UNIVERSIDADE DO SAGRADO CORAÇÃO

JOSÉ LUCAS MIRANDA PEREIRA

IMPLEMENTAÇÃO DE RECONHECIMENTO DE VOZ PARA VALIDAÇÃO DE SENHA

BAURU

2015

JOSÉ LUCAS MIRANDA PEREIRA

IMPLEMENTAÇÃO DE RECONHECIMENTO DE VOZ PARA VALIDAÇÃO DE SENHA.

Trabalho de conclusão de curso apresentado ao Centro de Ciências Exatas e Sociais Aplicadas como parte dos requisitos para obtenção de titulo de Bacharel em Ciência da Computação, sob a orientação do Profº Dr. Elvio Gilberto da Silva.

BAURU

2015

JOSÉ LUCAS MIRANDA PEREIRA

IMPLEMENTAÇÃO DE RECONHECIMENTO DE VOZ PARA VALIDAÇÃO DE SENHA.

Trabalho de conclusão de curso apresentado ao Centro de Ciências Exatas e Sociais Aplicadas como parte dos requisitos para obtenção de titulo de Bacharel em Ciência da Computação, sob a orientação do Profº Dr. Elvio Gilberto da Silva.

Banca examinadora:

_____

Prof. Esp. André Luiz Ferraz Castro
Universidade do Sagrado Coração

_____

Prof. Me. Patrick Pedreira Silva
Universidade do Sagrado Coração

_____

Prof. Dr. Elvio Gilberto da Silva
Universidade do Sagrado Coração

Bauru, 8 de dezembro de 2015

AGRADECIMENTOS

Gostaria de agradecer primeiramente a minha mãe, Francisca por me oferecer apoio durante o decorrer deste trabalho.

Ao meu orientador Elvio Gilberto pela paciência e por todos seus ensinamentos e opiniões que me auxiliaram na conclusão deste.

Agradeço também aos meus amigos que me alegraram e me fizeram por muitas vezes continuar quando na verdade tudo o que eu queria era desistir.

RESUMO

As técnicas de processamento de áudio vêm evoluindo rapidamente ao longo das últimas décadas, sendo aprimoradas através de algoritmos cada vez mais complexos e hardwares mais potentes, e com esse avanço, vêm se tornando cada vez mais eficazes as ferramentas de reconhecimento de voz. Estas ferramentas são capazes de auxiliar os usuários em rotinas comuns do dia a dia, facilitando ações cotidianas devido a sua praticidade e rapidez de resposta, assim como também são importantes aliadas para a inclusão de usuários que são incapazes de manipularem ferramentas eletrônicas de forma manual, como teclados, mouses e outros tipos de acessórios, devido a algum tipo de problema motor ou adverso. Devido a estas constatações, este trabalho teve como objetivo a criação de uma página de internet capaz de autenticar um usuário através de sua voz, para isto foi utilizada a ferramenta Java Speech e o kit de ferramentas de reconhecimento de voz CMU Sphinx para a criação de um software que, ao ser incluso em uma página pode ser inicializado através de qualquer navegador de internet, sendo este capaz de reconhecer os campos “nome de usuário” e “senha”, ditados pelo usuário através do microfone. Para a quantificação da eficiência, foram realizados testes de resposta através de tentativas de autenticação na página que mostraram um grau satisfatório de eficiência, comprovando a eficácia da ferramenta.

Palavras-chave: Reconhecimento de voz. Acessibilidade. Java Speech. CMU Sphinx. Java Web Start.

ABSTRACT

The audio processing techniques have evolved rapidly over the last few decades, being improved through increasingly complex algorithms and more powerful hardware, and with this advance, are becoming even more effective the voice recognition tools. These tools are able to assist users in common everyday routines, facilitating daily actions due to its convenience and speed of response, and are also important allies for the inclusion of users who are unable to manipulate electronic tools manually, such as keyboards, mice and other accessories, due to some motor issue or an adverse factor. Due to these findings, this study aimed the creation of a web page able to authenticate an user through his voice, for this it was used the Java Speech tool and the speech recognition toolkit CMU Sphinx for building a software that when included on a page can be booted via any web browser, which is able to recognize the user name field and the password, dictated by the user through the microphone, to quantify the efficiency have been conducted response tests through authentication attempts on the page that showed a high degree of efficiency, proving the effectiveness of the tool.

Key words: Voice recognition. Accessibility. Java Speech, CMU Sphinx, Java Web Start.

SUMÁRIO

1 INTRODUÇÃO **8**
1.1 OBJETIVOS 10
1.1.1 OBJETIVO GERAL 10
1.1.2 OBJETIVOS ESPECÍFICOS 10

2 REVISÃO DA LITERATURA **11**
2.1 FUNDAMENTOS PARA O PROCESSAMENTO DE VOZ 11
2.2 RECONHECIMENTO DE VOZ 12
2.3 O PROCESSAMENTO DOS SINAIS DE FALA 14
2.5 JAVA 18
2.5.1 JAVA SPEECH API 18
2.5.2 JAVA AWT 19
2.5.3 JSGF 20
2.6 PHP 20
2.7 SPHINX4 21
2.7.1 SPHINXTRAIN 22
2.8 BANCO DE DADOS 22
2.9 SQL 23
2.10 DESENVOLVIMENTO DE SOFTWARE 23
2.10.1 UML 24
2.10.2 TESTES 24

3 TRABALHOS CORRELATOS **26**
3.1 RECONHECIMENTO DE VOZ PARA PALAVRAS ISOLADAS 26
3.2 RECONHECIMENTO DE VOZ PARA COMANDOS DE DIRECIONAMENTO POR MEIO DE REDES NEURAIS 27
3.3 PARAKEET: A CONTINUOUS SPEECH RECOGNITION SYSTEM FOR MOBILE TOUCH-SCREEN DEVICES 28

4 METODOLOGIA **29**

5 RESULTADOS **35**
5.1 TESTES REALIZADOS 36

6 CONCLUSÃO E TRABALHOS FUTUROS **40**

REFERÊNCIAS **42**

LISTA DE FIGURAS

Figura 1 - Diagrama de processo de produção e captação da fala. 12
Figura 2 - Conversão de sinal analógico em digital. 15
Figura 3 - Exemplo de modelo oculto de Markov. 17
Figura 4 - Exemplo prático de Modelo oculto de Markov 17
Figura 5 - Arquitetura de aplicação utilizando o Java Speech. 19
Figura 6 - Modelo de funcionamento da linguagem PHP 21
Figura 7 - Modelo de desenvolvimento cascata 24
Figura 8 - Resultados dos experimentos de Silva. 26
Figura 9 - Redes criadas para teste de reconhecimento. 27
Figura 10 - Resultados dos testes das redes criadas. 28
Figura 11 - Diagrama de caso de uso 29
Figura 12 - Diagrama de caso de classe. 29
Figura 13 - Diagrama de atividades. 30
Figura 14 - Banco de dados de teste. 30
Figura 15 - Disposição de arquivos utilizados. 31
Figura 16 - Configurando a localização do arquivo e criando assinatura digital. 32
Figura 17 - Arquivos e bibliotecas utilizadas. 33
Figura 18 - Vocabulário da aplicação. 33
Figura 19 - Arquivo xml e algumas de suas configurações 34
Figura 20 - Versão final da página pronta para testes 35
Figura 21 - Adicionando a pagina a lista de exceções de confiança para testes. 36
Figura 22 - Índice de acertos do locutor 1. 37
Figura 23 - Índice de acertos do locutor 2. 37
Figura 24 - Índice de acertos do locutor 3. 38
Figura 25 - Índice de acertos do locutor 4. 38
Figura 26 - Índice médio de acertos. 39

1 INTRODUÇÃO

As pesquisas inerentes ao reconhecimento da fala foram iniciadas há muito tempo, e, segundo Louzada (2010) começaram a partir da década de 50 e atualmente devido ao seu alto grau de efetividade, já atuam em diversos dispositivos eletrônicos como celulares e computadores. À medida que as técnicas de reconhecimento se tornam mais sofisticadas, os objetivos dos pesquisadores se tornam cada vez mais complexos. Para Roe (1994), o objetivo final das pesquisas não é voltado para que o computador apenas o reconheça palavras e gere respostas prefixas, mas sim que também conheça o significado das mesmas, e talvez até mesmo perceba as emoções do locutor para baseado nelas, tomar a melhor medida de ações.

Embora existam avanços contínuos na área de reconhecimento de voz, ela permanece sendo trabalhada continuamente devido a sua importância como facilitadora, principalmente por razões sociais de inclusão, ou seja, ela serve como auxílio para pessoas que possuam algum tipo de deficiência que as impossibilite de usar não só computadores e seus periféricos como mouses e teclados, assim como, celulares, serviços bancários, entre outros. Outra vantagem para o reconhecimento de voz é a rapidez e a praticidade a qual pode se obter resposta através dessa tecnologia comparada aos movimentos motores, tornando-a um atrativo comercial para empresas, principalmente na área de portáteis como celulares, tablets e smart watches. Um bom exemplo disso é o Siri para Iphone, uma assistente virtual para os celulares Iphone que automatiza inúmeras tarefas através de comandos simples, dando maior eficiência e conforto ao usuário.

Em termos técnicos, a fala nada mais é que uma sequência de sinais que se diferenciam pela linguagem e características individuais daquele que fala, como por exemplo, uma voz de tom agudo ou grave ou até mesmo o sotaque. (BRESOLIN, 2003). Para se capturar estes sinais, um computador necessita de uma forma de recebê-los através de um dispositivo (um microfone, por exemplo), para que então possa processá-los e definir a partir do conteúdo processado as suas ações.

Este processamento que, segundo Furui (2004), por sua vez funciona transformando a entrada destes sinais acústicos em sinais digitais, e após a eliminação de ruídos e outros problemas externos de ambiente comparam estes dados com um banco de dados previamente moldado em busca de resultados compatíveis para então gerar um resultado.

Para que haja sucesso na realização do projeto faz-se então necessário o pleno conhecimento das técnicas não só de programação, assim como também é fundamental conhecer as técnicas de síntese e análise do som, as formas de produção, percepção e

entendimento da linguística e fonética para poder gerar resultados os mais próximos possíveis do desejado.

A comunicação através da fala é, segundo Furui (2004), uma das habilidades dentre as mais essenciais possuídas pelo ser humano. Levando isto em consideração, fica implícita a necessidade de se possuir uma forma de levar tal capacidade a um meio digital capaz de compreender e responder comandos, dando maior agilidade e autonomia aos usuários, algo indispensável nos dias de hoje.

Optou-se então pela criação de uma aplicação utilizando Java combinado ao kit de desenvolvimento Sphinx devido à rapidez, facilidade de aprendizagem e a amplitude de aplicações possíveis de serem criadas. Baseado nisso foi determinada a criação de uma página de internet que se utiliza dos recursos de reconhecimento de voz, pois este é um tipo de aplicação não utilizado nas páginas web em geral, podendo então se mostrar uma pesquisa útil para futuras aplicações na área. Foi também definida a criação de um modelo vocal próprio, ou seja, o banco de áudio comparativo a ser utilizado pela ferramenta, por questões de didática e de melhor aprendizagem da ferramenta.

1.1 OBJETIVOS

1.1.1 OBJETIVO GERAL

Desenvolver uma página de internet que permita ao usuário a autenticação do seu perfil através de comandos de voz.

