



**Departamento
de Engenharia
Informática**

Relatório Final do Trabalho Final de Curso

**LICENCIATURA EM ENGENHARIA
INFORMÁTICA E DE COMPUTADORES (LEIC)**

Ano Lectivo 2002 / 2003

Trabalho Nº 61

Utilização da fala para preenchimento de formulários num PDA

Professor Orientador:

João Paulo Neto

Co-Orientador:

Luís Caldas de Oliveira

Alunos:

nº 46813, Armando Jorge Cardoso Marques

nº 46985, Sofia Isabel dos Reis Valente

**Trabalho realizado no INESC-ID Lisboa no âmbito
do Laboratório de Língua Falada (L²F)**

Resumo

O objectivo do trabalho é usar a fala para preencher formulários num PDA. Como os métodos de entrada de dados neste dispositivo (teclado virtual, reconhecedor de letras, etc.) são pouco práticos, a fala surge como um método de entrada de dados mais intuitivo e fácil de utilizar. Os formulários a serem preenchidos através da fala são formulários médicos e correspondem a fichas de doentes. Estes formulários são implementados numa aplicação que permite preencher, ler, alterar e apagar fichas de doentes.

O trabalho compreende uma primeira fase em que se desenvolve uma aplicação de preenchimento de formulários médicos, ainda sem suporte de fala. A interface do formulário e a aquisição de dados são realizadas no PDA. No entanto, devido às limitações tecnológicas do PDA, o armazenamento e gestão de informação dos doentes é realizado num servidor externo. Nesta fase é dado ênfase à distribuição de tarefas e ao protocolo de comunicação.

Numa segunda fase pretende-se realizar o reconhecimento e síntese de fala. Começa-se por utilizar sistemas já existentes de reconhecimento e de síntese de fala para Português Europeu colocados num sistema central. Seguidamente pretende-se transferir um conjunto de tarefas associadas aos sistemas de reconhecimento e síntese de fala do sistema central para o PDA. Evoluindo de um processamento distribuído para um processamento local no dispositivo.

Palavras-chave

Reconhecimento de fala em dispositivos móveis, processamento automático de fala, preenchimento de formulários através da fala.

Agradecimentos

Em primeiro lugar os nossos agradecimentos são dirigidos ao nosso orientador Professor João Paulo Neto e ao nosso co-orientador Professor Luís Caldas de Oliveira a quem devemos a oportunidade de trabalhar na área do reconhecimento de fala. Queremos ainda agradecer toda a sua ajuda e colaboração ao longo deste trabalho.

Gostávamos de agradecer também, ao Eng. Rui Maia pelas sempre pacientes explicações fornecidas, fundamentais para a compreensão do sistema de reconhecimento de fala desenvolvido no INESC-ID.

Ao Eng. Renato Casaca, pelo apoio dado na construção dos mecanismos de suporte à gestão de informação.

Ao Eng. Hugo Meinedo, pelo seu esforço e empenho na resolução dos nossos problemas relacionados com o modelo acústico.

Ao Eng. David Matos, pelo auxílio na resolução dos problemas relacionados com a transição para o sistema AFS.

Ao INESC-ID (Instituto de Engenharia de Sistemas e Computadores) e ao IST (Instituto Superior Técnico).

Agradecemos à Fundação para Ciência e a Tecnologia (FCT) pelo apoio prestado através do projecto POSI/33846/2000 – “*Large vocabulary continuous speech recognition system for the European Portuguese language*”.

Aos nossos amigos e colegas de curso: Cláudio Oliveira, João Fiúza, e José Gomes, Pedro Sousa, Pedro Rolo e Sandra Conduto por todos os momentos de amizade e pelo apoio bem-humorado nas fases mais difíceis deste curso.

Um agradecimento muito especial aos nossos pais pelo carácter e determinação que sempre nos transmitiram e que nos permitiram chegar até aqui. A Sofia agradece ao Cláudio por todo o apoio, estímulo e atenção, sobretudo nos períodos mais difíceis; à Helena e ao Paulo pela preocupação e acompanhamento constantes; à Beatriz, à Clara e ao Luís pelo pilar educacional e pelo carinho demonstrado desde sempre. O Armando agradece a toda a sua família e amigos que o têm acompanhado desde sempre.

Índice

1	INTRODUÇÃO	1
1.1	ESTADO DA ARTE	2
1.2	OBJECTIVOS E DESCRIÇÃO DO PROBLEMA.....	2
1.3	ABORDAGEM DE RESOLUÇÃO DO PROBLEMA	4
2	COMUNICAÇÃO	11
3	APLICAÇÃO	15
3.1	MÓDULO DO PDA.....	15
3.1.1	<i>Requisitos</i>	16
3.1.2	<i>Interface</i>	17
3.1.3	<i>Gestão da informação</i>	19
3.1.4	<i>Acesso a recursos externos</i>	20
3.2	MÓDULO DO SERVIDOR DE ÁUDIO.....	20
3.3	MÓDULO DE RECURSOS EXTERNOS	20
3.3.1	<i>Servidor de Conteúdos</i>	21
3.3.2	<i>Servidor de Dados</i>	25
3.3.3	<i>HUB</i>	28
3.3.4	<i>TTS</i>	28
3.3.5	<i>Base de Dados</i>	28
4	RECONHECIMENTO DE FALA	31
4.1	DESCRIÇÃO DA AUDIMUS API	32
4.1.1	<i>Visão geral</i>	32
4.1.2	<i>Os blocos funcionais</i>	32
4.2	GERAR A BIBLIOTECA AUDIMUS NO PDA	34
4.2.1	<i>Principais obstáculos</i>	34
4.2.2	<i>Obstáculos ultrapassados</i>	35
4.2.3	<i>Obstáculos por ultrapassar</i>	36
4.2.4	<i>Revisão das funcionalidades do sistema</i>	36
4.3	CONFIGURAÇÃO DA AUDIMUS API.....	36
4.4	MODELO ACÚSTICO	40
4.5	MODELO DA LINGUAGEM.....	41
4.6	AQUISIÇÃO DE FALA NO PDA.....	41

4.7	SUORTE AO RECONHECIMENTO DE FALA NO PDA	44
4.8	INTERACÇÃO COM OS FORMULÁRIOS	45
4.9	SERVIDOR DE ÁUDIO	47
5	RESULTADOS.....	49
5.1	DESEMPENHO DO RECONHECIMENTO	49
5.1.1	<i>Tempos de reconhecimento de fala isolado</i>	<i>50</i>
5.1.2	<i>Tempos obtidos com o PDA a 1 m do ponto de acesso.....</i>	<i>51</i>
5.1.3	<i>Tempos obtidos com o PDA a 20 m do ponto de acesso.....</i>	<i>52</i>
5.1.4	<i>Comparação da capacidade de processamento entre o PC e o PDA.....</i>	<i>53</i>
5.1.5	<i>Diferenças de desempenho entre reconhecimento de fala remoto e distribuído.</i>	<i>53</i>
5.2	SUCESSO DO RECONHECIMENTO DE FALA	54
5.3	FALA VS ENTRADAS DE DADOS TRADICIONAIS DO PDA	57
6	CONCLUSÃO	59
7	BIBLIOGRAFIA.....	65
	ANEXOS	67

Lista de Figuras

FIGURA 1 – PRIMEIRA ABORDAGEM DA APLICAÇÃO.....	3
FIGURA 2 – INTERFACE DA APLICAÇÃO.....	4
FIGURA 3 – PRIMEIRA ABORDAGEM DO FLUXO DE INFORMAÇÃO ENTRE O PDA E O SERVIDOR EXTERNO.	5
FIGURA 4 – ARQUITECTURA INICIAL DO SISTEMA	6
FIGURA 5 – FLUXO FINAL DE INFORMAÇÃO TROCADA ENTRE O PDA E OS SERVIDORES.	8
FIGURA 6 – ARQUITECTURA DA APLICAÇÃO	9
FIGURA 7 – TESTE COM O FICHEIRO DE 570 KB	12
FIGURA 8 – TESTE COM O FICHEIRO DE 16 MB.....	12
FIGURA 9 – ARQUITECTURA DO MÓDULO DO PDA	16
FIGURA 10 – EXEMPLO DE UMA JANELA DE DIÁLOGO	18
FIGURA 11 – FORMULÁRIOS DA APLICAÇÃO	19
FIGURA 12 – FLUXO DE MENSAGENS ENTRE O SERVIDOR DE CONTEÚDOS COM O PDA, O SERVIDOR DE DADOS E O TTS	23
FIGURA 13 – DIAGRAMA DE ESTADOS DO SERVIDOR DE CONTEÚDOS.....	24
FIGURA 14 – DIAGRAMA DE CLASSES DO SERVIDOR DE DADOS.....	26
FIGURA 15 – DIAGRAMA DE ESTADOS DO SERVIDOR DE DADOS	27
FIGURA 16 – EXEMPLO DE UM DOENTE NA BASE DE DADOS.....	29
FIGURA 17 – REPRESENTAÇÕES DA PALAVRA "DOIS": A) REPRESENTAÇÃO ATRAVÉS DO GRÁFICO DE AMPLITUDE DO SINAL DE FALA COM O MICROFONE DO PC E B) REPRESENTAÇÃO DO RESULTADO DO END POINT APLICADO A A) COM OS PARÂMETROS DO PC.	38
FIGURA 18 – REPRESENTAÇÕES DA PALAVRA "DOIS": A) REPRESENTAÇÃO ATRAVÉS DO GRÁFICO DE AMPLITUDE DO SINAL DE FALA COM O MICROFONE DO PDA E B) REPRESENTAÇÃO DO RESULTADO DO END POINT APLICADO A A) COM OS PARÂMETROS DO PDA.	38
FIGURA 19 – PROCESSO DE AQUISIÇÃO DE FALA	43
FIGURA 20 – AQUISIÇÃO ATRAVÉS DE TAMPÕES.....	44
FIGURA 21 – INDICADOR DE CAMPO SELECIONADO.....	46

FIGURA 22 – ESQUEMA GERAL DO FUNCIONAMENTO DA APLICAÇÃO 48

Lista de Tabelas

TABELA 1 – CAMPOS DO FORMULÁRIO.....	17
TABELA 2 – CAMPOS COM VALORES PRÉ-DEFINIDOS E RESPECTIVAS DEFINIÇÕES..	17
TABELA 4 – PARÂMETROS DA AMOSTRAGEM.....	42
TABELA 5 – TIPOS DE MENSAGENS DA APLICAÇÃO.	45
TABELA 6 – ESTADOS DA APLICAÇÃO	45
TABELA 7 – CANAIS DE COMUNICAÇÃO	47
TABELA 8 – TEMPOS DE RECONHECIMENTO DE FALA OBTIDOS PARA CADA UM DOS TRÊS FICHEIROS, USANDO A AUDIMUS API NUM PROGRAMA ISOLADO.	50
TABELA 9 – RESULTADOS DOS TEMPOS OBTIDOS PARA A FRASE “DOIS”	51
TABELA 10 – RESULTADOS DOS TEMPOS OBTIDOS PARA A FRASE “DOIS ARMANDO SALGUEIRO MASCULINO”	51
TABELA 11 – RESULTADOS DOS TEMPOS OBTIDOS PARA A FRASE “DOIS JOÃO SOUSA MASCULINO GRAVE OSTEOPOROSE”	52
TABELA 12 – RESULTADOS DOS TEMPOS OBTIDOS PARA A FRASE “DOIS JOÃO SOUSA MASCULINO GRAVE OSTEOPOROSE” COM O PDA A 20 M DE DISTÂNCIA DO PONTO DE ACESSO.....	53
TABELA 13 – RESULTADOS DOS TEMPOS OBTIDOS PARA A FRASE 3 COM A AUDIMUS API A EXECUTAR-SE DE FORMA DISTRIBUÍDA ENTRE O PDA E O SERVIDOR DE ÁUDIO.....	54
TABELA 14 – TEMPOS DE EXECUÇÃO DA TAREFA 1 E O NÚMERO DE REPETIÇÕES DE TEXTO FALADO PARA CADA UM DOS CAMPOS EM ANÁLISE.....	55
TABELA 15 – TEMPOS DE EXECUÇÃO DA TAREFA 2 E O NÚMERO DE REPETIÇÕES DE TEXTO FALADO PARA CADA UM DOS CAMPOS EM ANÁLISE.....	55
TABELA 16 – TEMPOS DE EXECUÇÃO DA TAREFA 3 E O NÚMERO DE REPETIÇÕES DE TEXTO FALADO PARA CADA UM DOS CAMPOS EM ANÁLISE.....	56
TABELA 17 – PERCENTAGEM DO NÚMERO DE REPETIÇÕES NO PREENCHIMENTO DOS CAMPOS.....	56
TABELA 18 – TEMPOS DISPENDIDOS NA TAREFA 1 USANDO O TECLADO VIRTUAL ..	57
TABELA 19 – TEMPOS DISPENDIDOS NA TAREFA 2 USANDO O TECLADO VIRTUAL ..	57
TABELA 20 – TEMPOS DISPENDIDOS NA TAREFA 3 USANDO O TECLADO VIRTUAL ...	57

TABELA 21– TEMPOS DISPENDIDOS NA TAREFA 1 USANDO O MODO DE ENTRADA RECONHECEDOR DE LETRAS	57
TABELA 22 – TEMPOS DISPENDIDOS NA TAREFA 2 USANDO O MODO DE ENTRADA RECONHECEDOR DE LETRAS	58
TABELA 23 – TEMPOS DISPENDIDOS NA TAREFA 3 USANDO O MODO DE ENTRADA RECONHECEDOR DE LETRAS	58

Lista de Siglas

PC: Personal computer

PDA: Personal device assistant

XML: Extensible Markup Language

MFC: Microsoft Foundation Classes

ASR: Automatic Speech Recognition

TTS: Text to Speech

MKL: Math Kernel Library

1 Introdução

A necessidade de informação é uma característica que acompanha os seres humanos desde sempre. A informação é um meio para auxiliar a tomada de decisões, solucionar problemas ou localizar recursos.

O acesso a informação actualizada, exacta e num curto espaço de tempo constitui uma exigência na sociedade actual. De forma a cumprir esta exigência é necessário obter informação em qualquer lugar e em qualquer hora. Nesta perspectiva, os dispositivos móveis constituem uma solução tecnológica ao problema da mobilidade e disponibilidade da informação. Estes dispositivos podem ser desde telemóveis, agendas electrónicas, *Personal Device Assistant (PDA)*, computadores portáteis entre outros.

Em relação ao PDA, a sua utilização tem crescido significativamente nos últimos anos. Este crescimento está associado ao facto deste dispositivo ter as características de uma agenda electrónica e herdar muitas das funcionalidades dos computadores pessoais. A capacidade do PDA de gerir informação aliada à sua portabilidade, torna este dispositivo indispensável no quotidiano.

No entanto, apesar do PDA ser alvo de uma constante evolução tecnológica, ainda apresenta muitas limitações ao nível do armazenamento, processamento e interacção com o utilizador, em relação aos computadores pessoais. Ao nível da interacção com o utilizador, os métodos de entrada (reconhecimento de letras e teclado virtual) de dados no PDA são limitados e pouco práticos. Uma interface baseada em fala é apontada como a solução ideal para este tipo de dispositivos. A fala é um meio natural e universal de comunicação que ultrapassa algumas das limitações impostas pelos métodos de entrada de dados baseados na escrita. Futuramente, com os avanços tecnológicos no campo dos PDAs, será possível que estes possuam as capacidades necessárias para realizarem o reconhecimento e síntese de fala, de uma forma eficiente. Com a eliminação destas barreiras tecnológicas, a utilização da fala no PDA poderá sobrepor-se aos métodos de entrada de dados baseados na escrita como o teclado virtual e o reconhecedor de letras.

1.1 Estado da Arte

Apesar dos progressos significativos, o reconhecimento automático de fala ainda é uma área em desenvolvimento. Actualmente as aplicações desenvolvidas nesta área são limitadas e aplicadas a domínios específicos. As principais limitações prendem-se com a dimensão do vocabulário e a necessidade do utilizador realizar um treino para que o sistema de reconhecimento se adapte à sua voz.

No que diz respeito aos PDAs, actualmente existem no mercado inúmeras aplicações, desde de processadores de texto a jogos. No entanto o número de programas que utilizam a fala como interface é reduzido. A utilização da fala nos PDAs, através do seu reconhecimento ou da sua síntese é uma área que ainda se encontra na fase de investigação. Apesar de despertar o interesse da indústria informática, a fala ainda não se constitui como uma forte aposta no mercado dos PDAs. Existem algumas soluções como o MiPad¹ que englobam um grande número de funcionalidades. No entanto, estas aplicações estão em desenvolvimento ou ainda não ganharam grande importância no mundo dos PDAs.

1.2 Objectivos e Descrição do Problema

O PDA recebeu uma forte aceitação na comunidade médica devido à sua portabilidade e características tecnológicas. O PDA é utilizado na gestão da sempre crescente quantidade de informação que os médicos necessitam de saber (ex.: informação de fármacos) e para realizar cálculos médicos. Actualmente, o PDA começa a ser utilizado como um meio de visualizar e manipular informações sobre os pacientes no local de prestação de cuidados médicos. O que permite uma diminuição da burocracia e um aumento da qualidade dos serviços prestados aos doentes.

O objectivo deste trabalho é desenvolver uma aplicação para um dispositivo PDA com uma interface de formulários que poderão ser preenchidos através da fala, para além dos métodos de entrada de dados já existentes. Genericamente, o programa tem uma interface muito aproximada de um formulário médico. A aplicação deve

¹ <http://research.microsoft.com/srg/mipad.aspx>

permitir o preenchimento, consulta, alteração e eliminação de formulários. Para além da informação dos campos do formulário este pode conter referências a outros documentos relacionados com o doente, tais como radiografias, relatórios entre outros.

Para além dos métodos tradicionais de entrada de dados é oferecida ao utilizador a possibilidade de preencher os campos do formulário através de fala. Adicionalmente o utilizador tem a possibilidade de ouvir documentos escritos associados a um doente.

O PDA tem recursos limitados no que respeita à capacidade de processamento e armazenamento de informação (ver anexo IV). A aplicação exige muitos recursos ao nível do processamento de fala e armazenamento de informação dos doentes. Assim torna-se necessário distribuir a aplicação por um computador, designado por servidor externo. Esta entidade tem como funções guardar a informação dos doentes e realizar o processamento de fala. A utilização do servidor externo tem como objectivo vencer as limitações do PDA. Numa fase posterior pretende-se transferir algumas tarefas relacionadas com o processamento de fala do servidor para o PDA. A figura seguinte ilustra a arquitectura da aplicação.

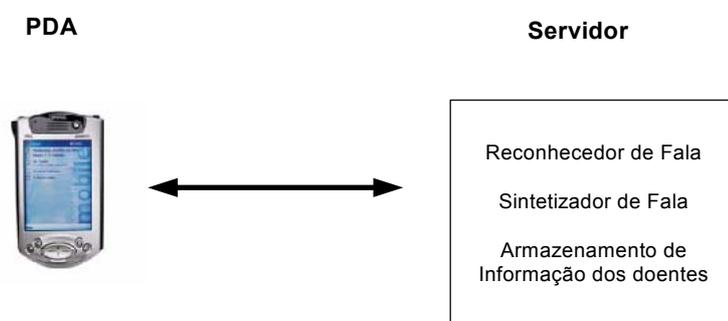


Figura 1 – Primeira abordagem da aplicação.

