332	3,77%	13

Tabela 4. Identificação do Sexo - Algoritmo Cepstro

Para ilustrar melhor o reconhecimento da identificação do sexo com o algoritmo Cepstro, a Tabela 5 mostra a matriz de confusão dos algoritmos da classe Lazy, IBk-3 e o IBK-5.

	Masculina	Feminina
Masculina	131	4
Feminina	9	201

Tabela 5. Matriz de confusão - Melhor algoritmo Cepstro na identificação do sexo

Avaliando a Tabela 5 verifica-se:

- Das 135 amostras masculinas presentes no banco de dados, ele acertou 131, errando apenas quatro;
- Das 210 amostras femininas no banco de dados, ele acertou 201, errando apenas nove.

A Tabela 6, mostra a matriz de confusão dos algoritmos, considerados os piores nessa análise, que foi o algoritmo NBTree da classe Tree.

	Masculina	Feminina
Masculina	127	8
Feminina	12	198

Tabela 6. Matriz de confusão - Pior algoritmo Cepstro na identificação do sexo

Avaliando a Tabela 6 verifica-se:

- Das 135 amostras masculinas presentes no banco de dados, ele acertou 127, errando oito amostras;
- Das 198 amostras femininas no banco de dados, ele acertou 198, errando doze amostras.

3.1.2. Algoritmo Espectro

Com o algoritmo Espectro na identificação do Sexo, a assertividade é mostrada na Tabela 7.

Tipo	Algoritmo	Tempo (Segundos)	% Assertividade	Assertividade	% Erro	Erro
Trees	BFTree	1,00	84,93%	293	15,07%	52
Trees	DecisionStump	0,06	81,74%	282	18,26%	63
Trees	FT	0,50	97,39%	336	2,61%	9
Trees	J48	0,14	87,83%	303	12,17%	42
Trees	J48graft	0,26	87,25%	301	12,75%	44
Trees	LADTree	1,40	88,12%	304	11,88%	41
Trees	LMT	8,83	99,13%	342	0,87%	3
Trees	NBTree	37,86	83,19%	287	16,81%	58
Trees	RandomForest	0,42	86,67%	299	13,33%	46
Trees	RandomTree	0,01	68,41%	236	31,59%	109
Trees	REPTree	0,18	86,96%	300	13,04%	45
Trees	SimpleCart	1,05	82,90%	286	17,10%	59
Bayes	NaiveBayes	0,08	67,83%	234	32,17%	111
Lazy	IBk -1	0	85,80%	296	14,20%	49
Lazy	IBk -2	0	81,16%	280	18,84%	65
Lazy	IBk -3	0	81,45%	281	18,55%	64
Lazy	IBk -4	0	82,90%	286	17,10%	59
Lazy	IBk -5	0	79,42%	274	20,58%	71

Tabela 7. Identificação do Sexo - Algoritmo Espectro

Na Tabela 7 é possível identificar que:

- A soma de amostras é de 345 (somatória da coluna assertividades com a coluna de erros);
- Os algoritmos da classe Lazy possuíram o menor tempo de execução, todos com a execução instantânea;
- O algoritmo NBTree da classe Trees, possuiu o maior tempo de execução, com 37,86 segundos;
- O melhor algoritmo em termos de assertividades é o LMT da classe com 99,13%;
- O pior algoritmo em termos de assertividade, é o algoritmo NaiveBayes da classe Bayes, com 32,17% de erros;
- Portanto o melhor algoritmo é o FT da classe Tree, mesmo não possuindo o melhor tempo, e nem mesmo a melhor taxa de assertividade, mas a junção de um tempo de execução de 0,5 segundos, com a assertividade de 97,39% o torna o melhor algoritmo desse módulo de reconhecimento;
- O pior algoritmo, devido ao tempo de execução, é o algoritmo NBTree da classe Trees.

Para ilustrar melhor o reconhecimento da identificação do sexo com o algoritmo Espectro, a Tabela 8 mostra a matriz de confusão do algoritmo da classe Tree, o FT.

	Masculina	Feminina
Masculina	134	1
Feminina	8	202

Tabela 8. Matriz de confusão - Melhor algoritmo Espectro na identificação do sexo

Avaliando a Tabela 8 verifica-se:

- Das 135 amostras masculinas presentes no banco de dados, ele acertou 134, errando apenas uma;
- Das 210 amostras femininas no banco de dados, ele acertou 202, errando apenas oito.

A Tabela 9, mostra a matriz de confusão dos algoritmos, considerados os piores nessa análise, que foi o algoritmo NBTree da classe Tree.