1.1.2 OBJETIVOS ESPECÍFICOS

- Aprender sobre as características vocais, a nível acústico e fonético e sua a forma de propagação em formato analógico e digital para captura de dados;
- Conhecer a API de reconhecimento de voz Java Speech assim como suas bibliotecas que serão utilizadas para captura do áudio e o seu processamento;
- Gravar modelos de voz através do software Sphinx para serem utilizados como base de comparação quando a página receber dados de voz de acesso;
- Desenvolver uma página com a linguagem PHP e HTML genérica que permita a realização da autenticação do usuário através de login e senha;
- Realizar testes com grupos de pessoas com senhas e usuários diferentes para definir uma porcentagem de efetividade do projeto;

2 REVISÃO DA LITERATURA

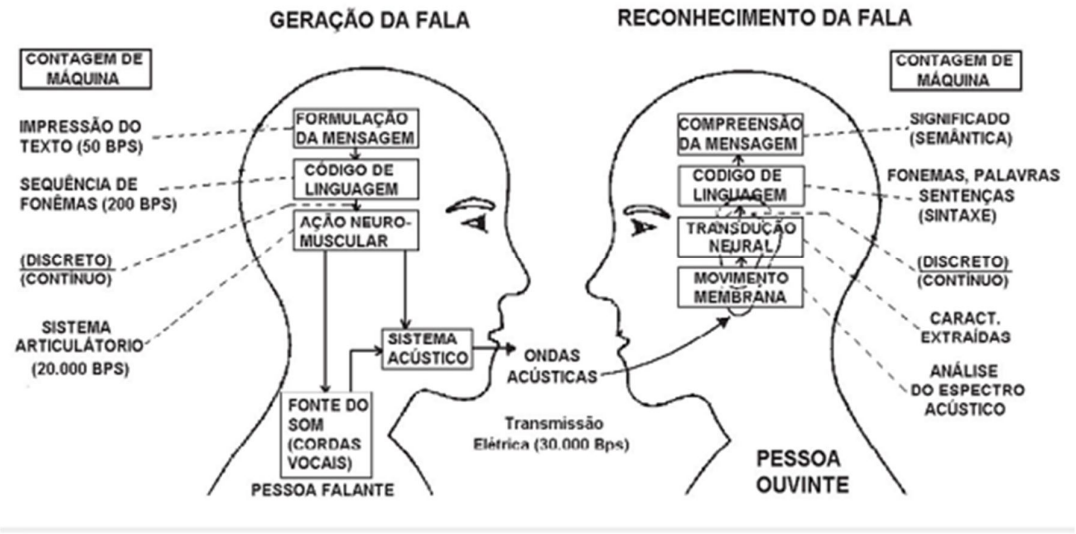
2.1 FUNDAMENTOS PARA O PROCESSAMENTO DE VOZ

Para realização deste trabalho fez se fundamental o estudo do funcionamento de ferramentas, técnicas e definições capazes de proporcionar a criação de um software capaz de reconhecer a voz de um indivíduo, a seguir temos uma análise destas que demonstram e o seu papel específico neste trabalho.

Segundo Furui (2004), a voz não é só a forma mais simples de comunicação entre os seres humanos como também é a mais utilizada, a partir de um sinal de voz é possível descobrir não apenas o significado do que foi dito como várias outras informações, como por exemplo, a posição de onde está sendo enviada a voz, as emoções do locutor, como por exemplo, tristeza ou alegria. Em outras palavras, o conceito de fala é baseado na ideia de se repassar informação a um ouvinte, e, embora para uma pessoa todas essas características sejam captadas de forma natural, os computadores atuais ainda não possuem robustez suficiente para compreender estas inúmeras definições.

Para Denes e Pinson (1963), ao se falar, o propósito principal assumido é de que existe algo ou alguém para ouvi-lo, sendo assim, é notável a relação necessária entre a fala e a escuta, a execução deste processo é realizado através do envio de ondas sonoras propagadas pelos órgãos vocais através do ar até o ouvinte, que capta a mensagem em seus ouvidos e a envia até seu sistema nervoso, responsável pela compreensão, permitindo que a informação enviada seja entendida pelo seu receptor e pelo próprio propagador lhe proporcionando um feedback como demonstrado na Figura 1, este sistema é conhecido como corrente de fala.

Figura 1 - Diagrama de processo de produção e captação da fala.



Fonte: BRESOLIN (2003).

Segundo Bresolin (2003) o processo passo a passo é feito da seguinte maneira, o locutor responsável pela fala formula mensagens em seu cérebro transformando essas ideias em sua linguagem, a partir disso o cérebro gera estímulos para as cordas vocais e musculatura bucal que combinadas emitem o som em formato de ondas acústicas, estes sons emitidos podem ser divididos em dois tipos, os vozeados e não vozeados, que são respectivamente os sons que fazem e não fazem as cordas vocais vibrarem, a vibração das cordas a partir destes sons específicos faz com que a glote, um órgão próximo a laringe se expanda aumentando a passagem de ar, e a partir destas alterações são gerados sinais com harmonia nas frequências das cordas de vibração. Estas ondas são por sua vez captadas pela membrana do tímpano do ouvinte, que através da transdução transforma essa onda em impulsos elétricos e as envia para o cérebro que, primeiramente reúne os fonemas (a divisão da palavra em pequenos segmentos sonoros) tornando-os palavras e, a partir disso, tenta tomar conclusões lógicas a respeito da informação.

2.2 RECONHECIMENTO DE VOZ

De forma semelhante à humana, o reconhecimento de voz para um computador tem como tarefa principal obter a fala em forma de ondas acústicas de seu usuário e produzir através de comparação uma saída correspondente, embora este objetivo não esteja próximo da completa eficiência, são inúmeros os avanços nas ultimas décadas na área, tanto pelo

desenvolvimento da microeletrônica, como o avanço nos algoritmos capazes de solucionar problemas com a captação da voz. (MARTINS, 1997).

A persistência na área é principalmente devida às inúmeras vantagens obtidas através de softwares capazes de trabalharem com ela, dentre algumas, estão à rapidez em comparação a digitação, a não necessidade de treinamento do usuário, pois para um ser humano, a fala é algo intrínseco de sua aprendizagem (ROE, 1994), o não requerimento da utilização das mãos, que possibilita utilização das tecnologias enquanto se move ou caso exista algum problema que o impossibilite de utilizá-las, como por exemplo, portadores de deficiência. (LOUZADA, 2010).

Existem também algumas desvantagens a serem levadas em consideração, existem inúmeros problemas a serem enfrentados na área, como por exemplo, a interferência de ruídos e outras fontes de som no ambiente, que podem ser, por exemplo, a fala de terceiros, tráfego de veículos ou outras fontes de som que dificultam a capacidade de entendimento da aplicação (BRESOLIN, 2003), outro problema também a se destacar são as inúmeras variações de voz de pessoa para pessoa, que podem se mostrar alteradas devido a tratos vocais, sotaque, problemas de dicção, condições físicas e emocionais. (FURUI, 2004).

As utilizações comuns para o reconhecimento de voz são muitas, podendo ser citadas, o acesso à informação, em forma de menus o qual o usuário interage, operações bancarias, controle de opções no celular, reconhecimento e autenticação de usuários através de suas características vocais, entre outras.

O funcionamento geral de um sistema de reconhecimento de voz pode ser definido como a entrada dos sinais que ao entrar no sistema é comparado a modelos previamente gravados, ao encontrar estes padrões, o sistema avalia a semântica e a sintaxe dando sentido ao comando gerando o processamento final e enviando a resposta ao usuário. Para a análise destes dados podem ser utilizados diversos princípios, dependendo assim da necessidade da aplicação, de forma geral, estas diferentes aproximações podem ser divididas em três tipos, que segundo Bresolin (2003) são:

- Acústico-fonética.
- Reconhecimento de padrões.
- Inteligência artificial.

A aproximação fonética procura a partir das características distintas das unidades fonéticas um conjunto de propriedades dos sinais emitidos, primeiramente dividindo estes sinais onde cada propriedade é representada por uma classe ou fonética. (BORGES, 2009). Após isso esta sequência é comparada na tentativa de validação destes sons.

A aproximação a partir do reconhecimento de padrões tenta reconhecer apenas os padrões de sinais sem necessariamente buscar pelo seu significado, podendo ser dividido em fase de treinamento e comparação, ele busca a partir dos dados previamente inseridos padrões repetíveis dentre as propriedades dos sinais recebidos. (BORGES, 2009). Este tipo de reconhecimento é o mais utilizado devido a sua praticidade e alto desempenho.

A aproximação por Inteligência Artificial (I.A.) é uma forma combinada das duas ultimas, tendo como conceito principal a adaptação e aprendizagem constante. A I.A. usa redes neurais para compreender as ligações entre os eventos fonéticos (LOUZADA, 2010) assim como para verificação das classes de sons.

Não obstante, existem também algumas definições de termos necessárias a se definir para que se possa compreender o processo de criação de um sistema de voz (MARANGONI, 2006). Os principais sendo os fonemas, modelos acústicos, expressões, gramática, e treinamento.

Os fonemas são constituídos pela sonoridade em sua menor unidade, é valido se atentar que essa lógica é voltada ao som, tal que, por exemplo, sílabas diferentes podem ser foneticamente iguais, como por exemplo, “ca” e “ka”. (SEARA, 2011).

O modelo acústico é uma fonte de dados vocais, criada para ser utilizada como base de comparação, servindo como paradigma da forma como uma palavra deve ser pronunciada ao ser recebida pelo reconhecedor de voz.

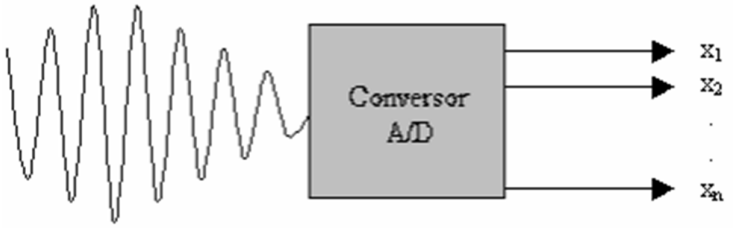
As expressões são as sequencias de voz entre os períodos de silêncio, podendo ser palavras, ou frases inteiras.

O treinamento é o processo o qual o sistema passa a identificar palavras fora de seu modelo base (FURUI, 2004), como por exemplo, passando a aceitar uma mesma palavra de uma pessoa que possui uma diferente pronuncia.

2.3 O PROCESSAMENTO DOS SINAIS DE FALA

A fala é um conjunto de vibrações acústicas que se manifesta em nosso ambiente de forma analógica, já os computadores trabalham com a informação em formato digital, para que seja possível então que uma aplicação entenda uma fala é necessário que ela seja convertida, quando um locutor fala no microfone, este age como um transdutor que transforma esses sinais acústicos em impulsos elétricos que serão traduzidos, essa conversão é composta de três etapas, sendo elas a amostragem, a quantização e a normalização como são mostradas na Figura 2. (FERNANDES, 2009).

Figura 2 - Conversão de sinal analógico em digital.



Fonte: VALIATI (2000)

A amostragem é a extração de amostras de um sinal recebido de forma contínua em determinados períodos de tempo, esse sinal pode ser calculado em hertz, sendo cada 1hz uma amostra dentro deste segundo, quão maior o numero de hertz, maior é o numero de amostragens dentro de um segundo, capitando assim maior informação desse sinal, no entanto captá-lo totalmente não é necessário, pois com um número suficiente de amostragens, os valores não calculados podem ser estimados, uma boa taxa de amostragem está em 10000hz. (VALIATI, 2000).

