



Escola Superior de Tecnologia
Instituto Politécnico de Castelo Branco
Licenciatura em Engenharia Informática

Microsoft Speech API (SAPI)

Processamento de Voz

Engenharia Informática

Nome:	Número	e-mail
António Afonso	177/01	aafonso@areadeservico.com
Nelson Vicente	206/01	nelsonvicente@gaveta.net
Ricardo Antunes	160/01	ric_antunes@areadeservico.com
Versão Online: http://sapi.areadeservico.com		

Maio 2007

Índice

Índice de Figuras	3
Índice de Tabelas.....	3
1. Introdução.....	4
1.1. Recursos Utilizados	4
2. Microsoft Speech API (SAPI)	5
2.1. Introdução ao SAPI.....	5
2.2. Vista Geral	6
2.3. Arquitectura	7
2.4. Visão da Microsoft.....	10
2.5. Breve Comparação entre as várias Releases	11
2.6. Projectos - Microsoft Speech Research Group	14
2.7. Ferramentas de Desenvolvimento.....	19
2.8. Programa Demonstrativo Desenvolvido	23
3. Conclusão	25
4. Bibliografia.....	26
5. Referências WWW.....	27

Índice de Figuras

Figura 1 – Vista Geral do SAPI	6
Figura 2 – Arquitectura do Microsoft Speech SDK	8
Figura 3 – MiPAD	15
Figura 4 – Interface da Aplicação Demonstrativa	24

Índice de Tabelas

Tabela 1 – Exemplos da linguagem de marcação do SAPI 4	12
Tabela 2 – Exemplos da linguagem de marcação do SAPI 5	13
Tabela 3 – Comparação entre várias API's	21

1. Introdução

A rápida disseminação dos computadores pessoais, hoje facilmente acessíveis para a maioria das pessoas, conjuntamente com a sua grande evolução e influência, faz com que exista uma séria necessidade de facilitar a comunicação e interação entre o homem e a máquina.

As técnicas de síntese e reconhecimento de voz, têm sido introduzidas cada vez mais em aplicações, com o intuito de melhorar as interfaces com os utilizadores, de produzir novos mecanismos de interação com computadores e ainda com o intuito de abranger um maior número de utilizadores.

Um dos maiores desafios de um conversor texto-fala prende-se com a necessidade de interpretar correctamente as palavras e o seu contexto, de modo a que lhe seja permitido concretizar um pré-processamento eficaz e livre de erros, assim como a aplicação com padrões adequados. Toda esta lógica contida num conversor texto-fala implica custos adicionais de processamento, e torna-se mais falível à medida que se necessita de mais inteligência por parte do conversor.

1.1. Recursos Utilizados

Para a realização deste relatório foram utilizados vários recursos, nomeadamente o computador, internet, o Speech SDK 5.1e o Visual Studio.NET 2003.

2. Microsoft Speech API (SAPI)

2.1. Introdução ao SAPI

O Speech Application Programming Interface, ou SAPI, é uma API desenvolvida pela Microsoft para permitir o uso do Speech Recognition e Speech Synthesis nas aplicações Windows.

A Microsoft fundou um grupo de desenvolvimento de aplicações de fala em 1993. Em 1995, este grupo publica o SAPI1 como uma plataforma de desenvolvimento de aplicações baseada na fala para o Windows. Seguiu-se o SAPI 2, SAPI 3 e SAPI 4 em 1998. Nesta altura o grupo foi transferido para o núcleo de desenvolvimento de Speech.NET, onde publicou o SAPI 5 e o SAPI 5.1 em 2001. O grande objectivo destas aplicações é a função de interface com a plataforma do Windows, fornecendo o serviço de motores de conversão texto-fala e reconhecimento de fala. Actualmente o Windows 2000 e o Windows XP já integram nas suas distribuições a plataforma SAPI.

De entre todas as versões distribuídas pela Microsoft, o SAPI4 e o SAPI5 são os mais conhecidos e utilizados entre a comunidade empresarial e científica.

Algumas das aplicações que usam o SAPI são o Microsoft Office, o Microsoft Agent, o Microsoft Speech Server entre muitas outras. O SAPI é um componente distribuído gratuitamente que pode ser usado em qualquer aplicação do Windows que necessite de tecnologia de reconhecimento de voz. Várias versões (contudo, nem todas) de reconhecimento de voz e de sintetizadores são também distribuídas gratuitamente.