Para garantir a interacção da aplicação têm de existir um meio de comunicação entre o PDA e o servidor externo de modo a não prejudicar a mobilidade do PDA.

1.3 Abordagem de resolução do problema

Definidos os objectivos começou-se por estudar as funcionalidades e capacidades do PDA. Na realização deste trabalho foi utilizado um Compaq iPAQ H3870 equipado com um processador StrongARM. O sistema operativo deste dispositivo é o Microsoft Pocket PC 2000. O PDA está equipado com Bluetooth integrado e com um módulo de expansão, que pode albergar uma placa de rede wireless. A especificação do PDA está no anexo IV.

A fase seguinte envolveu a escolha do meio de comunicação entre o PDA e o servidor externo. A rede wireless Wi-Fi apresentou-se como sendo a melhor opção, porque tem um ritmo de transmissão máximo (teórico) de 11 Mbps e não restringe a mobilidade. Porém, a distância entre o PDA e o ponto de acesso à rede wireless não pode exceder os 100 metros. De seguida procedeu-se à definição do protocolo de comunicação entre o PDA e o servidor externo e posteriormente à sua implementação. No Capítulo 2 analisa-se mais detalhadamente a comunicação.

Definida a estrutura de suporte à comunicação, chegou a altura de proceder ao desenvolvimento da aplicação. Numa primeira fase definiu-se os campos do formulário médico e os valores que estes podiam conter. Com base nesta definição foi implementada a interface com o utilizador (figura 2). Esta tarefa possibilitou o preenchimento e alteração de fichas de doentes recorrendo ao teclado virtual ou o reconhecedor de letras.



Figura 2 – Interface da aplicação

Surge então o problema de como e onde guardar as fichas dos doentes. Como já foi referido, o PDA tem restrições ao nível do armazenamento de informação e foi necessário recorrer a um servidor externo. Neste ponto tornou-se indispensável definir a arquitectura do sistema que abrange o PDA e o servidor externo. Para além do armazenamento de fichas de doentes o servidor terá que suportar todos os componentes do processamento de fala. Com base neste requisitos surge o seguinte modelo de interacção entre o PDA e o servidor externo. O PDA efectua um pedido, que contém fala do utilizador, ao servidor externo. A resposta que este envia ao PDA pode ser o texto resultante do reconhecimento, a fala sintetizada, fichas de doentes ou documentos. Assim, o servidor externo recebe a fala do PDA e realiza o reconhecimento da mesma. O resultado do reconhecimento é interpretado e redireccionado para a acção correspondente. A figura 3 ilustra o fluxo de informação entre o PDA e o servidor externo.

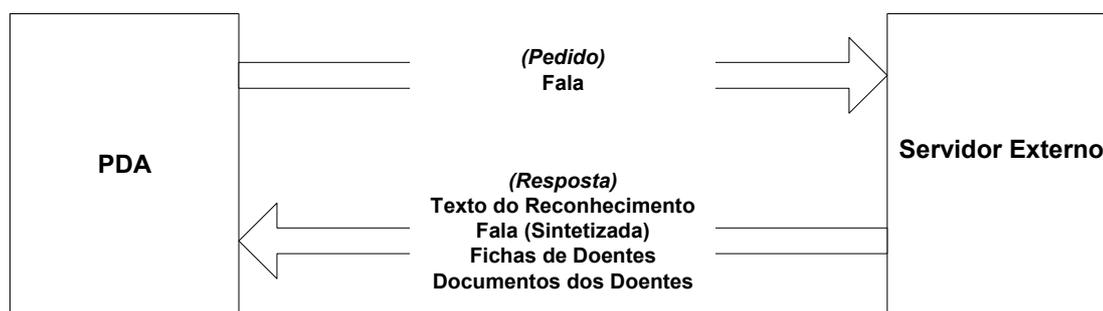


Figura 3 – Primeira abordagem do fluxo de informação entre o PDA e o servidor externo.

Com base neste modelo de interacção definiu-se uma primeira arquitectura da aplicação, ilustrada na figura 4.

O sistema divide-se em dois módulos lógicos principais: o PDA e o servidor. O módulo do PDA é composto pela aplicação que contém a interface com o utilizador, o sistema de entrada e saída de fala e a interface de comunicação com o módulo de servidor. A aplicação realiza a aquisição e reprodução de fala e contém uma interface de comunicação para trocar informação com os dois servidores: servidor de conteúdos e servidor de áudio.

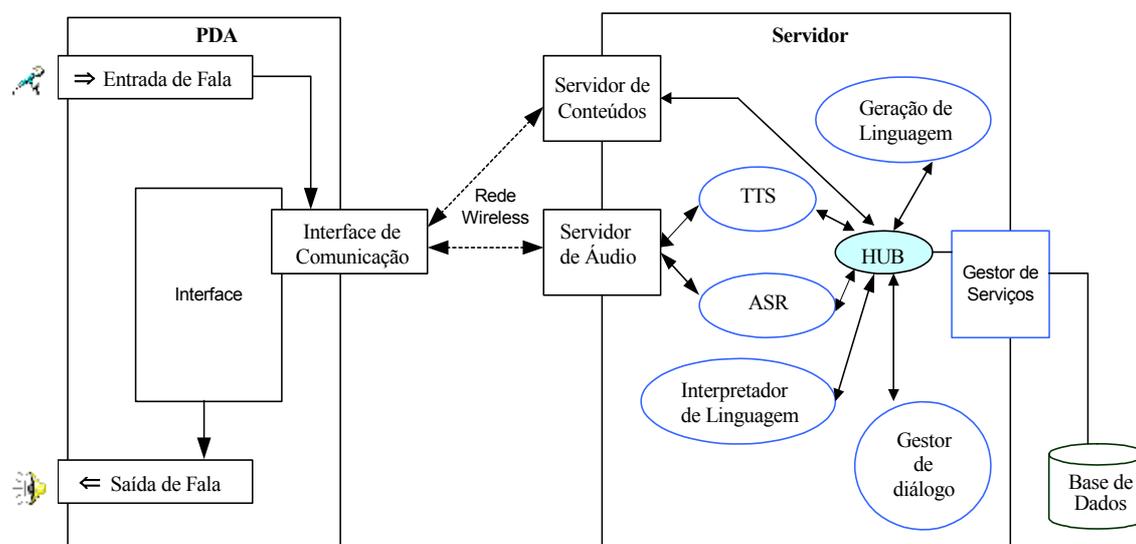


Figura 4 – Arquitetura inicial do sistema

O módulo do servidor é constituído pelo servidor de áudio e pelo servidor de conteúdos. O servidor de áudio foi baseado na arquitectura de um sistema de processamento de fala e possui os componentes necessários ao reconhecimento e síntese de fala. Os componentes são os seguintes:

- *ASR (Automatic Speech Recognition)*: realiza o reconhecimento automático de fala, convertendo fala em texto.
- *TTS (Text to Speech)*: realiza a síntese de fala, convertendo texto em fala.
- *Interpretador de Linguagem*: efectua a análise semântica do resultado do ASR.
- *Gestor de diálogo*: efectua a ligação das acções do utilizador com os serviços prestados pelo sistema. (ex. base de dados)
- *Geração de linguagem*: formula as frases para o TTS.

O PDA realiza a aquisição de fala do utilizador através da *Entrada de Fala*. A fala é transmitida pela *Interface de Comunicação* para o *Servidor de Áudio*. O servidor de áudio recebe a fala do PDA e envia-a para o *ASR*, onde é realizado o reconhecimento de fala. O resultado do *ASR* é enviado para o *Interpretador de Linguagem*, através do

HUB. O *Interpretador de Linguagem* analisa o resultado do *ASR* e distingue se este é apenas fala para ser reconhecida, ou se é um pedido de informação. No primeiro caso, a resposta consiste no resultado do *ASR* e é enviada, através do *HUB*, para o *Servidor de Conteúdos*. No segundo caso, o pedido de informação é enviado para o *Gestor de Serviços*, através do *HUB*. O *Gestor de Serviços* realiza o acesso à *Base de Dados* de forma a obter a resposta ao pedido recebido. A resposta do *Gestor de Serviços* é então enviada para o *Servidor de Conteúdos* através do *HUB*. Em ambos os casos o *Servidor de Conteúdos* envia a resposta para PDA. A resposta é recebida no PDA através da *Interface de Comunicação* e é exibida na *Interface*.

No entanto, esta arquitectura é muito abrangente considerando os objectivos do trabalho. Em primeiro lugar o preenchimento dos campos do formulário deve ser realizado quer com os métodos de entrada de dados já existente, quer através de fala. O reconhecimento de fala apenas será utilizado no preenchimento dos campos dos formulários. A restante manipulação da interface (escolher o campo, mudar a secção do formulário, etc.) é realizada através dos outros métodos de entrada do PDA. Deste ponto de vista, só é importante obter o resultado do reconhecimento e direccioná-lo para o respectivo campo do formulário. O uso da fala para dialogar com a aplicação introduz demasiada complexidade para ser abordada neste trabalho.

Como futuramente, pretende-se que o reconhecimento de fala (*ASR*) possa ser efectuada localmente no PDA, houve a necessidade de separar o reconhecimento de fala do processamento dos pedidos relacionados com os doentes. Assim, foi criado um módulo específico para o reconhecimento de fala: o módulo do servidor de áudio. O tratamento de pedidos de informação relacionada com os doentes foi integrado num outro módulo: módulo de Recursos Externos. A síntese de fala também ficou inserida neste último módulo porque não é um requisito essencial para a aplicação e é independente do reconhecimento de fala. A interacção entre as três entidades (PDA, servidor de áudio e módulo de recursos externo) está ilustrada na figura 5. O ponto 1 do esquema representa a fluxo de informação entre o PDA e o servidor de áudio. Esta informação compreende um pedido que contém fala do utilizador para ser reconhecida e uma resposta que consiste do texto resultante desse reconhecimento. Em relação ao ponto 2 do esquema, é representado o fluxo de informação entre o PDA e o servidor de conteúdos. Neste caso a informação consiste num pedido em texto e a resposta

pode ser uma ficha de um doente, um documento de um doente ou o resultado da síntese de fala. Os dois fluxos de informação são independentes, é possível efectuar um pedido ao servidor de conteúdos sem utilizar a fala.

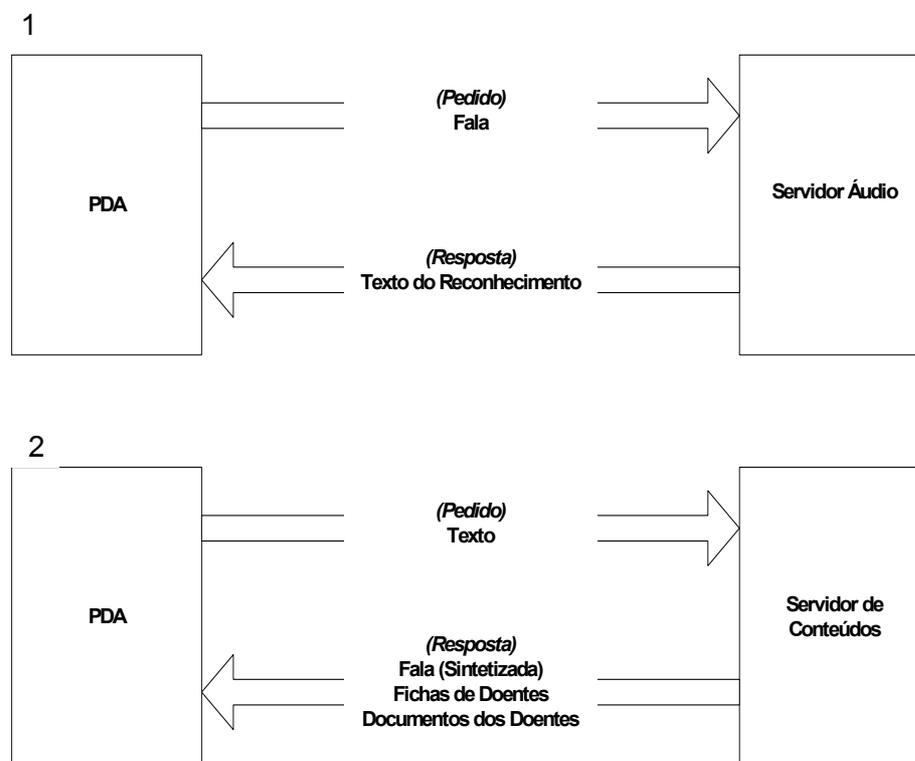


Figura 5 – Fluxo final de informação trocada entre o PDA e os servidores.

Na figura 6 é possível visualizar a arquitectura final da aplicação. No Capítulo 3 descreve-se detalhadamente a implementação da aplicação.

Por último foi abordado o problema do reconhecimento de fala. Para realizar o reconhecimento de fala foi utilizado um sistema de reconhecimento de fala já existente, a Audimus API (*Application Program Interface*). A Audimus API foi desenvolvida no Laboratório de Língua Falada do INESC-ID para realizar o reconhecimento de fala em computadores pessoais. Numa primeira fase todo o reconhecimento foi realizado no módulo do servidor de áudio. Posteriormente algumas tarefas do reconhecimento foram migradas para o PDA. Assim, existem duas configurações para o reconhecimento: reconhecimento remoto, totalmente realizado no servidor de áudio e reconhecimento distribuído, onde o processamento é distribuído entre o PDA e o servidor de áudio.

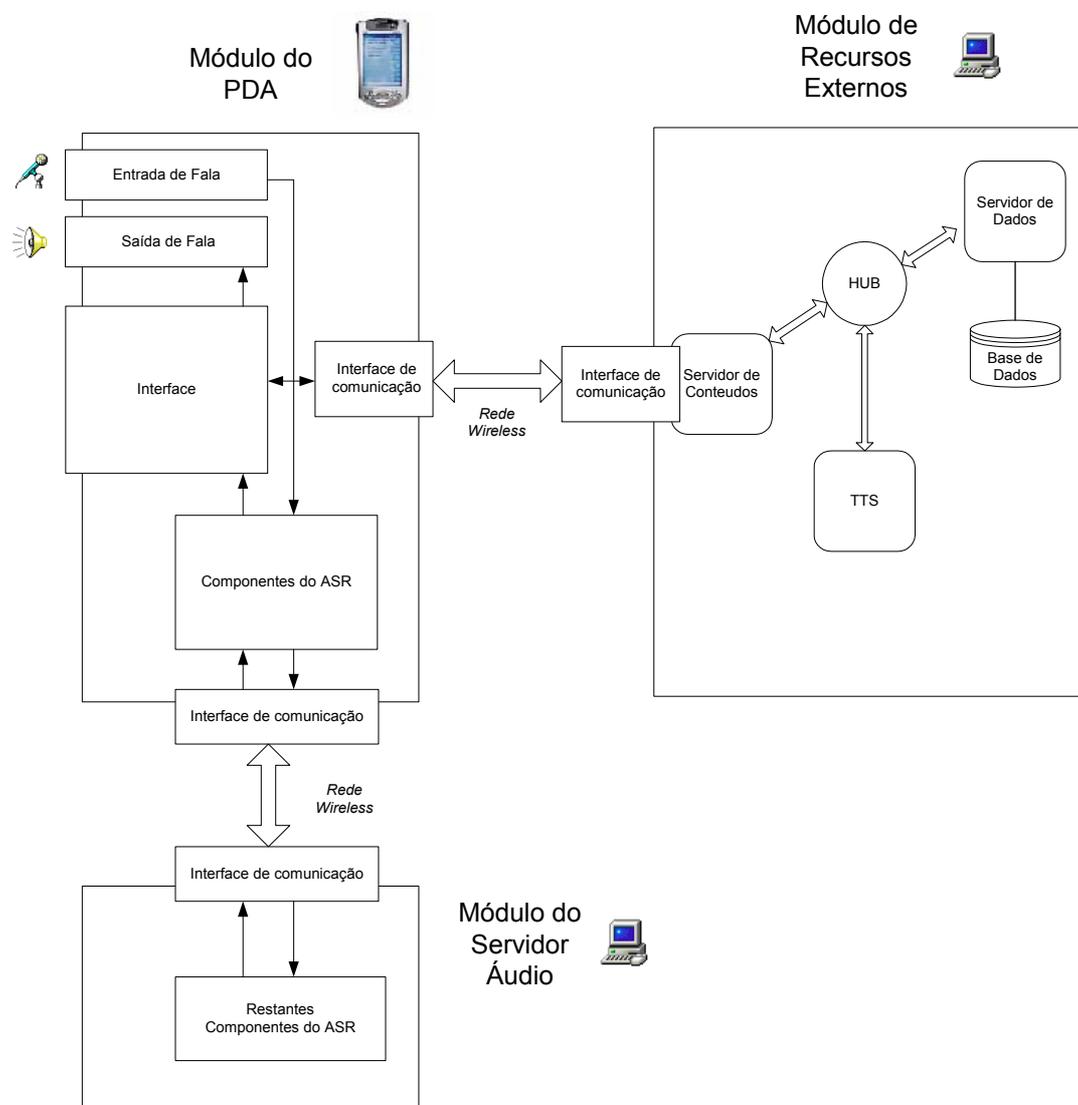


Figura 6 – Arquitectura da aplicação

O reconhecedor de fala para produzir resultados satisfatórios, têm de ser configurado tendo em conta o domínio da aplicação e o ambiente onde esta é utilizada. Torna-se necessário adaptar o mecanismo do reconhecedor de fala que procura detectar os instantes de tempo correspondentes ao início e ao fim de uma frase. Este mecanismo é denominado de End point e tem que ser adaptado às características de ruído ambiente do lugar onde a aplicação é utilizada. Esta adaptação permite diminuir o tempo de reconhecimento porque evita o processamento de ruídos

e silêncios. O reconhecedor de fala também necessita de ser configurado relativamente ao domínio da aplicação. O modelo de linguagem do reconhecedor é o conjunto limitado das palavras que este consegue reconhecer, pelo que tem de ser construído com base nos vocábulos utilizados na aplicação. Por último é necessário adequar o reconhecedor às características acústicas do utilizador. Cada utilizador é caracterizado no reconhecedor pelo seu modelo acústico. Quanto melhor for a adaptação do modelo acústico ao utilizador, menor será o tempo de processamento e maior será o sucesso do reconhecimento.

A integração do reconhecimento de fala na aplicação está detalhada no Capítulo 4.

2 Comunicação

Como foi referido no Capítulo 1, um dos objectivos deste trabalho é usar um meio de comunicação entre o PDA e o servidor externo que não prejudique a mobilidade do PDA. A utilização de redes *wireless* (sem-fios) é a solução que actualmente oferece mais mobilidade às aplicações. A escolha de uma rede com fios esteve fora de questão porque impõem grandes restrições à mobilidade. No que respeita à tecnologia *Bluetooth*, esta é mais adequada para ligar dispositivos, não como um meio de comunicação. A escolha da rede Wi-Fi (IEEE 802.11b) foi condicionada pelo facto desta solução estar disponível no L²F.