	Masculina	Feminina
Masculina	103	32
Feminina	26	184

Tabela 9. Matriz de confusão - Pior algoritmo Espectro na identificação do sexo

Na Tabela 9 é possível identificar:

- Das 135 amostras masculinas presentes no banco de dados, ele acertou 103, errando 32 amostras;
- Das 210 amostras femininas no banco de dados, ele acertou 184, errando 26 amostras.

3.1.3. Comparação dos valores

Com os resultados das análises, é possível determinar que o melhor algoritmo para a identificação do sexo do locutor, é o algoritmo do Audacity Espectro com algoritmo FT da classe tree.

3.2. Identificação da direção

A identificação da direção é um dos objetivos do trabalho. Pois com o reconhecendo de direção através de comandos de voz, é possível locomover movimentar objetos ou locomover-se no eixo XY. Será possível controlar os movimentos de uma cadeira de rodas por exemplo.

Dentre os comandos definidos no projeto, vale ressaltar que o comando “Pare” ou “Para” pode também representar um comando de emergência.

O item 4.2.1 descreve a utilização e validação dos algoritmos do Audacity (Cepstro e o Espectro) através dos dezoito algoritmos do Weka.

3.2.1. Algoritmo Cepstro

Com o algoritmo Cepstro, a Tabela 10 mostra os índices de assertividade obtidos, onde é possível identificar:

- A soma de amostras é de 345 (somatória da coluna assertividades com a coluna de erros);
- Os algoritmos da classe Lazy, possuíram o menor tempo de execução, todos com a execução instantânea;
- O algoritmo NBTree da classe Trees, possuiu o maior tempo de execução, com 605,69 segundos (mais de 10 minutos de espera);
- Por tanto, o melhor algoritmo é o RandomForest da classe Trees, possuindo uma taxa de reconhecimento de 68,7% demorando 0,57 segundos para o reconhecimento;

O algoritmo que apresentou piores resultados foi o NBTree da classe Trees, possuindo uma taxa de erro de 50,43% demorando 605,69 segundos para o processamento da direção.

Tipo	Algoritmo	Tempo (Segundos)	% Assertividade	Assertividade	% Erro	Erro
Trees	BFTree	4,90	52,75%	182	47,25%	163
Trees	DecisionStump	0,25	36,52%	126	63,48%	219
Trees	FT	4,46	75,65%	261	24,35%	84
Trees	J48	1,36	48,12%	166	51,88%	179
Trees	J48graft	2,24	52,17%	180	47,83%	165
Trees	LADTree	7,50	64,93%	224	35,07%	121
Trees	LMT	43,36	74,78%	258	25,22%	87
Trees	NBTree	605,69	50,43%	174	49,57%	171
Trees	RandomForest	0,57	68,70%	237	31,30%	108
Trees	RandomTree	0,01	31,30%	108	68,70%	237
Trees	REPTree	0,27	52,75%	182	47,25%	163
Trees	SimpleCart	1,26	54,49%	188	45,51%	157
Bayes	NaiveBayes	0,37	60,29%	208	39,71%	137
Lazy	IBk -1	0	36,81%	127	63,19%	218
Lazy	IBk -2	0	38,84%	134	61,16%	211
Lazy	IBk -3	0	43,48%	150	56,52%	195
Lazy	IBk -4	0	46,67%	161	53,33%	184
Lazy	IBk -5	0	48,41%	167	51,59%	178

Tabela 10. Identificação da Direção - Algoritmo Cepstro

As Tabelas 11, 12, 14, 15, 18, 20 e 21 concentram grandes quantidades de informações essenciais para o entendimento mais amplo do experimento. Portanto, é necessário, nesse ponto do trabalho, explicar de antemão como os dados nelas contidos devem ser interpretados para que a leitura possa fluir melhor.

Tomando como exemplo da Tabela 11 a seguir, existe uma coluna à esquerda com os *comandos enviados* e uma linha em que estão dispostos os *comandos reconhecidos*. Para esta Tabela há como referência um número de 69 amostras, valor indicado também na coluna da esquerda. Assim sendo, todo número localizado num campo em que ocorra uma zona de intersecção entre dois parâmetros dispostos nas colunas e linhas devem ser interpretados como *reconhecidos* pelo software. Um exemplo seria a zona de intersecção *para* e *para* em que encontra-se no campo o número 63. Isso significa que o software *reconheceu* o comando 63 vezes. A zona de intersecção *para* e *direita* em que se encontra no campo o número 02 significa que o software *confundiu* o comando *para* com *direita* 02 vezes.