2.2. Vista Geral

De uma forma geral todas as versões da API foram projectadas para que um programador de software possa desenvolver uma aplicação que permita o reconhecimento da voz usando um conjunto standard de funcionalidades disponíveis para várias linguagens de programação. Além disso, é possível uma empresa desenvolver os seus próprios Speech Recognition e motores Text-To-Speech ou adaptar os motores já existentes para trabalhar com o SAPI.¹

O SAPI API fornece uma relação de alto nível entre uma aplicação e os motores de voz. O SAPI executa todos os detalhes de baixo nível que são necessários para controlar e monitorizar as operações em tempo real dos vários motores de voz. Os dois tipos básicos de motores SAPI são: text-to-speech (TTS) e Speech Recognizers(SR).

Os sistemas de TTS sintetizam frases escritas e ficheiros em áudio usando vozes sintéticas. Os identificadores de voz convertem a voz de um humano em frases (strings) e ficheiros.

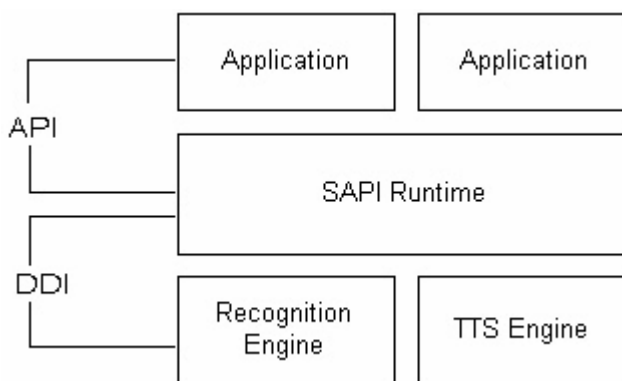


Figura 1 – Vista Geral do SAPI

¹ <http://msdn2.microsoft.com/en-us/library/ms720151.aspx>

2.3. Arquitectura

O Speech API pode ser visto como uma interface ou um middleware, ou seja, situa-se entre as aplicações e os motores de voz. Nas versões 1 a 4 do SAPI, as aplicações podiam comunicar directamente com os motores, a API continha uma definição de interface abstracta que as aplicações e os motores tinham de utilizar. As aplicações podiam também usar objectos higher-level simplificados que permitiam chamar directamente métodos nos motores.²

No SAPI 5, as aplicações e os motores não comunicam directamente um com o outro. Em vez disso cada conversa esta relacionada com um componente runtime (sapi.dll). Existe uma API implementada por este componente que é usado pelas aplicações, e um outro conjunto de interfaces usado para os motores.

Tipicamente no SAPI 5 as aplicações emitem chamadas para a API (por exemplo: iniciar o reconhecimento de voz; ou fornecer o texto para ser sintetizado), o componente sapi.dll runtime interpreta estes comandos e processa-os, onde se torna necessário chamar o Speech Recognizers através do motor de interfaces (por exemplo, carregar uma palavra de um ficheiro é feito no runtime, mas os dados da palavra são passados ao Speech Recognizers). Os motores de reconhecimento e de síntese geram também eventos ao processar esses comandos.

Os seguintes componentes estão incluídos na maioria das versões do Speech SDK:

- API definition files – C ou C++ header files,
- Runtime components – exemplo: sapi.dll,
- Control Panel applet – para seleccionar e configurar o Speech Recognizer,
- Text-To-Speech engines em várias línguas,
- Speech Recognition engines em várias línguas,
- Redistributable components para permitir aos programadores utilizarem os vários motores nas suas aplicações,
- Códigos Exemplos,
- Engines exemplos,

² http://en.wikipedia.org/wiki/Speech_Application_Programming_Interface

- Documentação.

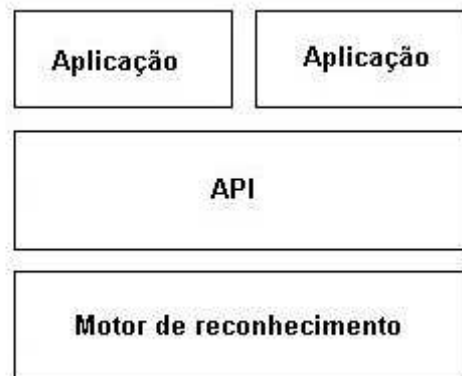


Figura 2 – Arquitectura do Microsoft Speech SDK

2.3.1. API Para Text-To-Speech

A classe `ISpVoice` que se encontra disponível na SAPI permite controlar o processo de síntese de voz. Para isso é necessário criar um objecto para instanciar esta classe. De seguida basta invocar o método `ISpVoice::Speak` para poder reproduzir o áudio a partir de um texto ou de um ficheiro.³

Através do método `SetVoice` podemos definir as características da voz que irá ser reproduzida pelo sintetizador, nomeadamente a língua / idioma, sexo e idade.