Uma rede Wi-Fi opera apenas com um conjunto pequeno de fronteiras: geralmente abrange um edifício ou uma pequena área de edifícios. A rede Wi-Fi apresenta velocidades de transmissão altas em comparação, por exemplo, com o GPRS/GSM. O GPRS (*General Packet Radio Service*) é uma solução utilizada por telemóveis para permitir o acesso de alto débito a conteúdos da Internet e a outros serviços baseados em transmissão de pacotes. O GSM (*Global System for Mobile Communications*) é o sistema celular mais utilizado na Europa e na Ásia e permite a transmissão de voz e dados. O GPRS funciona sobre redes GSM e suporta débitos de 14 kbps no envio e de 28-64 kbps na recepção de dados.

A rede Wi-Fi (IEEE 802.11b) usa um espectro de 2.4 GHz tal como os auscultadores sem fios topo de gama. A norma Wi-Fi é capaz de transmitir dados a uma velocidade máxima (teórica) de 11Mbps em boas condições de sinal.

As redes Wi-Fi são as redes *wireless* mais populares usadas hoje em dia devido ao seu baixo custo relativo e ao seu ritmo de transmissão razoável. No entanto, duas novas normas, a IEEE 802.11a e IEEE 802.11g, revelam-se promissoras. A norma 802.11a permite uma velocidade máxima de transmissão (teórica) de 54 Mbps e usa um espectro de 5 GHz. A norma 802.11g usa o mesmo espectro que o 802.11b mas através de Multiplexagem da Divisão de Frequência Ortogonal (OFDM) consegue atingir uma velocidade máxima (teórica) de transmissão 54Mbps.

De forma a avaliar a capacidade da rede *wireless* realizaram-se testes de comparação com a rede Ethernet. Também foi comparado o uso da rede *wireless* quer pelo PDA quer pelo PC, de forma a comparar o desempenho das comunicações, nestas duas plataformas.

O teste consiste em enviar um ficheiro do dispositivo (PDA e PC) para o servidor externo e receber o mesmo ficheiro do servidor. O envio e recepção são operações disjuntas. O tamanho dos ficheiros utilizados foi de 570 Kb e de 16 Mb. Os resultados obtidos apresentam-se nos seguintes gráficos, com ritmo de transmissão estimado:

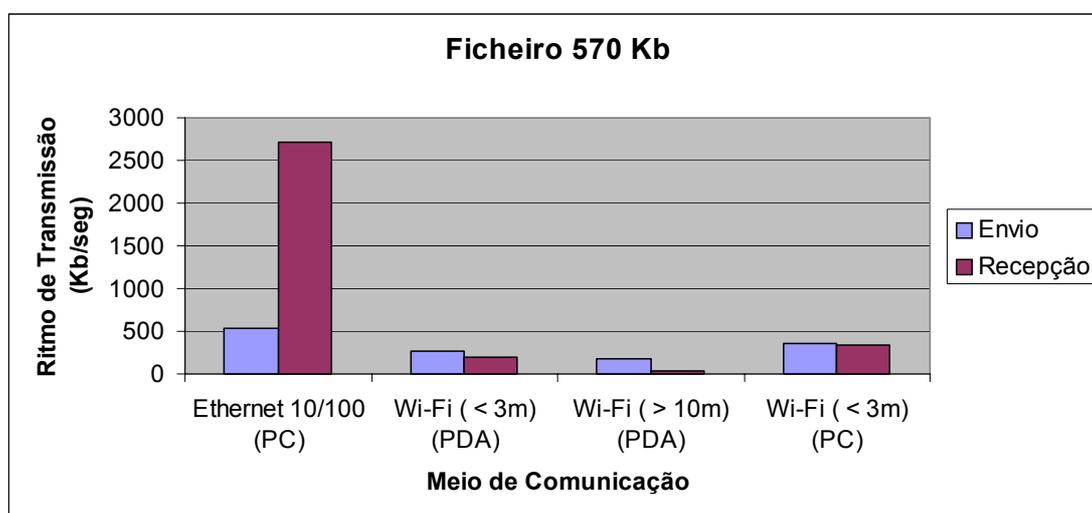


Figura 7 – Teste com o Ficheiro de 570 Kb

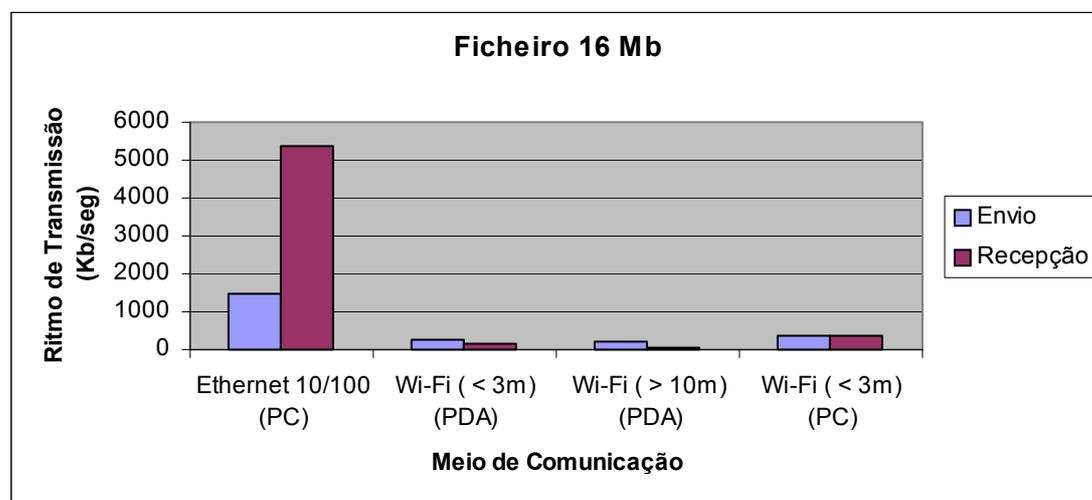


Figura 8 – Teste com o Ficheiro de 16 Mb

Como se pode observar nos gráficos da figura 7 e 8, a rede Ethernet tem resultados muito superiores aos da rede *wireless* (Wi-Fi), porque tem um ritmo de transmissão de 100 Mbps. No que diz respeito à rede *wireless* é visível um decaimento do desempenho da rede com o aumento da distância ao ponto de acesso. Por outro lado, comparando a utilização de rede *wireless* pelo PDA com o PC, verifica-se que no PDA o ritmo de envio é muito superior ao de recepção, o que não acontece no PC. Este facto poderá estar relacionado com o suporte da placa de rede do PDA.

Após a escolha do meio de comunicação entre o PDA e o servidor, o próximo passo é escolher o protocolo da camada rede a utilizar. Para o transporte de dados de doentes entre o PDA e o módulo de recursos externos, utilizou-se o protocolo TCP/IP. Esta é a solução ideal para o transporte fiável de informação em redes IP. Na ligação entre o PDA e o servidor de áudio também se utilizou o protocolo TCP/IP. A alternativa seria o protocolo UDP, mas esta envolvia a implementação de um mecanismo de recuperação de pacotes perdidos durante a comunicação. Esta operação seria justificada caso o tempo de transferência de fala tornasse impraticável a utilização do sistema de reconhecimento de fala.

3 Aplicação

No capítulo anterior definiram-se as estruturas que suportam a comunicação entre o PDA e o módulo do servidor de áudio, bem como entre o PDA e o módulo do servidor de conteúdos.

O presente capítulo pretende descrever a construção de todos os módulos que compõem a aplicação. Em primeiro lugar é descrito o módulo do PDA, a nível dos requisitos da aplicação e da interface. Seguidamente detalha-se a implementação do servidor de áudio, ao nível das suas funcionalidades e interacção com o módulo do PDA. Por último descreve-se o módulo de recursos externos e todos os seus componentes, bem como estes interagem entre si e com o PDA.

3.1 Módulo do PDA

O módulo da aplicação no PDA é o responsável por toda a interacção com o utilizador, peça fundamental no preenchimento de formulário através da fala.

Este módulo compreende as seguintes funcionalidades: aquisição de fala do utilizador, interface com utilizador, gestão dos dados dos formulários e reconhecimento de fala. Denota-se ainda, que a aplicação não suporta comandos de fala, limita-se a realizar o preenchimento dos campos.

A coordenação e independência de todas estas funcionalidades foram aspectos fundamentais na construção deste módulo. A figura 9 apresenta uma visão geral da arquitectura do módulo do PDA. Nos pontos seguintes descrevem-se os elementos que compõe o módulo do PDA.

Módulo do PDA

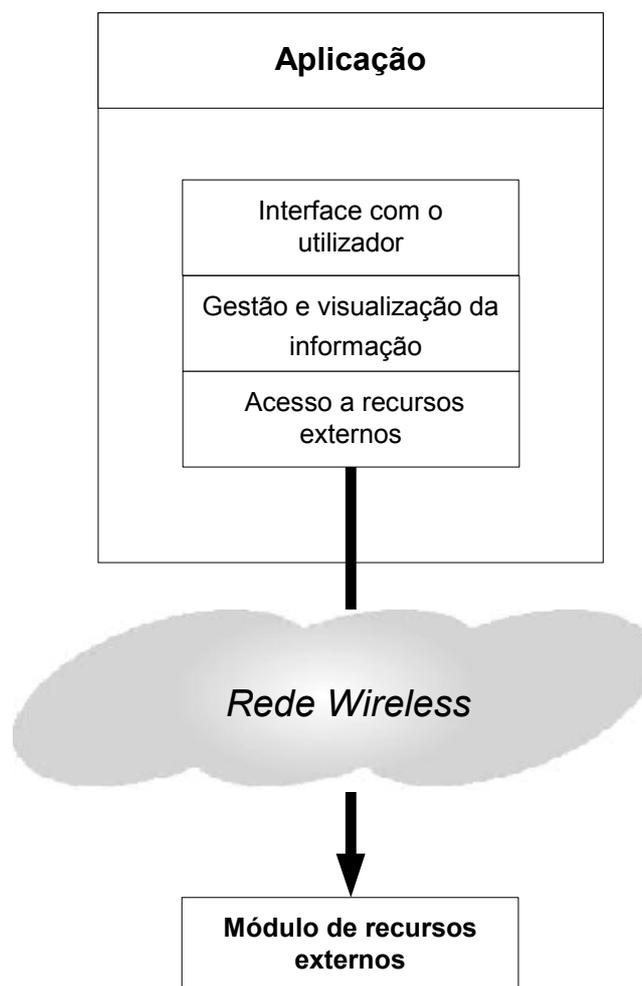


Figura 9 – Arquitectura do módulo do PDA

3.1.1 Requisitos

A interface foi desenvolvida mediante os seguintes requisitos:

- A interface deve implementar um formulário médico. O formulário é uma adaptação livre de alguns campos de uma ficha de um doente.
- Implementação das operações de criar, alterar, e eliminar fichas de doentes.
- Permitir visualizar documentos referenciados na ficha do doente (imagens, relatórios, gravações áudio).
- Os campos desse formulário devem poder ser preenchidos através da fala.

Os campos definidos para o formulário encontram-se na tabela seguinte:

Campo	Tipo	Valores pré-definidos
Nome	Cadeia de Caracteres	Não
Sexo	Cadeia de Caracteres	Sim
Idade	Numérico	Não
Nº Seg. Social	Numérico	Não
Sala	Cadeia de Caracteres	Não
Piso	Numérico	Sim
Doença	Cadeia de Caracteres	Sim
Estado	Cadeia de Caracteres	Sim
Observações	Cadeia de Caracteres	Não
Outros Documentos	Cadeia de Caracteres	-

Tabela 1 – Campos do formulário

Os campos com valores pré-definidos podem ter os seguintes valores:

Sexo	Piso
não foi definido	não foi definido
Feminino	9
Masculino	8
Estado	7
não foi definido	6
Estável	5
Grave	4
Cadáver	3
	2
	1
	0

Tabela 2 – Campos com valores pré-definidos e respectivas definições.

3.1.2 Interface

A interface é baseada em janelas de diálogo (*dialog boxes*): um componente gráfico das MFC. As janelas de diálogo podem conter controlos (botões, caixas de

texto, caixas de selecção, etc.) que permitem visualizar informação. Um exemplo de uma janela de diálogo está na figura 10.

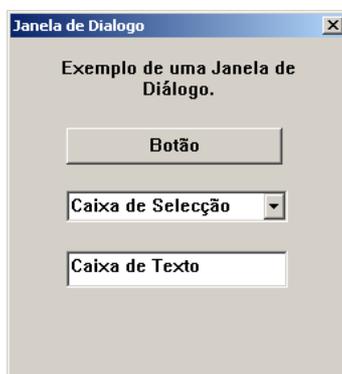


Figura 10 – Exemplo de uma janela de diálogo

Uma janela de diálogo está sempre associada à classe que a implementa. Nesta classe é possível adicionar funções para manipular e interagir com os componentes da janela de diálogo. Considerar-se uma janela de diálogo como sendo a combinação entre os componentes gráficos e a classe de suporte.

As fichas de doentes foram concretizadas sobre a forma de formulários, implementados em diversas janelas de diálogo. Foram utilizados dois tipos de controlos para os campos dos formulários:

- Caixas de Texto (“*EDIT BOX*”) para campos que não possuam valores pré-definidos.
- Caixas de Selecção (“*COMBO BOX*”) para campos com valores pré-definidos.

A figura 11 mostra os formulários da aplicação. Na aplicação existe sempre uma Janela de Diálogo activa. A Janela de Diálogo activa é a que está visível para o utilizador, recebe as mensagens de eventos da manipulação dos controlos pelo utilizador (cliques, selecções, pressão de teclas, ...). Para além destas mensagens podem ser definidas outras para fins específicos da aplicação.

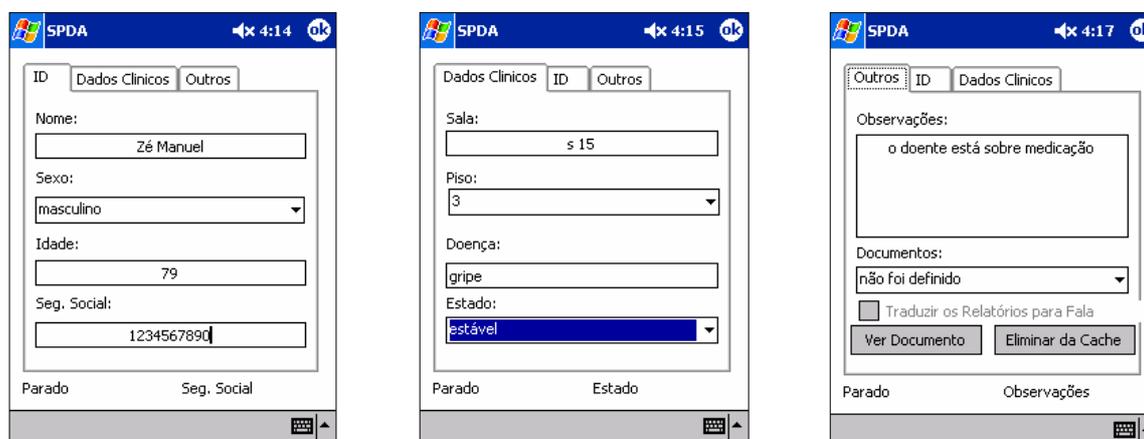


Figura 11 – Formulários da aplicação

3.1.3 Gestão da informação

As informações do formulário são mantidas numa estrutura de dados, visível a todas as janelas de diálogo que contenham campos do formulário. As janelas de diálogo quando são iniciadas lêem os dados da estrutura, que correspondem aos campos que possuem, actualizando o valor dos seus campos com essa informação. Quando as janelas de diálogo são fechadas, caso os campos tenham sido alterados, essas alterações são propagadas para a estrutura de dados.

No que diz respeito aos documentos referenciados nos formulários (campo “*documentos*”) esses documentos são guardados temporariamente no PDA, de forma a tornar mais rápida a sua visualização. Esses documentos ao serem lidos da base de dados são copiados para uma directoria do PDA, onde é adicionado ao nome do documento o número da cama do doente. Quando o utilizador estiver a visualizar um documento que esteja no PDA, o documento é lido a partir da directoria não havendo necessidade de o requisitar.

Os documentos de texto podem ser sintetizados para fala, permitindo ao utilizador ouvir o seu conteúdo. O documento que resulta da síntese de fala é um documento de áudio, independente do documento de origem. O nome do documento sintetizado é igual ao nome do documento antigo mais um sufixo para evitar possíveis conflitos com o nome.

São utilizados três tipos de documentos: documentos de áudio (WAV), documentos de imagens (JPEG) e documentos de texto (TXT). Os documentos de imagens e os documentos de texto são visualizados com o auxílio de outros programas do Pocket PC: PocketWord para os documentos de texto e o Internet Explorer para as imagens.

3.1.4 Acesso a recursos externos

A secção de acesso a recursos externos têm como principal função, servir de intermediário entre a aplicação e o módulo de recursos externos. O que permite aceder às informações da base de dados e às funcionalidades do sintetizador de fala.

Esta secção é o suporte local das seguintes funcionalidades, que se efectuem remotamente:

- Leitura e escrita das informações dos formulários.
- Acesso aos documentos.
- Síntese de fala correspondentes a documentos de texto.

Estas funcionalidades pressupõem todo um conjunto de operações, como o envio e recepção de pedidos, bem como o envio e leitura de documentos através da rede.

3.2 Módulo do Servidor de Áudio

O servidor de áudio tem como função efectuar o processamento relativo ao reconhecimento de fala. O PDA envia a fala para o servidor de áudio e recebe como resultado o texto correspondente a essa fala. A implementação do módulo do servidor de áudio está relacionada com reconhecimento de fala. Este assunto está detalhado no Capítulo 4.

3.3 Módulo de Recursos Externos

O principal objectivo do módulo de recursos externos é receber e responder a pedidos da aplicação do PDA. Assim distinguem-se três operações fundamentais:

receber os pedidos; obter a resposta inerente ao pedido e enviá-la para a aplicação do PDA.

O módulo de recursos externos contém três componentes: o servidor de conteúdos, o servidor de dados e o TTS. A comunicação entre os diversos componentes realiza-se através do Hub. O servidor de conteúdos tem como funções receber os pedidos do canal de comunicação, distinguir e direccionar cada pedido para a acção adequada. O servidor de dados tem como função servir os pedidos do servidor de conteúdos que envolvam o acesso à base de dados e resolver toda a interacção com a base de dados. O TTS tem como função sintetizar para fala os documentos enviados pelo PDA para tal efeito.

No anexo II está especificado em aspecto de tabela a forma das mensagens trocadas entre os componentes do módulo de recursos externos, como também com o PDA.

3.3.1 Servidor de Conteúdos

3.3.1.1 Análise dos Requisitos do Servidor de Conteúdos

O principal objectivo do servidor de conteúdos é receber e responder aos pedidos da aplicação do PDA. Os tipos de pedidos enviados pelo PDA são os seguintes:

- Leitura – pedido de um formulário de um determinado doente.
- Escrita – pedido de inserção de uma nova ficha de doente ou de uma actualização de uma ficha já existente.
- Informação – pedido de um documento associado a um determinado doente.
- Apagar – pedido de remoção de um doente da Base de Dados.
- Sintetizar – pedido para ouvir um exame escrito de um determinado doente.

