

UNIVERSIDADE SAGRADO CORAÇÃO

GUILHERME TREVISAN CANTRO

**DESENVOLVIMENTO DE UM SISTEMA DE
RECONHECIMENTO DE VOZ PARA PESSOAS COM
DEFICIÊNCIA MOTORA**

BAURU

2012

GUILHERME TREVISAN CANTRO

**DESENVOLVIMENTO DE UM SISTEMA DE
RECONHECIMENTO DE VOZ PARA PESSOAS COM
DEFICIÊNCIA MOTORA**

Trabalho de Conclusão de Curso
apresentado ao Centro de Ciências
Exatas como parte dos requisitos para
obtenção do título de bacharel em Ciência
da Computação, sob orientação da Prof^a.
Ms. Patricia Bellin Ribeiro.

BAURU

2012

C233d	<p data-bbox="565 1178 902 1209">Cantro, Guilherme Trevisan</p> <p data-bbox="565 1241 1284 1367">Desenvolvimento de um sistema de reconhecimento de voz para pessoas com deficiência motora / Guilherme Trevisan Cantro -- 2012. 44f. : il.</p> <p data-bbox="602 1398 1166 1430">Orientadora: Profa. Me. Patricia Bellin Ribeiro.</p> <p data-bbox="565 1461 1284 1556">Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) - Universidade Sagrado Coração - Bauru - SP</p> <p data-bbox="565 1587 1284 1650">1. Software. 2. Reconhecimento de voz. 3. Deficiência motora. 4. Fala. I. Ribeiro, Patricia Bellin. II. Título.</p>
-------	---

GUILHERME TREVISAN CANTRO

DESENVOLVIMENTO DE UM SISTEMA DE RECONHECIMENTO DE VOZ PARA PESSOAS COM DEFICIÊNCIA MOTORA

Trabalho de Conclusão de Curso apresentado ao Centro de Ciências Exatas e Sociais Aplicadas da Universidade do Sagrado Coração como parte dos requisitos para obtenção do título em Bacharel em Ciência da Computação sob orientação da Prof^a. Ms. Patricia Bellin Ribeiro.

Banca examinadora:

Prof^a. Ms. Patricia Bellin Ribeiro
Universidade do Sagrado Coração

Prof^a. Ms. Larissa Pavarini Da Luz
Universidade do Sagrado Coração

Prof^o. Ms. Patrick Pedreira Silva
Universidade do Sagrado Coração

Bauru, 6 de dezembro de 2012.

RESUMO

Com o avanço da tecnologia aumenta gradativamente as possibilidades de melhorar a qualidade de vida das pessoas, através de sistemas que possam executar funções pré-programadas. Pensando nisso torna-se objetivo deste trabalho desenvolver um software que utilize as linguagens de programação Flash, JavaScript e C# junto com as técnicas de reconhecimento de voz. Este software tem o propósito de facilitar o acesso de pessoas com deficiência motora ao sistema de busca hoje utilizado nos totens da Universidade do Sagrado Coração(USC) utilizando comandos de voz. Os testes realizados mostram que o sistema consegue identificar com sucesso o que foi falado, chegando em alguns casos a atingir 90 % de acerto. Melhorando assim cada vez mais a interação destas pessoas com os computadores e diminuindo as diferenças entre elas e as pessoas que não possuem nenhuma deficiência.

Palavras-chave: Software, Reconhecimento de voz, Deficiência motora, Fala.

ABSTRACT

With the advancement of technology gradually increases the possibilities of improving the quality of life of people through systems that can perform preprogrammed functions. Thinking about it becomes objective of this study to develop a software that uses Flash programming languages, JavaScript and C # along with voice recognition techniques. This software is designed to facilitate access for people with motor disabilities to the search system now used in the totems of the University of the Sacred Heart (USC) using voice commands. The tests show that the system can successfully identify what was spoken, and in some cases reaching 90% accuracy. Thus improving increasingly interact with computers these people and decreasing the differences between them and people who have no disabilities.

Key Words: Software, Voice recognition, Motor disability, Speaks.

Índice de Ilustrações

Figura 2.1 – Gráfico da forma da onda.....	13
Figura 2.2 – Dados digitais separados em faixas de frequência	14
Figura 2.3 – Reconhecimento da fala.....	15
Figura 5.1 – Tela de configuração.....	21
Figura 5.2 – Tela inicial	22
Figura 5.3 – Tela de login.....	23
Figura 5.4 – Tela de consulta.....	26

SUMÁRIO

1 INTRODUÇÃO	8
1.1 Problema	9
1.2 Justificativa	9
1.3 Objetivo Geral.....	9
1.4 Objetivos Específicos.....	10
2 DEFICIÊNCIA MOTORA.....	11
2.1 Deficiência motora	11
2.2 Voz.....	11
2.2.1 Fisiológica	12
2.2.2 Acústica.....	12
2.2.3 Perceptiva	13
3 RECONHECIMENTO DA FALA	14
3.3.1 Transformando fala em dados.....	14
4 MATERIAIS E MÉTODOS.....	17
4.1 Local	17
4.2 Materiais	17
4.2.1 Headphone.....	17
4.2.2 Computador.....	17
4.2.3 Notebook.....	18
4.2.4 Microsoft Visual Studio 2010.....	18
4.2.5 Microsoft Speech Platform	18
4.2.6 Flash	19
4.3 Metodologia	20
4.3.1 Inicial.....	20
4.3.2 Login	20
4.3.3 Consulta	20
5 RESULTADOS FINAIS.....	21
5.1 Configuração	21
5.2 Inicial.....	21
5.3 Login	23
5.4 Consulta.....	25
6 CONSIDERAÇÕES FINAIS	29

7 REFERÊNCIAS	30
ANEXO A – GETMIC.....	32
ANEXO B – SETVOLUME	33
ANEXO C – STARTRECORDING.....	33
ANEXO D – RECONHECEDOR	34
ANEXO E – PLAYBACK.....	37
ANEXO F – ONCLICKBTNPROFESSOR	38
ANEXO G – ONCLICKBTNALUNO.....	38
ANEXO H – ONCLICKBTNVOLTAR32.....	38
ANEXO I – ONCLICKBTNAPAGAR.....	39
ANEXO J – ENTRAR	39
ANEXO K – PESQUISAR.....	41

1 INTRODUÇÃO

No mundo atual a tecnologia caminha a passos largos, com isso cada vez mais são desenvolvidos sistemas que executam funções pré-programadas para facilitar a vidas das pessoas, através de interfaces visuais muito agradáveis e que se utilizam de sistemas de reconhecimento de voz para melhorar a interação entre homem-máquina.

Segundo Apaydin (2002 apud DAMASCENO,2005,p.15),

[...] a nova geração de interfaces homem-computador será a de interfaces de fácil aprendizado e de alta acessibilidade destacando o uso das interfaces de voz, das quais seriam o equivalente a um terço de todas as interfaces criadas para os ambientes computacionais de alta complexidade. Isto por três motivos principais: primeiro a facilidade no aprendizado do ambiente através do uso de voz conforme Zelter e Johnson (1994 apud DAMASCENO,2005,p.15), em segundo pela liberdade que o usuário pode ter, conforme destaca Tatham (2002 apud DAMASCENO,2005,p.15), que descreve que a geração e a utilização de mensagens ou comandos falados e interpretados pelo computador, ou seja, síntese e reconhecimento da voz humana, permitindo que os olhos ou mãos estejam livres para a realização de outras tarefas e por final, é uma forma mais intuitiva de comunicação, pois o ser humano já utiliza a voz para se comunicar.

Ainda de acordo com Apaydin (2002 apud DAMASCENO,2005,p.15) estes softwares de reconhecimento de voz são muito uteis para o desenvolvimento de sistemas para pessoas com deficiência nos membros superiores, porém eles necessitam de uma máquina cujo hardware seja mais robusto, pois em máquinas simples eles não conseguem ser implementados já que necessitam de alto processamento de dados.

1.1 Problema

Nos dias atuais tem-se a necessidade de cada vez mais melhorar a qualidade de vida das pessoas através de sistemas computacionais, mas as pessoas com deficiência motora nos membros superiores acabam sempre sendo desconsideradas quando se é levantado os pré-requisitos de um sistema (DAMASCENO, 2005).

Segundo Damasceno (2005) muitas vezes os aparelhos de entrada de informação em um sistema, como teclado e mouse, acabam não sendo suficientes para as pessoas com deficiência, já que muitas vezes eles não possuem uma completa coordenação motora de seus membros superiores.

Com isso surgiu-se a necessidade de se pesquisar e desenvolver uma interface, através de reconhecimento da fala, que possa facilitar a interação destas pessoas que possuem deficiência.