Para ilustrar melhor o reconhecimento da identificação da direção com o algoritmo Cepstro, a Tabela 11 mostra a matriz de confusão do algoritmo da classe Tree, o RandomForest.

COMANDOS ENVIADOS (69)	COMANDOS RECONHECIDOS				
	Para	Direita	Esquerda	Frente	Trás
Para	63	2	2	1	1
Direita	1	47	11	10	0
Esquerda	5	27	28	5	4
Frente	4	15	3	43	4
Trás	7	2	3	1	56

Tabela 11. Matriz de confusão - Melhor algoritmo Cepstro na identificação da Direção

A Tabela 12, mostra a matriz de confusão do pior algoritmo para a identificação da direção, que é o algoritmo NBTree da classe Tree.

COMANDOS ENVIADOS (69)	COMANDOS RECONHECIDOS				
	Para	Direita	Esquerda	Frente	Trás
Para	41	5	8	8	7
Direita	1	30	20	13	5
Esquerda	11	16	28	7	7
Frente	6	14	8	33	8
Trás	8	0	8	11	42

Tabela 12. Matriz de confusão - Pior algoritmo Cepstro na identificação da direção

3.2.2. Algoritmo Espectro

Com o algoritmo Espectro, a Tabela 13 contém os índices de assertividade obtidos, onde é possível identificar:

- A soma de amostras é de 345 (somatória da coluna assertividades com a coluna de erros);
- Os algoritmos da classe Lazy, possuíram o menor tempo de execução, todos com a execução instantânea;
- O algoritmo NBTree da classe Trees, possuiu o maior tempo de execução, com 246,61 segundos (mais de 4 minutos de espera);
- Por tanto, o melhor algoritmo é o FT da classe Trees, possuindo uma taxa de reconhecimento de 85,8% demorando 1,31 segundos para o reconhecimento;

Já o pior algoritmo, é o NBTree da classe Trees, possuindo uma taxa de erro de 31,88% demorando 246,61 segundos para o processamento da direção.

Tipo	Algoritmo	Tempo (Segundos)	% Assertividade	Assertividade	% Erro	Erro
Trees	BFTree	1,23	65,51%	226	34,49%	119
Trees	DecisionStump	0,07	37,10%	128	62,90%	217
Trees	FT	1,31	85,80%	296	14,20%	49
Trees	J48	0,26	67,25%	232	32,75%	113
Trees	J48graft	0,55	68,70%	237	31,30%	108
Trees	LADTree	2,09	69,86%	241	30,14%	104
Trees	LMT	16,28	88,12%	304	11,88%	41
Trees	NBTree	246,61	68,12%	235	31,88%	110
Trees	RandomForest	0,47	79,71%	275	20,29%	70
Trees	RandomTree	0,01	60,29%	208	39,71%	137
Trees	REPTree	0,26	66,96%	231	33,04%	114
Trees	SimpleCart	1,09	66,96%	231	33,04%	114
Bayes	NaiveBayes	0,06	59,42%	205	40,58%	140
Lazy	IBk -1	0	73,04%	252	26,96%	93
Lazy	IBk -2	0	73,33%	253	26,67%	92
Lazy	IBk -3	0	77,10%	266	22,90%	79
Lazy	IBk -4	0	79,42%	274	20,58%	71
Lazy	IBk -5	0	79,42%	274	20,58%	71

Tabela 13. Identificação da Direção - Algoritmo Espectro

Para ilustrar melhor o reconhecimento da identificação da direção com o algoritmo Espectro, a Tabela 14 mostra a matriz de confusão do algoritmo da classe Trees, o RandomForest.

COMANDOS ENVIADOS (69)	COMANDOS RECONHECIDOS				
	Para	Direita	Esquerda	Frente	Trás
Para	61	0	0	1	7
Direita	0	56	2	11	0
Esquerda	0	1	61	6	1
Frente	2	13	3	51	0
Trás	2	0	0	0	67

Tabela 14. Matriz de confusão - Melhor algoritmo Espectro na identificação da Direção

A Tabela 15, mostra a matriz de confusão do pior algoritmo para a identificação da direção, que é o algoritmo NBTree da classe Tree.

COMANDOS ENVIADOS (69)	COMANDOS RECONHECIDOS				
	Para	Direita	Esquerda	Frente	Trás
Para	54	3	0	3	9
Direita	3	40	4	21	1
Esquerda	2	6	51	7	3
Frente	2	12	12	39	4
Trás	8	3	4	3	51

Tabela 15. Matriz de confusão - Pior algoritmo Espectro na identificação da direção

3.2.3. Comparação dos valores

Com os resultados das duas análises, é possível determinar que o melhor algoritmo para a identificação da direção, é o algoritmo do Audacity Espectro com algoritmo FT da classe tree.