A quantização é a amplitude do sinal amostrado nos intervalos determinados de tempo, sendo esta calculada em bits (VALIATI, 2000), por exemplo, uma quantização de 8 bits equivale a 2^8 quantizações, quanto maior o numero de bits maior a resolução, porém maior a quantidade de dados a serem processados. Após a quantização é também realizado um processo de codificação, ou seja, a transformação desse sinal em código binário.

A normalização é a etapa que realiza a tentativa de regularizar a intensidade do sinal recebido das amostras (VALIATI, 2000), quando um sinal recebido é maior que o pré-definido a normalização acontece, restringindo estes valores.

2.4 MODELOS PARA PROCESSAMENTO DO SOM

Dentre os modelos mais utilizados para o reconhecimento de voz estão os algoritmos baseados em métodos estatísticos (YNOGUTI, 1999), sendo os de maiores destaques a utilização de redes neurais artificiais e os modelos ocultos de Markov, existem também aplicações hibridas de ambos que tentam extrair as vantagens de cada uma das técnicas.

2.4.1 UNIDADES FUNDAMENTAIS DE RECONHECIMENTO

Em sistemas que reconhecem poucas palavras geralmente se utiliza palavras completas como unidades fundamentais a serem reconhecidas, para que um sistema esteja apto a reconhecer uma palavra, é necessário que ele seja treinado múltiplas vezes para reconhecer cada uma dessas palavras, já em sistemas que possuem quantidades mais expressivas de palavras é necessário buscar uma diferente abordagem, pois realizar o mesmo treinamento para cada uma dessas diversas palavras se tornaria inviável e custoso. (YNOGUTI, 1999).

Utiliza-se então o treinamento a partir das subunidades fonéticas, que podem ser os fonemas, sílabas, demissílabas, e etc., essa divisão torna o processo de treinamento muito mais rápido pela possibilidade de reutilização dessas subunidades em palavras distintas (YNOGUTI, 1999), por exemplo, digamos que seja necessário que um sistema reconheça as palavras “casa”, “caca” e “saca”, ao realizar o treinamento por palavra e por exemplo sendo necessário utilizar 5 gravações de voz para cada treinamento, seriam necessários 15 treinos, já utilizando um treinamento de subunidades fonéticas, poderiam ser treinadas as silabas “sa” e “ca” 5 vezes cada, totalizando 10 treinos e podendo serem recombinadas para criar as mesmas três palavras.

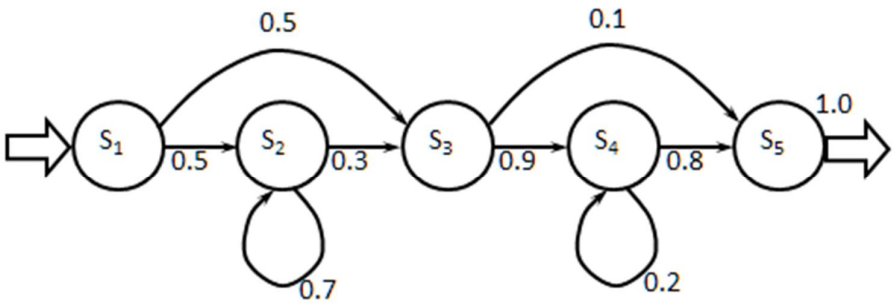
Embora o exemplo demonstre utilização de sílabas o método mais utilizado é o de concatenação fonética, ou seja, a representação sonora das unidades de fala (KAFKA, 2002), esses fonemas geralmente são armazenados em conjuntos de unidades, que podem ser difones, trifones ou polifones, por exemplo, o difone “da” é a combinação do som da letra “d”, concatenado ao som da letra “a”, funcionando de forma semelhante para os trifones e polifones. A partir deste armazenamento é então possível a utilização de técnicas que saibam diferenciar estes sons e a partir deles definir quais palavras estão sendo ditadas.

2.4.2 MODELO OCULTO DE MARKOV

O modelo oculto de Markov é uma técnica não utilizada apenas no reconhecimento de voz, mas em diversas outras áreas, uma das suas principais características é que ela não possui memória, tendo todas as outras ações resumidas no valor atual do seu processo (SILVA, 2009), foi desenvolvida na década de 60 e apenas a partir da década de 70 foi introduzida na área de reconhecimento de voz, sendo hoje em dia um dos recursos mais utilizados da área.

O modelo de Markov é uma máquina de estados que possui transições ligadas a um processo determinado por probabilidades de transição para cada estado como demonstrando na Figura 3.

Figura 3 - Exemplo de modelo oculto de Markov.

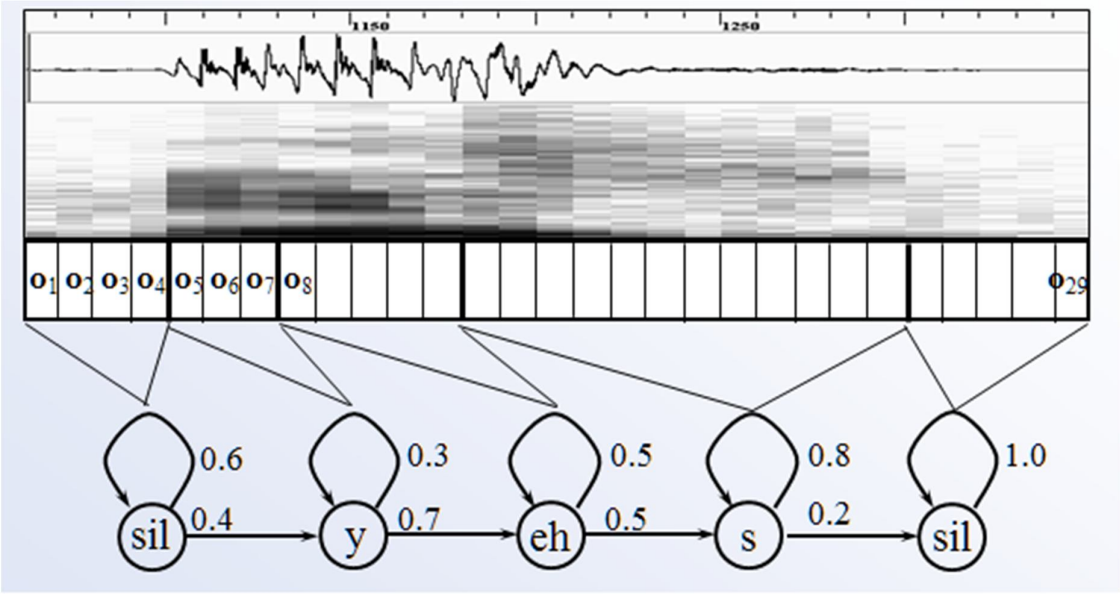


Fonte: HOSOM (2011).

Outros fatores inerentes do modelo são a transição entre os estados, que é realizada em amostras de tempo pré-determinadas, podendo estar em apenas um estado por vez, ao contrário das máquinas de estado típicas, as ações no modelo oculto de Markov não ocorrem durante as transições, mas sim durante cada estado. (HOSOM, 2011).

Em uma aplicação voltada ao reconhecimento de voz, no geral cada fonema é representado por um modelo oculto de Markov, e sua ocorrência durante o treinamento define a sua probabilidade de mudança de um estado para uma de suas possibilidades, isso pode ser demonstrado no exemplo da Figura 4.

Figura 4 - Exemplo prático de Modelo oculto de Markov



Fonte: HOSOM (2011).

O exemplo demonstra a gravação de um treinamento de um modelo oculto de Markov para a palavra “yes”, sendo os períodos O1 até O29 a divisão em amostragens do som, ao observar o modelo percebe-se a transição inicial “sil”, que nada mais é que a ausência de som, seguida dos fonemas correspondentes e novamente do silêncio, que finaliza a expressão, é possível também perceber que aos sons de maior ocorrência a probabilidade atribuída a ele de acontecer novamente é aumentada em relação aos sons mais discretos.

2.5 JAVA

A linguagem utilizada para a criação deste projeto, sendo orientada a objetos, em outras palavras, uma linguagem que através da abstração busca assimilar características ao mundo humano, os objetos propriamente ditos são obtidos através de suas classes, que servem como moldes para a criação. As classes por sua vez são definidas por variáveis de dados e por métodos, que são as ações e funções que as compõe. (DEITEL, 2010).

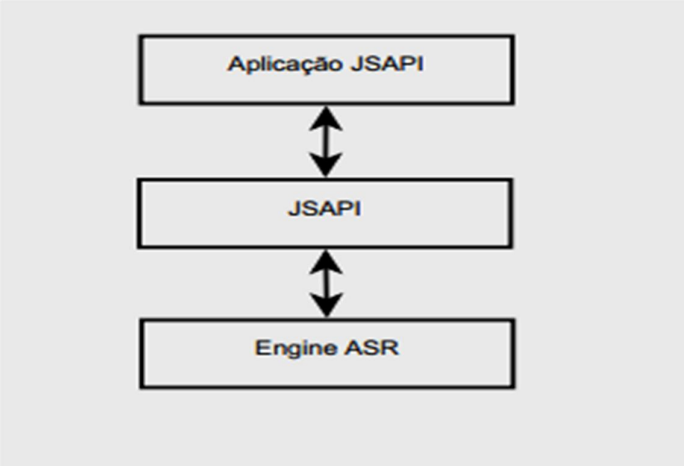
O Java possui uma rica coleção de bibliotecas, que possuem implementações que facilitam a criação de novas aplicações. Dentre elas o enfoque está para a utilizada neste projeto, a API Java Speech (JSAPI), que possibilita a incorporação das tecnologias de voz no ambiente Java, nesta biblioteca estão classes que serão responsáveis pela comunicação entre a página e o software de reconhecimento de voz responsável pela captura dos dados. Outra classe a ser utilizada é Java AWT, que possui ferramentas que auxiliam na manipulação da interface a ser utilizada.

2.5.1 JAVA SPEECH API

Esta API é capaz de incorporar tecnologias de voz ao ambiente Java e aos softwares criados a partir desta, surgiu da parceria de grandes empresas como Apple Computer, AT&T, IBM entre outras. (BATISTA, 2011).

Ela é independe de plataformas, e tem suporte para criação de sistemas de comando, controle e síntese de voz. Em termos gerais, o JSAPI é utilizado como interface entre o programador e software de reconhecimento de voz, possibilitando manipula-la através desta, a Figura 5 demonstra a arquitetura de camadas e a forma como são trocadas as informações entre a aplicação.

Figura 5 - Arquitetura de aplicação utilizando o Java Speech.



Fonte: BATISTA (2011).

O JSAPI trabalha capturando os dados recebidos pelo motor de reconhecimento de voz e processa esses dados e os envia para a aplicação que é a responsável pelo fornecimento do resultado.

Em outras palavras, embora não seja possível a partir desta API a captação da voz do usuário, a partir dela poderão ser desenvolvidos o processamento do áudio e a criação de um projeto de gramática, ou seja, as palavras capturadas pelo reconhecedor de voz, assim como gerar o resultado a partir do processamento da mesma.

2.5.2 JAVA AWT

O Java AWT (Abstract Window Toolkit) é um pacote contendo inúmeras ferramentas para o desenvolvimento de interfaces e para criação de gráficos e imagens.

Dentre estas, existe a classe Robot, que tem a finalidade de gerar entradas nativas de teclado e mouse com o objetivo de automação. (ORACLE, 2015). Os métodos associados a ela e mais utilizados para automatização são os seguintes:

- keyPress: simula o pressionamento de qualquer tecla do teclado;
- keyRelease: remove o pressionamento da tecla;
- mousePress: realiza um clique do mouse em um píxel específico da tela;
- mouseRelease: remove o pressionamento do botão do mouse;

2.5.3 JSGF

Os sistemas de reconhecimento de voz atuais permitem que os computadores escutem palavras ditas pelos usuários e determinem o que foi dito, no entanto não existe tecnologia atualmente que permita um reconhecimento de fala não delimitado (HUNT, 2000), devido a isto existe a necessidade dos reconhecedores utilizarem gramáticas.