1.2 Justificativa

A criação deste trabalho terá o intuito de beneficiar pessoas que possuam algum tipo de deficiência motora, pois através dele será desenvolvido um software que permitirá que estas pessoas possam realizar funções, que para pessoas sem deficiência são simples, mas que para elas seriam muito difíceis de realizarem através de um simples comando de voz.

1.3 Objetivo Geral

O software desenvolvido neste trabalho terá como objetivo executar os mesmos comandos dos totens da Universidade do Sagrado Coração, que hoje são através do toque, através da fala. Para garantir que pessoas com deficiências motoras que não conseguem utilizar os totens com total perfeição possam através da fala facilitar a utilização dos mesmos.

1.4 Objetivos Específicos

Os objetivos específicos deste trabalho são:

- Estudar os métodos de reconhecimento de voz;
- Pesquisar os softwares necessários para se fazer o desenvolvimento do sistema;
- Criação do sistema utilizando os softwares pesquisados e os métodos estudados;
- Testar e avaliar o desempenho do sistema.

2 DEFICIÊNCIA MOTORA

Neste capítulo será apresentado os conceitos utilizados para a elaboração deste trabalho, como deficiência motora e reconhecimento da fala.

2.1 Deficiência motora

Segundo Mattos (2010) pessoas com deficiência motora são aquelas que possuem algum distúrbio da estrutura ou da função, que interfere na movimentação e/ou locomoção.

Ainda de acordo com Mattos (2010) a pessoa para ser considerado um deficiente motor, de caráter permanente, deve possuir incapacidade igual ou superior a 60% tanto sendo nos membros superiores ou inferiores. A deficiência motora pode ser ocasionada por vários motivos como por exemplo: problemas no parto, erros médicos, acidentes de trabalho, acidentes de trânsito, traumatismos, entre outros.

Segundo Mattos (2010) os tipos de deficiência motora podem ser classificados como: monoplegia: paralisia em um membro do corpo, hemiplegia: paralisia em um dos lados do corpo, paraplegia: paralisia da cintura para baixo, diplegia: membros inferiores mais afetados que os superiores, quadriplegia: quatro membros afetados de forma semelhante, dupla hemiplegia: quatro membros afetados, sendo um hemicorpo mais afetado.

2.2 Voz

Segundo Horta (2001) a voz são sons produzidos através do relacionamento dos pulmões com a traquéia, a laringe, a faringe, as cavidades nasais e a cavidade oral.

Segundo Head e Kent (1992 apud HORTA, 2001) a fala nada mais é do que sons auditivamente perceptíveis, podendo ser dividida em três áreas: fisiológica (fisiologia fonética), acústica (fonética acústica) e perceptiva (fonética perceptiva).

2.2.1 Fisiológica

Segundo Cristófal-Silva (2009) a fisiologia fonética ou fonética articulatória é o estudo dos aspectos fisiológicos e articulatorios da produção da fala, ela é encarregada de analisar descrição, classificação e transcrição dos sons produzidos.

Ainda de acordo com Cristófal-Silva (2009) os sons produzidos são chamados de fones ou segmentos, eles podem ser divididos em três grupos: consoantes, vogais e semivogais.

Ainda de acordo com Cristófal-Silva (2009) as consoantes são sons produzidos com algum tipo de obstrução no trato vocal caracterizando algum tipo de impedimento da passagem do ar. Esses sons são classificados conforme os critérios de modo de articulação, lugar de articulação, vozeamento e nasalidade/oralidade. As vogais e as semivogais são sons produzidos sem a obstrução caracterizada na consoante, deixando a passagem do sem nenhum impedimento. Os sons são classificados conforme os critérios de altura da língua, anterioridade/posterioridade, arredondamento dos lábios e nasalidade/oralidade.

2.2.2 Acústica

Segundo Cristófal-Silva (2009) é uma parte da fonética que trabalha com aspectos acústicos, propriedades físicas dos sons da fala. Através desta pesquisa é possível identificar além do conteúdo da fala características da pessoa que esta emitindo o som como: sexo, porte físico, idade, origem sócio-cultural, problemas de saúde e estado emocional.

Ainda de acordo com Cristófal-Silva (2009) a fonética acústica analisa as propriedades da amplitude, duração, frequência fundamental e conteúdo espectral da onda sonora através de espectrogramas, gráficos da forma de onda, trajetórias de formantes e da frequência fundamental.

A figura 2.1 mostra o gráfico da forma de onda que possuem mesma amplitude com frequências diferentes.

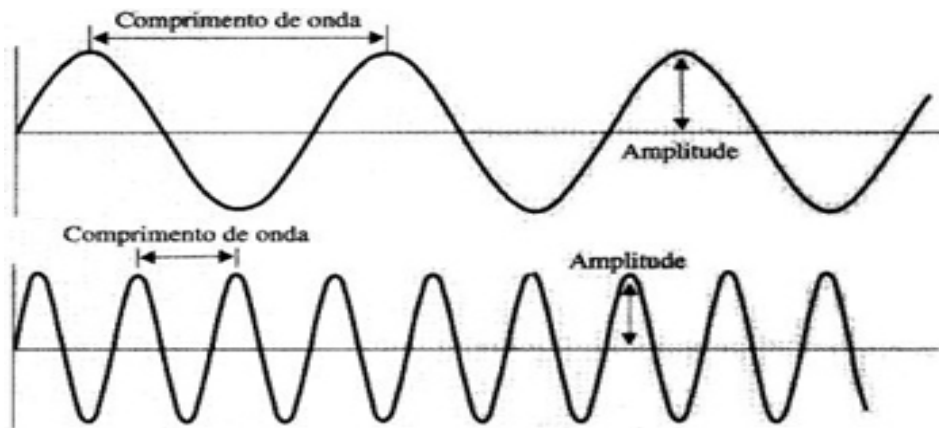


Figura 2.1 – Gráfico da forma da onda. Fonte: (CRISTÓFARO-SILVA, 2009).

2.2.3 Perceptiva

Segundo Hora e Collischon (2003) muitas vezes percebemos o mesmo som de formas diferentes, por isso precisamos analisá-lo de forma mais específica e a área responsável por fazer este estudo é a fonética perceptiva, esta análise dos sons é feita através da audição.

3 RECONHECIMENTO DA FALA

Segundo a Microsoft (2012) o *Microsoft Speech Server Platform* é uma tecnologia utilizada principalmente para o reconhecimento da fala.

Ainda de acordo com a Microsoft (2012) através desta tecnologia pode-se melhorar a interação entre o usuário e o computador, pois não seria necessário nenhum dispositivo de entrada (teclado ou mouse) para o usuário utilizar o computador apenas inserir comandos de voz.

3.3.1 Transformando fala em dados

Segundo Grabianowski (2012) para se transformar fala em um texto o computador executa vários passos complexos. Primeiramente quando é emitido algum som se criam vibrações no ar, então o Conversor Analógico-Digital (CAD) captura estas vibrações e as converte em dados digitais separando-os em varias faixas de freqüência, como pode ser visualizado na figura 2.2

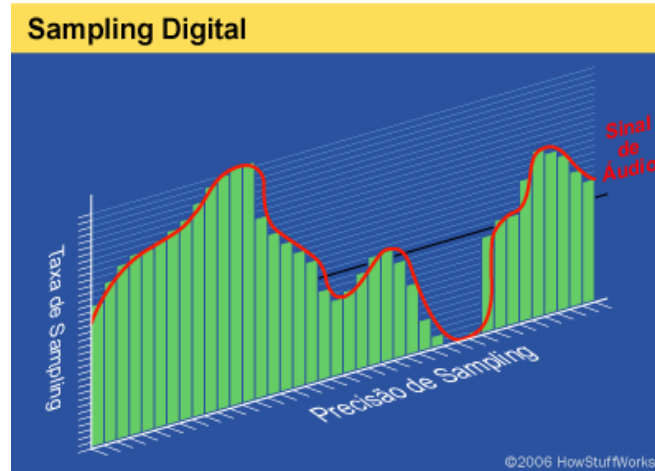


Figura 2.2 – Dados digitais separados em faixas de frequência. Fonte: (GRABIANOWSKI, 2012).

Ainda de acordo com Grabianowski (2012) após converter as vibrações em dados digitais, estes então são divididos em segmentos de centésimos ou até milésimos de segundos, cada segmento é comparado a um fonema do idioma desejado.