3.3. A identificação do vetor e do sexo

A identificação do vetor pronunciado e do sexo do falante é importante ao trabalho, pois além de identificar a orientação de locomoção, ela é capaz de adequar o discurso ao gênero do interlocutor, podendo proporcionar interação homem-máquina de modo mais próximo ao humano.

Conforme o que foi documentado, o reconhecimento do comando do “Pare” ou “Para” é o comando de maior significação, por se tratar do comando de emergência.

A seguir, mostra o estudo feito com os dois algoritmos do Audacity (Cepstro e o Espectro) usando os dezoito algoritmos do Weka.

3.3.1. Algoritmo Cepstro

Com o algoritmo Cepstro, a Tabela 16 mostra os índices de assertividade obtidos.

Tipo	Algoritmo	Tempo (Segundos)	% Assertividade	Assertividade	% Erro	Erro
Trees	BFTree	2,27	43,19%	149	56,81%	196
Trees	DecisionStump	0,09	21,45%	74	78,55%	271
Trees	FT	4,42	66,96%	231	33,04%	114
Trees	J48	0,44	39,13%	135	60,87%	210
Trees	J48graft	0,85	40,29%	139	59,71%	206
Trees	LADTree	3,77	55,07%	190	44,93%	155
Trees	LMT	101,86	69,28%	239	30,72%	106
Trees	NBTree	720,91	39,71%	137	60,29%	208
Trees	RandomForest	0,74	60,58%	209	39,42%	136
Trees	RandomTree	0,01	20,00%	69	80,00%	276
Trees	REPTree	0,33	42,90%	148	57,10%	197
Trees	SimpleCart	1,64	39,13%	135	60,87%	210
Bayes	NaiveBayes	0,08	64,06%	221	35,94%	124
Lazy	IBk -1	0	35,07%	121	64,93%	224
Lazy	IBk -2	0	34,49%	119	65,51%	226
Lazy	IBk -3	0	41,16%	142	58,84%	203
Lazy	IBk -4	0	44,64%	154	55,36%	191
Lazy	IBk -5	0	43,48%	150	56,52%	195

Tabela 16. Identificação da Direção e do Sexo - Algoritmo Cepstro

Na Tabela 16 é possível identificar:

- A soma de amostras é de 345 (somatória da coluna assertividades com a coluna de erros);
- Os algoritmos da classe Lazy, possuíram o menor tempo de execução, todos com a execução instantânea;
- O algoritmo NBTree da classe Trees, possuiu o maior tempo de execução, com 720,91 segundos (mais de 12 minutos de espera);
- O algoritmo que apresentou melhor resultado foi o NaiveBayes da classe Bayes, apresentando uma taxa de reconhecimento de 64,06% e demorando 0,08 segundos para o reconhecimento.
- Já o pior algoritmo, é o NBTree da classe Trees, possuindo uma taxa de erro de 39,71 % demorando 720,91 segundos para o processamento da direção.

Para ilustrar melhor o reconhecimento da identificação da direção com o algoritmo Cepstro, a Tabela 17 mostra a matriz de confusão do algoritmo da classe Bayes, o NaiveBayes.

Comandos Enviados Masc (27) Fem (42)	Comandos Reconhecidos									
	Direita Masc	Direita Fem	Esquerda Masc	Esquerda Fem	Frente Masc	Frente Fem	Para Masc	Para Fem	Trás Masc	Trás Fem
Direita Masc	17	0	5	1	4	0	0	0	0	0
Direita Fem	0	32	0	4	0	6	0	0	0	0
Esquerda Masc	6	0	14	0	6	0	0	0	1	0
Esquerda Fem	0	9	0	26	0	5	0	0	0	2
Frente Masc	7	0	3	1	12	0	0	0	3	1
Frente Fem	0	10	0	3	2	27	0	0	0	0
Para Masc	0	0	3	0	0	0	20	0	4	0
Para Fem	0	1	0	4	0	4	1	17	0	15
Trás Masc	1	0	4	0	2	0	1	0	19	0
Trás Fem	0	0	1	1	0	1	0	2	0	37

Tabela 17. Matriz de confusão - Melhor algoritmo Cepstro na identificação da Direção e Sexo

A Tabela 18, mostra a matriz de confusão do pior algoritmo para a identificação da direção e do sexo, que é o algoritmo NBTree da classe Trees.