Como mostra o exemplo a seguir, podemos configurar o volume e a velocidade de saída do som através dos métodos `SetVolume` e `SetRate`:

```
SpVoice Voice = new SpVoice();  
Voice.Volume = 50; Voice.Rate = 5;  
Voice.Speak("Hello", SpeakFlags.SVSFlagsAsync);  
Voice.Speak("C:/Teste.txt", SpeakFlags.SVSFIIsFilename);
```

³ <http://msdn2.microsoft.com/en-us/library/ms720151.aspx>

2.3.2. API para reconhecimento de voz

Uma aplicação de reconhecimento de voz necessita de ter uma gramática, definida à priori, para que quando lhe for enviada uma palavra, esta possa ser reconhecida.

O reconhecimento de voz é feito através da interface ISpRecoContext, depois de criado o objecto a aplicação passa a ter todo o controle do conteúdo e da gramática de reconhecimento. Esta tem também a capacidade de parar e voltar ao reconhecimento.⁴

No fim da gramática estar definida, a aplicação necessita de saber o que foi reconhecido, para isso utiliza-se o evento RecoContext.Recognition.

⁴ <http://msdn2.microsoft.com/en-us/library/ms720151.aspx>

2.4. Visão da Microsoft

Na perspectiva da Microsoft a integração de engenhos de reconhecimento e síntese de voz em equipamentos informáticos num número cada vez maior de dispositivos portáteis tais como PDAs (Personal Digital Assistant), Tablet PCs, smart phones, livros digitais e até telemóveis, tem sido um desafio e ao mesmo tempo uma vitória.⁵

A visão da Microsoft inclui também a junção de sistemas de voz com a Web numa só infra-estrutura, usando ferramentas de desenvolvimento estandardizadas e baseadas na Web, que consigam correr na nossa actual infra-estrutura Web.

⁵ <http://www.microsoft.com/speech/evaluation/overview/default.aspx>

2.5. Breve Comparação entre as várias Releases

Existem três grandes diferenças entre as duas distribuições:

- **Arquitetura** – No SAPI5 o módulo de conversão texto-fala encontra-se separado do módulo que guarda as propriedades e regras da voz. No SAPI4 não existe separação destes módulos.
- **Linguagem de marcação** – No SAPI5 a linguagem de marcação é baseada em XML, enquanto que no SAPI4 é baseada numa linguagem própria.
- **Painel de controlo** – No SAPI5 existe um painel de controlo centralizado com as propriedades genéricas do conversor texto-fala. O mesmo não acontece no SAPI4.

2.5.1. Release 4

As características principais da API incluem:⁶

- **Voice Command** - objectos do Speech Recognition para o comando e controle por voz,
- **Voice Dictation**- objectos do Speech Recognition para reconhecimento contínuo do discurso.
- **Voice Talk** - objectos para a síntese do discurso.
- **Voice Telephony** - objectos para aplicações reconhecimento de voz pelo telefone.
- **Direct Speech Recognition** - objectos para o controle directo do motor de reconhecimento.
- **Direct Text To Speech** - objectos para o controle directo do motor de síntese.
- **Audio objects** - objecto para ler de um dispositivo áudio ou de um ficheiro.

⁶ http://en.wikipedia.org/wiki/Speech_Application_Programming_Interface

Propriedade	Uso da Marca
Volume	\vol=65535\ (valor máximo)
Velocidade	\spd=10\ , número de palavras por minuto
Frequência F0	\pit=200\ , valor em Hertz
Pausa	\pau=1000\ , número de milissegundos

Tabela 1 – Exemplos da linguagem de marcação do SAPI 4

2.5.2. Release 5

As características principais da API incluem:^{7 8}

- Shared Recognizer - Para aplicações do ambiente trabalho.
- In-proc recognizer - Para aplicações que requerem um controlo específico no processo de reconhecimento.
- Grammar objects - Gramática que é definida à priori, para que quando lhe for enviada uma palavra, esta possa ser reconhecida.
- Voice object - Executa a síntese do discurso, produzindo áudio de um texto.
- Áudio interfaces - O runtime inclui objectos para executar o discurso de input do microfone ou o discurso de output aos altifalantes.
- User lexicon object - Isto permite a um utilizador ou aplicação introduzir palavras personalizadas.