O software analisa cada fonema dentro do conjunto de fonemas e compara com um conjunto de frases já conhecidas para determinar o que foi dito, como pode ser visualizado na figura 2.3

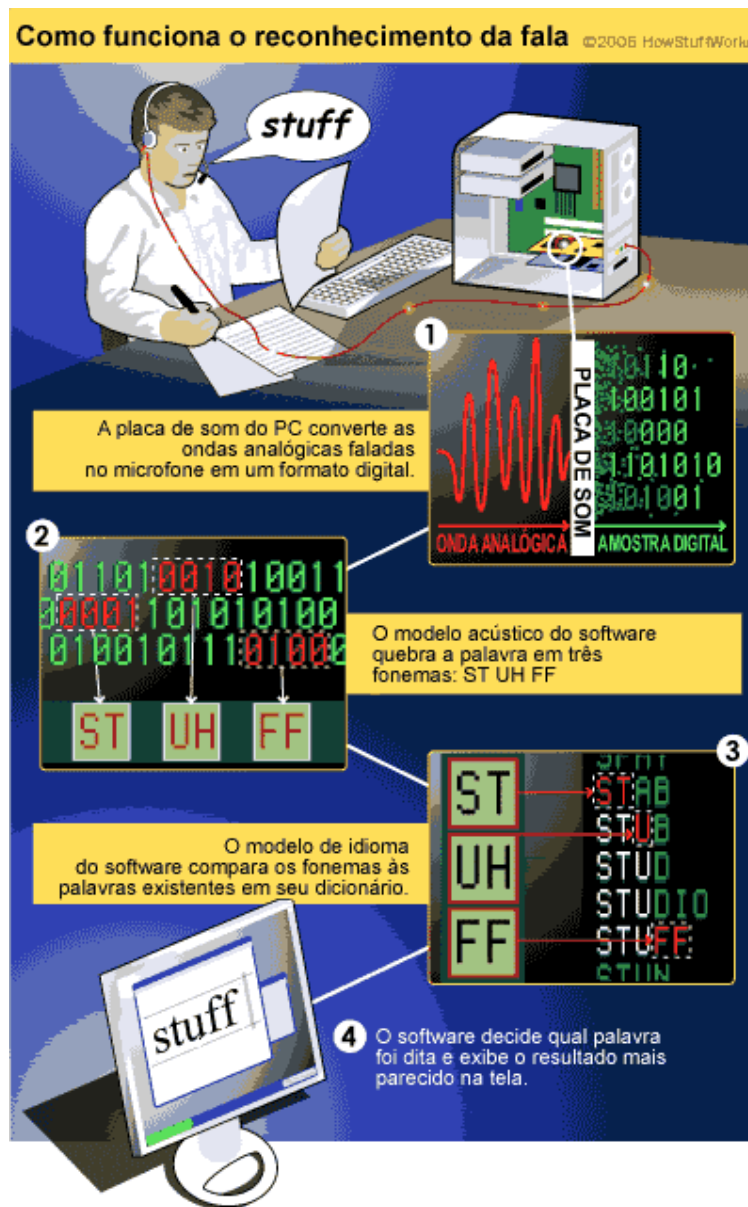


Figura 2.3 – Reconhecimento da fala. Fonte: (GRABIANOWSKI, 2012).

A figura 2.3 mostra como é feita a identificação do que foi falado, primeiramente é feita a conversão do som para um array, onde cada item é uma seqüência binária, então cada item deste array é comparado a um fonema do idioma desejado e por fim estes fonemas são agrupados formando assim a palavra que foi falada.

Mas segundo Grabianowski (2012) nenhum sistema é 100% perfeito, pois vários fatores podem atrapalhar a captura de som do computador como ruídos, que podem ser do próprio som ambiente ou de microfones de má qualidade que não possuem proteção contra sinais elétricos. Podem ocorrer problemas quando duas ou mais pessoas falam ao mesmo tempo, pois o computador não consegue identificar quais as palavras que foram ditas entre os intervalos de tempo.

Outro problema são os homônimos, que são as palavras escritas de forma diferente e com significado diferente mas que possuem o mesmo som, pois o computador não consegue identificar qual é a palavra pronunciada pois os sons são iguais (GRABIANOWSKI, 2012).

E por fim Grabianowski (2012) diz que os computadores devem possuir um alto poder de processamento pois eles trabalham sempre no limite para poder conseguir identificar a palavra que foi pronunciada.

4 MATERIAIS E MÉTODOS

Os materiais e métodos utilizados na pesquisa foram de total importância para o sucesso da pesquisa.

4.1 Local

Todos os testes e captura de som foram realizados na Universidade do Sagrado Coração, na cidade de Bauru-SP, em locais onde não se concentram grandes quantidades de alunos para que não ocorra muita interferência sonora, estes testes foram realizados com todos os equipamentos necessários.

4.2 Materiais

4.2.1 Headphone

O headphone utilizado será um com fone de ouvido de impedância de 32 Ohm/ 1KHz, resposta de frequência de 20 ~ 20 KHz e microfone de impedância 32 Ohm/ 1KHz e resposta de frequência de 30 ~ 16 KHz.

4.2.2 Computador

Especificações Gerais:

- Processador Core 2 Duo 2.93GHz
- HD 500 Gb Interno
- Placa mãe Intel DZ68DB
- Sistema Operacional Windows 7 64 Ultimate
- Memória 4 Gb

Foi utilizado o computador com estas especificações porque o sistema necessita de uma grande quantidade de processamento.

4.2.3 Notebook

Especificações Gerais:

- HP Compaq
- Processador Core 2 Duo 2.5 GHz
- HD 160 Gb
- Sistema Operacional Windows 7 64 Ultimate
- Memória 4 Gb

Foi utilizado o computador com estas especificações porque o sistema necessita de uma grande quantidade de processamento.

4.2.4 Microsoft Visual Studio 2010

Segundo a Microsoft (2012) o Microsoft Visual Studio é uma plataforma desenvolvida pela Microsoft que permite desenvolver aplicativos locais ou web em diversas linguagens de programação como:

- Visual Basic;
- C;
- C++;
- C#;
- J#.

Ainda de acordo com a Microsoft (2012) a plataforma utiliza o .NET framework, que é um componente integral do Windows que oferece suporte a criação de aplicativos, ele contém uma biblioteca de classes com métodos prontos para facilitar o desenvolvimento.

4.2.5 Microsoft Speech Platform

Segundo a Microsoft (2012) o *Microsoft Speech Platform* SDK (Software Development Kit) versão 11, é uma plataforma que permite a criação de aplicativos para facilitar a interação do usuário com a máquina, pois ao invés do usuário utilizar o mouse ou teclado ele utilizara apenas a voz.

Ainda de acordo com a Microsoft (2012) o *Microsoft Speech Platform SDK 11* é uma biblioteca da plataforma Microsoft Visual Studio, são classes escritas em linguagem de programação C#, esta biblioteca tem a intenção de facilitar para o programador o desenvolvimento de sistemas que utilizem a voz, tanto como comando de entrada como comando de saída, esta biblioteca esconde todo o código complexo desenvolvido para utilizar se o reconhecimento da fala, habilitando apenas para o desenvolvedor a chamada de seus métodos sem que ele veja o “passo-a-passo” da execução do código.

O *Microsoft Speech Platform SDK 11* suporta tanto comandos únicos de voz, uma palavra de cada vez, como também ditados. É uma biblioteca simples de se instalar e também de utilizar, pois acompanha um manual bem didático, e pode-se integrar com outras API's desde que pertençam a plataforma Microsoft Visual Studio.

4.2.6 Flash

Segundo a Adobe (2012) o Flash é um tipo de linguagem de programação criada pela Macromedia que possibilita uma interação visual mais agradável com o usuário, já que ela serve para criar animações e websites dinâmicos.

Ainda de acordo com a Adobe (2012) pode-se descrever as vantagens do flash da seguinte forma: os sites que são construídos em flash carregam mais rápidos que os construídos em outras linguagens, os sites são mais dinâmicos, conseguem ser interpretados por qualquer browser ou sistema operacional, diferente das outras linguagens que dependendo do sistema operacional ou browser não funciona.

4.3 Metodologia

4.3.1 Inicial

Os testes foram realizados utilizando as palavras “professor” e “aluno”, pois cada palavra representa uma área diferente do sistema, então o objetivo é identificar qual área o usuário deseja acessar, cada palavra foi pronunciada 10 vezes, foi estipulada esta métrica para poder analisar qual a margem de aproveitamento do sistema, e então após a fala do usuário era observado se o sistema conseguia identificar ou não o que foi falado.

4.3.2 Login

Os testes foram realizados utilizando as palavras “login”, “senha”, “apagar”, “voltar”, “entrar”, “1”, “2” e “3”, pois cada palavra representa uma função do sistema, cada palavra foi pronunciada 10 vezes, foi estipulada esta métrica para poder analisar qual a margem de aproveitamento do sistema, e então após a fala do usuário era observado se o sistema conseguia identificar ou não o que foi falado.