Comandos Enviados Masc (27) Fem (42)	Comandos Reconhecidos									
	Direita Masc	Direita Fem	Esquerda Masc	Esquerda Fem	Frente Masc	Frente Fem	Para Masc	Para Fem	Trás Masc	Trás Fem
Direita Masc	9	1	4	2	4	1	3	2	0	1
Direita Fem	2	23	0	10	1	5	0	1	0	0
Esquerda Masc	5	0	8	1	2	0	1	2	2	6
Esquerda Fem	2	10	1	14	1	9	0	3	0	2
Frente Masc	4	0	2	2	10	1	1	1	3	3
Frente Fem	1	10	2	4	3	15	0	3	1	3
Para Masc	1	0	2	0	0	0	16	2	4	2
Para Fem	1	0	1	10	0	5	2	16	1	6
Trás Masc	2	0	4	1	5	1	0	2	9	3
Trás Fem	2	3	0	6	1	5	1	5	2	17

Tabela 18. Matriz de confusão - Pior algoritmo Cepstro na identificação da direção e Sexo

3.3.2. Algoritmo Espectro

Com o algoritmo Espectro, a Tabela 19 mostra os índices de assertividade obtidos.

Onde é possível identificar que:

- A soma de amostras é de 345 (somatória da coluna assertividades com a coluna de erros);
- Os algoritmos da classe Lazy, apresentaram o menor tempo de execução, todos com a execução instantânea;
- O algoritmo NBTree da classe Trees, apresentou o maior tempo de execução, com 511,33 segundos (mais de 8 minutos de espera);
- Portanto, o melhor algoritmo é o FT da classe Trees, que apresenta uma taxa de reconhecimento de 88,99% demorando 3,39 segundos para o reconhecimento;
- Já o pior algoritmo, é o NBTree da classe Trees, apresenta uma taxa de erro de 51,88% demorando 511,33 segundos para o processamento da direção.

Tipo	Algoritmo	Tempo (Segundos)	% Assertividade	Assertividade	% Erro	Erro
Trees	BFTree	1,94	55,07%	190	44,93%	155
Trees	DecisionStump	0,11	22,03%	76	77,97%	269
Trees	FT	3,39	88,99%	307	11,01%	38
Trees	J48	0,37	55,07%	190	44,93%	155
Trees	J48graft	0,66	57,97%	200	42,03%	145
Trees	LADTree	3,48	60,58%	209	39,42%	136
Trees	LMT	31,81	88,99%	307	11,01%	38
Trees	NBTree	511,33	51,88%	179	48,12%	166
Trees	RandomForest	0,69	65,51%	226	34,49%	119
Trees	RandomTree	0,01	41,45%	143	58,55%	202
Trees	REPTree	0,31	49,86%	172	50,14%	173
Trees	SimpleCart	1,44	54,20%	187	45,80%	158
Bayes	NaiveBayes	0,07	44,93%	155	55,07%	190
Lazy	IBk -1	0	63,48%	219	36,52%	126
Lazy	IBk -2	0	60,29%	208	39,71%	137
Lazy	IBk -3	0	60,29%	208	39,71%	137
Lazy	IBk -4	0	65,51%	226	34,49%	119
Lazy	IBk -5	0	66,09%	228	33,91%	117

Tabela 19. Identificação do Direção e Sexo - Algoritmo Espectro

Para ilustrar melhor o reconhecimento da identificação da direção e do sexo com o algoritmo Espectro, a Tabela 20 mostra a matriz de confusão do algoritmo da classe Trees, o FT.

Comandos Enviados Masc (27) Fem (42)	Comandos Reconhecidos									
	Direita Masc	Direita Fem	Esquerda Masc	Esquerda Fem	Frente Masc	Frente Fem	Para Masc	Para Fem	Trás Masc	Trás Fem
Direita Masc	24	1	0	0	2	0	0	0	0	0
Direita Fem	1	38	0	0	0	3	0	0	0	0
Esquerda Masc	1	0	24	0	1	0	0	0	1	0
Esquerda Fem	0	0	0	40	0	1	0	0	0	1
Frente Masc	2	1	0	0	21	3	0	0	0	0
Frente Fem	0	6	0	0	0	33	0	1	2	0
Para Masc	0	0	0	0	0	0	26	0	1	0
Para Fem	0	0	0	0	0	1	0	39	0	2
Trás Masc	0	1	0	0	0	0	0	1	24	1
Trás Fem	0	0	0	0	0	0	0	4	0	38

Tabela 20. Matriz de confusão - Melhor algoritmo Espectro na identificação da Direção e do Sexo

A Tabela 21, mostra a matriz de confusão do pior algoritmo para a identificação da direção e do Sexo, que é o algoritmo NBTree da classe Trees.