⁷ <http://www.w3.org/TR/speech-synthesis/>

⁸ http://msdn2.microsoft.com/en-us/library/ms723627.aspx#New_Interfaces

Propriedade	Uso da Marca
Volume	<volume level="100"> teste </volume> 0 - mínimo 100 - máximo +/- é possível (volume relativo)
Velocidade	<rate abspeed="-5"> teste </rate> valor entre -10 (1/3 do valor por defeito) e 10 (3x do valor por defeito) 0 - coloca o valor por defeito
Frequência F0	<pitch absmiddle="5" > teste </pitch> valor entre -10 (1/3 do valor por defeito) e 10 (3x do valor por defeito) 0 - coloca o valor por defeito
Pausa	<silence msec="3000" /> número de milissegundos desde 0 a 65535

Tabela 2 – Exemplos da linguagem de marcação do SAPI 5

2.6. Projectos - Microsoft Speech Research Group

A Microsoft tem actualmente dois centros de investigação e desenvolvimento de tecnologias da fala. Estes centros estão localizados em Redmond, nos Estados Unidos da América e outro em Beijing, na China e trabalham em conjunto para melhorar tecnologias da fala.

O objectivo principal deste grupo é criar aplicações que possibilitem o uso de computadores em toda a parte e que trabalhem com plataformas e tecnologias de Speech Recognizing. Ou seja, pretende-se com todo este esforço, criar um computador inteiramente Speech Enabled, através da interacção Homem - Máquina.

Para melhor compreensão da visão da Microsoft nesta área, temos alguns vídeos. É aconselhável a sua visualização para melhor compreensão. Estes vídeos demonstram alguns dos projectos desenvolvidos ou ainda em fase de desenvolvimento. Estão disponíveis em: <http://research.microsoft.com/stg/videos/>

2.6.1. Projectos Desenvolvidos

No passado, este grupo de desenvolvimento trabalhou em alguns dos projectos disponíveis actualmente. Estes projectos foram completados com sucesso e estão disponíveis para download ou equipam produtos específicos, por exemplo Tablet's PC, PDA ou Smartphones.

Alguns exemplos são:

- MiPAD – Multimodal Interactive Pad⁹

Foi o primeiro protótipo multimodal. O grupo de pesquisa começou o trabalho do MiPAD em 1998 e teve uma primeira demonstração pública por 2000.

⁹ <http://research.microsoft.com/stg/mipad.aspx>

Esta aplicação permitia, através da fala, efectuar diversas tarefas, como por exemplo: enviar um correio electrónico, criar uma entrada na agenda, usar uma calculadora, efectuar uma chamada, entre outras.

Imaginemos alguém com problemas a nível das mãos. Todos nós sabemos o quanto é complicado usar um PDA e a sua respectiva caneta. Esta aplicação era perfeita para estes casos. No entanto, tal como os autores indicam, ficaram por resolver problemas ao nível do ruído, acentos, etc. Foi o primeiro passo!

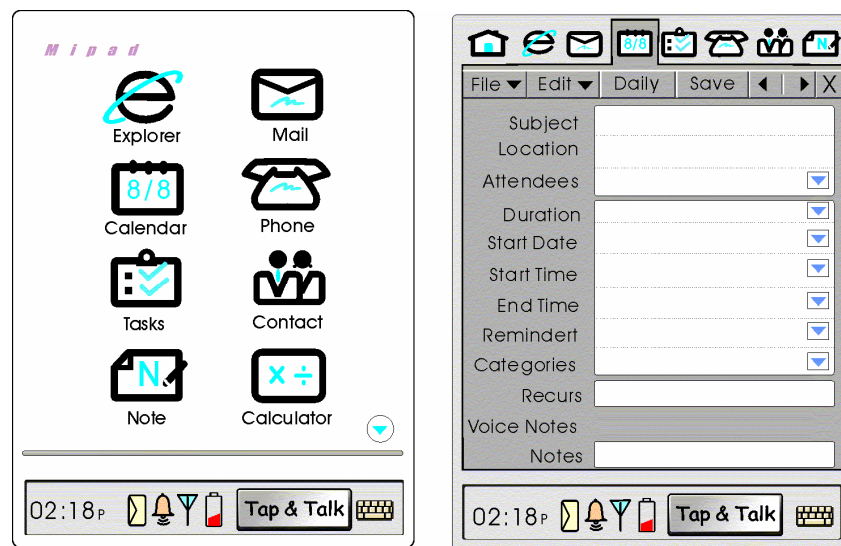


Figura 3 – MiPAD

Vídeo demonstrativo:

http://research.microsoft.com/stg/videos/MIPADDemo_4min_300k.wmv

- Whisper – Speech Recognition¹⁰

Nome de Código: Windows Highly Intelligent SPEech Recognizer)

A ideia é aperfeiçoar os sistemas de modo a que o computador consiga compreender qualquer tipo de fala. Este motor foi usado no Microsoft Phone, Microsoft Agent, Microsoft Encarta, Windows 2000, Office XP e Windows XP.