4.3.3 Consulta

Os testes foram realizados utilizando as palavras “nome”, “professor”, “aluno”, “apagar”, “pesquisar”, “a” e “gui”, pois cada palavra representa uma função do sistema, cada palavra foi pronunciada 10 vezes, foi estipulada esta métrica para poder analisar qual a margem de aproveitamento do sistema, e então após a fala do usuário era observado se o sistema conseguia identificar ou não o que foi falado.

5 RESULTADOS FINAIS

5.1 Configuração

A tela de configuração serve para verificar se o microfone esta instalado. Ela contém um botão que permite o usuário liberar ou não o acesso ao microfone. Tela de configuração conforme a figura 5.1

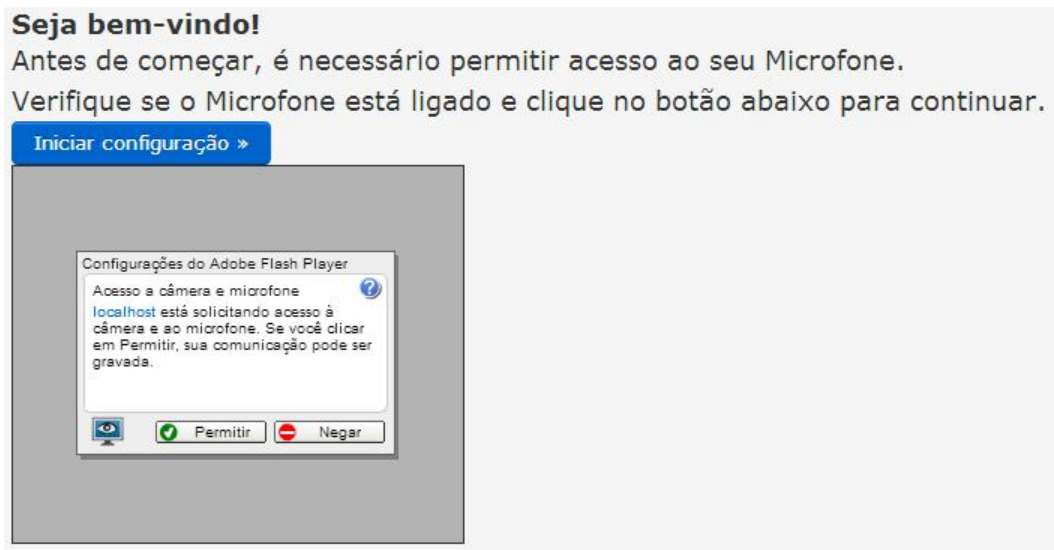


Figura 5.1 – Tela de configuração.

A tela consta com um botão Iniciar configuração que verifica se o microfone esta instalado.

Ao ser clicado o botão chama o método `getMic()`, este método verifica se o microfone esta instalado, então ele ativa o uso do microfone no sistema, que pode ser visualizado no Anexo A.

5.2 Inicial

A tela serve para o usuário escolher se deseja entrar na área do professor ou do aluno. Tela inicial conforme a figura 5.2



Figura 5.2 – Tela inicial.

Ela consta de seis botões sendo eles:

“Volume microfone”, serve para regular o volume de captura de som do microfone, ao ser clicado ele chama o método `setVolume`, que pode ser visualizado no Anexo B.

“Clique aqui e fale a palavra”, serve para chamar o método `startRecording` que captura a voz, o método pode ser visualizado no Anexo C. Foi utilizado este botão para realizar a captura porque ele vira a se tornar um futuro botão físico onde o usuário ira apertá-lo e então ira ativar a captura da voz.

“Parar”, serve para parar a captura da voz, ao ser clicado ele chama o método `stopRecording`, se o botão não for clicado em 7 segundos o método `stopRecording` é chamado automaticamente, foi utilizado este período de 7 segundos para que mesmo que o usuário tenha uma dificuldade na fala ele possa ter tempo para falar. Após ser finalizada a gravação do áudio é chamado o método `reconhecer` que realiza a descompactação do áudio e chama o método `reconhecerWAV`, este verifica se já esta instalado o SDK e o runtime do `SpeechPlataform`, após isso é feita a inserção das palavras que seram comparadas com o áudio para verificar qual delas foi pronunciada, a inserção é feita através de um XML no formato padrão do `SpeechPlataform`, logo em seguida é feita a conexão com o `WebService` que retorna então o que foi falado no áudio em forma de texto, após isso é chamado o método `WordDone` que aciona o método correto dependendo de cada palavra que foi pronunciada, os métodos podem ser visualizados no Anexo D.

“Ouvir” o que eu falei, serve para reproduzir o que foi falado enquanto a captura da voz estava aberta, ao ser clicado ele chama o método `playback`, que pode ser visualizado no Anexo E.

“Professor”, serve para direcionar o usuário para a área do professor, ao ser clicado ele chama o método `onClickBtnProfessor`, que pode ser visualizado no Anexo F.

“Aluno”, serve para direcionar o usuário para a área do aluno, ao ser clicado ele chama o método `onClickBtnAluno`, que pode ser visualizado no Anexo G.

Após testes realizados na Universidade do sagrado Coração foi possível observar que a tela teve um aproveitamento de 90% na parte da manhã e da tarde, pois nesses horários não se encontravam muitos alunos, deixando assim o ambiente mais silencioso. Já no período noturno a tela teve um aproveitamento de 50%, pois a concentração de alunos era maior e isso ocasionava uma poluição sonora maior, interferindo assim nos comandos de voz passados ao microfone. Os testes tiveram uma margem boa de aproveitamento pois a tela inicial só possui dois comandos, tornando-se assim fácil de se utilizar.

5.3 Login

A tela serve para que o professor ou o aluno façam o login no sistema. Tela de login conforme a figura 5.3



Por favor fale novamente.

Volume Microfone

Clique aqui e fale a palavra

Ouvir o que eu falei

Aluno

Id(Login)

Senha

Voltar Apagar Entrar

Figura 5.3 – Tela de login.

Ela consta de sete botões sendo eles:

“Volume microfone”, serve para regular o volume de captura de som do microfone, ao ser clicado ele chama o método `setVolume`, que pode ser visualizado no Anexo B.

“Clique aqui e fale a palavra”, serve para chamar o método `startRecording` que captura a voz, o método pode ser visualizado no Anexo C. Foi utilizado este botão para realizar a captura porque ele vira a se tornar um futuro botão físico onde o usuário irá apertá-lo e então irá ativar a captura da voz.

“Parar”, serve para parar a captura da voz, ao ser clicado ele chama o método `stopRecording`, se o botão não for clicado em 7 segundos o método `stopRecording` é chamado automaticamente, foi utilizado este período de 7 segundos para que mesmo que o usuário tenha uma dificuldade na fala ele possa ter tempo para falar. Após ser finalizada a gravação do áudio é chamado o método `reconhecer` que realiza a descompactação do áudio e chama o método `reconhecerWAV`, este verifica se já está instalado o SDK e o runtime do `SpeechPlatform`, após isso é feita a inserção das palavras que serão comparadas com o áudio para verificar qual delas foi pronunciada, a inserção é feita através de um XML no formato padrão do `SpeechPlatform`, logo em seguida é feita a conexão com o `WebService` que retorna então o que foi falado no áudio em forma de texto, após isso é chamado o método `WordDone` que aciona o método correto dependendo de cada palavra que foi pronunciada, os métodos podem ser visualizados no Anexo D.

“Ouvir o que eu falei”, serve para reproduzir o que foi falado enquanto a captura da voz estava aberta, ao ser clicado ele chama o método `playback`, que pode ser visualizado no Anexo E.

“Voltar”, serve para redirecionar o usuário para a segunda tela, ao ser clicado ele chama o método `onClickBtnVoltar32`, que pode ser visualizado no Anexo H.

“Apagar”, serve para apagar a última letra do campo que está com o foco, ao ser clicado ele chama o método `onClickBtnApagar`, que pode ser visualizado no Anexo I.

“Entrar”, serve para validar se os dados de login e senha estão corretos, ao ser clicado ele chama o método `onClickBtnEntrar`, este por sua vez verifica se os campos foram preenchidos e chama o método `Entrar` no javascript, este realiza uma conexão Ajax e chama o método `Entrar` no C# que verifica através de um XML se o login e senha estão corretos, este XML foi desenvolvido pensando em uma possível integração deste sistema com o sistema da Universidade do Sagrado Coração, foi utilizado o XML pela facilidade que ele apresenta com relação a integração de diferentes sistemas diferentemente do banco de dados que é mais complexo, os métodos podem ser visualizados no Anexo J.