Os pedidos agrupam-se em três conjuntos: o primeiro conjunto contém os pedidos que envolvem o servidor de dados (leitura, escrita e remoção); o segundo conjunto contém os pedidos de documentos que envolve apenas o servidor de conteúdos; o terceiro conjunto contém os pedidos de síntese para fala de documentos escritos.

3.3.1.2 Funcionamento

O servidor de conteúdos trata cada um dos três conjuntos de pedidos de forma diferente. No caso de pedidos de Leitura, Escrita e Remoção de fichas de doentes, o procedimento seguido é enviar o pedido para o servidor de dados. Este último realiza todas as operações necessárias que envolvem a interação com a base de dados e devolve a resposta ao servidor de conteúdos.

Por outro lado, o tratamento dos pedidos de documentos envolvem retirar o documento de um doente e enviá-lo para a aplicação do PDA. Os documentos associados aos doentes não estão na Base de Dados porque não foi possível inserir o conteúdo de ficheiros na Base de Dados XML. Estes documentos encontram-se em directorias indexadas pelo nº de cama, cuja gestão é efectuada pelo servidor de conteúdos.

Por último, resta o pedido para ouvir um relatório que está na forma escrita. Neste caso o servidor de conteúdos vai buscar o respectivo documento de texto do doente, envia-o para o TTS e aguarda pela resposta. Quando esta chega (na forma de documento de áudio) envia-a para a aplicação do PDA.

A figura 12 ilustra o fluxo de informação trocada entre o servidor de conteúdos com o PDA, o servidor de dados e o TTS.

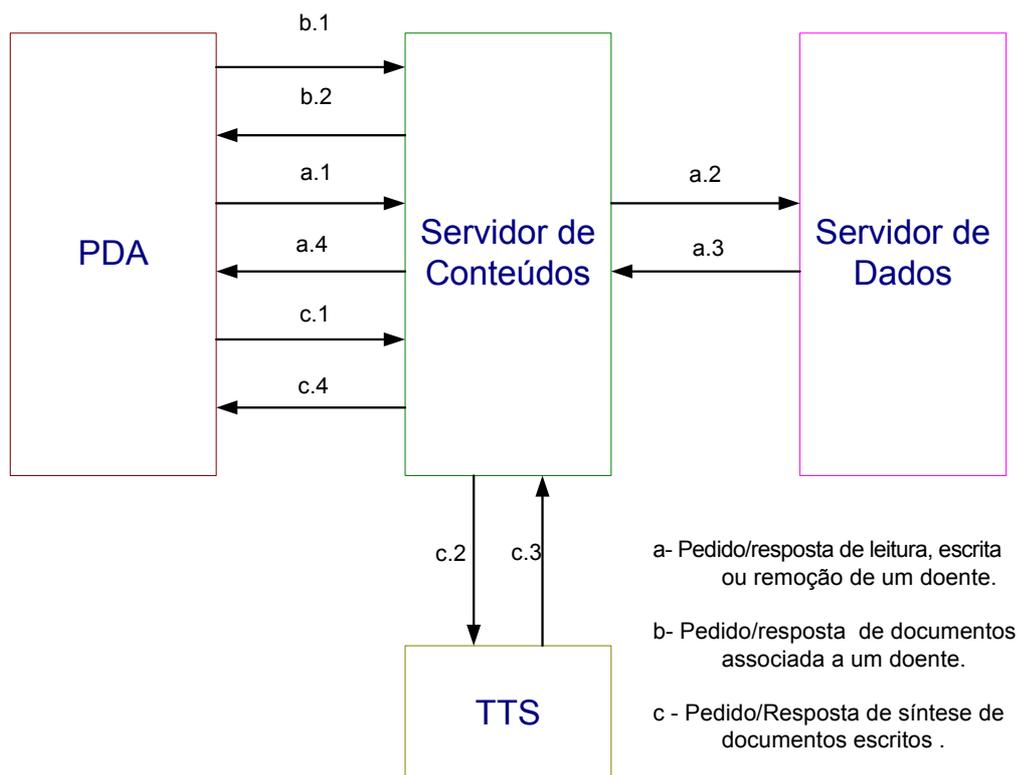


Figura 12 – Fluxo de mensagens entre o servidor de conteúdos com o PDA, o servidor de dados e o TTS

Em síntese, o diagrama de estados do servidor de conteúdos encontra-se na figura seguinte.

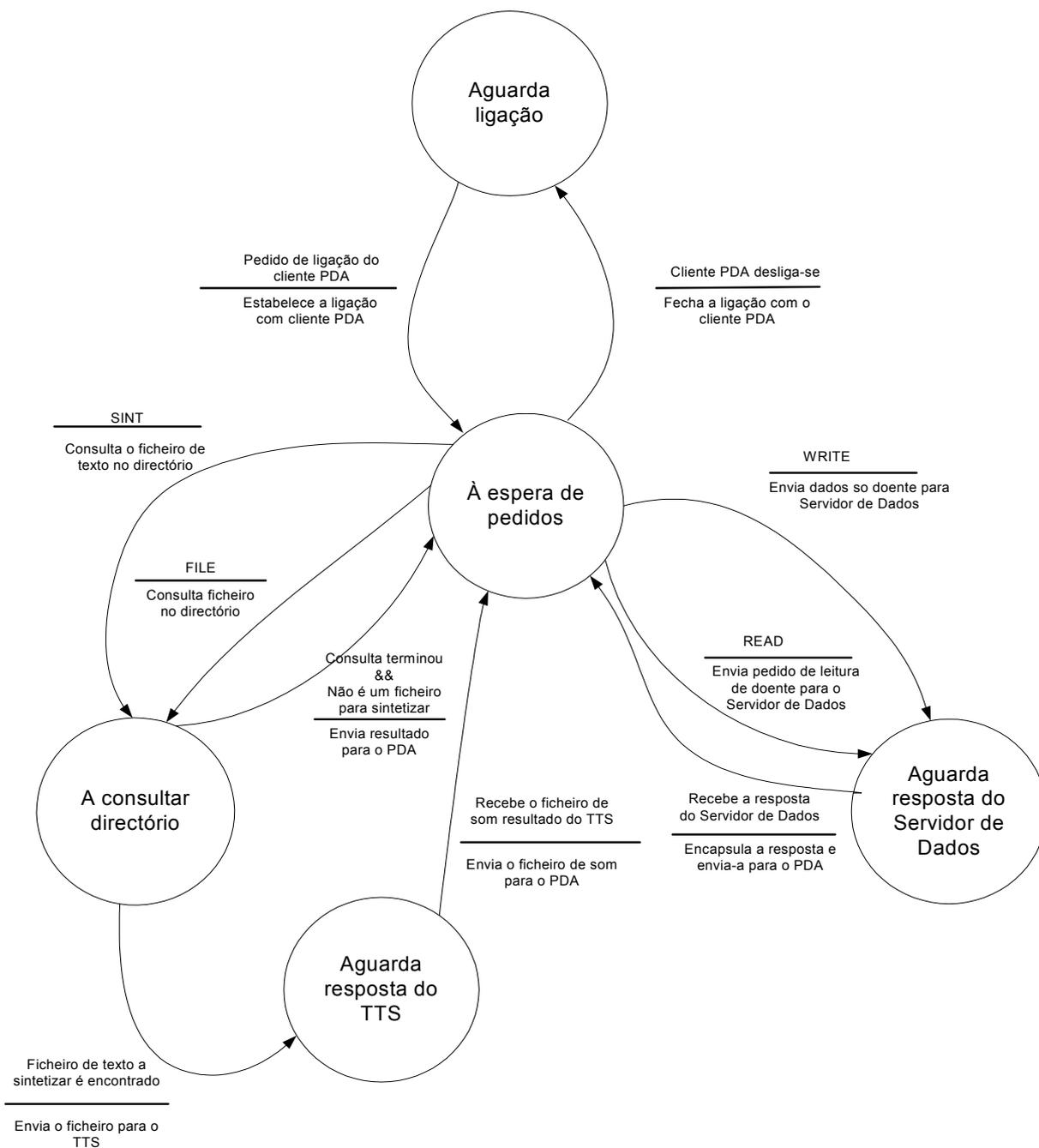


Figura 13 – Diagrama de estados do servidor de conteúdos

3.3.2 Servidor de Dados

3.3.2.1 Análise dos Requisitos do Servidor de Dados

O servidor de dados recebe e trata os pedidos do servidor de conteúdos que envolvem acesso à base de dados. Os tipos de pedidos possíveis são os seguintes:

- Inserir uma nova ficha de um doente: o objectivo é inserir na base de dados os dados de um novo doente.
- Ler uma ficha de um doente: o objectivo é consultar na base de dados os dados de um doente.
- Retirar uma ficha de um doente: o objectivo é retirar da base de dados os dados de um doente.
- Actualizar uma ficha de um doente: o objectivo é modificar os dados de um doente.

3.3.2.2 Organização dos Dados

O primeiro desafio inerente à construção do servidor de dados foi o de estruturar todos os dados que este terá que lidar. A informação manipulada é a dos campos do formulário de um doente. Assim criou-se uma classe doente que contém todos os tipos de informação associada a um doente: número da cama, nome, idade, sexo, número de segurança social, sala, piso, doença, estado, observações e um campo (*outros*) com a referência de todos os documentos associados ao doente. O armazenamento dos documentos de cada doente envolve guardar individualmente o nome e local onde estes estão localizados, já que estes não se encontram na base de dados. Assim criou-se uma classe documento que contém o nome do documento e o seu local de armazenamento, bem como as funções que permitem manipular estas duas informações. A relação entre a classe Doente e a classe Documento é de agregação, já que um doente tem documentos associados, mas também pode não ter. Na figura 14 encontra-se o diagrama de classe que serviu a programação do servidor de dados.

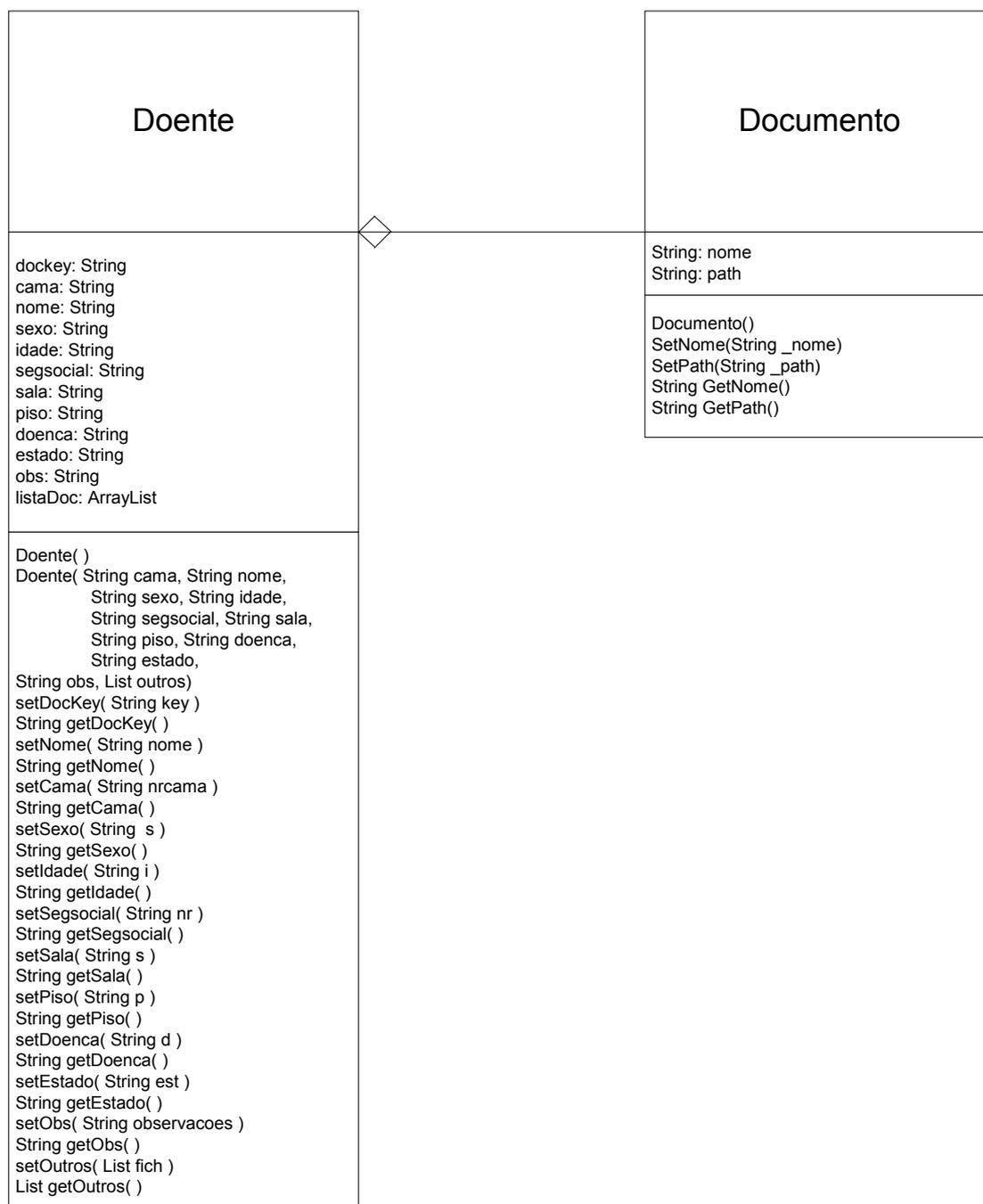


Figura 14 – Diagrama de classes do servidor de dados

3.3.2.3 Funcionamento

O servidor de dados recebe os pedidos do servidor de conteúdos e direcciona o pedido para a acção adequada. Existem quatro acções que o servidor de dados pode executar: ler, inserir, remover e alterar um doente da base de dados. No caso da leitura e remoção de um doente, a chave de indexação é o número de cama.

Quando se trata de inserir um novo doente, o pedido contém todos os valores dos campos da ficha desse doente. Com base nestes valores é então construído um ficheiro em formato XML e enviado para a base de dados.

No caso de se tratar de uma leitura de um doente, o servidor de dados usa o número da cama que lhe é enviado na mensagem de pedido para consultar a base de dados. Se o doente existir, o resultado da consulta vem em formato XML. Este resultado é então convertido para o formato da mensagem de resposta que o servidor de conteúdos aguarda. Caso ocorra algum erro, o servidor de conteúdos é informado da situação.

Na remoção de um doente retira-se o número da cama respectivo da mensagem do pedido. Este valor permite identificar o doente na base de dados e proceder à sua remoção. Caso ocorra alguma exceção na remoção, tal incoerência também é reportada ao servidor de conteúdos. Em síntese, o diagrama de estados do servidor de conteúdos encontra-se na figura seguinte.

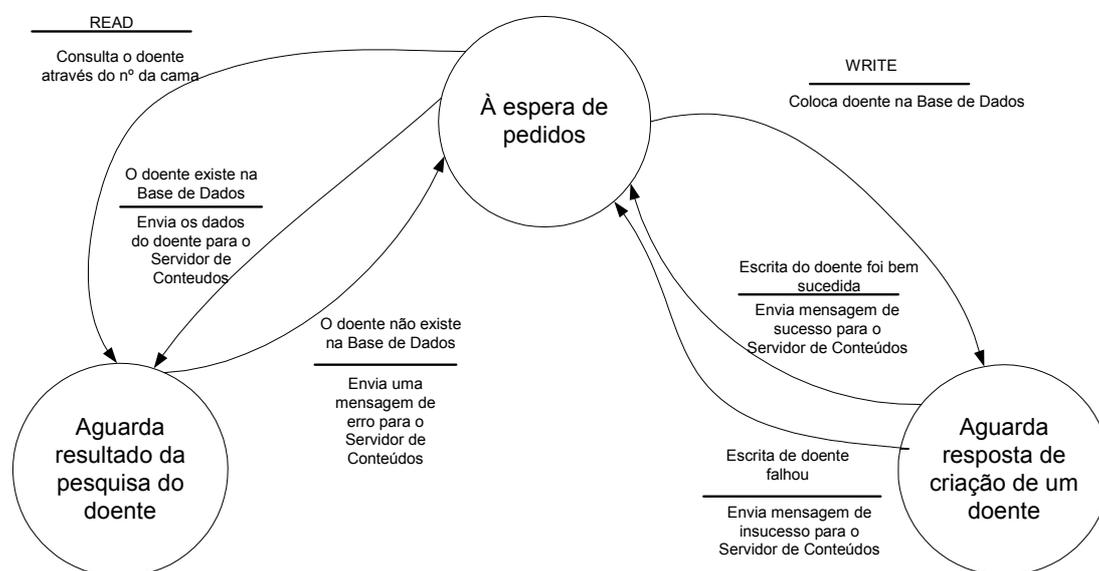


Figura 15 – Diagrama de estados do servidor de dados

3.3.3 HUB

O hub é um programa que permite a comunicação e a coordenação entre os diversos servidores. Trata-se de um módulo de software do projecto Galaxy Communicator. No caso específico deste trabalho, o hub permite a comunicação do servidor de conteúdos com o servidor de áudio e com o TTS.

O hub é responsável pelo carácter modular do sistema já que encapsula todas as informações associadas ao tráfego de mensagens. A reconfiguração do fluxo de mensagens implica apenas a alteração do ficheiro de configuração associado ao hub, não sendo necessário qualquer alteração adicional em nenhum dos servidores. A única invariante que estes devem respeitar é o nome das mensagens enviadas. O nome associado à mensagem será usado pelo hub como selector do destino que lhe será atribuído.

3.3.4 TTS

O TTS é responsável por converter as frases (texto) em ondas sonoras, interpretadas como fala pelo utilizador. Este programa usa o motor de síntese de fala Festival, com a voz do projecto Dixi+ em desenvolvimento no Laboratório de Língua Falada do INESC-ID.

Quando o servidor de conteúdos recebe do PDA um pedido para sintetizar um ficheiro de texto, envia o ficheiro para o TTS. O TTS converte o ficheiro recebido num ficheiro de áudio e envia-o para o servidor de conteúdos. O conteúdo do ficheiro final corresponde à sintetização de todas as frases do ficheiro de texto original. Por fim, o servidor de conteúdos envia o ficheiro de som para a aplicação do PDA, onde se procederá à audição do ficheiro.

3.3.5 Base de Dados

A Base de Dados contém um ficheiro XML para cada doente. Cada ficheiro tem um identificador associado que o permite distinguir dos restantes.