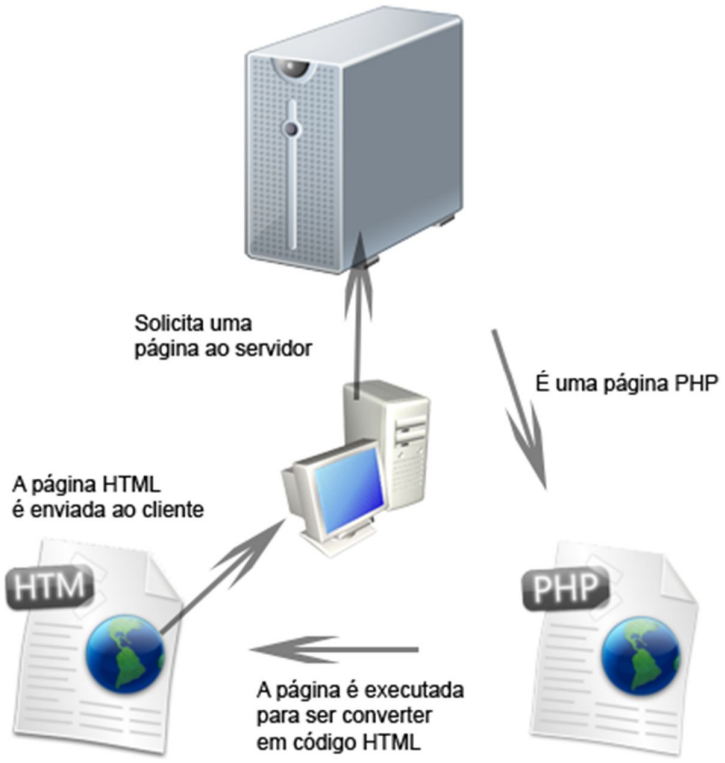
O JSGF (JSpeech Grammar Format) define um formato de regras gramaticais em representação textual que pode ser compreendido tanto pelo desenvolvedor quanto pelo computador.

O JSGF define um conjunto de expressões que um locutor pode vir a dizer, essas expressões são definidas através da utilização de chaves, cada expressão possui um conjunto de símbolos terminais, nesse caso, as palavras contidas na gramática. Por exemplo, a expressão <abrir porta> = abrir | fechar;, define que o comando abrir porta, possui as opções abrir e fechar, tendo então as opções gramaticais divididas pela barra.

2.6 PHP

O PHP é uma linguagem criada em 1994 por Rasmus Ledorf, voltada para a criação de sites dinâmicos, possibilitando assim a interação com o usuário (WELLING e THOMSOM, 2009), linguagem esta que é executada no servidor e retornado ao cliente apenas o código HTML, conforme demonstrado na Figura 6.

Figura 6 - Modelo de funcionamento da linguagem PHP



Fonte: ROCHA (2012).

Devido a ter sua utilização focada no lado do servidor, o enfoque da utilização será a coleta de dados do formulário para autenticação do usuário, realizando a conexão com o banco de dados e comparando os dados para validação, para isto sendo utilizado o comando "$_POST".

Para que não haja autenticação indevida, também é utilizado o comando "$_SESSION" que impede que um usuário adentre uma página sem estar autenticado.

2.7 SPHINX4

É um software de síntese e reconhecimento de voz produzido pela empresa CMU Sphinx de licença BSD, em outras palavras, código aberto, possui um vocabulário amplo que aumenta constantemente assim como possui ferramentas que permitem a criação de sistemas de reconhecimento de voz (SPHINX, 2014), compatível com java e linguagem C/C++ este servirá como software de reconhecimento de voz para a aplicação, embora em inglês é possível utilizar suas ferramentas de treinamento para adapta-lo ao reconhecimento de qualquer vocabulário. (SPHINX, 2014).

Para criação de uma aplicação Java utilizando o Sphinx pode se utilizá-lo em forma de API, sendo então apenas necessária a manipulação de suas classes de forma conveniente, dentre as mais utilizadas estão as seguintes:

- Frontend: é utilizada para manipular a entrada de informação através de, por exemplo, o microfone;
- Recognizer: é utilizada para inicializar e alocar recursos para o reconhecimento dos dados recebidos;
- ConfigurationManager: serve para configurar e selecionar os arquivos que serão utilizados, como o arquivo com a gramática e o modelo acústico;
- Result: utilizada para receber os resultados processados em uma string.

Existe também o arquivo de gramática com a extensão ".gram", responsável pelas palavras capazes de serem geradas pela aplicação através de um modelo feito utilizando JSGF (Java Speech Grammar Format), e o arquivo de configuração que contém o caminho do modelo acústico e da gramática, e configurações relevantes quanto a forma de processamento do áudio, sendo este em linguagem de marcação XML.

2.7.1 SPHINXTRAIN

Sphinxtrain é uma ferramenta auxiliar do software Sphinx4 e tem como objetivo realizar o treinamento de modelos acústicos pelo próprio usuário, essa ferramenta é útil caso seja necessário desenvolver um treinamento fora da linguagem a qual o Sphinx, que é em inglês, ou então caso seja necessário desenvolver um dicionário diferenciado (SPHINX, 2014).

A forma mais fácil de criação de um modelo acústico é através da criação de um corpus de áudio, ou seja, uma base de áudios gravados e transcritos, também é necessário um dicionário fonético que associará os fonemas as passagens de áudio e um modelo linguístico, uma lista de palavras que validará as palavras resultantes da gramática.

2.8 BANCO DE DADOS

Devido a necessidade de um local de armazenamento das informações modelo comparativas para a análise de voz assim como os dados de login dos usuários, torna-se necessária a utilização dos recursos assim como uma breve explicação de alguns conceitos

utilizados para que se possa dar clareza aos objetivos. O projeto de um banco de dados pode ser dividido em três partes (HEUSER, 1998):

- Modelagem conceitual.
- Projeto lógico.
- Projeto físico.

Na modelagem conceitual são definidos os conceitos num modelo entidade-relacionamento, aqui são colhidas as necessidades em termos de armazenamento. Ao se realizar o projeto lógico é definida a forma a qual será implementado o banco de dados em um sistema especifico. O modelo físico é a fase em que se enriquece o banco com detalhes que influenciam no desempenho do banco de dados, obtendo-se assim a forma final.

2.9 SQL

A linguagem SQL (Structured Query Language) é uma linguagem criada especificamente para manipulação e interação em um banco de dados especifico, criada pela IBM e sendo lançado comercialmente a partir de 1979 (ORACLE, 2014), onde o objetivo principal da linguagem é o armazenamento de dados.

Segundo Heuser (1998) o funcionamento organizacional do SQL é em formato de tabelas a qual a informação é guardada em seus nodos, o comando para a criação de uma tabela é definido por CREATE sendo então definidos os campos e os seus tipos de dado, é importante definir tais tipos com cautela pois essa escolha poderá limitar a futura inclusão de itens específicos em seus devidos campos.

A criação é definida pela inclusão de novos itens em uma tabela a partir do comando INSERT. A exclusão é definida pela deleção de itens indesejados na tabela, o comando para tal função é o DELETE. A alteração é definida pela necessidade de se atualizar um dado já adicionado a tabela sem exclui-lo, o comando utilizado para isto é o UPDATE. (MONTEIRO, 2010).

2.10 DESENVOLVIMENTO DE SOFTWARE

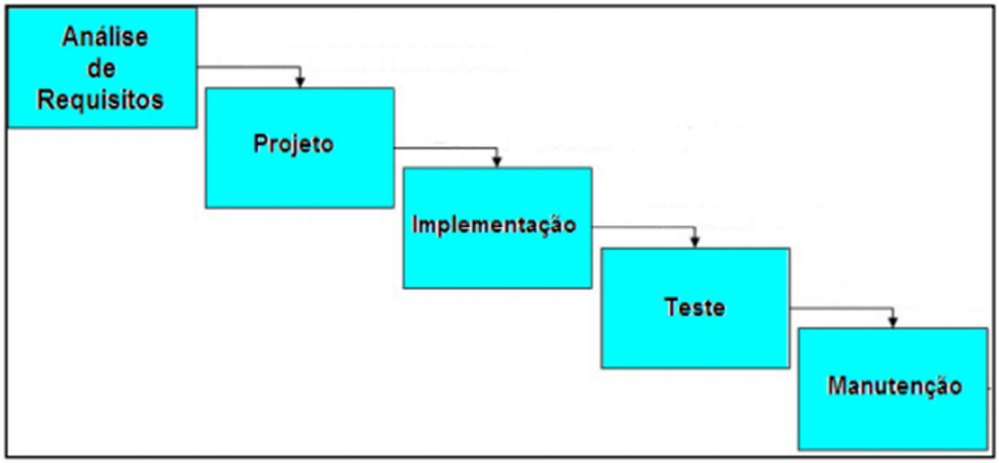
A vida de um software pode ser dividida em ciclos, esta divisão tem como objetivo organizar o sistema para que ele alcance seus objetivos determinados na sua área de atuação, a fase de testes pertence portanto ao ciclo inicial, o de desenvolvimento (FUKUMORI, 2008).

Os testes consistem na realização de avaliações na aplicação através de metodologias pré-estabelecidas com o objetivo de eliminar erros não percebidos durante o desenvolvimento,

e que caso permaneçam no software podem causar problemas nas próximas etapas do desenvolvimento e tendo de retornar etapas, ou durante o próprio uso da aplicação final, em ambos os casos causando prejuízos.

Dentre os modelos de desenvolvimento, existe o modelo cascata, que divide as etapas de desenvolvimento de forma sequencial, como mostrado na Figura 7:

Figura 7 - Modelo de desenvolvimento cascata



Fonte: FUKUMORI (2008).

2.10.1 UML

O padrão UML (Unified Modeling Language) é uma linguagem que apresenta um padrão que define o formato de um produto através da sua diagramação, facilitando uma visão do que está sendo criado para todos os envolvidos no projeto (FERNANDES, 2010), os diagramas mais utilizados são os:

- Casos de uso: Descrevem de maneira visual os requisitos funcionais do software, como eles interagem entre si e com entidades alheias ao software, como por exemplo, os usuários;
- Diagrama de classe: Serve para identificar o relacionamento entre as classes e as suas dependências;
- Diagrama de atividades: Tem como objetivo demonstrar o fluxo de atividades dos processos mostrando suas dependências

2.10.2 TESTES

Para assegurar que todos os componentes de um software sejam testados o processo de testes é dividido em várias etapas e pode ser realizado de diversas formas, dependendo da

necessidade do desenvolvedor, no geral, as formas mais utilizadas são os testes de unidade, que testam os menores componentes do software de diversas maneiras garantindo sua funcionalidade, e os testes de integração (FUKUMORI, 2008), que são realizados a partir da união de todas as unidades programadas, analisando então como elas se comportam trabalhando interligadas.

Tanto no teste de unidade quanto de integração, são utilizadas as técnicas de teste funcional, também conhecida como teste de caixa preta, que serve para verificar a funcionalidade do sistema, ou seja, se ele cumpre com todos os requisitos do sistema, também é utilizado o teste de estrutura, ou teste de caixa branca, que é voltado para o código, e checa se ele possui algum comprometimento lógico que pode vir a causar problemas.