Após testes realizados na Universidade do sagrado Coração foi possível observar que a tela teve um aproveitamento de 70% na parte da manhã e da tarde, pois nesses horários não se encontravam muitos alunos, deixando assim o ambiente mais silencioso. Já no período noturno a tela teve um aproveitamento de 40%, pois a concentração de alunos era maior e isso ocasionava uma poluição sonora maior, interferindo assim nos comandos de voz passados ao microfone. Os testes não tiveram uma margem tão boa de aproveitamento quanto a tela inicial pois esta tela de login é mais complexa e necessita de mais comandos de voz.

5.4 Consulta

A tela serve para realizar a busca dos horários de aula dos alunos ou dos professores. Tela de consulta conforme a figura 5.4



Figura 5.4 – Tela de consulta.

Ela consta de seis botões sendo eles:

“Volume microfone”, serve para regular o volume de captura de som do microfone, ao ser clicado ele chama o método `setVolume`, que pode ser visualizado no Anexo B.

“Clique aqui e fale a palavra”, serve para chamar o método `startRecording` que captura a voz, o método pode ser visualizado no Anexo C. Foi utilizado este botão para realizar a captura porque ele vira a se tornar um futuro botão físico onde o usuário ira apertá-lo e então ira ativar a captura da VOZ.

“Parar”, serve para parar a captura da voz, ao ser clicado ele chama o método stopRecording, se o botão não for clicado em 7 segundos o método stopRecording é chamado automaticamente, foi utilizado este período de 7 segundos para que mesmo que o usuário tenha uma dificuldade na fala ele possa ter tempo para falar. Após ser finalizada a gravação do áudio é chamado o método reconhecer que realiza a descompactação do áudio e chama o método reconhecerWAV, este verifica se já está instalado o SDK e o runtime do SpeechPlatform, após isso é feita a inserção das palavras que serão comparadas com o áudio para verificar qual delas foi pronunciada, a inserção é feita através de um XML no formato padrão do SpeechPlatform, logo em seguida é feita a conexão com o Webservice que retorna então o que foi falado no áudio em forma de texto, após isso é chamado o método WordDone que aciona o método correto dependendo de cada palavra que foi pronunciada, os métodos podem ser visualizados no Anexo D.

“Ouvir o que eu falei”, serve para reproduzir o que foi falado enquanto a captura da voz estava aberta, ao ser clicado ele chama o método playback, que pode ser visualizado no Anexo E.

“Apagar”, serve para apagar a última letra do campo que está com o foco, ao ser clicado ele chama o método onClickBtnApagar, que pode ser visualizado no Anexo I.

“Pesquisar”, serve para realizar a pesquisa conforme os dados preenchidos na tela, ao ser clicado ele chama o método onClickBtnPesquisar, este por sua vez verifica se os campos foram preenchidos e chama o método Pesquisar no javascript, este realiza uma conexão Ajax e chama o método Pesquisar no C# que verifica através de um XML quais seriam os horários de aula do nome pesquisado, este XML foi desenvolvido pensando em uma possível integração deste sistema com o sistema da Universidade do Sagrado Coração, foi utilizado o XML pela facilidade que ele apresenta com relação a integração de diferentes sistemas diferentemente do banco de dados que é mais complexo, os métodos podem ser visualizados no Anexo K.

Após testes realizados na Universidade do sagrado Coração foi possível observar que a tela teve um aproveitamento de 70% na parte da manhã e da tarde, pois nesses horários não se encontravam muitos alunos, deixando assim o ambiente mais silencioso. Já no período noturno a tela teve um aproveitamento de 50%, pois a concentração de alunos era maior e isso ocasionava uma poluição sonora maior, interferindo assim nos comandos de voz passados ao microfone. Os testes não tiveram uma margem tão boa de aproveitamento quanto a tela inicial pois esta tela de consulta é mais complexa e necessita de mais comandos de voz.

6 CONSIDERAÇÕES FINAIS

Com o avanço da tecnologia são criadas cada vez mais novas ferramentas que visam facilitar a vida das pessoas, ferramentas estas que proporcionam uma interação entre usuário e sistema tornando-se assim essenciais no dia a dia.

Mas na maioria das vezes as pessoas com deficiência acabam sendo desconsideradas quando se é levado em conta os pré-requisitos de um sistema, pois os aparelhos de entrada de informação, como teclado e mouse, acabam não sendo suficientes para as pessoas com algum tipo de deficiência, já que elas não possuem uma completa coordenação motora.

Assim neste trabalho desenvolveu-se um sistema com as funções do sistema utilizado nos totens da Universidade do Sagrado Coração, com a opção das funções serem realizadas através do comando de voz, proporcionando as pessoas com deficiência os mesmos acessos de uma pessoa que não possua nenhuma deficiência motora, melhorando a interação social dentro da Universidade do Sagrado Coração.

O software possui algumas limitações quanto à identificação de fonemas parecidos, nem sempre retornando a palavra desejada, e também a captura de ruídos, que dependendo da poluição sonora pode atrapalhar e muito a identificação da palavra.

Portanto, percebeu-se que a nova cultura social visa à igualdade entre todas as pessoas, assim a utilização desta ferramenta sistêmica facilita a vida das pessoas com deficiência dentro do campus da universidade, deixando-as em um patamar de igualdade.

7 REFERÊNCIAS

CRISTÓFARO-SILVA, Thaís ; YEHIA, Hani Camille . Sonoridade em Artes, Saúde e Tecnologia. Belo Horizonte: Faculdade de Letras, 2009. Disponível em <http://fonologia.org>. ISBN 978-85-7758-135-1.

DAMASCENO, E. F. AVALIAÇÃO DAS BIBLIOTECAS DE RECONHECIMENTO E SÍNTESE DE FALA EM AMBIENTES VIRTUAIS, 129p. Dissertação (Mestrado em Ciências da Computação)- FUNDAÇÃO DE ENSINO EURÍPIDES DA ROCHA SOARES, UNIVEM, Marília, 2005.

FAGUNDES, R. D. R. Reconhecimento de Voz, Linguagem Contínua, usando Modelos de Markov. Tese de Mestrado, Escola Politécnica da USP, 1993.

GRABIANOWSKI, E. Como funciona o reconhecimento de voz. Howstuffworks, 2012. Disponível em: <<http://informatica.hsw.uol.com.br/reconhecimento-de-voz.htm>>. Acesso em: 21 mar. 2012.

HORA, Dermeval da ; COLLISCHON, Gisela. *Teoria lingüística: fonologia e outros temas*. João Pessoa: Editora Universitária, 2003.

HORTA, F. L. Um método de investigação dos distúrbios da fala e voz: a espectrografia vocal. Fonoaudiologia, 2001. Disponível em: <<http://www.fonoaudiologia.com/trabalhos/artigos/artigo-027/artigo-027-objetivo.htm>>. Acesso em: 30 mar. 2012.

JAVASCRIPT REFERÊNCIA DE LINGUAGEM. Microsoft, 2012. Disponível em: <[http://msdn.microsoft.com/pt-br/library/d1et7k7c\(v=vs.94\).aspx](http://msdn.microsoft.com/pt-br/library/d1et7k7c(v=vs.94).aspx)>. Acesso em: 24 mar. 2012.

JOHNSON, Glenn. MCTS Self-Paced Training Kit (Exam 70-516): Accessing Data with Microsoft .Net Framework 4. Redmond: Ed. Microsoft GmbH, 2011. 647p.

MATTOS, Elisabeth de. Deficiências e seus tipos: atividades motoras, esportivas e recreativas para portadores de deficiência motora, 2010. Disponível em: <<http://www.luzimarteixeira.com.br/wp-content/uploads/2010/05/atividades-motoras-e-esportivas-na-deficiencia-fisica.pdf>>. Acesso em: 30 mar. 2012.

MICROSOFT SPEECH PLATFORM FOR SERVERS. Microsoft, 2012. Disponível em: <<http://www.microsoft.com/en-us/Tellme/developers/default.aspx#tab=server>>. Acesso em: 23 mar. 2012.

NORTHRUP, Tony; Northrup, Anthony; Snell, Mike. McTs Self-Paced Training Kit (Exam 70-515): Web Applications Development with Microsoft .Net Framework 4: Web Applications Development with Microsoft .Net Framework 4. Redmond: Ed. Microsoft GmbH, 2010. 967p.

PUBLICAÇÃO DE DOCUMENTOS FLASH. Adobe, 2012. Disponível em: <http://help.adobe.com/pt_BR/Flash/10.0/UsingFlash/Wd60f23110762d6b883b18f10cb1fe1af6-7bd4a.html#Wd60f23110762d6b883b18f10cb1fe1af6-7bd3a>. Acesso em: 23 mar. 2012.