O formato do ficheiro XML de cada doente tem todos os campos do formulário do doente em *tags*. Um exemplo de um ficheiro XML de um doente encontra-se na figura 16.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<doente>
  <cama>2</cama>
  <nome>Filipa de Sousa Alves</nome>
  <sexo>Feminino</sexo>
  <idade>25</idade>
  <segsocial>88896544</segsocial>
  <sala>8</sala>
  <pisos>1</pisos>
  <doenca>Gravidez</doenca>
  <estado>Estável</estado>
  <obs>Feto regular e bem formado.</obs>
  <outros>
    <documento>
      <nomefich>Ecografia</nomefich>
      <path>Ecografia.wav</path>
    </documento>
  </outros>
</doente>
```

Figura 16 – Exemplo de um doente na Base de Dados.

4 Reconhecimento de fala

A área do reconhecimento de fala tem evoluído muito significativamente nos últimos anos. Um dos aspectos da investigação do L²F nesta área prende-se com o desenvolvimento de uma biblioteca de reconhecimento de fala para Português Europeu: a Audimus API. Com o objectivo de permitir utilizar a fala em diferentes aplicações, independentemente do sistema operativo.

Para conseguir que a aplicação do PDA utilize a fala, como método de entrada de dados, foi necessário usar a Audimus API. Com o intuito de descrever o funcionamento e modo de utilização da Audimus API, torna-se necessário introduzir conceitos importantes na área do reconhecimento de fala.

A base do reconhecimento de fala é o sinal de fala: sinal acústico que se propaga no ar através de uma onda sonora. Estas ondas sonoras podem ser representadas através de um gráfico de amplitude que representa a pressão do ar ao longo do tempo [Neto, 1998]. Após a aquisição do sinal de fala, o próximo passo consiste na extracção de informação relevante de uma forma eficiente: o processamento de sinal. Um outro conceito importante é o de reconhecimento de padrões: conjunto de algoritmos usados para agrupar dados, criar um ou mais padrões de um conjunto de dados e comparar um par de padrões nas bases de medidas características dos padrões.

Para completar esta breve introdução aos conceitos de reconhecimento de fala define-se modelo acústico e modelo da linguagem. Segundo o modelo acústico existem unidades fonéticas finitas e diferentes na linguagem falada. Estas unidades são amplamente caracterizadas por um conjunto de propriedades que são manifestadas no sinal da fala, ou no seu espectro, através do tempo. Por modelo da linguagem define-se o conjunto de todas as palavras possíveis de serem proferidas e as respectivas possibilidades de repetição. Estes dois modelos têm um papel fundamental no sucesso do reconhecimento de fala.

O presente capítulo pretende demonstrar como foi conseguido o objectivo de usar a fala na aplicação desenvolvida. No ponto 4.1 procura-se explicar os componentes e funcionamento da Audimus API. O estudo do funcionamento da Audimus API justifica-se no objectivo de migrar componentes do reconhecimento de fala para o PDA. No ponto 4.2 descreve-se como se conseguiu incluir a Audimus API no PDA, bem como quais os módulos que se executam no PDA. No ponto 4.3 estão explicados e justificados os parâmetros usados para configurar a biblioteca de reconhecimento. O modo de construção destes modelos encontra-se nos pontos 4.4 e 4.5. Por último, nos pontos 4.6, 4.7, 4.8 e 4.9 descreve-se como a aplicação do PDA usa as ferramentas de reconhecimento de fala descritas nos pontos anteriores.

4.1 Descrição da Audimus API

O objectivo deste ponto é apresentar uma visão geral do funcionamento da Audimus API.

4.1.1 Visão geral

A Audimus API é uma biblioteca de funções que permitem realizar reconhecimento de fala numa qualquer aplicação que se execute em Windows ou em Linux para processadores Intel x86. O modelo que está na sua base contém cinco blocos funcionais: AudioIn, User, Grammar, ResultSet e Engine. Os dois últimos encontram-se conceptualmente num nível mais elevado. Originalmente estes blocos estavam isolados e funcionavam de forma independente, mas com a implementação da Audimus API, foram unidos de forma a funcionarem de forma homogénea.

4.1.2 Os blocos funcionais

AudioIn

Tem como entrada um sinal de fala e com saída a representação do sinal numa forma mais compacta. É composto pelos sub blocos seguintes:

- Audio/End Point: Recebe o sinal de fala vindo da entrada do sistema e calcula a energia do sinal para decidir se trata de fala ou de silêncio, extraíndo a fala e eliminando o silêncio.
- Rasta: Faz a extração de características de cada bloco de som e representa-os na forma compacta. Para tal utiliza um algoritmo matemático de predição linear, neste caso uma Perceptual Linear Prediction (PLP), de ordem 12 da qual resultam 12 coeficientes mais a energia, perfazendo um total de 13 coeficientes. Estimando as derivadas temporais destes coeficientes obtêm-se um total de 26 coeficientes que são passados ao subloco seguinte, o Buffer.
- Buffer²: Este subloco tem a função de, no caso de existir retardamento na acção do bloco User, guardar a informação recebida para garantir a continuidade da acção do Audio/End Point e do Rasta.

User

Este bloco calcula a probabilidade de cada fone³ de um determinado idioma (no nosso caso, 40 fonemas para o Português) estar presente em cada bloco analisado. É composto unicamente pelo subloco Forward:

- O Forward implementa uma rede neuronal, o modelo do perceptrão multi-camada, tendo três níveis principais de informação. Um nível para entrada para diversas tramas de informação vindas do bloco anterior, uma camada escondida ou intermédia e a camada de saída correspondente às probabilidades dos fonemas de um idioma.

O vector de saída representa as probabilidades de na trama que se encontra ao centro no vector de entrada, estarem presentes cada um dos fones da língua portuguesa ou o silêncio.

² Buffer – Tampão. Zona de memória utilizada para guardar registos de informação temporariamente.

³ Fone –Unidade básica sonora. Som.

É o Forward que faz com que o resultado do sistema de reconhecimento seja diferente consoante o utilizador. Esta diferenciação é devida aos diferentes pesos presentes no perceptrão multi-camada, que basicamente são matrizes usadas para o cálculo dos valores da camada intermédia e a de saída. A cada utilizador correspondem pesos específicos.

Grammar

Este bloco finaliza o processo de reconhecimento de fala recebendo como entrada as probabilidades dos fones (vindo do bloco User) e tem como saída a frase mais provável que o orador terá proferido.

Engine

Permite ao utilizador lançar a execução de um motor de reconhecimento de fala e manipular o seu estado, bem como especificar o comportamento deste motor a nível de retorno de resultados.

ResultSet

Define um conjunto de funções que servem de interface entre o programador e o que este pretende obter do motor de reconhecimento de fala. [Cassaca e Maia, 2002]

4.2 Gerar a biblioteca Audimus no PDA

De modo a compilar a biblioteca Audimus API no PDA, procedeu-se à compilação sequencial de cada um dos módulos que a compõem. Os blocos foram inseridos de forma gradual, de modo a atingir um equilíbrio entre o desempenho e a independência da aplicação face a processamento externo.

4.2.1 Principais obstáculos

Na tentativa de compilar a Audimus API no Pocket PC, verificou-se que as bibliotecas do sistema do Pocket PC 2000 apresentam algumas diferenças em relação às bibliotecas do Windows 32 bits. De seguida apresenta-se a listagem de problemas que ocorreram na compilação de cada bloco.

Problemas comuns a todos os blocos:

- O Pocket PC 2000 não suporta o mecanismo de comunicação entre processos conhecido por *pipes*.

O bloco AudioIn apresentou os seguintes problemas:

- Não foram reconhecidos os seguintes cabeçalhos: `process.h`, `io.h`, `fcntl.h`, `pthread.h`, `time.h`, `sys/types.h` e `errno.h`.

O bloco User apresentou o seguinte problema:

- Não é possível usar as bibliotecas MKL (Math Kernel Library) da Intel no Pocket PC 2000.

O bloco Decoder apresentou o seguinte problema:

- Não foram reconhecidas várias funções e tipos de dados do C++.

4.2.2 Obstáculos ultrapassados

Apenas foram ultrapassados os obstáculos referentes ao bloco AudioIn. O que significa que a Audimus API irá funcionar de duas formas: distribuído e remoto. No caso do processamento distribuído, o bloco AudioIn é executado no PDA e o resultado vai ser enviado para o servidor de áudio, que contém os blocos User e Grammar. No caso do processamento remoto o reconhecimento é efectuado totalmente no servidor de áudio.

Apesar de os cabeçalhos mencionados anteriormente, não existirem as funções encontram-se implementadas no Pocket PC. No entanto, no caso da manipulação de *threads* houve a necessidade de substituir as funções de criação e terminação das mesmas.

A resolução do problema dos pipes envolveu a criação de funções que simulam o funcionamento dos pipes. Para isso foram utilizados os procedimentos de manipulação de ficheiros do Pocket PC.

4.2.3 Obstáculos por ultrapassar

Os blocos User e Grammar não funcionam no Pocket PC. No caso do bloco User este precisa de uma conjunto de bibliotecas para realizar operações matemáticas e otimizadas para o processador Pentium. Estas bibliotecas são da Intel e denominam-se por MKL. As bibliotecas em causa contêm funções matemáticas que permitem, por exemplo a manipulação de matrizes, imprescindíveis no reconhecimento de padrões através de redes neuronais. Neste momento não existe uma versão destas bibliotecas para Pocket PC. Como a implementação desta biblioteca no Pocket PC está fora do âmbito deste trabalho, não foi possível colocar o bloco User no PDA.

Quanto ao bloco Grammar, este está codificado na linguagem C++ (os restantes blocos estão em C). Contudo, existem inúmeras funções que a biblioteca de C++ do Pocket PC não suporta. A resolução seria implementar cada uma delas, o que constituiria também um trabalho demasiado complexo.

4.2.4 Revisão das funcionalidades do sistema

Devido aos problemas de compilação e execução da Audimus API no Pocket PC o reconhecimento de fala não será totalmente realizado no PDA, apenas serão executados os blocos que funcionam neste dispositivo. O bloco AudioIn executa-se no PDA e o resultado será enviado para o bloco User que se executa no servidor de áudio. Porém continua a haver a possibilidade de realizar o reconhecimento de fala, de uma forma totalmente remota, em que o servidor de áudio processa todos os blocos.

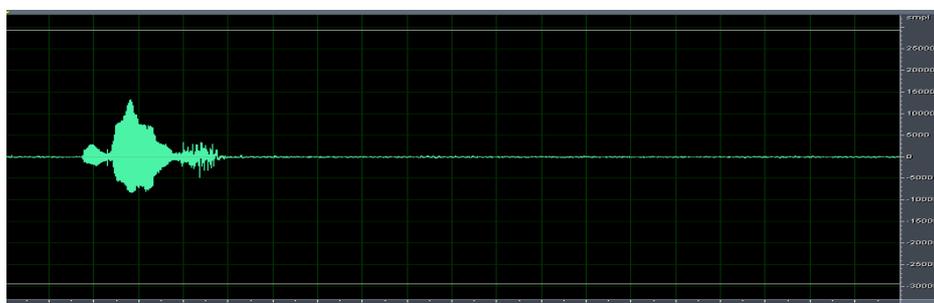
4.3 Configuração da Audimus API

Como foi referido no ponto 4.1.2, o bloco End Point recebe o sinal de som do microfone do PDA e calcula a energia do sinal para decidir se trata de fala ou de silêncio. Para que a decisão entre fala e silêncio seja a mais acertada possível foi necessário adequar os parâmetros utilizados no algoritmo do End Point às características do sinal de fala produzido no PDA. O funcionamento do detector baseia-se numa máquina de estados onde o troço de sinal de fala é definido por três

estados: HEAD, BODY e TAIL. As mudanças de estado são controladas por limiares de energia (thresholds) e por intervalos de tempo nos quais a energia terá que permanecer acima dos respectivos limiares. Existem três limiares de energia: *head threshold*, *body threshold* e *tail threshold*. Quando o sinal de fala atinge o *head threshold* e permanece acima deste limiar durante um intervalo de tempo denominado por *head max time*, existe a possibilidade de neste intervalo de tempo corresponder a um início de uma frase. O trecho do sinal está assim no estado HEAD. Para que o trecho do sinal seja considerado como frase é necessário que a energia do sinal atinja o limiar *body threshold* e permaneça acima deste limiar durante um intervalo mínimo de tempo denominado por *seg min time*. Neste caso, o sinal passa para o estado BODY. Para determinar o final da frase é necessário que o sinal de fala fique abaixo de um valor de energia imposto pelo limiar *tail threshold*, durante um intervalo de tempo denominado por *seg extended time*. Nesta situação, o sinal transita para o estado de TAIL.

Para uma melhor compreensão do trabalho realizado neste ponto convém analisar as diferenças entre os sinais produzidos pelo microfone do PDA e pelo microfone do PC. O sinal de referência é o sinal produzido pelo microfone do PC. O objectivo é conseguir um resultado de End Point para o sinal do PDA o mais aproximado possível daquele que é apresentado para o microfone do PC. As figuras seguintes ilustram as diferenças entre os dois sinais e os resultados da aplicação do End Point.

a)



b)

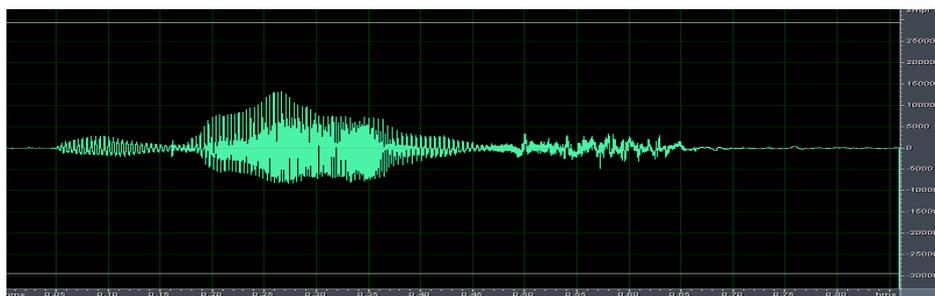
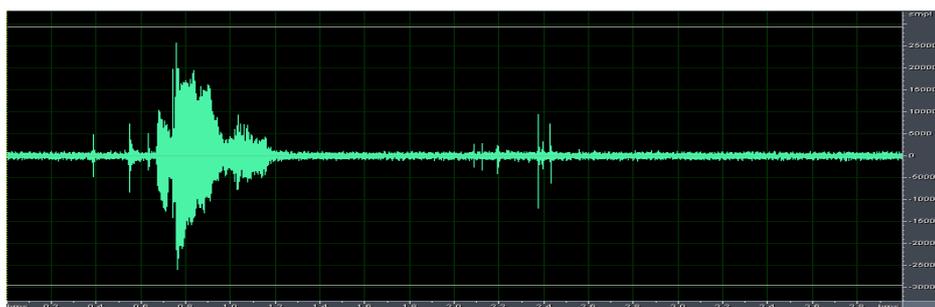


Figura 17 – Representações da palavra "dois": a) representação através do gráfico de amplitude do sinal de fala com o microfone do PC e b) representação do resultado do End Point aplicado a a) com os parâmetros do PC.

a)



b)

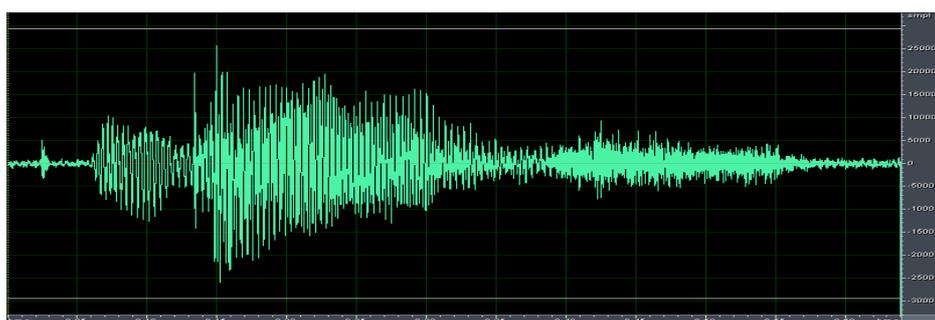


Figura 18 – Representações da palavra "dois": a) representação através do gráfico de amplitude do sinal de fala com o microfone do PDA e b) representação do resultado do End Point aplicado a a) com os parâmetros do PDA.

A frase utilizada para ilustrar as diferenças é constituída apenas pela palavra “Dois”. Na figura 17 a) e 18 a) encontra-se representado o gráfico de amplitude do

sinal de fala para a palavra “dois” produzido com o microfone do PC e produzido com o microfone do PDA. As diferentes amplitudes dos dois sinais são evidentes. A amplitude do ruído no sinal do PDA é superior em relação à amplitude do ruído do sinal do PC. O que demonstra que o End Point terá que usar valores diferentes nos seus parâmetros para o PDA.

Os valores atribuídos aos limiares e intervalos de tempo estão nas tabelas seguintes, para o microfone do PC e para o microfone do PDA.

Microfone do PC	
Parâmetros	Valores
Head Threshold	3.0 db
Body Threshold	9.0 db
Tail Threshold	5.0 db
Head Max Time	0.1 s
Seg Min Time	0.25 s
Seg Extended Time	0.15 s

Tabela 3 – Valores dos parâmetros de End Point para o microfone do PC.

Microfone do PDA	
Parâmetros	Valores
Head Threshold	4.0 db
Body Threshold	15.0 db
Tail Threshold	9.0 db
Head Max Time	0.04 s
Seg Min Time	0.5 s
Seg Extended Time	0.04 s

Tabela 4 – Valores dos parâmetros de End Point para o microfone do PDA.

Como se pode observar nas tabelas 3 e 4 existem diferenças entre os valores dos limiares de energia para o sinal de fala produzido pelo microfone do PC e do PDA. Os limiares (*head threshold*, *body threshold* e *tail threshold*) têm valores superiores para o PDA porque a energia do sinal produzido pelo microfone do PDA é superior em relação àquela que é produzida pelo microfone do PC. As diferenças de energia entre o microfone do PC e do PDA são visíveis nos sinais ilustrados nas figuras 18a) e 19a). A duração máxima do início e do fim do segmento de fala (*head max time* e *seg extended time*) para a configuração do PDA é inferior em relação à configuração do PC. Isto deve-se ao facto do sinal produzido pelo microfone do PDA apresentar um ruído de fundo com um maior nível de energia e com mais variações em relação ao que é apresentado pelo sinal do PC.