3 TRABALHOS CORRELATOS

Nesta seção são apresentados trabalhos que de forma semelhante realizaram estudos na área de reconhecimento de voz proposto através deste trabalho, embora existam vários outros, foram selecionados os seguintes projetos por se assemelharem no formato da pesquisa e de sua aplicação: “Reconhecimento de voz para palavras isoladas” (SILVA, 2009); “Reconhecimento de voz para comandos de direcionamento por meio de redes neurais” (VALIATI, 2000) e “Parakeet: A Continuous Speech Recognition System for Mobile Touch-Screen Devices” (VERTANEN, 2009).

3.1 RECONHECIMENTO DE VOZ PARA PALAVRAS ISOLADAS

O trabalho de Silva (2009) apresenta um sistema de reconhecimento de palavras isoladas baseado nos Modelos ocultos de Markov, sendo este capaz de reconhecer dígitos de 0 a 9 e as palavras “sim” e “não”, com a possibilidade de se expandir este vocabulário caso necessárias. O sistema é dividido em quatro blocos, a aquisição do sinal da fala, o pré processamento, a extração dos parâmetros e utilização do modelo oculto de Markov na obtenção do resultado final. Para a criação do sistema foi utilizada a ferramenta MATLAB. Para a elaboração dos resultados foi criada uma base de dados a partir de 13 locutores, sendo 10 do sexo masculino e 3 do feminino para então a realização de testes. Através dos experimentos realizados foram obtidos resultados expressivos como demonstrados na Figura 8:

Figura 8 - Resultados dos experimentos de Silva.

	Taxa de acerto(%)
0 (zero)	100
1 (um)	98,48
2 (dois)	97,73
3 (três)	83,17
4 (quatro)	100
5 (cinco)	88,61
6 (seis)	90,13
7 (sete)	95,41
8 (oito)	99,22
9 (nove)	100
Sim	96,21
Não	99,24
Média	95,68

Fonte: SILVA (2009).

3.2 RECONHECIMENTO DE VOZ PARA COMANDOS DE DIRECIONAMENTO POR MEIO DE REDES NEURAIS

O trabalho de Valiati (2000), visou o processamento de sinais de fala através da utilização de técnicas de processamento de sinais e de redes neurais, para a o treinamento e classificação foram utilizadas redes do tipo Backpropagation e Fuzzy ARTMAP. Através da coleta de diversas amostragens de voz de usuários distintos foi criado um sistema com alto índice de acerto. O sistema é capaz por sua vez da realização de instruções através de um vocabulário fechado, que reconhece as palavras: “esquerda”, “direita”, “siga”, “pare” e “recue”. Após a realização dos treinamentos do sistema criado, foram realizados testes para definir a eficiência alcançada pelas redes criadas demonstradas na Figura 9:

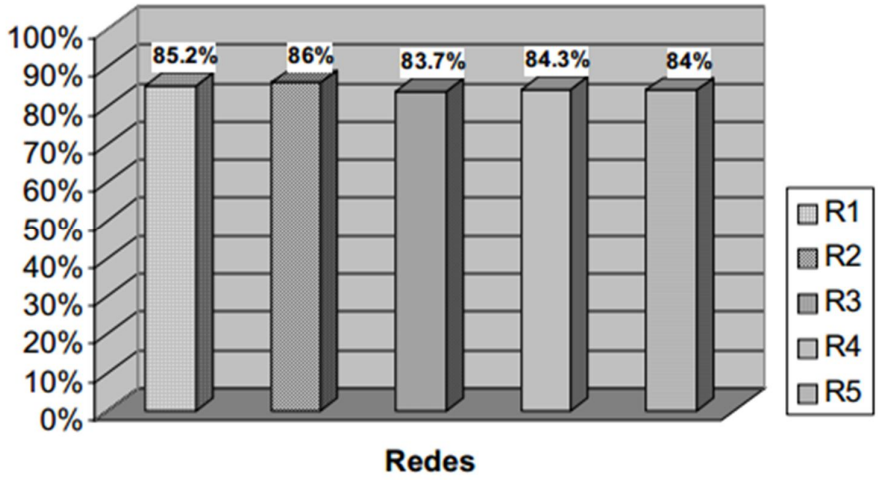
.
Figura 9 - Redes criadas para teste de reconhecimento.

Especificações / Redes	número de neurônios na camada intermediária	EMQ alcançado	Número de iterações
R1	30	0,00301818	2994
R2	20	0,00462638	2977
R3	60	0,00563109	2509
R4	40	0,00464447	3000
R5	10	0,00403261	3000

Fonte: VALIATI (2000).

Cada uma das redes por sua vez possuindo um número de neurônios, erro médio quadrado e número de iterações distinta afim de buscar um melhor índice de resultados, que mostraram um desempenho geral apropriado mostrados na Figura 10:

Figura 10 - Resultados dos testes das redes criadas.



Fonte: VALIATI (2000).

3.3 PARAKEET: A CONTINUOUS SPEECH RECOGNITION SYSTEM FOR MOBILE TOUCH-SCREEN DEVICES

Vertanen (2010) trabalha com o desenvolvimento de um sistema de reconhecimento de voz continua para equipamentos móveis (tablets e celulares), focado na transcrição de mensagens de textos de entrada, como envio de sms e comandos variados utilizando inglês americano, para criação do sistema foi utilizado o PocketSphinx (versão móvel do CMU Sphinx utilizado neste trabalho), e realizados testes de entrada em um dispositivo tablet com sistema operacional Linux, tendo as entradas de voz recebidas através de um microfone bluetooth. Foram realizados testes com quatro participantes mostrando taxa de assertividade de 96%, no geral apresentando problemas em ambientes abertos devido a ruídos adversos.

4 METODOLOGIA

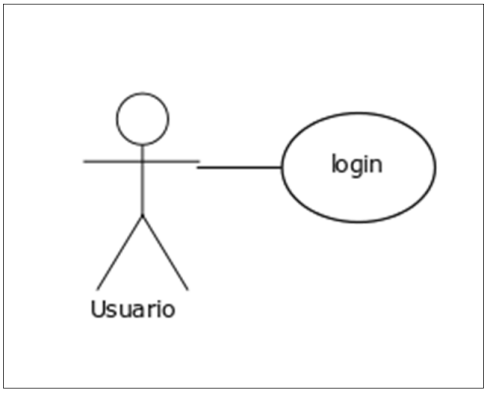
A metodologia é parte chave da execução deste trabalho, pois é uma explicação minuciosa de todas as ações geradas a partir do processo de pesquisa (KAUARK, 2010).

A princípio, o foco principal deste trabalho iniciou-se na hipótese de criação de um sistema capaz de reconhecer a voz de um usuário e utiliza-la validar a entrada em um site, assim melhorando sua acessibilidade.

Para checar a possibilidade de realização e caso viável, os processos e as ferramentas a serem utilizadas para tornar isso possível, foram realizadas pesquisas em sites especializados, livros e materiais relacionados a processamento de dados e reconhecimento de voz que permitiram a criação de um conceito o qual pôde ser utilizado para esse fim assim como as ferramentas que combinadas puderam ser utilizadas na busca pelo resultado final.

Para facilitação da abstração da aplicação foram desenvolvidos casos de uso, de classe, e de atividade, abaixo, sendo demonstrando na Figura 11 o caso de uso.

Figura 11 - Diagrama de caso de uso



Fonte: Elaborada pelo autor.

Como não existe processo algum além da autenticação do usuário na página e não existe nenhuma ação posterior a esta, o diagrama de caso de uso demonstra apenas isso.

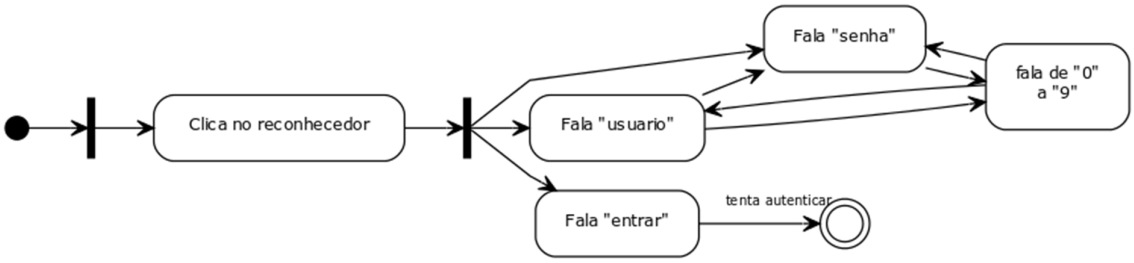
Figura 12 - Diagrama de caso de classe.

Fonte: Elaborada pelo autor.

O diagrama de caso de uso da Figura 12 descreve a relação entre o reconhecedor que preenche os dados do usuário.

O diagrama de atividades da Figura 13 demonstra o passo a passo do programa ao ser executado:

Figura 13 - Diagrama de atividades.



Fonte: Elaborada pelo autor.

Foi desenvolvido um banco de dados utilizando a ferramenta phpMyAdmin devido a sua praticidade e por já vir acompanhada da ferramenta de desenvolvimento XAMPP que seria utilizada mais adiante para os testes de código PHP. Este banco foi criado com o nome de “bancotcc” e nele criada uma tabela nomeada “usuário” com três campos, sendo o campo “id” uma chave-primária auto incrementada, e os campos “nome” e “senha”, ambos do tipo inteiro com capacidade para até 10 caracteres cada, para realização dos testes iniciais foram criados apenas dois usuários distintos de 4 caracteres tanto para o nome quanto para a senha como mostrado na Figura 14:

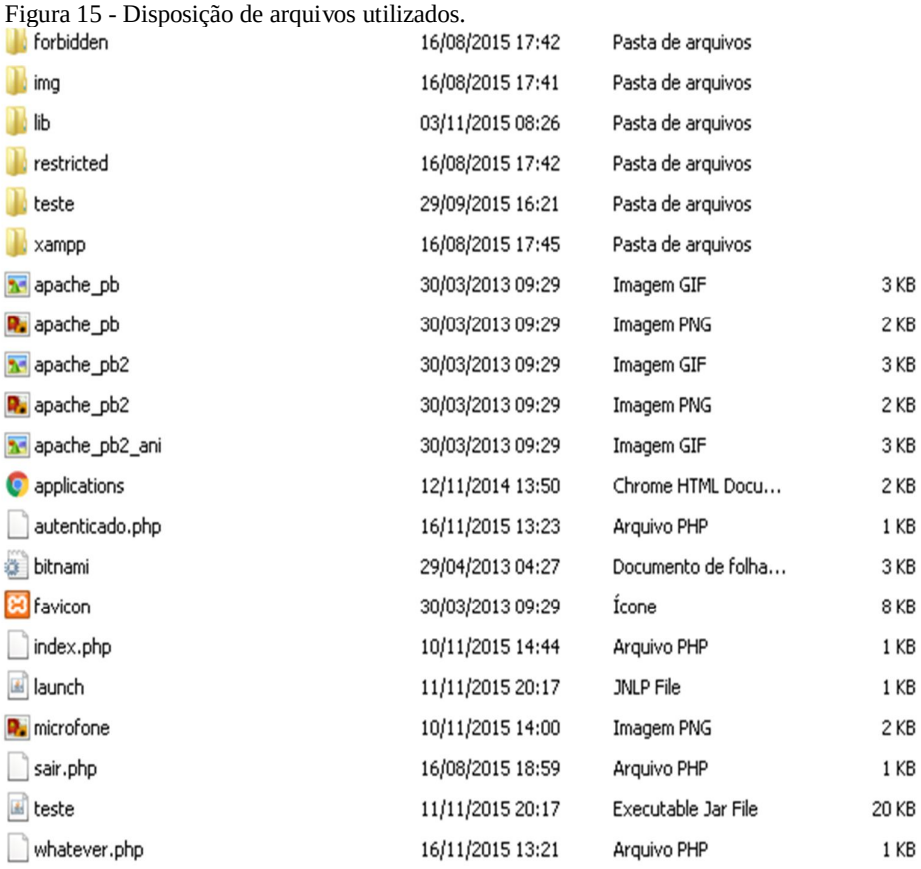
Figura 14 - Banco de dados de teste.