SEBESTA, R. Conceitos de Linguagens de Programação, 3ª ed., Rio de Janeiro, BookMan 2002

VISÃO GERAL CONCEITUAL DO .NET FRAMEWORK. Microsoft, 2012. Disponível em: <<http://msdn.microsoft.com/pt-br/library/zw4w595w.aspx>>. Acesso em: 23 mar. 2012.

ANEXO A – GETMIC

```

// Permissão do microfone;
public function getMic()
{
    log("on getMic");
    mic.addEventListener(StatusEvent.STATUS, onMicStatus);
    recorder.record();
    recorder.gain = 50;

    recorder.microphone.setSilenceLevel(0, 400);
    recorder.microphone.addEventListener(ActivityEvent.ACTIVITY, this.Desativar);
    recorder.stopOnMicSilent = true;

    // Status
    function onMicStatus(event:StatusEvent):void
    {
        log("onMicStatus");
        if (event.code == "Microphone.Unmuted")
        {
            log("Microphone access was allowed.");
            recorder.stop();
            ExternalInterface.call("step1()");
        }
        else if (event.code == "Microphone.Muted")
        {
            log("Microphone access was denied.");
            ExternalInterface.call("error()");
        }
    }
    log("Microfone rate: " + recorder.rate.toString());
    log("Nível para parar gravação: " + activLevel.toString());
}

function Desativar(event:ActivityEvent)
{
    log("activating= " + event.activating + ", activityLevel= " + mic.activityLevel);
}

```

ANEXO B – SETVOLUME

```

public function setVolume(volume)
{
    try
    {if ((volume >= 0) && (volume <= 100))
        {recorder.gain = int(volume);
            log("Microfone gain: "+recorder.gain);
            if (recorder.gain <= 10){activLevel = 5;}
            else if (recorder.gain <= 15){activLevel = 8;}
            else if (recorder.gain <= 20){activLevel = 9;}
            else if (recorder.gain <= 25){activLevel = 10;}
            else if (recorder.gain <= 30){activLevel = 11;}
            else if (recorder.gain <= 35){activLevel = 13;}
            else if (recorder.gain <= 40){activLevel = 15;}
            else if (recorder.gain <= 45){activLevel = 18;}
            else if (recorder.gain <= 50){activLevel = 22;}
            else if (recorder.gain <= 55){activLevel = 26;}
            else if (recorder.gain <= 60){activLevel = 32;}
            else if (recorder.gain <= 65){activLevel = 39;}
            else if (recorder.gain <= 70){activLevel = 48;}
            else if (recorder.gain <= 75){activLevel = 59;}
            else if (recorder.gain <= 80){activLevel = 73;}
            else if (recorder.gain <= 85){activLevel = 91;}
            else{activLevel = 100;}}
        else{recorder.gain = 50;
            activLevel = 40;}}
    catch (err){log("Erro Setar volume: " + err.toString());}
}

```

ANEXO C – STARTRECORDING

```

public function startRecording()
{
    recording = true;
    recorder.microphone.codec = SoundCodec.NELLYMOSER;
    recorder.record();
    tempoGravacao.start();
    btnOuvir.label = "Parar";
    log("Gravação iniciada");
    gravou = false;
    log("Nível para gravar: " + activLevel.toString());
    timerMonitor.start();
}

```

ANEXO D – RECONHECEDOR

```

public function stopRecording(atividade:int)
{
    tempoGravacao.stop();
    tempoGravacao.reset();
    timerMonitor.stop();
    recording = false;
    recorder.stop();
    btnOuvir.label = "Gravar";
    log("Gravação interrompida");
    log("Tamanho sem compressão: " + recorder.output.length.toString());
    recorder.output.deflate();
    log("Tamanho com compressão: " + recorder.output.length.toString());
    var request:URLRequest = new URLRequest("../Request.ashx?atividade=" + atividade.toString());
    request.contentType = 'application/octet-stream';
    request.method = URLRequestMethod.POST;
    request.data = recorder.output;
    var loader:URLLoader = new URLLoader ;
    loader.addEventListener(Event.COMPLETE,uploadComplete);
    loader.addEventListener(ProgressEvent.PROGRESS,uploadProgress);
    loader.addEventListener(SecurityErrorEvent.SECURITY_ERROR, onSecurityError);
    loader.addEventListener(IOErrorEvent.IO_ERROR, onIoError);
    loader.load(request);
    function uploadProgress(e:ProgressEvent):void{log("Enviando o arquivo para o servidor: " + e.bytesLoaded + "%");}
    function uploadComplete(e:Event):void{
        var loader:URLLoader = URLLoader(e.target);
        var resultado:String = loader.data;
        ExternalInterface.call("wordDone", resultado);
    }
    function onSecurityError(event:SecurityErrorEvent):void
        trace("securityErrorHandler: " + event);
        ExternalInterface.call("fatalError", "securityErrorHandler: " + event);
    }
    function onIoError(event:IOErrorEvent):void
        trace("ioErrorHandler: " + event);
        ExternalInterface.call("fatalError", "ioErrorHandler: " + event);
    }
}

[WebMethod]
[System.Web.Script.Services.ScriptMethod()]
public string reconhecer(byte[] arquivo, Int64 lesson = 0)
{
    if (arquivo.LongLength <= 0)
    {
        return "0x3421"; // arquivo vazio
    }

    // Descompacta o arquivo de audio.
    MemoryStream streamCompactada = new MemoryStream(arquivo);
    DeflateStream stream = new DeflateStream(streamCompactada, CompressionMode.Decompress);
    MemoryStream streamDescompactada = new MemoryStream();
    this.CopyStream(stream, streamDescompactada);

    return this.reconhecerWAV(streamDescompactada.GetBuffer(), lesson, false);
}

```

```

public string reconhecerWAV(byte[] arquivoWAV, Int64 lesson = 0, bool isPCM = false)
{
    RecognitionResult result;
    // Verifica a existencia de reconhecedores instalados na maquina servidor.
    if (SpeechRecognitionEngine.InstalledRecognizers().Count < 1){
        throw new Exception("Sem reconhecedores instalados.");
    }
    string arquivoGramatica = Server.MapPath("/Gramaticas/PalavrasPortugues.grxml");
    if (!File.Exists(arquivoGramatica)){
        throw new Exception("Arquivo de gramática não encontrado");
    }
    SpeechRecognitionEngine recognizer = new SpeechRecognitionEngine(new System.Globalization.CultureInfo("pt-br"));
    MemoryStream arquivo = new MemoryStream(arquivoWAV);
    StringBuilder resultado = new StringBuilder();
    try
    {
        recognizer.SetInputToWaveStream(arquivo);
        Grammar grammar = new Grammar(arquivoGramatica);
        grammar.Enabled = true;
        grammar.Name = "Gramatica 1";
        recognizer.LoadGrammar(grammar);
        result = recognizer.Recognize();
        if (result != null){
            int numAlternativas = result.Alternates.Count >= 3 ? 3 : result.Alternates.Count;
            for (int i = 0; i < numAlternativas; i++){
                resultado.Append(" " + result.Alternates[i].Text);
            }
        }
        else{
            resultado.Append("erro");
        }
    }
    catch (Exception e){
        resultado.AppendLine(e.Message);
    }
    finally {
        recognizer.UnloadAllGrammars();
        arquivo.Flush();
        arquivo.Dispose();
        recognizer.Dispose();
    }
    return resultado.ToString();
}
}

<grammar version="1.0" xml:lang="pt-BR" mode="voice" root="Palavras" xmlns="http://www.w3.org/2001/06/grammar">
  <rule id="Palavras" scope="public">
    <item>
      <one-of>
        <item>professor</item>
        <item>aluno</item>
        <item>senha</item>
        <item>login</item>
        <item>voltar</item>
        <item>entrar</item>
        <item>nome</item>
        <item>pesquisar</item>
        <item>apagar</item>
        <item>0</item><item>1</item>
        <item>2</item><item>3</item>
        <item>4</item><item>5</item>
        <item>6</item><item>7</item>
        <item>8</item><item>9</item>
        <item>a</item><item>b</item>
        <item>c</item><item>d</item>
        <item>e</item><item>f</item>
        <item>g</item><item>h</item>
        <item>i</item><item>j</item>
        <item>k</item><item>l</item>
        <item>m</item><item>n</item>
        <item>o</item><item>p</item>
        <item>q</item><item>r</item>
        <item>s</item><item>t</item>
        <item>u</item><item>v</item>
        <item>w</item><item>x</item>
        <item>y</item><item>z</item>
      </one-of>
    </item>
  </rule>
</grammar>