A duração mínima de um segmento de fala (*seg min time*) é maior para o sinal do PDA comparativamente com o do PC. Tal justifica-se porque para realizar a aquisição de fala no PDA é necessário carregar num botão e o respectivo ruído é integralmente apanhado pelo microfone do PDA. Assim, o valor do parâmetro *seg min time* corresponde a um intervalo de tempo superior à duração do barulho associado ao clicar do botão, com o objectivo deste ruído ser eliminado.

A escolha dos valores exactos atribuídos a cada parâmetro foi determinada experimentalmente e envolveu um elevado número de tentativas até encontrar uma configuração adequada.

A figura 18 b) mostra o resultado do End Point usando os valores por nós definidos. Verifica-se um corte significativo das zonas de ruído do sinal original. Verifica-se também que a qualidade do microfone do PDA é baixa, pois o sinal de fala produzido apresenta muito ruído.

4.4 Modelo Acústico

Para adaptar o modelo acústico usado na Audimus API à fala a um dos elementos executantes deste trabalho foi necessário criar um novo ficheiro de pesos para a rede neuronal que compõe o bloco User. O programa que permitiu realizar esta tarefa foi o Recolha que tem por base o programa Neural Network Simulator. A utilização do programa Recolha envolve gravar um conjunto de frases, ricas foneticamente, com um microfone. O programa Recolha quando foi utilizado neste trabalho só possuía uma versão para o Windows NT. Como era necessário adaptar o modelo acústico ao microfone do PDA e compilar o programa Recolha, para o PDA, é uma tarefa demasiado complexa. Foi criada uma aplicação para o PDA que permite efectuar a gravação das frases do programa Recolha e aplicar o algoritmo de End Point nos ficheiros gravados. Estas operações estão dependentes do microfone e do dispositivo e têm de ser feitas no PDA. Seguidamente, os ficheiros gravados no PDA foram copiados para o programa Recolha de forma a realizar o modelo acústico do orador. As frases do programa Recolha foram gravadas simultaneamente no PDA e no PC de forma a comparar o resultado da adaptação dos dois modelos de linguagem como o mesmo orador.

O resultado obtido no final do treino na tarefa de validação cruzada foi de 60.93% de frases correctas para o PDA e 78.43% de frases correctas para o PC. Como o resultado obtido para o PDA é inferior em mais de 10 pontos percentuais em relação ao obtido para o PC, resolveu-se adaptar o modelo acústico usando novamente o programa Neural Network Simulator, mas agora partindo de pesos obtidos no modelo acústico de um Telejornal, já que o sinal de fala do Telejornal apresenta muito ruído em situações de reportagens exteriores. Os resultados obtidos foram agora de 77.64% de frases correctas, o que constitui um valor mais aproximado do valor percentual obtido pelas frases gravadas com o microfone do PC.

4.5 Modelo da linguagem

Como os campos do formulário incluem nomes de pessoas, números, doenças e texto corrente foi necessário gerar um modelo de linguagem específico para a aplicação do PDA. Para gerar este modelo existem duas alternativas:

1. Geração de um modelo genérico que inclua todas as possíveis palavras a serem usadas no formulário;
2. Geração de quatro modelos de linguagem diferentes: números, nomes, patologias e vocabulário corrente.

A alternativa que conduzirá a melhores resultados é a segunda. No entanto será necessário que a Audimus API permita a troca de modelos de linguagem, que até ao momento não é possível.

A alternativa de um modelo genérico poderá degradar o desempenho e sucesso do reconhecimento, já que aumenta o número de possibilidades de diferentes resultados. Optou-se pela solução do modelo genérico, ou seja, criou-se um modelo de linguagem com cerca de 100 palavras que englobam desde nomes, apelidos a doenças.

4.6 Aquisição de Fala no PDA

A aquisição da fala é realizada através do microfone incorporado no PDA, utilizando a *Low level Digital Audio* como biblioteca de áudio. As rotinas de

aquisição de áudio foram baseadas num programa exemplo, o WaveIn, que efectua a gravação digital de áudio usando dois Tampões. [WaveIn, 2002]

A amostragem do sinal de áudio é feita através de PCM (*Pulse Code Modulation*), com os parâmetros da tabela seguinte:

Variável	Valor
Nº de canais (Mono/Stereo)	1
Nº de amostras recolhidas por segundo	16000
Nº de bits que compõem uma amostra	16

Tabela 4 – Parâmetros da amostragem

O processo de aquisição é iniciado através da abertura de um dispositivo de entrada de áudio com os seguintes parâmetros: uma estrutura que contém os valores que definem a amostragem PCM e um apontador para uma *thread* de retorno. Não é necessário indicar qual o dispositivo a utilizar porque a biblioteca de áudio faz uma procura por todos os dispositivos de entrada de áudio que suportem os parâmetros de amostragem desejados. Apenas o primeiro dispositivo a ser encontrado é utilizado, mas como o PDA possui apenas um microfone este será de certo utilizado.

A *thread* de retorno recebe as amostras do dispositivo e faz o respectivo processamento. A comunicação entre a *thread* de retorno e o dispositivo é feita com recurso a tampões, que são trocados entre as duas partes através de mensagens. Durante a aquisição, o dispositivo realiza a amostragem e guarda o resultado num tampão. Quando o tampão estiver completo é enviado para a *thread* de retorno através de uma mensagem.

O tampão é recebido pela *thread* de retorno que faz o respectivo processamento conforme o estado da aplicação. No final o tampão é associado novamente ao dispositivo. O tampão fica então disponível para receber novas amostras do dispositivo.

A amostragem é um processo que não pode ser interrompido, devem portanto, ser utilizados no mínimo dois tampões. Quando a *thread* de retorno estiver a

processar um tampão existe outro disponível para o dispositivo gravar as amostras.

A figura 19 ilustra o processo de aquisição de fala.

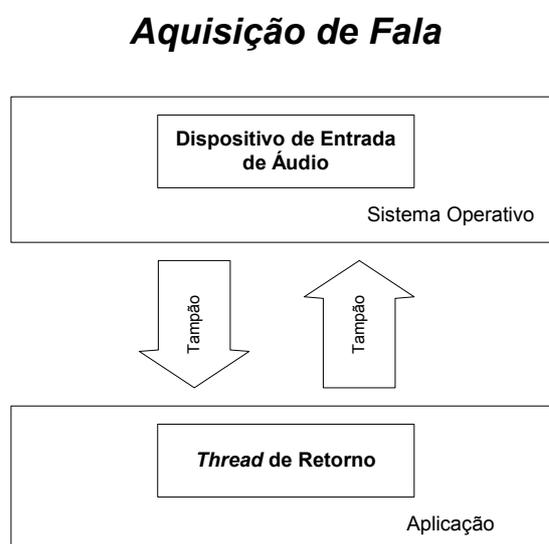


Figura 19 – Processo de aquisição de fala

Na aplicação, a aquisição de fala só é efectuada quando se está a preencher um dado campo do formulário. Assim é necessário limitar a aquisição a estes períodos, de forma a evitar processamento e transmissões de dados desnecessárias.

Para tal é feita uma selecção, na *thread* de retorno, dos tampões que chegam do dispositivo. Essa selecção é alterada pelo utilizador quando carrega no botão de gravação para preencher um campo do formulário.

Porém este processo apresenta o inconveniente de ser pouco preciso. Porque cada tampão pode conter uma grande quantidade de amostragens (32.000). Ou seja, em casos extremos a fala do utilizador pode ficar cortada no final porque o tampão que veio do dispositivo chegou pouco tempo depois de o utilizador ter terminado a aquisição. Para remediar este problema no final da aquisição é sempre aceite mais um tampão. A figura 20 ilustra esta situação. Se não se aceitar o tampão nº 3 na aquisição de fala, fala fica cortada no final.

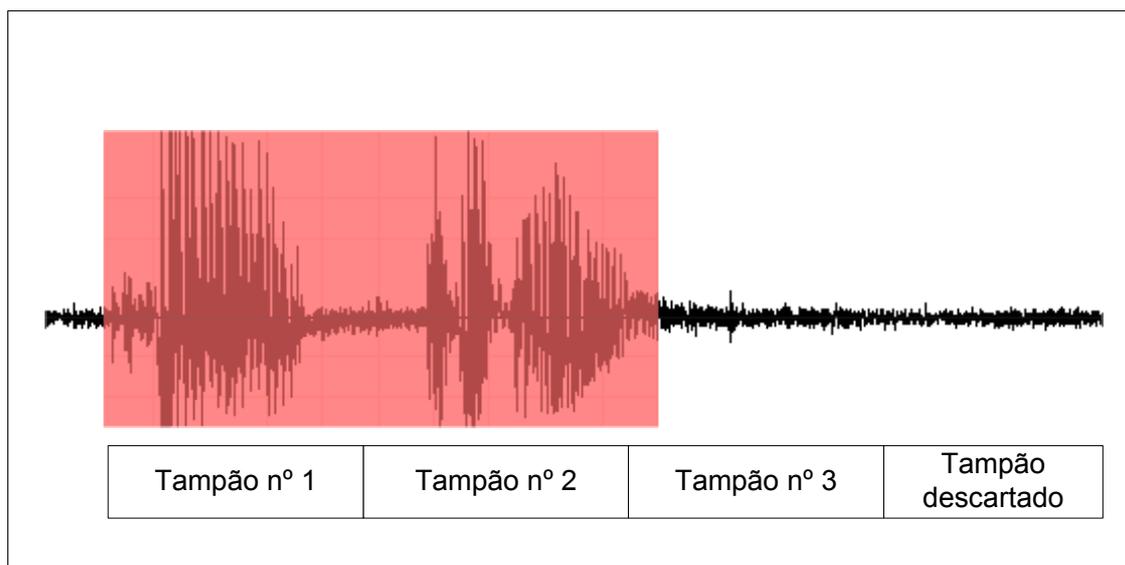


Figura 20 – Aquisição através de tampões.

4.7 Suporte ao reconhecimento de fala no PDA

O reconhecimento de fala é realizado, quase na sua totalidade, de forma remota no servidor de áudio. Assim, foi necessário criar uma interface entre o módulo do PDA e o servidor de áudio que permitisse realizar o reconhecimento de fala de uma forma transparente e coordenada. O processo envolve o envio e recepção de dados descritos nos seguintes pontos:

- **Envio de dados para o servidor de áudio**

Como grande parte do reconhecimento de fala é feito remotamente no servidor de áudio, é necessário enviar os dados do PDA para servidor de áudio.

- **Recepção de resultados**

A recepção de resultados permite ao PDA receber do servidor de áudio o resultado do reconhecimento de fala. Este resultado é recebido, palavra a palavra e no final quando se chega ao fim da frase é devolvido um indicador. Logo após a recepção das palavras, estas são enviadas através de mensagens para a aplicação. Caso seja um fim de frase é enviada uma mensagem indicando que o processamento da frase finalizou.

4.8 Interação com os formulários

O formulário médico tem os seus campos dispersos por várias janelas de diálogo. É da responsabilidade da janelas de diálogo activa receber os resultados provenientes da *recepção de resultados* e actualizar os campos dos formulários com essa informação.

Estas informações são enviadas para as janelas de diálogo através de mensagens de eventos. As janelas de diálogo recebem os dois tipos de mensagem ilustrados na tabela seguinte:

Mensagem	Descrição
WM_RECORD_CONTROL	Mensagem que contém informações sobre o estado da aplicação. Esta mensagem é enviada pela <i>Aquisição de Fala</i> para a <i>Interface</i> .
WM_SPEECH_RECON	Mensagem com o resultado do reconhecimento. Esta mensagem é enviada da <i>Recepção de Resultados</i> para a <i>Interface</i> .

Tabela 5 – Tipos de mensagens da aplicação.

As mensagens de WM_RECORD_CONTROL indicam à janela de diálogo qual o estado da aplicação. Os estados possíveis encontram-se na tabela 6.

Mensagem	Descrição
Parado	Não se está a fazer aquisição de fala e nem se está a efectuar nenhum processamento.
Em Gravação	A aplicação está a fazer aquisição da fala.
Processar	A aplicação está a processar.

Tabela 6 – Estados da aplicação

As mensagens de WM_SPEECH_RECON contêm a palavra resultante do reconhecimento. A janela de diálogo activa pega nessa palavra e actualiza o campo seleccionado pelo utilizador. Se o campo é uma Caixa de Texto, adiciona essa palavra ao texto inserido da caixa. Caso seja uma Caixa de Selecção é feita uma pesquisa pelos valores dessa caixa. O valor a ser seleccionando é o que for igual à palavra que resultou do reconhecimento, caso não haja nenhum valor igual é seleccionado o primeiro valor (“não foi definido”).

O utilizador selecciona o campo, que deve ser preenchido através de um clique. Existe sempre um campo seleccionado, as janelas de diálogo quando são iniciadas seleccionam automaticamente um campo e essa selecção só é alterada quando outro campo é seleccionado. A figura seguinte ilustra o indicador de campo seleccionado.

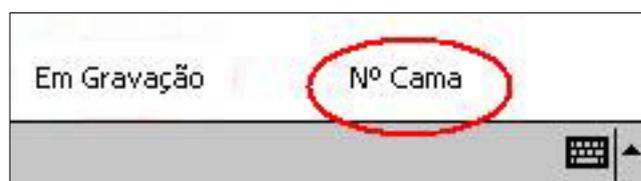


Figura 21 – Indicador de campo seleccionado

No que diz respeito a este trabalho a fala do utilizador é utilizada para preencher campos de um formulário, logo os resultados do processamento praticamente não precisam de ser interpretados, salvo algumas traduções de números por extenso para dígitos.

Neste trabalho o processamento dos resultados do reconhecimento de fala que é realizado ao nível da interface está implementado nas janelas de diálogo. Isto porque esse processamento está relacionado com os campos da interface, que variam de Janela de Diálogo para Janela de Diálogo.

4.9 Servidor de áudio

No caso do reconhecimento distribuído, o servidor de áudio só realiza o processamento relativo aos últimos dois blocos da Audimus API (User e Grammar). No caso de reconhecimento remoto, efectua o processamento de todos os blocos.

O servidor de áudio tem uma implementação muito simples, é constituído pela biblioteca Audimus API e pela interface de comunicação com o PDA. Entre o PDA e servidor de áudio existem 2 ligações, descritas na tabela abaixo:

Canal	Descrição
Ligação de Fala	Usada pelo PDA para enviar dados para o Servidor de Audio.
Ligação de Resultados	Usado pelo Servidor de Audio para enviar resultados para o PDA.

Tabela 7 – Canais de comunicação

O PDA envia os dados para o servidor áudio através da ligação de fala. Estes dados podem ser amostras PCM ou resultados do Rasta caso o PDA esteja a realizar processamento distribuído. Os dados são lidos do socket e copiados para um pipe ligado à Audimus API. Os resultados do processamento da Audimus API são expedidos através de uma thread, para o PDA através da ligação de Resultados.

Foi necessário reescrever algumas funções da Audimus API, para que fosse possível realizar o processamento só com dois blocos, recebendo os dados do Rasta como entrada.

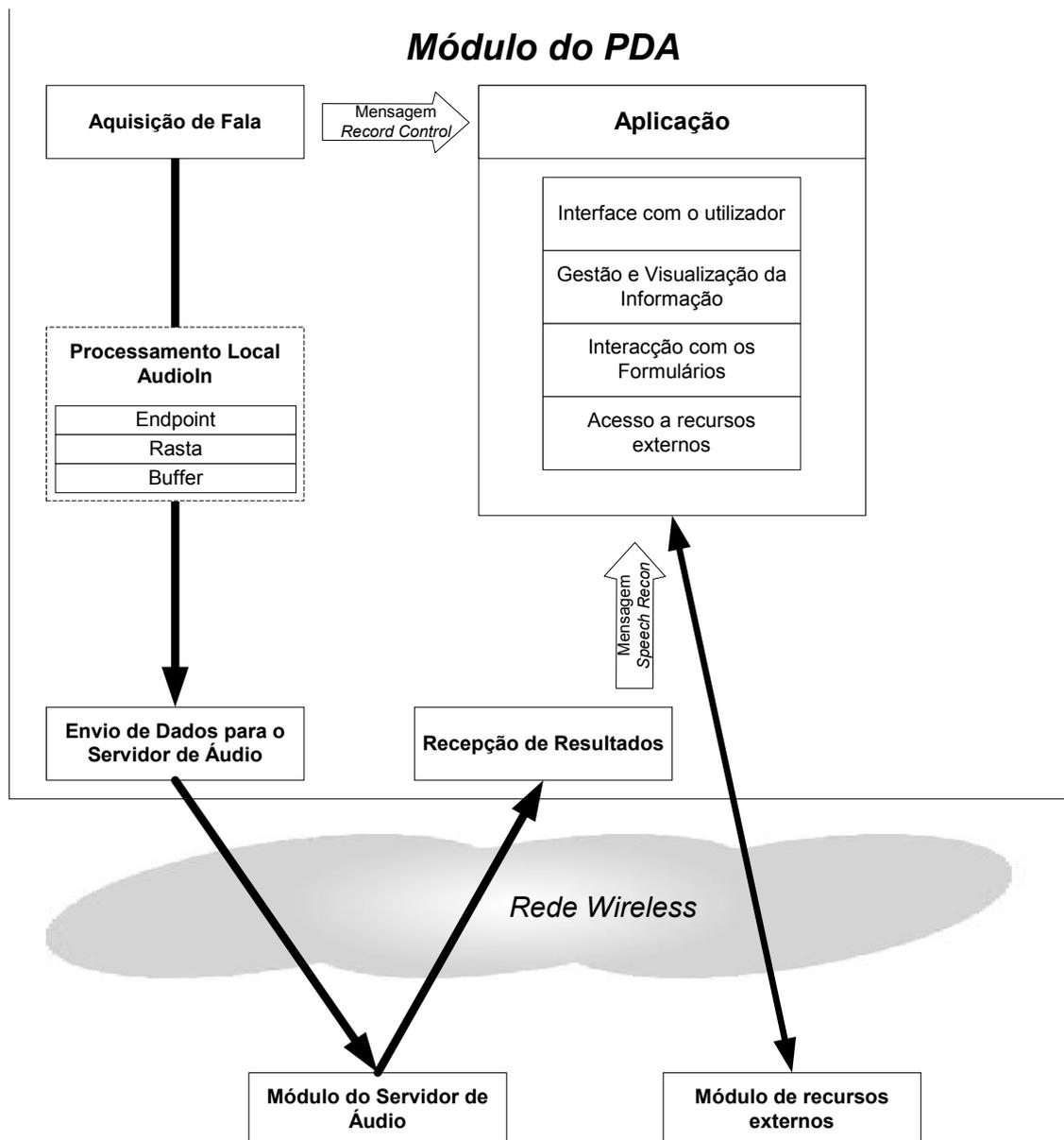


Figura 22 – Esquema geral do funcionamento da aplicação

5 Resultados

Os testes seguintes têm como objectivo avaliar o desempenho do sistema nas suas várias configurações, comparar o uso do novo método de preenchimento de formulários (fala) com os métodos anteriores. Finalmente, pretende-se também avaliar o sucesso do reconhecimento da fala.

5.1 Desempenho do Reconhecimento

O uso da aplicação desenvolvida mostra que a acção de preencher um campo do formulário não é instantânea. Desde o momento que o utilizador fala até aparecer a frase correspondente no campo é necessário um tempo de espera. Assim, convém determinar o atraso introduzido pelo tempo de transmissão na rede wireless e pelo reconhecimento da fala. As entidades envolvidas neste teste são o módulo do PDA e o servidor de áudio.

No PDA, é determinado o intervalo de tempo que começa com o envio da fala e acaba na recepção do resultado em texto. A este tempo vamos chamar T_{total} .

Para determinar o tempo dispendido no reconhecimento de fala foi necessário recorrer a um programa auxiliar. Este programa contém apenas o processamento da Audimus API. O tempo aqui obtido domina-se por $T_{Audimus}$. Outra alternativa seria no Servidor de Áudio determinar o tempo total dispendido no reconhecimento da fala recebida. No entanto, este tempo não corresponde apenas a reconhecimento, também se pode traduzir em esperas de dados a processar devido ao atraso dos pacotes de fala.

O tempo gasto na transmissão de dados, pelo facto da aplicação ser distribuída, ($T_{distribuido}$) inclui a soma dos tempos de aquisição da fala no PDA, de transmissão e propagação. Este valor é obtido em termos absolutos, através da diferença do tempo medido PDA com o tempo de processamento do Reconhecimento.

A estratégia para realizar este teste foi a seguinte:

- Criar três ficheiros de fala com frases típicas de preenchimento dos campos da aplicação:
 - Frase 1: 2 (Tamanho = 125 Kbytes)

- Frase 2: 2 *Armando Salgueiro Masculino* (Tamanho = 500 Kbytes)
- Frase 3: 2 *João Sousa Masculino Grave Osteoporose* (Tamanho = 750 Kbytes)
- Retirar cinco registos dos tempos T_{total} e $T_{Audimus}$ para cada ficheiro.
- Repetir o teste para um dos ficheiros posicionando o PDA a 20m do ponto de acesso.

5.1.1 Tempos de reconhecimento de fala isolado