Fonte: Elaborada pelo autor.

Após a criação do banco de dados de teste foi criada uma página utilizando técnicas de CSS para formatação centralizada e tabulação dos elementos em tela, a página é composta

pelas páginas “index.php”, que possui a estrutura de login, com o formulário e o botão de ativação do sensor de voz, a página “autenticado.php” é a página a qual é atingida ao se validar uma sessão, a estrutura “sair.php” possui um código para destruição de sessão caso o usuário deseje sair da página. Os arquivos java por sua vez são o “launch.jnlp” que é utilizado para baixar a aplicação java no navegador, sendo o arquivo “teste.jar” o programa a ser baixado, a pasta “lib” possui todas as bibliotecas utilizadas pelo arquivo, como o dicionário utilizado pelo reconhecimento de voz e outras API’s que foram utilizadas na criação do software.

Figura 15 - Disposição de arquivos utilizados.

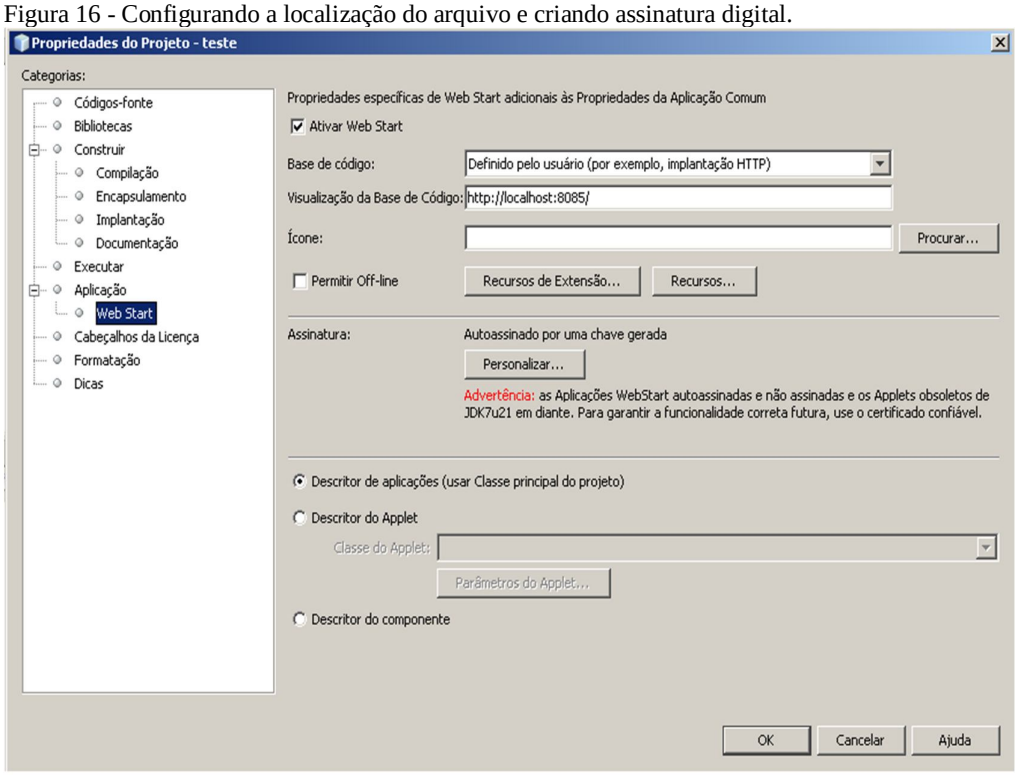


Fonte: Elaborada pelo autor.

Ao desenvolver a aplicação Java foi criada uma aplicação nomeada teste e definida como uma aplicação Java Web Start, para ser visualizada na própria máquina local na porta 8085, devido a restrições de segurança, existe a necessidade de criação de uma chave de assinatura para que se consiga implementar este código na página posteriormente, o papel deste assinatura digital é garantir a procedência do documento e proteger os usuários contra

crimes digitais diversos, por se tratar de uma aplicação para testes pessoais sem nenhum objetivo comercial foi optado pela criação de uma chave gerada auto assinada, que embora não possui uma validade de 6 meses e pode ser utilizada para fins de pesquisa, o processo de criação de uma chave auto assinada e seu local de visualização pode ser feito no ambiente de desenvolvimento utilizado, Netbeans, através da aba de aplicação Web Start, como mostrado na Figura 16.

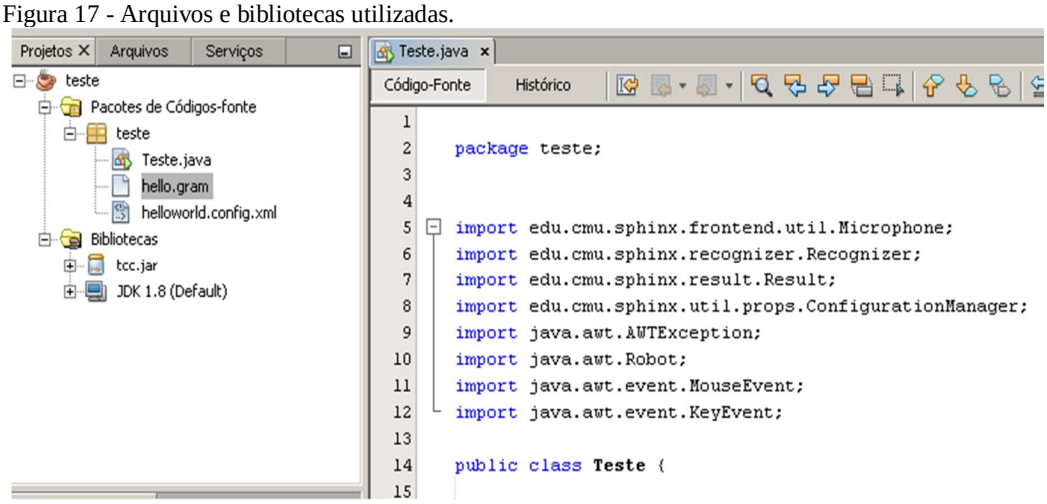
Figura 16 - Configurando a localização do arquivo e criando assinatura digital.



Fonte: Elaborada pelo autor.

A criação do aplicativo foi divida na criação de 3 arquivos, o arquivo base é foi nomeado de “teste.java”, nele foram importados as bibliotecas do Sphinx que servem para a captação, a filtragem do ruído e recuperação do resultado, existe também entre elas a biblioteca de configuração do Sphinx, que serve para indicar os arquivos de gramática e modelos vocais que serão utilizados na aplicação. Outra biblioteca utilizada é a AWT que serviu para emular movimentações do mouse e teclado, a fim de recriar as ações requisitadas pelo locutor durante a utilização do aplicativo. Para maior facilidade de manipulação destas bibliotecas, foi realizada uma alteração no arquivo de construção do aplicativo, compilando todos estes no arquivo “tcc.jar”.

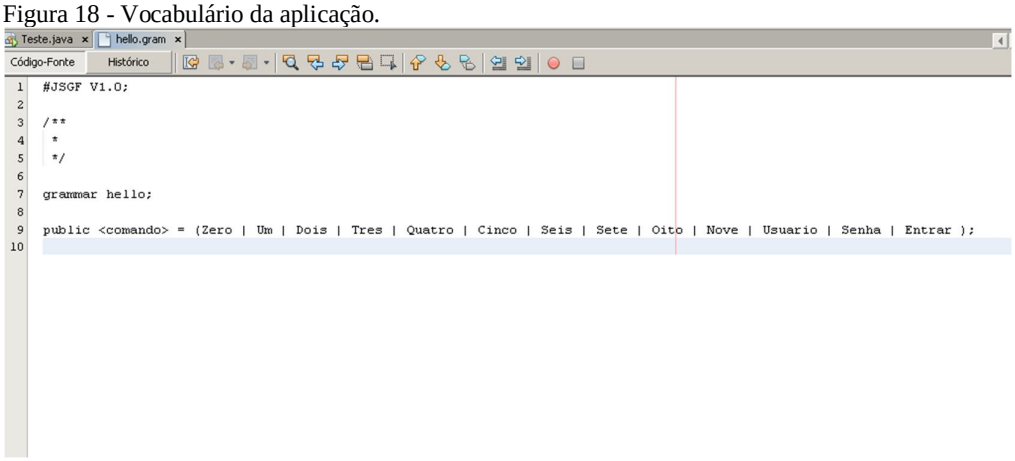
Figura 17 - Arquivos e bibliotecas utilizadas.



Fonte: Elaborada pelo autor.

O arquivo “hello.gram” é um arquivo que possui a gramática utilizada pelo programa, neste caso, o token “comando” pode gerar os números de 0 a 9, e as palavras “usuário”, “senha” e “entrar”, elas são representadas em formato JSGF.

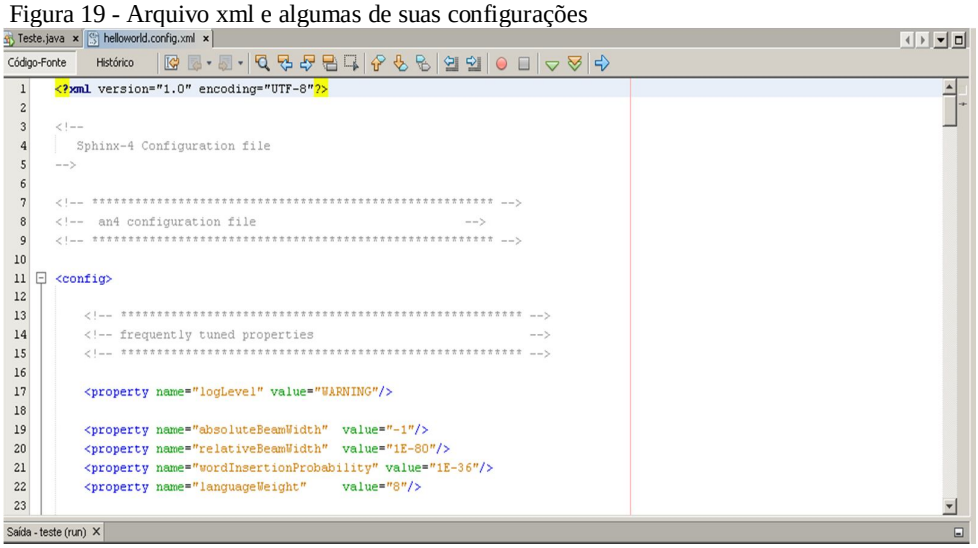
Figura 18 - Vocabulário da aplicação.



Fonte: Elaborada pelo autor.

O arquivo helloworld.config.xml mostrado na Figura 19, é um arquivo de marcação que é utilizado na localização dos arquivos utilizados para o reconhecimento, como a localização do arquivo de áudio e a gramática (nesse caso o “hello.gram”), nele também é possível alterar a forma como é captado o áudio, e alterar algumas técnicas de processamento de ruído na tentativa de se obter melhores resultados.