Permite-nos falar para o computador e este escreve o que nós ditamos.

¹⁰ <http://research.microsoft.com/stg/srproject.aspx>

Temos um vídeo demonstrativo da versão do Office XP em Japonês. Neste vídeo é possível demonstrar muitas das funcionalidades deste motor. Podemos praticamente criar um documento do Word, editá-lo e imprimi-lo apenas com o recurso à nossa voz.

Vídeo demonstrativo:

http://research.microsoft.com/stg/videos/MGBSDN_Japan_500k_take1_large.wmv

- Whistler – Text to Speech¹¹

Nome de Código: Windows Highly Intelligent STochastic taLkER

Um sintetizador de voz, de modo que um computador possa comunicar com um humano.

- WhisperID – Speaker Identification¹²

Esta aplicação pretende responder a perguntar: “Quem é que está a falar?”.

Cada ser humano tem uma voz diferente.

- Speech Application Programming Interface (SAPI) Development Toolkit¹³

Whisper speech recognizer, pode ser usado por programadores, na produção de aplicações que usem reconhecimento de voz.

2.6.2. Projectos em Desenvolvimento

Actualmente este grupo de desenvolvimento está a trabalhar em projectos inovadores nomeadamente:

- Noise Robustness¹⁴

Como fazer com que o sistema funcione quando está presente ruído de fundo?

¹¹ <http://research.microsoft.com/stg/ssproject.aspx>

¹² <http://research.microsoft.com/stg/whisperid.aspx>

¹³ <http://research.microsoft.com/stg/sapi.aspx>

¹⁴ <http://research.microsoft.com/stg/robust.aspx>

- Microphone Arrays¹⁵

Redução de ruído através de microphone arrays.

Um sistema com vários microfones devidamente posicionados pode ajudar a clarificar o sinal capturado. Este sistema com “closely-positioned microphones” é chamado microphone arrays. O Windows Vista já vem preparado para este sistema.

- Dereverberation¹⁶

Combater a problemática das ondas paralelas e das ondas reflectidas.

Por exemplo num debate, o sistema deverá saber qual é a voz do orador principal.

- Acoustic Modeling¹⁷

Como nós modelamos telefones e variações acústicas?

- Language Modeling¹⁸

Como são provavelmente as palavras?

- Automatic Grammer Induction¹⁹

O investigador deste projecto, Ye-Yi Wang, quer ter mais tempo para férias, assim está a ensinar o seu computador a fazer algum trabalho por ele.

Pretende-se desenvolver tecnologias capazes de geração automática da gramática, aprendizagem da anotação semântica e detectar e adaptar os constrangimentos de cada linguagem.

¹⁵ <http://research.microsoft.com/users/ivantash/MicrophoneArrayProject.aspx>

¹⁶ <http://research.microsoft.com/users/ivantash/DereverberationProject.aspx>

¹⁷ <http://research.microsoft.com/stg/acoustic-modeling.aspx>

¹⁸ <http://research.microsoft.com/stg/language-modeling.aspx>

¹⁹ <http://research.microsoft.com/stg/grammar.aspx>

- SALT (Speech Enabled Language Tags)²⁰

Markup Language para a Web (Multimodal).

“SALT extends existing Web Markup languages to enable multimodal and telephony Access to Web”

- Multimodal Conversational User Interface²¹
- Personalized Language Model for improved accuracy²²

²⁰ <http://research.microsoft.com/stg/salt.aspx>

²¹ <http://research.microsoft.com/stg/slu.aspx>

²² <http://research.microsoft.com/stg/PersonalizedLM.aspx>

2.7. Ferramentas de Desenvolvimento

Existem no mercado um grande número de API's que permitem aos programadores adicionarem às suas aplicações dispositivos de síntese de voz. Para escolher qual a API a usar numa determinada aplicação, é necessário definir quais as necessidades e efectuar uma avaliação em determinados critérios.