A Audimus API executa-se num programa isolado no PC que tem como única função o reconhecimento de fala. O programa tem como entrada um ficheiro de fala e apresenta como saída o resultado do reconhecimento dessa mesma fala. O objectivo é determinar o desempenho da Audimus API com um processamento e entradas locais. Os dados de fala não provêm da rede, como acontece na aplicação que envolve o PDA. Os resultados obtidos para cada uma das frases encontram-se na tabela seguinte.

Registo	Tempo Total (ms)		
	Frase 1	Frase 2	Frase 3
1	251	1202	792
2	121	1251	791
3	220	631	791
4	260	631	1773
5	250	1241	1552
Média (ms)	220,4	991,2	1139,8
Desvio Padrão (ms)	57,6	329,3	483,5

Tabela 8 – Tempos de reconhecimento de fala obtidos para cada um dos três ficheiros, usando a Audimus API num programa isolado.

Da análise da tabela 7, verifica-se que o tempo de reconhecimento aumenta com o aumento do tamanho dos dados de fala a reconhecer.

5.1.2 Tempos obtidos com o PDA a 1 m do ponto de acesso

As tabelas seguintes apresentam os resultados obtidos para cada um dos ficheiros, com o PDA situado a 1 m de distância do ponto de acesso.

a)

Frase 1	
Registo	T_total (ms)
1	525
2	503
3	522
4	506
5	509
Média (ms)	513
Desvio Padrão (ms)	9,87

b)

T_total (ms)	T_audimus(ms)	T_distribuido (ms)
513	220	293

Tabela 9 – Resultados dos tempos obtidos para a frase “Dois”

a)

Frase 2	
Registo	T_total (ms)
1	2459
2	2535
3	2541
4	2439
5	2404
Média (ms)	2475,6
Desvio Padrão (ms)	60,30

b)

T_total (ms)	T_audimus(ms)	T_distribuido (ms)
2476	991	1485

Tabela 10 – Resultados dos tempos obtidos para a frase “Dois Armando Salgueiro Masculino”

a)

Frase 3	
Registo	T total (ms)
1	3590
2	3597
3	3608
4	3570
5	3555
Média (ms)	3584
Desvio Padrão (ms)	21,31

b)

T_total (ms)	T_audimus(ms)	T_distribuido (ms)
3584	1140	2444

Tabela 11 – Resultados dos tempos obtidos para a frase “Dois João Sousa Masculino Grave Osteoporose”

Através da análise da tabela 8 verifica-se que o tempo gasto pelo reconhecimento de fala é 43% do tempo total. No caso da tabela 9 este valor é de 40% e na tabela 10 é de 32%. Assim, constata-se que em relação ao tempo de total de processamento remoto, a transmissão de dados é a operação que ocupa mais tempo. O tempo de transmissão também inclui as operações de leitura e envio de blocos do ficheiro.

5.1.3 Tempos obtidos com o PDA a 20 m do ponto de acesso

A tabela seguinte apresenta os resultados obtidos para o ficheiro três, com o PDA situado a 20 m de distância do ponto de acesso.

a)

Frase 3	
Registo	T_total (ms)
1	3588
2	4084
3	3655
4	3796
5	3633
Média (ms)	3751,2
Desvio Padrão (ms)	201.63

b)

T_total (ms)	T_audimus (ms)	T_distribuido (ms)
3751	1140	2611

Tabela 12 – Resultados dos tempos obtidos para a frase “Dois João Sousa Masculino Grave Osteoporose” com o PDA a 20 m de distância do ponto de acesso.

Os dados da tabela 12 em comparação com os da tabela 11 revelam que o tempo de processamento distribuído aumenta com o aumento da distância do PDA ao ponto de acesso.

5.1.4 Comparação da capacidade de processamento entre o PC e o PDA

O objectivo é testar e comparar quais as capacidades do PC e do PDA ao nível das operações matemáticas e ao nível do sistema de ficheiros. Já que toda a aplicação desenvolvida envolve inúmeras operações matemáticas (especialmente no processamento de fala) e acessos a ficheiros (implementação dos *pipes*). Os resultados obtidos neste ponto estão no anexo V.

5.1.5 Diferenças de desempenho entre reconhecimento de fala remoto e distribuído

Como foi referido no ponto 3.2 a biblioteca de reconhecimento de fala Audimus API pode funcionar de forma distribuída. O bloco AudioIn executa-se na aplicação do PDA e os restantes blocos executam-se no servidor de áudio. No entanto, existe também a possibilidade do reconhecimento ser totalmente realizado remotamente no

Servidor de Áudio. Assim torna-se útil analisar a diferença de desempenho entre o reconhecimento de fala realizado de forma distribuída e remotamente.

Os testes realizados no ponto anterior usaram o reconhecimento de fala remoto. Isto é, a execução da Audimus API é totalmente realizada no servidor de áudio. O que significa que só é necessário realizar o teste ao reconhecimento de fala distribuído. Para tal utilizou-se a frase três. Os resultados obtidos para o reconhecimento de fala distribuído encontram-se na tabela seguinte:

a)

Frase 3	
Registo	T total (ms)
1	37008
2	36193
3	36190
4	36115
5	33814
Média (ms)	35864
Desvio Padrão (ms)	1202,99

b)

T_total (ms)	T_audimus (ms)	T_distribuido (ms)
35864	1140	24641

Tabela 13 – Resultados dos tempos obtidos para a frase 3 com a Audimus API a executar-se de forma distribuída entre o PDA e o servidor de áudio.

Como era de esperar tendo em conta a grande diferença de desempenhos entre o PDA e o PC (ver Anexo V), o tempo total nesta configuração é 10 vezes superior em relação ao tempo de processamento remoto. O subloco Rasta do bloco AudioIn ocupa a maior fracção de processamento da Audimus API no PDA.

5.2 Sucesso do reconhecimento de fala

Como o objectivo desta aplicação é preencher campos do formulário através da fala, torna-se imprescindível analisar a qualidade do reconhecimento de fala. A estratégia utilizada foi a de criar três tarefas de dificuldade gradual. Cada tarefa compreende o preenchimento de um determinado número de campos com frases

previamente estabelecidas. Para cada tarefa é determinado o tempo de execução e o número de falhas de reconhecimento em cada campo. A melhor forma de especificar este último parâmetro é a através da contagem do número de repetições da frase falada que o utilizador realiza para cada campo. Os resultados obtidos encontram-se nas tabelas seguintes. Neste teste foi utilizada a configuração de reconhecimento remoto, porque é a que tem melhor desempenho.

Tarefa 1: Mudar a doença do doente de uma cama

	1ª vez	2ª vez	3ª vez	4ª vez	5ª vez	
Campo	Repetições	Repetições	Repetições	Repetições	Repetições	Média
Doença	0	0	0	2	1	1
Tempo total (s)	4	4	5	10	5	6

Tabela 14 – Tempos de execução da tarefa 1 e o número de repetições de texto falado para cada um dos campos em análise.

Tarefa 2: Alterar o nome para “João Carmo”, a idade para “32”, o sexo para “masculino” e por último a doença para “sarampo”.

	1ª vez	2ª vez	3ª vez	4ª vez	5ª vez	
Campo	Repetições	Repetições	Repetições	Repetições	Repetições	Média
Nome	2	3	0	0	0	1
Idade	2	4	3	4	0	3
Sexo	0	0	0	0	0	0
Doença	0	0	5	1	6	2
Tempo total (s)	148	140	121	181	71	132

Tabela 15 – Tempos de execução da tarefa 2 e o número de repetições de texto falado para cada um dos campos em análise.

Tarefa 3: Colocar um novo doente com os seguintes valores: nome é “Pedro Sousa”, o sexo é “masculino”, a idade é “65”, o nº de segurança social é “77777777”, a sala é

a “7”, o piso é o “4”, a doença é “sarampo”, o estado é “estável” e as observações são “o doente encontra-se em recuperação”.

	1ª vez	2ª vez	3ª vez	4ª vez	5ª vez	
Campo	Repetições	Repetições	Repetições	Repetições	Repetições	Média
Cama	2	3	1	2	0	2
Nome	0	0	1	0	2	1
Sexo	0	0	0	1	0	0
Idade	0	2	2	4	2	2
Segurança social	2	6	3	2	4	3
Sala	0	0	2	1	1	1
Piso	5	3	4	2	1	3
Doença	8	0	1	0	2	2
Estado	0	3	2	0	1	1
Obs.	1	3	2	1	2	2
Tempo total (s)	248	342	328	255	350	300

Tabela 16 – Tempos de execução da tarefa 3 e o número de repetições de texto falado para cada um dos campos em análise.

Repetições	%
0	31
1	17
2	23
-	29

Tabela 17 – Percentagem do número de repetições no preenchimento dos campos.

Analisando os resultados anteriores, verifica-se que em 31% dos casos o reconhecedor reconheceu correctamente o que o utilizador disse. Em 17% dos casos é necessário repetir uma segunda vez e em 23% dos casos uma terceira vez. A partir destes resultados é possível inferir que o reconhecimento de fala fica aquém das expectativas. Em muitos casos é necessário realizar várias repetições, o que se traduz num acréscimo de tempo dispendido no preenchimento desse campo. Em alguns casos é frequente que o texto reconhecido contenha palavras a mais em relação ao que era esperado, principalmente, quando o ambiente contém muito ruído. Todavia estes resultados são uma consequência da fraca qualidade do microfone do PDA e de se estar a utilizar um modelo de linguagem genérico.

5.3 Fala vs Entradas de dados tradicionais do PDA

Nesta secção pretende-se comparar as diferenças de desempenho dos métodos usuais de entrada do PDA com a fala. Para isso, foram realizadas as tarefas do ponto anterior usando os métodos de entrada de dados do PDA: o teclado virtual e o reconhecedor de letras. O objectivo é a anotar o tempo dispendido para a realização de cada tarefa para cada um dos dois métodos. Neste teste foi utilizada a configuração de reconhecimento remoto, porque é a que tem melhor desempenho.

Modo de entrada: Teclado virtual

Tarefa 1	1ª vez	2ª vez	3ª vez	4ª vez	5ª vez	Média (s)
T_total (s)	86	75	55	60	61	67

Tabela 18 – Tempos dispendidos na tarefa 1 usando o teclado virtual

Tarefa 2	1ª vez	2ª vez	3ª vez	4ª vez	5ª vez	Média (s)
T_total (s)	94	80	62	58	59	71

Tabela 19 – Tempos dispendidos na tarefa 2 usando o teclado virtual

Tarefa 3	1ª vez	2ª vez	3ª vez	4ª vez	5ª vez	Média (s)
T_total (s)	127	123	126	127	130	127

Tabela 20– Tempos dispendidos na tarefa 3 usando o teclado virtual

Modo de entrada: Reconhecedor de letras

Tarefa 1	1ª vez	2ª vez	3ª vez	4ª vez	5ª vez	Média (s)
T_total (s)	127	133	129	120	120	127

Tabela 21– Tempos dispendidos na tarefa 1 usando o modo de entrada reconhecedor de letras

Tarefa 2	1ª vez	2ª vez	3ª vez	4ª vez	5ª vez	Média (s)
T_total (s)	150	135	170	148	120	145

Tabela 22 – Tempos dispendidos na tarefa 2 usando o modo de entrada reconhecedor de letras

Tarefa 3	1ª vez	2ª vez	3ª vez	4ª vez	5ª vez	Média (s)
T_total (s)	334	318	314	310	270	309

Tabela 23 – Tempos dispendidos na tarefa 3 usando o modo de entrada reconhecedor de letras

Tempos Médio (ms)	Fala	Teclado Virtual	Reconhecedor de Letras
Tarefa 1	6	67	127
Tarefa 2	132	71	145
Tarefa 3	300	127	309

Tabela 24 – Comparação dos tempos médios dos três métodos de entrada de dados em análise.

Através da análise da tabela anterior, conclui-se que a fala como métodos de entrada de dados apresenta tempos aproximados com o reconhecedor de letras. Em comparação com o teclado virtual revela-se um processo mais lento. No entanto, no caso da primeira tarefa, como a palavra utilizada é facilmente reconhecida, a fala revela-se um método mais rápido em relação aos restantes. Destaca-se ainda que a fala é um método mais fácil de utilizar do que o teclado virtual e o reconhecedor de letras.

6 Conclusão

A utilização da fala nos PDAs, ainda se encontra na fase inicial de investigação. Falta percorrer um longo caminho até que o reconhecimento de fala e a síntese de fala no PDA atinjam o mesmo nível de desenvolvimento e performance, em relação ao que existe actualmente para o PC.

O processamento de fala no PDA está limitado por uma série de dificuldades. Considerando apenas a família de PDAs para qual este trabalho foi realizado: dispositivos com o processador StrongARM e que utilizam o Pocket PC como sistema operativo. Observa-se que existem limitações no hardware, a nível do sistema de áudio e principalmente na capacidade de processamento. Fazendo com que o processamento de fala tenha de ser realizado com o auxílio de computadores externos com maiores capacidades de processamento. Em relação ao sistema operativo Pocket PC 2000 não existe nenhum suporte para o processamento de fala. Por exemplo a biblioteca de suporte ao processamento de fala da Microsoft, a MS Speech API, ainda não foi implementada neste sistema operativo, de forma a ser utilizada nos PDAs. Conclui-se então, que para criar uma aplicação que utilize a fala no PDA, como método de entrada de dados, é necessário implementar todo o processamento de fala a partir de raiz. O que se estende desde da aquisição da fala até ao preenchimento dos campos com resultado do reconhecimento.

No entanto, o uso da fala no preenchimento de formulários apresenta resultados positivos. A utilização da fala é uma mais valia, apesar de estar condicionada por algumas limitações como a utilização de um vocabulário reduzido, ou o uso de um modelo acústico especialmente treinado para o utilizador.

Comunicação

As estruturas de comunicações são muito importantes porque as transmissões de dados representam uma grande parte do tempo no preenchimento de formulários através da fala. Podem ser tomadas algumas medidas para otimizar a comunicação

como utilizar um protocolo mais adequado a este tipo de dados, mais rápido e menos complexo (ex. UDP). A redução do volume de dados transmitidos através da compressão de dados ou realizando um pré-processamento também é um factor a ter em conta.

A utilização das redes *wireless* por dispositivos móveis como o PDA, é uma solução que permite ultrapassar algumas das limitações destes dispositivos e que não restringe a mobilidade, dentro da área de acção.

Aplicação

A utilização da fala como meio de preenchimento de formulários revelou-se como sendo uma boa solução. Através dos resultados dos testes efectuados, observa-se que está ao nível dos restantes métodos de entrada de dados do PDA. A fala é uma forma de interacção natural, flexível, fácil de utilizar e que não precisa de nenhum tipo de aprendizagem por parte do utilizador. O preenchimento dos campos através da fala aumenta a usabilidade da aplicação e torna-a menos dependente dos outros métodos de entrada de dados

A arquitectura de uma aplicação que utilize a fala necessita de funções e estruturas de suporte ao processamento de fala. O suporte ao processamento de fala deve ser independente da aplicação de forma a ser utilizado em diferentes domínios. No que diz respeito a este trabalho, o suporte ao processamento de fala não está dependente do domínio da aplicação de formulários médicos, pode ser utilizado em aplicações de outros domínios. Para serem utilizadas mais facilmente, as funções e estruturas de suporte ao processamento de fala devem ser agrupadas numa API.

A gestão distribuída da informação pretende servir dois propósitos: vencer as limitadas capacidades tecnológicas do PDA e tornar a informação disponível para vários utilizadores, em qualquer lugar. Esta solução tem ganho cada vez mais importância com a crescente necessidade de mobilidade. Devem ser equacionados alguns aspectos que não foram abordados neste trabalho, como permitir o acesso à informação a um grande número de utilizadores simultaneamente. Assim como, gerir a coerência e integridade da informação através da aplicação de mecanismos

transaccionais que implementem a atomicidade, consistência, isolamento e durabilidade da informação.

Reconhecimento de fala

Neste trabalho foi dada uma grande ênfase ao reconhecimento de fala, tarefa vital no preenchimento de formulários através da fala. A síntese de fala não teve uma importância crítica e foi tratada como algo mais acessório. As soluções utilizadas neste trabalho para o reconhecimento de fala seguem a mesma orientação que outras aplicação para esta área. A utilização de um servidor externo para realizar o reconhecimento de fala, tendo uma rede *wireless* como meio de comunicação, é uma solução que também é utilizada pela Microsoft na sua aplicação MiPad⁴.