Comandos Enviados Masc (27) Fem (42)	Comandos Reconhecidos									
	Direita Masc	Direita Fem	Esquerda Masc	Esquerda Fem	Frente Masc	Frente Fem	Para Masc	Para Fem	Trás Masc	Trás Fem
Direita Masc	15	2	1	0	5	4	0	0	0	0
Direita Fem	6	20	1	0	4	9	0	0	0	2
Esquerda Masc	1	1	15	5	1	4	0	0	0	0
Esquerda Fem	4	4	3	25	2	1	0	0	1	2
Frente Masc	9	1	4	0	9	4	0	0	0	0
Frente Fem	1	10	4	4	2	21	0	0	0	0
Para Masc	3	0	1	0	0	0	13	9	1	0
Para Fem	2	2	0	0	0	2	7	23	0	6
Trás Masc	4	0	3	1	0	2	1	0	14	2
Trás Fem	4	0	1	5	0	0	2	4	2	24

Tabela 21. Matriz de confusão - Pior algoritmo Espectro na identificação da direção e do sexo

3.3.3. Comparação dos valores

Com os resultados das duas análises, é possível determinar que o melhor algoritmo para a identificação da direção e sexo, é o algoritmo do Audacity Espectro com algoritmo FT da classe tree.

3.4. ESTUDO E PROPOSTA DE IMPLEMENTAÇÃO

A implementação desse projeto para o reconhecimento dos algoritmos do Weka, pode usar a documentação e a própria API de consumo do projeto. Tal API, desenvolvida na linguagem de programação Java tornando seus exemplos didáticos, de fácil entendimento e devido que a linguagem é a líder do mercado segundo ranking feito pela instituição Spectrum do IEEE. Na Figura 4, segue o Top 10 das linguagens de programação classificadas pela instituição.

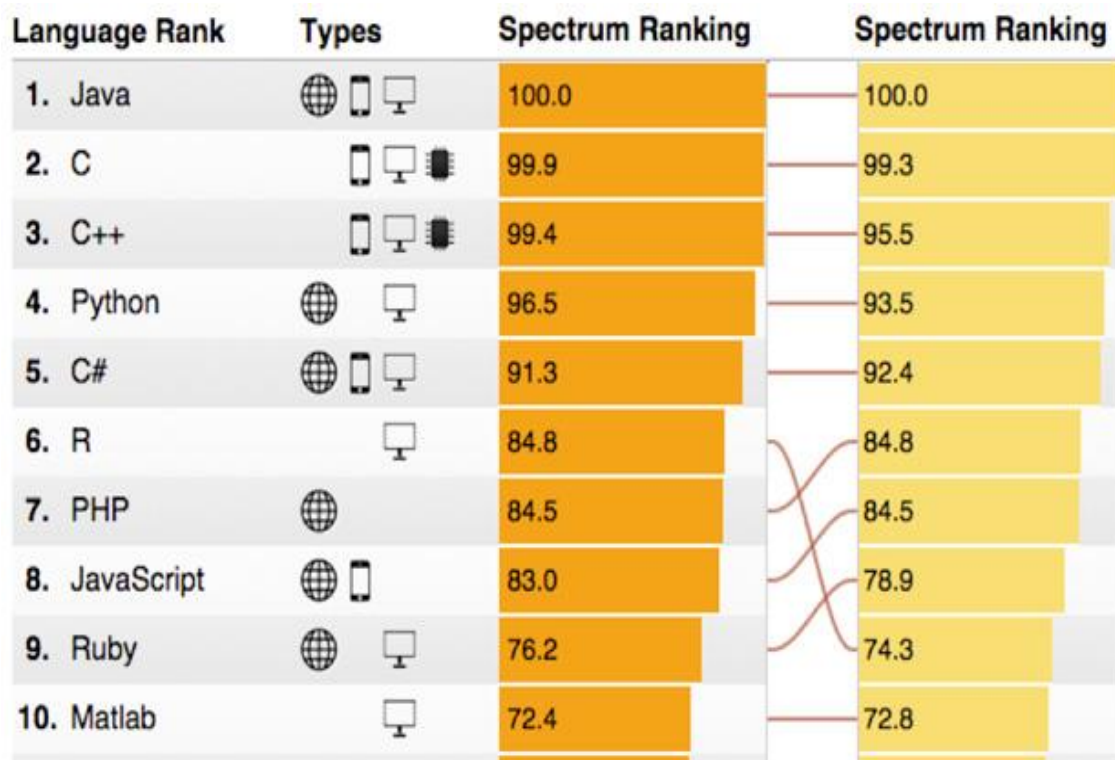


Figura 4. Ranking linguagem de programação de 2015 em relação ao Ranking de 2014. [7]

Para o uso da API do Weka, é necessário a biblioteca do Weka disponibilizada em seu próprio site [8] e um ambiente com o Java instalado para o consumo da biblioteca.