Estes critérios podem ser:

- Output das amostras de áudio – é importante que a API permita o redireccionamento das amostras de áudio obtidas dos sintetizadores de voz (ex. microfone) para além da saída de áudio padrão (ex. ficheiros).
- Facilidade de uso – É também uma mais valia a API escolhida ser de fácil integração e utilização. Desta maneira torna-se possível a obtenção de um máximo partido da API.
- Configuração de parâmetros - É de referenciar a necessidade de configuração de parâmetros de síntese de voz como velocidade, afinação, timbre, língua, etc.
- Linguagem de programação – Esta propriedade define o tipo de linguagem em que esta API foi desenvolvida e para que linguagens se destina, ou possa ser embutida.
- Portabilidade - É ideal que a API escolhida possua propriedades de portabilidade, para que esta possa ser utilizada em qualquer tipo de sistema operativo.

2.7.1. Comparativo entre API's

Dentro daquelas disponíveis no mercado foi feita uma avaliação das APIs Java Speech [JS98], MS SAPI (Microsoft Speech API) [Microsoft00], e ECI (Eloquence Command Interface) [IBM00] levando-se em conta os aspectos supracitados. Alguns critérios tais como a qualidade dos engenhos de síntese de voz que podem ser manipulados com a API e facilidade de uso foram avaliados de forma subjectiva.

A tabela seguinte sintetiza os resultados obtidos.

Esta tabela foi adaptada de um artigo disponível na internet.²³

Critério	JavaSpeech	MS SAPI	ECI
As amostras de áudio podem ser redireccionadas para arquivos e/ou outros dispositivos?	Não	Sim	Sim
A API está acoplada a um único engenho de síntese de voz?	Não	Não	Sim
Como é classificada a qualidade dos resultados produzidos pelos engenhos de síntese de voz que podem ser manipulados com a API?	Boa	Boa	Boa
Como é classificado o aprendizado e o emprego da API no desenvolvimento de aplicações?	Fácil	Difícil	Muito fácil
A API permite que os parâmetros do engenho de síntese de voz sejam configurados?	Sim	Sim	Sim
Qual a linguagem na qual a API	Java	C++	C

²³ <http://www.cin.ufpe.br/~tg/2000-2/sma.doc>

foi desenvolvida?			
Para quais linguagens a API possui bindings?	Nenhuma linguagem	Qualquer ambiente de desenvolvimento que possa manipular componentes COM (Componente Object Model)	Nenhuma linguagem
Em quais ambientes de desenvolvimento a API pode ser utilizada?	JDK (Java Development Kit)	Visual C++	Visual C++, C++ Builder e GCC
Em quais sistemas operacionais a API pode ser utilizada?	Qualquer sistema operacional para o qual exista uma máquina virtual Java	Windows	Windows e Linux

Tabela 3 – Comparação entre várias API's

2.7.2. Speech SDK 5.1

O Microsoft Speech SDK é um kit de desenvolvimento de software que permite a aplicações escritas para Windows, em diversas linguagens de programação (C/C++, C#, JavaScript e Visual Basic), terem o acesso a recursos como o reconhecimento e síntese de voz. Este kit esconde os pormenores de concretização de baixo nível ao programador, fazendo com que este se concentre apenas na lógica do reconhecimento das palavras.²⁴

A actual implementação desta biblioteca de programação suporta o reconhecimento de vocábulos na língua inglesa, chinesa e japonesa.

No Microsoft Speech SDK existem dois tipos de gramática: de ditado e de controlo.

Nas gramáticas de ditado não é necessário indicar as palavras a serem reconhecidas pois o sistema comporta-se como um analisador contínuo de discurso livre. Isto significa que conforme o utilizador vai falando o Speech Recognizer vai tentando reconhecer todas as palavras pronunciadas.

²⁴ <http://www.di.fc.ul.pt/disciplinas/pei/pei0405/conteudo/documentos/relatorios-0405/amadeus-dias-28300.pdf>

Já nas gramáticas de controlo o comportamento é deveras diferente. Neste tipo de reconhecimento de voz o programador tem que especificar todas as palavras que pretende que o motor de reconhecimento detecte. Apenas serão reconhecidas as palavras incluídas nesse leque e não outras quaisquer. Para esse efeito é definida uma gramática num ficheiro XML com uma sintaxe própria

2.7.3. Microsoft Speech Application Software Development Kit

Qualquer programador pode utilizar o Microsoft Speech Application SDK (SASDK) Versão 1.1, de uma forma rápida e eficaz, de modo a incorporar interfaces de TTS e STT às aplicações Web feitas na plataforma ASP.NET. As ferramentas de desenvolvimento do SASDK incluem suporte para a especificação SALT (Speech Application Language Tags).