```

```

function wordDone(result) {

    $("#ouvir-wrap").show();
    var rec = false;
    var palavra = "";
    result = result.split(' ');
    debug("palavra " + palavra);
    debug("result " + result);
    $(".loader").hide();

    for (var i = 0; i < result.length; i++) {
        if (result[i].trim().toLowerCase() == "professor"
            || result[i].trim().toLowerCase() == "aluno"
            || result[i].trim().toLowerCase() == "senha"
            || result[i].trim().toLowerCase() == "login"
            || result[i].trim().toLowerCase() == "voltar"
            || result[i].trim().toLowerCase() == "entrar"
            || result[i].trim().toLowerCase() == "nome"
            || result[i].trim().toLowerCase() == "pesquisar"
            || result[i].trim().toLowerCase() == "apagar"
            || result[i].trim().toLowerCase() == "0" || result[i].trim().toLowerCase() == "1"
            || result[i].trim().toLowerCase() == "2" || result[i].trim().toLowerCase() == "3"
            || result[i].trim().toLowerCase() == "4" || result[i].trim().toLowerCase() == "5"
            || result[i].trim().toLowerCase() == "6" || result[i].trim().toLowerCase() == "7"
            || result[i].trim().toLowerCase() == "8" || result[i].trim().toLowerCase() == "9"
            || result[i].trim().toLowerCase() == "a" || result[i].trim().toLowerCase() == "b"
            || result[i].trim().toLowerCase() == "c" || result[i].trim().toLowerCase() == "d"
            || result[i].trim().toLowerCase() == "e" || result[i].trim().toLowerCase() == "f"
            || result[i].trim().toLowerCase() == "g" || result[i].trim().toLowerCase() == "h"
            || result[i].trim().toLowerCase() == "i" || result[i].trim().toLowerCase() == "j"
            || result[i].trim().toLowerCase() == "k" || result[i].trim().toLowerCase() == "l"
            || result[i].trim().toLowerCase() == "m" || result[i].trim().toLowerCase() == "n"
            || result[i].trim().toLowerCase() == "o" || result[i].trim().toLowerCase() == "p"
            || result[i].trim().toLowerCase() == "q" || result[i].trim().toLowerCase() == "r"
            || result[i].trim().toLowerCase() == "s" || result[i].trim().toLowerCase() == "t"
            || result[i].trim().toLowerCase() == "u" || result[i].trim().toLowerCase() == "v"
            || result[i].trim().toLowerCase() == "w" || result[i].trim().toLowerCase() == "x"
            || result[i].trim().toLowerCase() == "y" || result[i].trim().toLowerCase() == "z") {
            rec = true;
            palavra = result[i].trim().toLowerCase();
            break;
        }
    }
}

```

```

if (rec) {
    $("#Message").hide();
    if (palavra == "erro") { $("#Message").show(); }
    else if (palavra == "professor") { onClickBtnProfessor(); }
    else if (palavra == "aluno") { onClickBtnAluno(); }
    else if ((area == "Professor" || area == "Aluno") && (entrar == "false" || entrar == "")) {
        if (foco == "senha" && (palavra == "0" || palavra == "1" || palavra == "2" || palavra == "3"
            || palavra == "4" || palavra == "5" || palavra == "6" || palavra == "7" || palavra == "8" || palavra == "9")) {
            $('input[id$="txtSenha"]').val($('input[id$="txtSenha"]').val() + palavra);
            $('input[id$="txtSenha"]').focus();
        }
        else if (foco == "RA" && (palavra == "0" || palavra == "1" || palavra == "2" || palavra == "3" || palavra == "4"
            || palavra == "5" || palavra == "6" || palavra == "7" || palavra == "8" || palavra == "9")) {
            $('input[id$="txtRA"]').val($('input[id$="txtRA"]').val() + palavra);
            $('input[id$="txtRA"]').focus();
        }
        else if (palavra == "senha") { foco = "senha"; $('input[id$="txtSenha"]').focus(); }
        else if (palavra == "login" || palavra == "i" || palavra == "d" || palavra == "b") {
            foco = "RA"; $('input[id$="txtRA"]').focus();
        }
    }
    else if (palavra == "voltar") { onClickBtnVoltar32(); }
    else if (palavra == "entrar") { onClickBtnEntrar(); }
    else if (palavra == "apagar") { onClickBtnApagar(); }
    else { $("#Message").show(); }
} else if (entrar == "true") {
    if (palavra == "professor") { $('input[id$="rblBusca"]').val("1"); busca = "1"; }
    else if (palavra == "aluno") { $('input[id$="rblBusca"]').val("2"); busca = "2"; }
    else if (palavra == "nome") { foco = "nome"; $('input[id$="txtNome"]').focus(); }
    else if (palavra == "apagar") { onClickBtnApagar(); }
    else if (palavra == "pesquisar") { onClickBtnPesquisar(); }
    else if (foco == "nome"
        && (palavra == "0" || palavra == "1" || palavra == "2" || palavra == "3" || palavra == "4"
            || palavra == "5" || palavra == "6" || palavra == "7" || palavra == "8" || palavra == "9"
            || palavra == "a" || palavra == "b" || palavra == "c" || palavra == "d" || palavra == "e"
            || palavra == "f" || palavra == "g" || palavra == "h" || palavra == "i" || palavra == "j"
            || palavra == "k" || palavra == "l" || palavra == "m" || palavra == "n" || palavra == "o"
            || palavra == "p" || palavra == "q" || palavra == "r" || palavra == "s" || palavra == "t"
            || palavra == "u" || palavra == "v" || palavra == "w" || palavra == "x" || palavra == "y" || palavra == "z")) {
        $('input[id$="txtNome"]').val($('input[id$="txtNome"]').val() + palavra);
        $('input[id$="txtNome"]').focus();
    }
    else { $("#Message").show(); } else { $("#Message").show(); } else { $("#Message").show(); }
}

```

ANEXO E – PLAYBACK

```

public function playback()
{
    try
    {
        var sound: PCMSound;
        recorder.output.inflate();
        sound = new WAVSound( recorder.output, onPCMSoundComplete );
    }
    catch (e:Error)
    {
        log(e.message);
    }
}

private function onPCMSoundComplete( sound: PCMSound ): void
{
    sound.play();
    recorder.output.deflate();
}

```

ANEXO F – ONCLICKBTNPROFESSOR

```
function onClickBtnProfessor() {
    $('step2').effect('slide', { direction: 'down', mode: 'hide' }, 200, function () {
        $('step3').effect('slide', { direction: 'up', mode: 'show' }, 200);
        area = "Professor";
        $("#spArea").text("");
        $("#spArea").append(area);
    });
}
```

ANEXO G – ONCLICKBTNALUNO

```
function onClickBtnAluno() {
    $('step2').effect('slide', { direction: 'down', mode: 'hide' }, 200, function () {
        $('step3').effect('slide', { direction: 'up', mode: 'show' }, 200);
        area = "Aluno";
        $("#spArea").text("");
        $("#spArea").append(area);
    });
}
```

ANEXO H – ONCLICKBTNVOLTAR32

```
function onClickBtnVoltar32() {
    $('step3').effect('slide', { direction: 'down', mode: 'hide' }, 200, function () {
        $('step2').effect('slide', { direction: 'up', mode: 'show' }, 200);
        $('input[id$="txtRA"]').val("");
        $('input[id$="txtSenha"]').val("");
        foco = "";
        area = "";
        $("#spArea").append(area);
    });
}
```


ANEXO I – ONCLICKBTNAPAGAR

```
function onClickBtnApagar() {
  if (foco == "RA") {
    $('input[id$="txtRA"]').val($('input[id$="txtRA"]').val().toString().substring(0, $('input[id$="txtRA"]').val().length - 1));
    $('input[id$="txtRA"]').focus();
  } else if (foco == "senha") {
    $('input[id$="txtSenha"]').val($('input[id$="txtSenha"]').val().toString().substring(0, $('input[id$="txtSenha"]').val().length - 1));
    $('input[id$="txtSenha"]').focus();
  } else if (foco == "nome") {
    $('input[id$="txtNome"]').val($('input[id$="txtNome"]').val().toString().substring(0, $('input[id$="txtNome"]').val().length - 1));
    $('input[id$="txtNome"]').focus();
  }
}
}
```