Ilustrando o processo descrito acima, a Figura 5 mostra o fluxograma de uso.

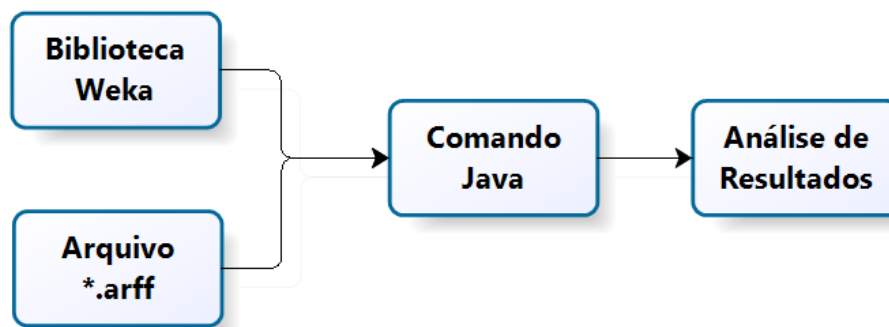


Figura 5. Fluxograma da Implementação

Para o teste de consumo do usaremos o arquivo *arff* gerado para a identificação do sexo com o algoritmo do espectro.

Para demonstrar os algoritmos, usaremos o Prompt de Comando do próprio Windows para ativa-los e avaliar os resultados gerados com o uso do aplicativo do Weka.

Para executar a função, é necessário o comando que pode ser visto na Figura 6:

```
C:\windows\system32\cmd.exe
C:\weka>java -cp weka.jar weka.classifiers.trees.J48 -t amostras_espectro_sexo.arff
```

A imagem mostra uma janela do Prompt de Comando do Windows. O título da janela é 'C:\windows\system32\cmd.exe'. O conteúdo da janela mostra o comando `java -cp weka.jar weka.classifiers.trees.J48 -t amostras_espectro_sexo.arff` sendo executado. Os números 1, 2 e 3 estão posicionados sob as partes correspondentes do comando: 'weka.jar' (1), 'weka.classifiers.trees.J48' (2) e 'amostras_espectro_sexo.arff' (3).

Figura 6. Comando para a execução do algoritmo

Onde:

1. Biblioteca do Weka
2. Algoritmo usado para a identificação do Sexo
3. Arquivo *arff* contendo valores das amostras

Executando esse comando ele fará uma validação cruzada usando 10 folders. Vale-se lembrar de que os testes anteriormente, eram usando 3 folders. O retorno é mostrado na Figura 7:

```

C:\windows\system32\cmd.exe

=== Stratified cross-validation ===
Correctly Classified Instances      308           89.2754 %
Incorrectly Classified Instances    37            10.7246 %
Kappa statistic                    0.774
Mean absolute error                 0.1071
Root mean squared error             0.3113
Relative absolute error             22.483 %
Root relative squared error         63.7815 %
Total Number of Instances          345

=== Confusion Matrix ===
 a  b  <-- classified as
115 20 | a = masculina
 17 193 | b = feminina

```

Figura 7. Retorno API do Weka

Na Figura 7, é possível verificar que:

- Foi acertado 308 vetores, com um percentual de 89,2754%;
- Errou em 37 vetores, 10,7246%;
- Já na análise de sua matriz de confusão, é possível verificar:
 - Acertou 115 vetores masculinos, dos 135 presentes;
 - Acertou 193 vetores femininos, dos 210 presentes.

Para comparação dos resultados, foi executada uma análise no aplicativo do Weka que é ilustrada Figura 8.

```

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      308           89.2754 %
Incorrectly Classified Instances    37            10.7246 %
Kappa statistic                    0.774
Mean absolute error                 0.1071
Root mean squared error             0.3113
Relative absolute error             22.483 %
Root relative squared error         63.7815 %
Total Number of Instances          345

=== Detailed Accuracy By Class ===

          TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
          0.852   0.081   0.871     0.852   0.861     0.931   masculina
          0.919   0.148   0.906     0.919   0.913     0.931   feminina
Weighted Avg.  0.893   0.122   0.892     0.893   0.893     0.931

=== Confusion Matrix ===

 a  b  <-- classified as
115 20 | a = masculina
 17 193 | b = feminina

```

Figura 8. Resultados com Validação pelo aplicativo do Weka

Na Figura 8, é possível verificar:

- Acertou 308 vetores, com um percentual de 89,2754%;
- Errou em 37 vetores, 10,7246%;
- Já na análise de sua matriz de confusão, é possível verificar:
 - Acertou 115 vetores masculinos, dos 135 presentes;
 - Acertou 193 vetores femininos, dos 210 presentes.