Figura 19 - Arquivo xml e algumas de suas configurações



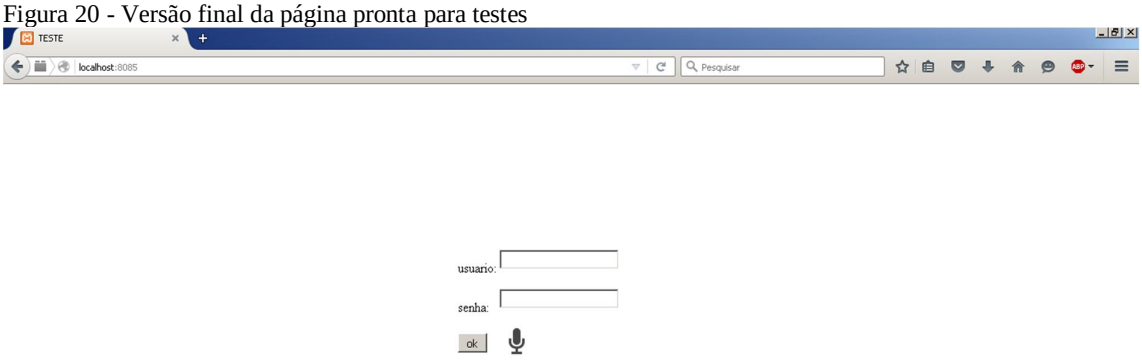
Fonte: Elaborada pelo autor.

Após a criação do aplicativo foi preciso alimentá-lo com um modelo acústico, para isto foi criado uma base com o programa SphinxTrain, de forma simples, bastando apenas, executar o programa e selecionar os arquivos de áudio utilizados, a voz gravada e o arquivo de texto com a gramática, o resultado final é um arquivo “WSJ_8gau_13dCep_16k_40mel_130Hz_6800Hz.jar” que também foi incluso a biblioteca, e selecionado como base no arquivo “helloworld.config.xml” terminando, assim, a etapa de desenvolvimento.

5 RESULTADOS

Após a configuração destes três arquivos, foi desenvolvido na classe “teste.java” um código que em loop recebe a voz do usuário e a identifica filtrando em 3 opções.(“usuário”, “senha” e “entrar”) sendo estas 3 posições especificadas, em pixels, e clicadas na tela, após a realização do comando. Caso o locutor, por exemplo, decida dizer “usuário”, o programa realiza um clique no campo usuário e libera para que o locutor dite, então, o seu numero de usuário, o mesmo ocorre para senha e caso diga entrar é então realizada a tentativa de autenticação.

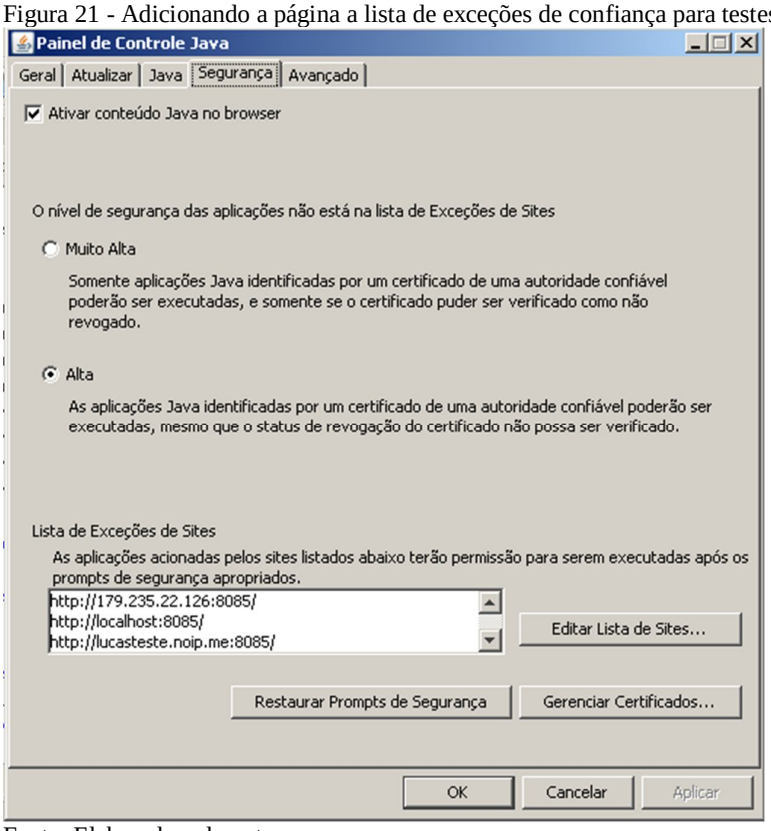
Figura 20 - Versão final da página pronta para testes



Fonte: Elaborado pelo autor.

Com a página pronta para testes, foi alimentado o banco de dados com 10 combinações de usuário e senha de números distintos e realizado um teste de eficiência com 4 pessoas, sendo elas três homens e uma mulher. Para que os usuários testassem do seu próprio local foi criado um servidor para acesso remoto através do XAMPP no IP 179.235.22.126 na porta 8085, devido a um erro de compatibilidade os testes não puderam ser realizados no navegador Chrome que já não mais permite os protocolos utilizados pelo Java web, sendo então feitos no Firefox e Internet Explorer. Outro empecilho foi a necessidade de liberação do endereço como confiável através das configurações do Java, que não considerou os certificados serem confiáveis por serem auto assinados.

Figura 21 - Adicionando a página a lista de exceções de confiança para testes.



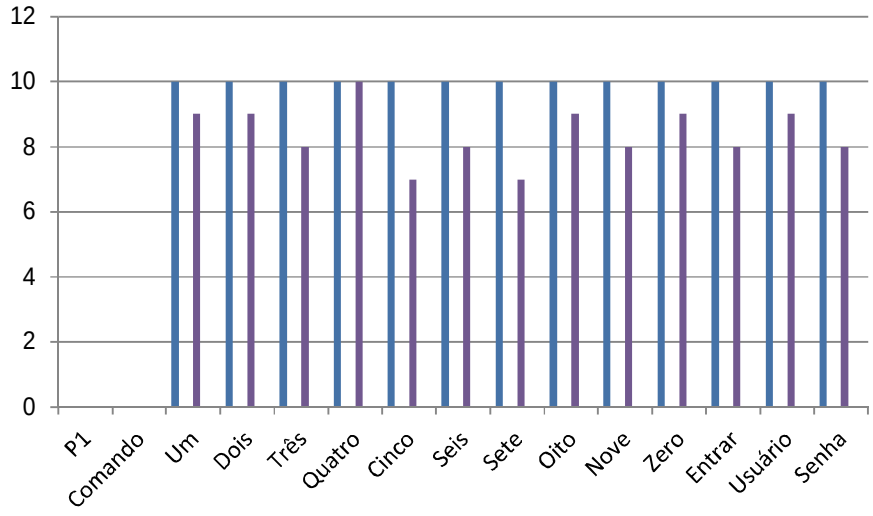
Fonte: Elaborado pelo autor.

5.1 TESTES REALIZADOS

Todos os testes foram realizados em ambiente fechado com 4 usuários, sendo eles 3 homens e uma mulher, cada um deles tentou realizar o login na página com as 10 combinações inseridas no banco, obtendo resultados expressivos. As tabelas geradas demonstram por individuo a taxa de acerto.

A Figura 22 mostra as tentativas de teste do locutor número um, homem, de 21 anos, a taxa de acertos mostra valores expressivos, tendo os piores resultados nas tentativas envolvendo o número cinco.

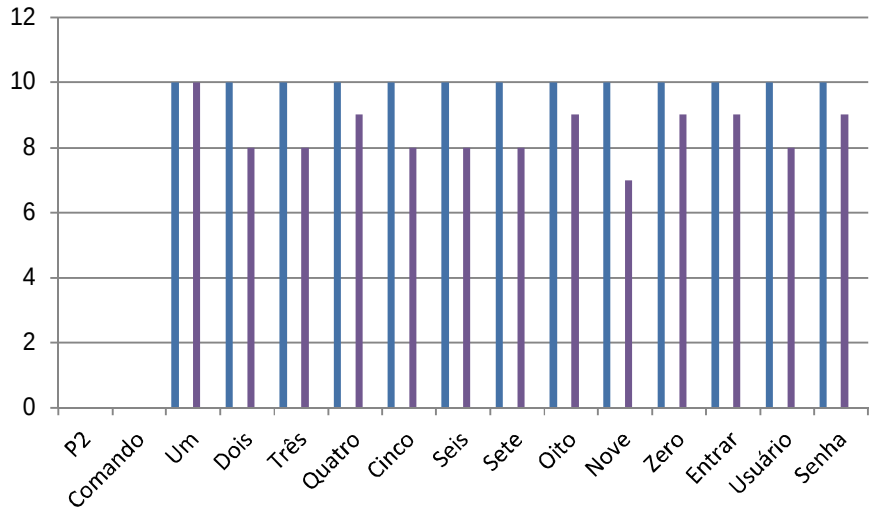
Figura 22 - Índice de acertos do locutor 1.



Fonte: Elaborada pelo o autor.

A Figura 23 mostra os resultados do locutor número dois, sendo esta uma mulher de 23 anos, de voz mais aguda, embora o modelo vocal que é utilizado como comparativo no reconhecimento tenha apenas a voz de um locutor masculino, a taxa de assertividade continua alta.

Figura 23 - Índice de acertos do locutor 2.

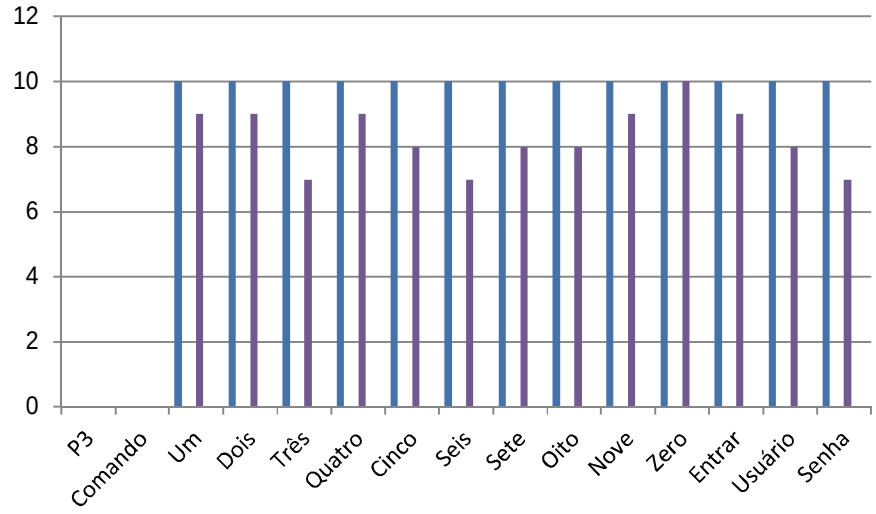


Fonte: Elaborada pelo o autor.

O locutor 3, um homem de 25 anos, também possui resultados relativamente expressivos como mostrado na Figura 24, algo digno de nota é que alguns dos erros são geralmente seguidos, possivelmente devido à análise de probabilidade do algoritmo de

Markov, que leva em consideração os fonemas que possuem a maior possibilidade de ocorrência.