ANEXO J – ENTRAR

```
function onClickBtnEntrar() {
  if ($('#input[id$="txtRA"]').val() == "") {
    $("#CorretamenteEntrar").show();
    $("#CorretamenteEntrar").text("");
    $("#CorretamenteEntrar").append("Preencha o campo RA.");
  } else if ($('#input[id$="txtSenha"]').val() == "") {
    $("#CorretamenteEntrar").show();
    $("#CorretamenteEntrar").text("");
    $("#CorretamenteEntrar").append("Preencha o campo Senha.");
  } else {
    Entrar(area);
  }
}
}
```

```

function Entrar(area) {
    $.ajax({
        type: "POST",
        url: "Inicio.aspx/Entrar", // url da pagina/nome do metodo
        contentType: "application/json; charset=utf-8",
        dataType: "json",
        data: "{area: '" + area + "',RA: '" +
            $('input[id$="txtRA"]').val() + "',senha: '"
            + $('input[id$="txtSenha"]').val() + '"", //parametros da função
        success: function (msg, status) {
            data = msg.d;
            if (data.indexOf("True") != -1) {
                $(''.step3').effect('slide', { direction: 'down', mode: 'hide' }, 200, function () {
                    $(''.step4').effect('slide', { direction: 'up', mode: 'show' }, 200);
                    entrar = "true";
                });
            } else if (data.indexOf("False") != -1) {
                $("#spArea").text("");
                $("#spArea").append(area + " - Preencha os campos corretamente.");
                entrar = "false";
            } else {
                $("#spArea").text("");
                $("#spArea").append(area + " - " + data);
                entrar = "false";
            }
        },
        complete: function () {
        }
    });
}

[WebMethod(EnableSession = true)]
[ScriptMethod(ResponseFormat = ResponseFormat.Json)]
public static string Entrar(string area, string RA, string senha)
{
    try
    {
        string xml = string.Empty;
        if (area == "Aluno")
        {
            xml = HostingEnvironment.MapPath("~/xml/Aluno.xml");
        }
        else if (area == "Professor")
        {
            xml = HostingEnvironment.MapPath("~/xml/Professor.xml");
        }

        var doc = XDocument.Load(xml);

        return
            doc.Descendants(area.ToUpper()).Any(
                order => order.Element("RA").Value == RA && order.Element("SENHA").Value == senha
                ? "True"
                : "False");
    }
    catch (Exception ex)
    {
        return ex.Message;
    }
}
}

```

```

<USUARIO>
  <ALUNO>
    <NOME>GUILHERME</NOME>
    <RA>123</RA>
    <SENHA>123456</SENHA>
  </ALUNO>
  <ALUNO>
    <NOME>EMERSON</NOME>
    <RA>456</RA>
    <SENHA>123456</SENHA>
  </ALUNO>
  <ALUNO>
    <NOME>VINICIUS</NOME>
    <RA></RA>
    <SENHA>789</SENHA>
  </ALUNO>
  <ALUNO>
    <NOME>ENEDY</NOME>
    <RA>111</RA>
    <SENHA>123456</SENHA>
  </ALUNO>
</USUARIO>

```

ANEXO K – PESQUISAR

```

function onClickBtnPesquisar() {
  if ($('#input[id$="txtNome"]').val() == "") {
    $("#Corretamente").show();
    $("#Corretamente").text("");
    $("#Corretamente").append("Preencha o campo nome.");
  } else if (busca == "") {
    $("#Corretamente").show();
    $("#Corretamente").text("");
    $("#Corretamente").append("Selecione professor ou aluno.");
  } else {
    $("#grid").text("");
    Pesquisar($('#input[id$="txtNome"]').val(), busca);
  }
}

```

```

function Pesquisar(nome,busca) {
$.ajax({
  type: "POST",
  url: "Inicio.aspx/Pesquisar", // url da pagina/nome do metodo
  contentType: "application/json; charset=utf-8",
  dataType: "json",
  data: "{nome: '" + nome + "', busca: '" + busca + "'}", //parametros da função
  success: function (msg, status) {
    data = msg.d;
    $("#grid").show();
    $("#Corretamente").hide();

    if (data != "") {
      //gridAlternatingRow
      $("#grid").append('<table class="grid">');
      $("#grid").append('<tr class="gridHeader"><th>Nome</th><th>Dia da semana</th><th>Disciplina</th><th>horário</th>');
      for (var i = 0; i < (data.split(',').length / 4); i++) {
        $("#grid").append("<tr class='gridRow'><td style='text-align:center'>" +
          data.split(',')[i * 4 + 0].toString().split(':')[1].toString().replace('"', ' ').replace("'", ' ') +
          "</td><td style='text-align:center'>"
          + data.split(',')[i * 4 + 1].toString().split(':')[1].toString().replace('"', ' ').replace("'", ' ') +
          "</td><td style='text-align:center'>"
          + data.split(',')[i * 4 + 2].toString().split(':')[1].toString().replace('"', ' ').replace("'", ' ') +
          "</td><td style='text-align:center'>"
          + data.split(',')[i * 4 + 3].toString().split(':')[1].toString().replace('"', ' ').replace("'", ' ') +
          "</td></tr>");
      }
      $("#grid").append("</table>");
    } else {
      $("#grid").append("<span><h2>Não foi localizado nenhum horário para este nome.</h2></span>");
    }
  },
  complete: function () {
  }
});
}

```

```

[WebMethod(EnableSession = true)]
[ScriptMethod(ResponseFormat = ResponseFormat.Json)]
public static string Pesquisar(string nome, string busca)
{
    try
    {
        string xml = HostingEnvironment.MapPath("~/xml/Horario.xml");
        var doc = XDocument.Load(xml);
        DataTable dtGrid = new DataTable();
        dtGrid.Columns.Add("nome");
        dtGrid.Columns.Add("dia");
        dtGrid.Columns.Add("disciplina");
        dtGrid.Columns.Add("horario");
        foreach (XElement order in doc.Descendants(busca == "1" ? "PROFESSOR" : "ALUNO").Descendants("AULA"))
        {
            if (order.Element("NOME").Value.ToLower().Contains(nome.ToLower()))
            {
                DataRow row = dtGrid.NewRow();
                row["nome"] = (string)order.Element("NOME");
                row["dia"] = (string)order.Element("DIA");
                row["disciplina"] = (string)order.Element("DISCIPLINA");
                row["horario"] = (string)order.Element("HORA");
                dtGrid.Rows.Add(row);
            }
        }
        List<Dictionary<string, object>> list = new List<Dictionary<string, object>>();
        foreach (DataRow row in dtGrid.Rows)
        {
            Dictionary<string, object> dict = new Dictionary<string, object>();
            foreach (DataColumn col in dtGrid.Columns)
            {
                dict[col.ColumnName] = row[col];
            }
            list.Add(dict);
        }
        JavaScriptSerializer serializer = new JavaScriptSerializer();
        return serializer.Serialize(list);
    }
    catch (Exception ex)
    {
        throw;
    }
}

```

```

<HORARIO>
  <ALUNO>
    <AULA><NOME>Guilherme</NOME>
      <DIA>Terca</DIA>
      <DISCIPLINA>Redes Neurais</DISCIPLINA>
      <HORA>19:00 - 20:30</HORA></AULA>
    <AULA><NOME>Guilherme</NOME>
      <DIA>Quinta</DIA>
      <DISCIPLINA>Topicos especiais III</DISCIPLINA>
      <HORA>19:00 - 22:15</HORA></AULA>
    <AULA><NOME>Emerson</NOME>
      <DIA>Terca</DIA>
      <DISCIPLINA>Redes Neurais</DISCIPLINA>
      <HORA>19:00 - 20:30</HORA></AULA>
    <AULA><NOME>Emerson</NOME>
      <DIA>Quinta</DIA>
      <DISCIPLINA>Topicos especiais III</DISCIPLINA>
      <HORA>19:00 - 22:15</HORA></AULA>
  </ALUNO>
  <PROFESSOR>
    <AULA><NOME>Larissa</NOME>
      <DIA>Terca</DIA>
      <DISCIPLINA>Orientacao</DISCIPLINA>
      <HORA>19:00 - 20:30</HORA></AULA>
    <AULA><NOME>Larissa</NOME>
      <DIA>Quinta</DIA>
      <DISCIPLINA>Topicos especiais III</DISCIPLINA>
      <HORA>19:00 - 22:15</HORA></AULA>
    <AULA><NOME>Patricia</NOME>
      <DIA>Terca</DIA>
      <DISCIPLINA>Redes Neurais</DISCIPLINA>
      <HORA>19:00 - 20:30</HORA></AULA>
    <AULA><NOME>Patricia</NOME>
      <DIA>Quinta</DIA>
      <DISCIPLINA>Orientacao</DISCIPLINA>
      <HORA>19:00 - 22:15</HORA></AULA>
  </PROFESSOR>
</HORARIO>

```