Como os dois resultados coincidiram, foi possível mostrar que tanto pelo aplicativo, quanto pela API, os resultados gerados foram os mesmos. Tornando a implementação desse projeto confiável e ágil, uma vez que podemos utilizar os algoritmos já desenvolvidos pela a comunidade do Weka.

4. CONCLUSÕES

Os objetivos iniciais do projeto foram parcialmente obtidos. No entanto, os resultados parciais obtidos são bastante consistentes e podem ser utilizados como parâmetros em projetos futuros.

A quantidade de Algoritmos utilizados para avaliar os cinco fonemas estudados nesse projeto foi bastante abrangente e possibilitou determinar que a assertividade obtida utilizando a metodologia de avaliação dos fonemas é válida e apresentou bons resultados. Também foi possível mostrar que para os três reconhecimentos testados, o algoritmo do Audacity de espectro de frequência Espectro é o mais eficiente nos três casos. Já o algoritmo FT, da classe Trees do Weka foi considerado o mais adequado para realizar as análises, pô apresentar os melhores resultados.

Ferramentas de reconhecimento de voz podem ser utilizadas em uma diversidade de áreas, dentre as que apresentam maior necessidade e aplicações estão: interface Homem-Máquina, Realidade Virtual, Realidade Aumentada, Jogos, Domótica, Medicina, dentre outras.

Para a continuidade desse projeto, sugere-se que os resultados obtidos sejam implementados os algoritmos que apresentaram os melhores resultados, para gerar ferramentas que realizem reconhecimento de voz, utilizando a diversidade de informações contidas no ALIB e, disponibilizando-as para utilização em projetos embarcados que necessitem de ferramentas de reconhecimento de voz.

5. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] BEDO, Marcos Vinícius Naves **Incluindo funções de distância e extratores de características para suporte a consultas por similaridade**. 2013. Dissertação (Mestrado em Ciências de Computação e Matemática Computacional) - Instituto de Ciências Matemáticas e de Computação, University of São Paulo, São Carlos, 2013. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/55/55134/tde-08112013-160506/>>. Acesso em: 14 mar. 2015.
- [2] BUGATTI, Pedro H.; TRAINA, Agra J.M; TRAINA JR, Caetano. Assessing the best integration between distance-function and image-feature to answer similarity queries. In: Proceedings of the 2008 ACM symposium on Applied computing. ACM, 2008. p. 1225-1230.
- [3] MANFIO, Edio Roberto. Processamento de Linguagem Natural, Processamento de Sinais da Fala, Geolinguística e um Naco de Humor. In: *Anais do X Seminário de Iniciação Científica Estudos Linguísticos e Literários - Sóletras*. UENP – Jacarezinho, 2013. Disponível em: <http://www.cj.uenp.edu.br/index.php/institucional/eventos/1-soletras/event_details>. Acesso em: 21 jun. 2014.
- [4] MELO, Bruno Mariani de. **Áudio sobre IP**. 2011. Monografia (Especialização - Pós-Graduação Lato Sensu em Sistemas de Telecomunicações) Escola Superior Aberta do Brasil –Esab, Vila Velha, Espírito Santo, 2011. Disponível em: <http://www.esab.edu.br/arquivos/monografias/monografia_brunomariani.pdf>. Acesso em: 20 nov. 2014.
- [5] CARDOSO, Suzana Alice Marcelino da Silva et al. *Atlas Linguístico do Brasil: Cartas Linguísticas I*. Vol. 2. Londrina: Eduel, 2014b.
- [6] **Projeto Atlas Linguístico do Brasi**. Disponível em : <<http://twiki.ufba.br/twiki/bin/view/Alib/AlibObjetivos> > Acesso em 16 de Outubro de 2015.
- [7] Top 10 de linguagens de programação. Disponível em: <<http://spectrum.ieee.org/computing/software/the-2015-top-ten-programming-languages> > Acesso em 16 de Outubro de 2015.
- [8] Documentação Oficial da API Weka. Disponível em: <<http://weka.wikispaces.com/> > Acesso em 16 de Outubro de 2015.