Este kit é facilmente integrado no Visual Studio .NET 2003, ferramenta bastante conhecida pelos programadores, permitindo criar aplicações Web speech-enabled para os mais variados dispositivos, por exemplo computadores pessoais, Tablet PC's, PDA, Smartphones, entre outros.

O SASDK disponibiliza:

- Gramática de voz bastante completa
- Ferramentas de Debugging
- Documentação
- Ferramentas de análise
- ASP.NET Speech Controls
- Exemplos e aplicações demonstrativas
- Speech Application Deployment Service (SADS)
- Speech Add-in para o Microsoft Internet Explorer

2.8. Programa Demonstrativo Desenvolvido

Para este trabalho foi desenvolvido um pequeno programa, no formato página Web. Com o auxílio do Visual Studio.NET e com o Speech SDK 5.1 fizemos uma pequena página HTML onde é possível testar a funcionalidade Text-To-Speech. Partimos de um exemplo disponível no Speech SDK 5.1.

Nós tentámos usar Microsoft Speech Application SDK (SASDK) com suporte para SALT, num Web site criado por nós, mas não conseguimos. Neste site pretendíamos testar tanto a funcionalidade TTS como STT.

O programa de instalação do SASDK dava sempre erro. Formata-mos várias vezes o PC, instalámos todos os pré-requisitos e mesmo assim não foi possível. Pensamos que talvez seja por alguma incompatibilidade de Hardware, nomeadamente com a placa de som.

No entanto o exemplo demonstra as capacidades do SAPI, no que diz respeito ao Text-To-Speech.