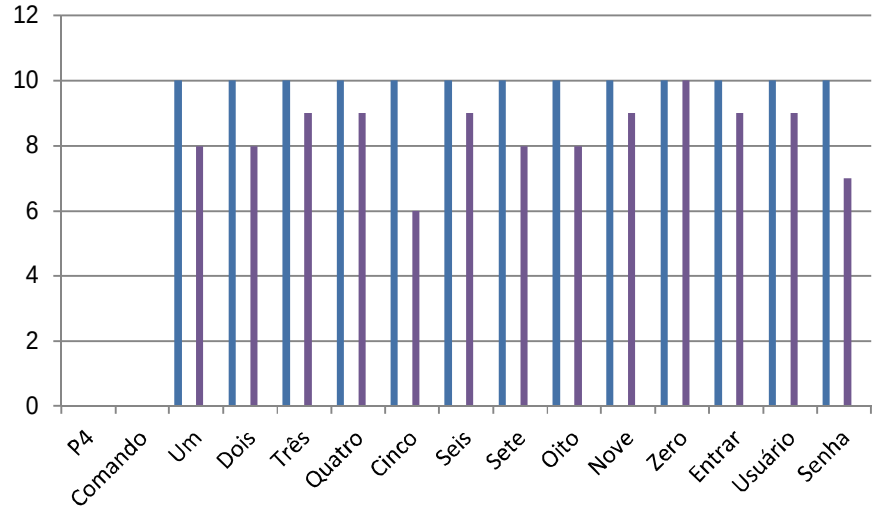
Figura 24 - Índice de acertos do locutor 3.



Fonte: Elaborado pelo o autor.

O último teste foi realizado com um homem de 25 anos, como mostrado na Figura 25, mostrando que no geral a ferramenta possui uma boa capacidade de reconhecimento independentemente do usuário, mesmo não possuindo uma extensa base para treinamento.

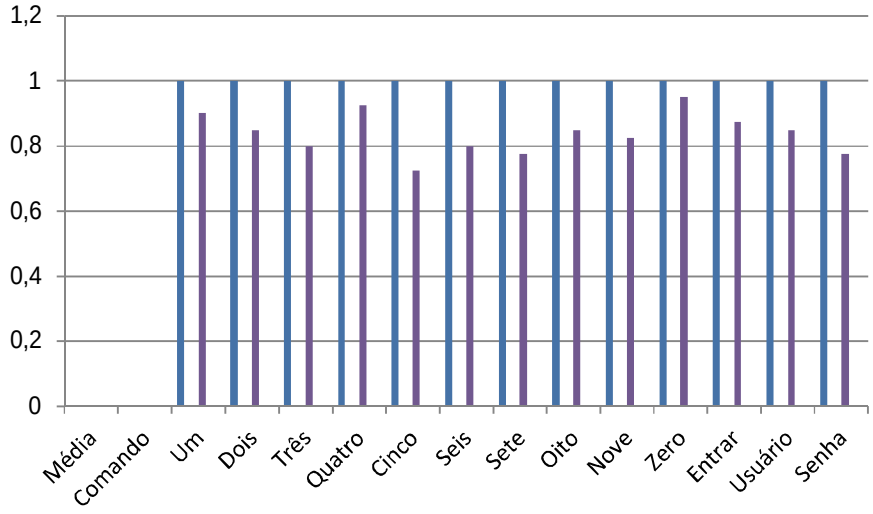
Figura 25 - Índice de acertos do locutor 4.



Fonte: Elaborado pelo o autor.

O índice médio de acertos foi calculado a partir do média aritmética dos resultados obtidos, como é possível observar, a taxa de acertos mínima de cada um do total de comandos é a de 72% referente ao comando “cinco” e o maior de 95% para o número “zero”.

Figura 26 - Índice médio de acertos.



Fonte: Elaborado pelo o autor.

6 CONCLUSÃO

Este trabalho cumpriu seu objetivo de criação de um aplicativo de reconhecimento de voz que possa ser utilizado para autenticação do usuário na internet.

Para que essa meta pudesse ser atingida foi necessário um estudo abrangente, explanando suas origens, as características necessárias para que um sistema deste funcione corretamente e alguns trabalhos relacionados à área de atuação que inspiraram a aplicação final.

Após este estudo foram definidos os métodos e técnicas a serem utilizados, na busca pela praticidade e compatibilidade, foi escolhido o kit de desenvolvimento Sphinx para a criação da aplicação por já possuir facilitadores de treinamento de voz e fácil incorporação à linguagem Java através de sua biblioteca.

Alguns empecilhos ocorreram durante a criação da aplicação, como por exemplo, a retirada do suporte a tecnologia NPAPI dos navegadores Chrome que impossibilitam que a aplicação seja utilizada neles, sendo hoje em dia um dos navegadores mais utilizados, a retirada desse suporte limita em muito os utilizadores dessa aplicação final.

De modo geral o resultado final pode ser considerado satisfatório, tendo como pior resultado a partir dos testes uma taxa de acerto não menor que de 72%, sendo este referente ao número cinco, que muitas vezes era confundido com o número “seis” ou com a ordem “senha” pelo sistema. Para melhora destes níveis existem várias possibilidades de trabalhos futuros, como por exemplo:

- Criação de um modelo acústico mais complexo que componha palavras ditas com diferente sonoridade;
- Utilização de técnicas que diminuam o ruído gerado melhorando a capacidade de interpretação do programa;
- Alterar o vocabulário para palavras de sonoridade diferenciada ou a utilização de alguma técnica diferente de reconhecimento destes fonemas para que sejam interpretados de forma diferente.

Em conclusão este trabalho demonstra de forma concisa uma pesquisa de desenvolvimento na área de reconhecimento de voz, que no Brasil, comparada a outros países é considerada algo novo e desafiador, apesar dos obstáculos encontrados o conhecimento obtido foi de validade infindável e despertou curiosidade e desejo de elaboração de projetos futuros na área.

É importante ressaltar que existe uma ampla gama de possibilidades de trabalhos futuros a partir deste projeto, além das alterações já citadas nestas considerações, ainda pode ser considerada a aplicação dessa forma de automação não apenas para páginas web, como por exemplo, a automação completa de um computador, a manipulação de diversos utensílios eletrônicos e o controle de iluminação e de outros componentes de ambiente.

REFERÊNCIAS

BATISTA, Pedro. **SpeechOO: Uma extensão de ditado para Libre Office**. 2011 11 f. Universidade Federal do Pará.

BORGES, Sidney Reys. **Reconhecimento de Voz aplicada a Interface de Sistemas Emergenciais Hospitalares**. Bacharelado em Informática. 2009, 86 f. Trabalho final de Curso – Universidade Católica de Salvador.

BRAGA, Petrônio L**., Reconhecimento de voz dependente do locutor utilizando Redes neurais artificiais**, 2006, 88f. Escola Politécnica de Pernambuco.

BRESOLIN, Adriano de Andrade. **Sistema de Reconhecimento de Voz para o acionamento de equipamentos elétricos via comandos em português**. Mestrado em automação Industrial, 2003, 107 f. Trabalho de dissertação de Mestrado – Universidade do Estado de Santa Catarina.

DAMASCENO, Eduardo Figueiras. **Implementação de Serviços de Voz em Ambientes Virtuais.** 2005 7 f. Universidade Federal do Mato Grosso do Sul.

DEITEL, Harvey e Paul. **Java como programar** .8e. 2010 1113 f. Pearson education do Brasil.

DENES, Peter B.; PINSON, Elliot. **The Speech Chain: The physics and biology of spoken language**, 2e. 1963. 246f. Oxford: W.H Freeman and Company.

ENDEN, Jarkko**. Java Speech API**, 66 f. 2001, Universidade de Helsink.

FERNANDES, Tales et al, **Do Analógico ao Digital: Amostragem, Quantificação e Codificação**, 3f. Universidade Federal do ABC.

FUKUMORI, Anderson T. **A importância da atividade de teste no desenvolvimento de software,** 2008, 68f. Centro Universitário Eurípides de Marilia.

FURUI, Sadaoki, **Speech Processing Systhesis and Recognition** 2e. 2004 111 f. Marcel Dekker Inc.

GUIMARÃES, Rita de F. R. et al, **GVOICER: Sistema de reconhecimento de voz para controle de aplicações de realidade virtual**, 2005, 4f. Faculdade de Campo Limpo Paulista.

HEUSER, Carlos Alberto, **Projeto de Banco de Dados**. 4e. 1998. 192 f. Instituto de Informática UFRGS.

HOSSOM, John-Paul**, Automatic Speech Recognition with Hidden Markov Models**, 2011. Disponível em:< http://www.cslu.ogi.edu/people/hosom/cs552/>. Acesso em 20/11/2015.

HUNT, Andrew, **JSpeech Grammar Format,** 2000, Disponível em: <http://www.w3.org/TR/jsgf/>. Acesso em 27/11/2015.

IBM**, IBM ViaVoice**. Embedded ViaVoice. 2007. Disponível em: <http://www-01.ibm.com/software/pervasive/embedded_viavoice/>. Acesso em 6/5/2014.

KAFKA, Sandra G. et al, **Utilização de Segmentos Transicionais Homorgânicos em Síntese de fala Concatenativa**,2002, 6f. Universidade Federal de Santa Catarina.

KAUARK, Fabiana. **Metodologia de Pesquisa: Um guia prático**. 2010. 89f. Via Literarum.ç

LATSCH, Vagner L, **Construção de banco de Unidades para Síntese da fala por concatenação no domínio temporal**, 2005, 133 f. Universidade Federal do Rio de Janeiro.

LOUZADA, Jailton Alckmin. **Reconhecimento automático de fala por computador**. 2010 52 f. Trabalho de Conclusão de Curso, PUC Goiás.

MARANGONI, Josemar Barone. **Reconhecimento e sintetização de voz usando Java Speech**. 2006, 10f. Faculdade de Ciências Jurídicas e Gerenciais de Garça.

MARTINS, José Antonio, **Avaliação de Diferentes técnicas para o reconhecimento de fala**. Tese de doutorado. 1997 153 f. Universidade Estadual de Campinas.

MONTEIRO, Emiliano S., **Projeto de Sistema de Banco de Dados** 2. Ed., 2010, 498 f. Edição do autor.

ORACLE, **Java Speech API**. Disponivel em: < http://www.oracle.com/technetwork/java/jsapifaq-135248.html> Acesso em 06/05/2014.

ROCHA, Alisson Nunes, **A História do PHP disponível** em: < http://alisson.eti.br/blog/?p=7> Acesso em 3/06/2014.

ROE. David B. **Voice Communication Between Humans And Machines**. 1994 548 f. National Academy of Sciences.

SEARA, Izabel Christine et al. **Fonética e Fonologia do Português Brasileiro**. 2011, 119 f. UFSC.

SILVA, Carlos P., **Um Software de reconhecimento de voz para o português Brasileiro**, 2010, 85f. Universidade Federal do Pará.

SILVA, Anderson G., **Reconhecimento de voz para palavras isoladas**, 2009, 60f. Universidade Federal de Pernambuco.

SPHINX, **CMUSphinx Tutorial For Developers.** 2014. Disponivel em: <http://cmusphinx.sourceforge.net/> Acesso em 25/11/2015.

UFRGS, **A linguagem PHP.** Disponível em:<http://www.ufrgs.br/engcart/PDASR/linguagens.html> Acesso em 7/05/2014.

VALIATI, João F., **Reconhecimento de voz para direcionamento por meio de redes neurais,** 2000, 129f. Universidade Federal do Rio Grande Do Sul.

VERTANEN, Keith et al, Parakeet: **A Continuous Speech Recognition System for Mobile Touch-Screen Devices,** 2009, 10f. University of Cambridge.

WELLING, Luke. THOMSOM, Laura. **PHP and MySQL Web Development** 4e. 2009. 968 f. Pearson Education Inc.

YNOGUTI, Carlos A., **Reconhecimento de fala contínua usando modelos ocultos de Markov**, 1999, 138 f. Universidade Estadual de Campinas.