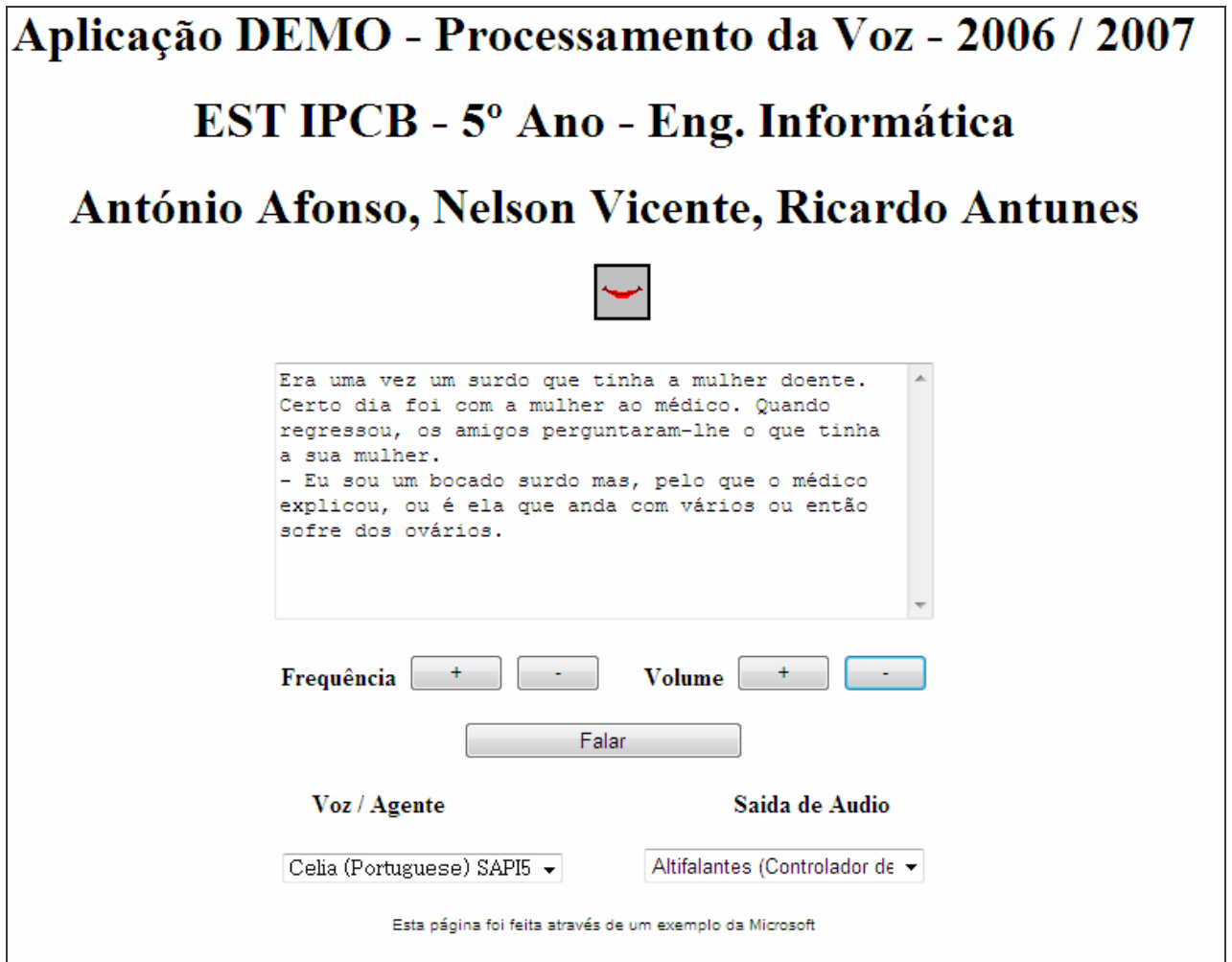


Figura 4 – Interface da Aplicação Demonstrativa

Nesta pequena aplicação demonstrativa podemos seleccionar o Agente / Voz, a Saída de Áudio, a Frequência e o Volume. Existe ainda uma área para escrever o texto que pretendemos que seja reproduzido.

3. Conclusão

Podemos verificar que esta é uma área cada vez mais em expansão, com o intuito de melhorar e facilitar as nossas tarefas quotidianas. É certo que ainda esta numa fase de evolução mas mesmo assim já consegue, e com algum grau de certeza, obter resultados bastante promissores.

A tecnologia de voz possui grande potencial para a criação de aplicativos que possibilitem uma eficaz interação humano-computador, através das tecnologias de síntese de voz e de reconhecimento de voz.

As técnicas de síntese e reconhecimento de voz têm sido empregadas com uma frequência cada vez maior, tanto na computação pessoal quanto na corporativa, com o intuito de melhorar as interfaces com os usuários existentes e de produzir novos mecanismos de interação com computadores.

4. Bibliografia

Não foram consultadas quaisquer fontes bibliográficas durante a realização deste trabalho.

5. Referências WWW

Lista de referências WWW consultadas ou apresentadas durante a realização do trabalho.

- [1] <http://msdn2.microsoft.com/en-us/library/ms720151.aspx>
- [2] http://en.wikipedia.org/wiki/Speech_Application_Programming_Interface
- [3] <http://msdn2.microsoft.com/en-us/library/ms720151.aspx>
- [4] <http://msdn2.microsoft.com/en-us/library/ms720151.aspx>
- [5] <http://www.microsoft.com/speech/evaluation/overview/default.aspx>
- [6] http://en.wikipedia.org/wiki/Speech_Application_Programming_Interface
- [7] <http://www.w3.org/TR/speech-synthesis/>
- [8] http://msdn2.microsoft.com/en-us/library/ms723627.aspx#New_Interfaces
- [9] <http://research.microsoft.com/stg/mipad.aspx>
- [10] <http://research.microsoft.com/stg/srproject.aspx>
- [11] <http://research.microsoft.com/stg/ssproject.aspx>
- [12] <http://research.microsoft.com/stg/whisperid.aspx>
- [13] <http://research.microsoft.com/stg/sapi.aspx>
- [14] <http://research.microsoft.com/stg/robust.aspx>
- [15] <http://research.microsoft.com/users/ivantash/MicrophoneArrayProject.aspx>
- [16] <http://research.microsoft.com/users/ivantash/DereverberationProject.aspx>
- [17] <http://research.microsoft.com/stg/acoustic-modeling.aspx>
- [18] <http://research.microsoft.com/stg/language-modeling.aspx>
- [19] <http://research.microsoft.com/stg/grammar.aspx>
- [20] <http://research.microsoft.com/stg/salt.aspx>
- [21] <http://research.microsoft.com/stg/slu.aspx>
- [22] <http://research.microsoft.com/stg/PersonalizedLM.aspx>
- [23] <http://www.cin.ufpe.br/~tg/2000-2/sma.doc>
- [24] <http://www.di.fc.ul.pt/disciplinas/pei/pei0405/conteudo/documentos/relatorios-0405/amadeus-dias-28300.pdf>