O reconhecimento de remoto fala, executado num servidor externo, é uma solução aceitável para as situações em que existe uma rede *wireless* disponível para o PDA. Neste caso é possível adaptar sistemas de reconhecimento de fala já existentes, utilizando os motores de reconhecimento de fala sobre a forma de servidores. Estes recebem os dados do PDA, processam-os através dos seus motores de reconhecimento e enviam os resultados para o PDA. Porém esta solução está limitada aos locais abrangidos pelas redes *wireless*.

Inicialmente pretendia-se transferir a totalidade do reconhecedor de fala para o PDA. Porém, surgiram dificuldades na migração da biblioteca de reconhecimento de fala (Audimus API) para o Pocket PC. Adicionalmente, o PDA revelou um desempenho decepcionante no processamento de fala, já que o tempo de execução do reconhecimento local é dez vezes superior ao tempo de reconhecimento remoto. Assim, conclui-se que não se pode simplesmente transferir um motor de reconhecimento de fala do PC para o PDA, é necessário modifica-lo, adaptando-o às características tecnológicas do dispositivo.

Em relação a este trabalho, a configuração do processamento de fala distribuído reparte o processamento entre o PC e o PDA. Esta configuração só será eficiente se o processamento na parte do PDA for reduzido. Uma possível solução

⁴ <http://research.microsoft.com/srg/mipad.aspx>

consiste em limitar o processamento no lado do PDA a tarefas pouco exigentes em cálculos numéricos, como a determinação do fim de uma frase (End Point).

O sucesso da utilização de fala, nesta aplicação, está dependente da adequação do reconhecedor de fala à aplicação e aos seus utilizadores, assim como ao ambiente onde esta é utilizada. O End Point constitui um elemento muito importante no reconhecimento de fala, porque a sua configuração tem uma grande influência no tempo de processamento. Tendo em conta que, neste trabalho, o End Point possui uma configuração fixa, o ideal seria ter uma configuração que se adequasse automaticamente às condições de ruído do meio ambiente. A utilização de um modelo de linguagem genérico foi influenciada pelas características da Audimus API não permitir outra opção. Verificou-se que os campos dos formulários, na sua maioria, admitem um conjunto limitado de valores. Uma forma de diminuir o tempo do reconhecimento de fala e diminuir os erros de reconhecimento, consiste em adoptar um modelo de linguagem para cada campo. Ao diminuir o modelo de linguagem, diminuem-se os resultados possíveis para o reconhecimento. O que se traduz automaticamente num menor tempo de processamento e numa maior probabilidade de sucesso. Em relação ao modelo acústico, este é fortemente influenciado pela qualidade do microfone. Neste caso o microfone do PDA conduziu a um modelo acústico pouco robusto. Uma possível solução passa pela utilização de um outro microfone, como por exemplo um microfone *Bluetooth*.

Relativamente ao sucesso do reconhecimento, com base nos resultados do ponto 5.2, conclui-se que o processo de reconhecimento de fala é razoável. As razões que conduziram a este resultado prendem-se com a geração do modelo acústico e com as características do modelo de linguagem. O modelo acústico está muito relacionado com a qualidade do microfone e com as condições de ruído ambiente. Por outro lado, um modelo de linguagem genérico aliado a um modelo acústico pouco robusto torna mais falível o sucesso do reconhecimento.

Conclui-se finalmente que existem um longo caminho a percorrer até que a fala possa ser utilizada num PDA, tão facilmente como é utilizada num PC. Para que esse objectivo seja concretizado, existem muitas dificuldades que têm de ser ultrapassadas. As características peculiares dos dispositivos móveis introduzem novos paradigmas, que levam a repensar as soluções actuais. Deve-se considerar este

trabalho como a etapa inicial de uma longa caminhada, onde se avaliam as dificuldades que têm de ser vencidas e onde se traça o caminho a seguir.

7 Bibliografia

[Neto, 1998] J. Neto (1998). *Reconhecimento da Fala Contínua com aplicação de técnicas de Adaptação ao Orador*. Tese de Doutoramento, Instituto Superior Técnico da Universidade Técnica de Lisboa, Lisboa (Portugal).

[Cassaca e Maia, 2002] R. Cassaca e R. Maia (2002). *Assistente Electrónica*. Relatório do Trabalho Final de Curso, Instituto Superior Técnico da Universidade Técnica de Lisboa, Lisboa (Portugal).

[Shilmover, 2002] B. Shilmover (2002). *War Walking: Detecting wireless networks*. Pocket Pc Magazine vol. 5, pp 40-41.

[Mitchell, 2002] D. Mitchell (2002). *Developer's Corner: The Challenges of Pocket PC Development*, Pocket Pc Magazine vol. 5, pp 54-56.

[Rój, 2003] M. Rój (2003). *Implementing H.323 Terminal: Real-Time Protocol-based Audio Engine*.

home.elka.pw.edu.pl/~mroj/homepage/works/mroj/html/audio/audio-book.htm

[WaveIn, 2002] Programa WaveIn (2002)

<http://www.borg.com/~jglatt/tech/lowaud.htm>

Anexos

Anexo I

Manual do utilizador

O texto seguinte tem como objectivo descrever o modo como se utiliza a aplicação desenvolvida no presente trabalho.

Começar o programa

Para começar o programa execute as tarefas seguintes:

- Ligue o Pocket PC, carregue em Start e escolha a aplicação SPDA2. Dentro de momentos aparecerá o menu de entrada da aplicação SpeechPDA (ver figura 13).
- No campo *Servidor* escreva o endereço IP da máquina onde se executa o servidor de áudio e o servidor de conteúdos e prima *Ligar*. Dentro de momentos aparecerá um menu onde poderá experimentar todas as funcionalidades da aplicação. A utilização deste menu é explicada no ponto seguinte.



Figura 23– Interface de entrada da aplicação Speech PDA (SPDA)

Falar para os campos

Para utilizar a fala no preenchimento dos campos, terá que realizar os seguintes passos:

- Seleccionar o campo que deseja preencher;
- Carregar na direcção inferior (para baixo) do botão de navegação do PDA
- Iniciar o discurso. Quando terminar de falar, largue o botão e aguarde que o texto correspondente ao seu discurso apareça no campo respectivo.

Ver ocupação da cama

Selecione o campo *Nº de cama* e diga o número de cama que pretende ver a ocupação. Quando o número que disse aparecer no campo respectivo prima o botão *Ver Ocupação* do menu (figura 14).

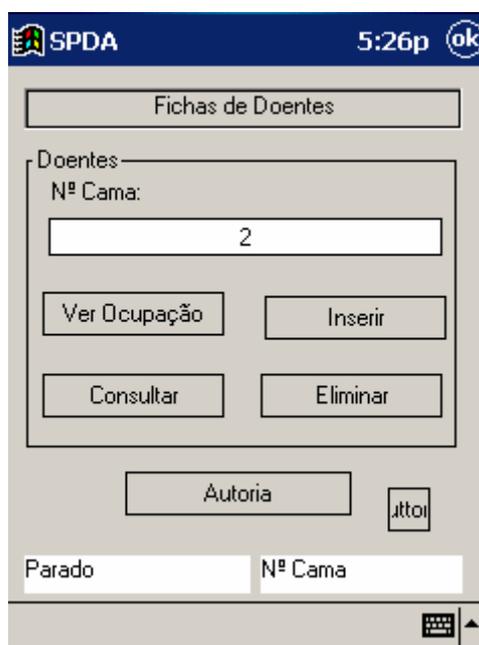
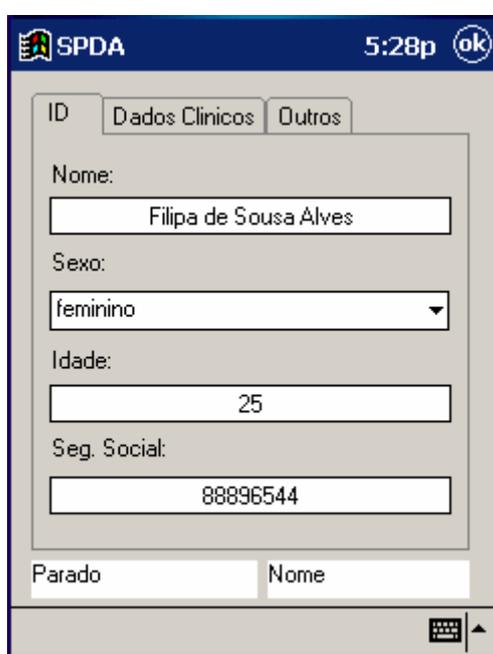


Figura 24 – Interface de operações da aplicação SPDA

Ver e alterar uma ficha de um doente

Selecione o campo *Nº de cama* e diga o número de cama do doente que pretende consultar. Quando o número que disse aparecer no campo respectivo prima o botão *Consultar* do menu (figura 14). Seguidamente ser-lhe-á apresentado um menu com três separadores onde poderá encontrar todos os campos da ficha de um doente. Cada um destes campos contém discriminada informação sobre o doente. A figura 15 apresenta um exemplo de uma ficha de um doente que ocupa a cama 2. Para alterar algum campo, basta seleccionar o campo, apagar a informação que não quer e dizer a nova informação. Para sair do formulário terá que carregar em *ok* no canto superior direito do ecrã. Se realizou alguma alteração será pedida a confirmação neste momento.



The image shows a screenshot of a PDA application interface. At the top, there is a status bar with the text 'SPDA' on the left, '5:28p' in the center, and an 'ok' button on the right. Below the status bar, there are three tabs: 'ID', 'Dados Clinicos', and 'Outros'. The 'ID' tab is selected. The form contains several input fields: 'Nome' with the value 'Filipa de Sousa Alves', 'Sexo' with a dropdown menu showing 'feminino', 'Idade' with the value '25', and 'Seg. Social' with the value '88896544'. At the bottom of the form, there are two more input fields labeled 'Parado' and 'Nome', and a small keyboard icon with an arrow pointing up.

Figura 25 – Exemplo de formulário de um doente exibido pela aplicação SPDA

Ver documentos de um doente

Consulte o último separador do formulário. Neste separador encontra-se um campo *Documentos* que contém a lista de todos os documentos associados ao doente. Selecione da lista o documento que pretende consultar e prima o botão *Ver Documento*. No caso de querer ouvir um documento que está na forma escrita terá que depois de seleccionar o documento, seleccionar também a opção *Traduzir os relatórios para fala* e de seguida premir o botão *Ver Documento*. Dentro de momentos ouvirá o relato do relatório que escolheu.

Criar uma ficha de um doente

Seleccione o campo *Nº de cama* e diga o número de cama onde pretende inserir o novo doente. Quando o número que disse aparecer no campo respectivo prima o botão *Inserir* do menu (figura 14). Apresentar-se-á uma ficha com os campos vazios de informação. Para preencher um campo, basta seleccionar o campo e dizer a nova informação. Quando todos os campos tiverem sido preenchidos carregue em *ok* no canto superior direito do ecrã e o novo doente é criado. Caso contrário ser-lhe-á mostrado a causa do insucesso da operação.

Apagar um doente

Seleccione o campo *Nº de cama* e diga o número de cama que pretende apagar. Quando o número que disse aparecer no campo respectivo prima o botão *Apagar* do menu (figura 14). Seguidamente será notificado do sucesso ou insucesso da operação.

Anexo II

Protocolo de Comunicação

Mensagens trocadas entre o servidor de conteúdos e o servidor de dados

Entidades	Tipo de mensagem	Forma da mensagem
Servidor de conteúdos para Servidor de dados	Pedido de um doente	pedido:cama:<nrº cama>
	Adicionar um doente	actualizacao:cama:<valor>:nome:<valor>: sexo:<valor>:idade:<valor>:segsocial:<valor>:sala:<valor>: piso:<valor>:doenca:<valor>:estado:<valor>:obs:<valor>: outros:<valor>
	Apagar um doente	remocao:cama:<nº da cama>
Servidor de dados para Servidor de conteúdos	Resposta ao pedido de um doente	pedido:cama:<valor>:nome:<valor>:sexo:<valor>:idade:<valor> :segsocial:<valor>:sala:<valor>:piso:<valor>:doenca:<valor> :estado:<valor>:obs:<valor>:outros:<valor>
	Resposta ao pedido de um doente numa situação de erro	pedido:<mensagem de erro>
	Resposta ao pedido para adicionar um doente	actualizacao:OK
	Resposta quando o adicionar do doente reporta um erro	actualizacao:<mensagem de erro>
	Resposta ao pedido de apagar um doente	remoção:OK
	Resposta quando o apagar do doente reporta um erro	remoção:<mensagem de erro>

Tabela 25 – Tabela com a especificação das mensagens trocadas entre o servidor de conteúdos e o servidor de dados

Mensagens trocadas entre o servidor de conteúdos e o PDA

Entidades	Tipo de mensagem	Forma da mensagem
PDA para Servidor de Conteúdos	Pedido de um doente	READ:<tamanho mensagem>:<nrº cama>
	Adicionar um doente	WRITE:<tamanho mensagem>:cama:<valor>:nome:<valor>:sexo:<valor>:idade:<valor>:segsocial:<valor>:sala:<valor>:pisso:<valor>:doenca:<valor>:estado:<valor>:obs:<valor>:outros:<valor>
	Pedido de documento	FILE:<tamanho mensagem>:<nrº cama>:<path>
	Apagar um doente	KILL: <tamanho mensagem>:<nº da cama>
	Pedido de documento escrito sintetizado.	SINT: <tamanho mensagem>:<nº da cama>:<path>
Servidor de Conteúdos para PDA	Resposta ao pedido de um doente	READ:<tamanho mensagem>:<estado>:cama:<valor>:nome:<valor>:sexo:<valor>:idade:<valor>:segsocial:<valor>:sala:<valor>:pisso:<valor>:doenca:<valor>:estado:<valor>:obs:<valor>:outros:<valor>
	Resposta ao pedido de um doente numa situação de erro	READ: <tamanhomensagem>:KO:<mensagem de erro>
	Resposta ao pedido para adicionar um doente	WRITE:<tamanhomensagem>:OK
	Resposta quando o adicionar do doente reporta um erro	WRITE:<tamanhomensagem>:KO:<mensagem de erro>
	Resposta ao pedido de apagar um doente	KILL: < tamanhomensagem >:OK
	Resposta quando o apagar do doente reporta um erro	KILL: < tamanhomensagem>: KO:<mensagem de erro>
	Resposta ao pedido de documento	FILE:<tamanhomensagem>:<tipo do ficheiro>:<dados do ficheiro>
	Resposta quando o tratamento do documento reporta um erro	FILE:< tamanhomensagem>: KO:<mensagem de erro>
	Resposta ao pedido de um documento sintetizado	SINT:<tamanhomensagem>:<tipo do ficheiro>:<dados do ficheiro>
	Resposta quando o tratamento do pedido do documento sintetizado reporta um erro.	SINT:< tamanhomensagem>: KO:<mensagem de erro>

Tabela 26 – Tabela com a especificação das mensagens trocadas entre o PDA e o Servidor de Conteúdos.

Mensagens trocadas entre o Servidor de Conteúdos e o TTS

Entidades	Tipo de mensagem	Forma da mensagem
Servidor de Conteúdos para TTS	Pedido de síntese de um ficheiro de texto.	PedidoTts:<caminho do ficheiro de texto>
TTS para Servidor de Conteúdos	Resposta ao pedido de síntese de um ficheiro de texto	PedidoTts:<nº da cama>:<caminho do ficheiro de texto>
	Resposta no caso de ocorrer algum erro.	PedidoTts:<mensagem de erro>

Tabela 27 – Tabela com a especificação das mensagens trocadas entre o Servidor de Conteúdos e o TTS.

Anexo III

Implementação dos pipes no Pocket PC

Para implementar os pipes no Pocket PC foi necessário realizar as principais funções que manipulam os pipes definidas na biblioteca pipes.h. Destacam-se as operações de criar, abrir, escrever, ler e fechar um pipe. A ideia base é criar o pipe como um ficheiro e tratar esse ficheiro como uma região crítica, em que as leituras e as escritas têm que ser disjuntas. No entanto, como a Audimus utiliza vários pipes em simultâneo foi necessário armazenar as referências de todos os ficheiros abertos até ao momento. O esquema seguinte ilustra o funcionamento do algoritmo de simulação dos pipes:

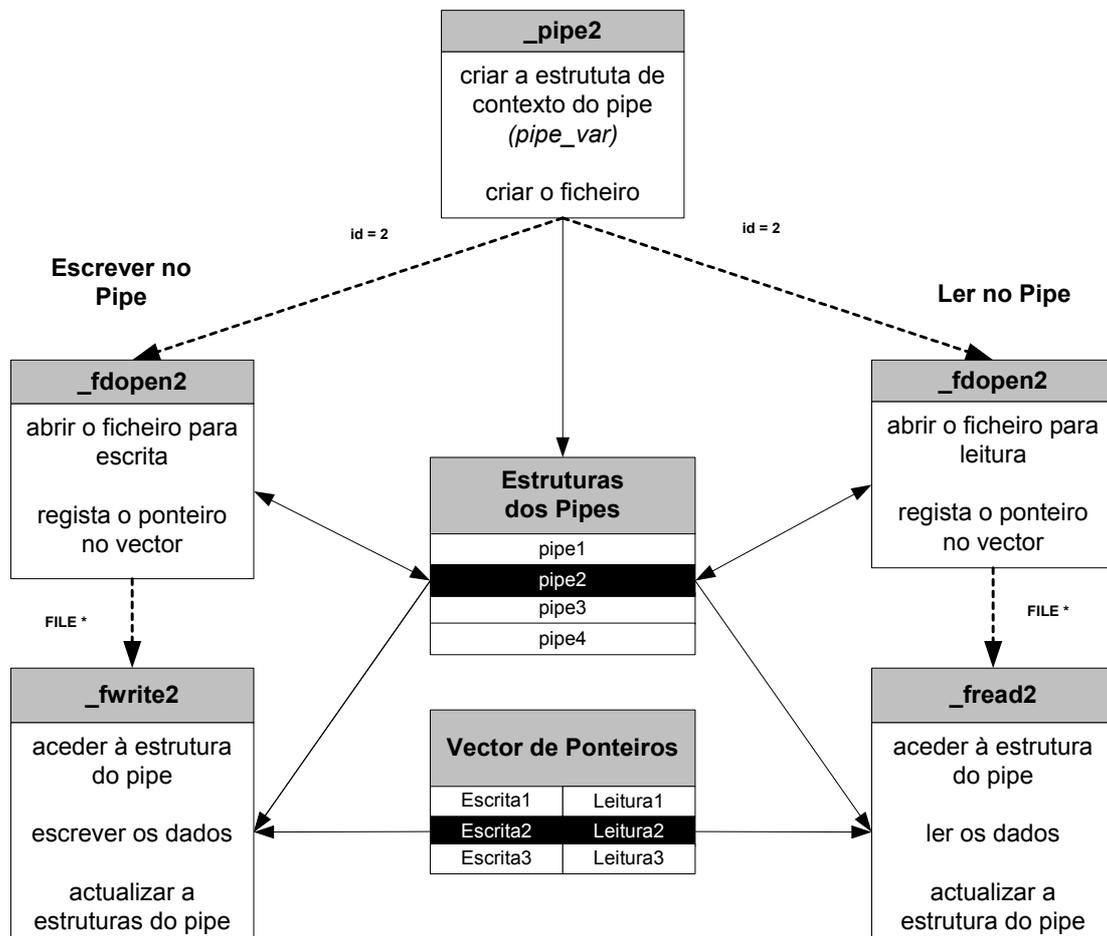


Figura 26 – Representação esquemática do algoritmo dos Pipes.

O primeiro objectivo na manipulação de pipes é conseguir criar um pipe. Para tal elaborou-se a função *_pipe2*. A função *_pipe2* cria o ficheiro que irá armazenar a

informação, coloca numa estrutura a informação deste pipe e armazena-a num *array* com todas as estruturas de pipes criados até então (*pipe_var*). A função *_fdopen2* abre o ficheiro criado em modo de leitura ou escrita e guarda a sua referência num local que contém as referências de todos os ficheiros criados (Vector de ponteiros).

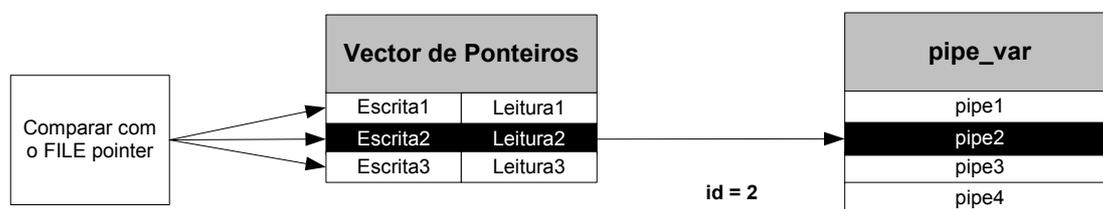


Figura 27 – Estruturas usadas na manipulação dos Pipes

O próximo passo é implementar a função de escrita no pipe. O algoritmo de escrita é apresentado esquematicamente na figura 6. A função recebe, para além dos dados a escrever, a informação do descritor de ficheiro para onde se irá colocar os dados. Com o descritor de ficheiro é fácil determinar qual o pipe que está a ser requerido recorrendo às estruturas atrás mencionadas. Para realizar a escrita no ficheiro é necessário entrar numa zona de exclusão mútua para que as escritas e leituras se façam de modo coerente.

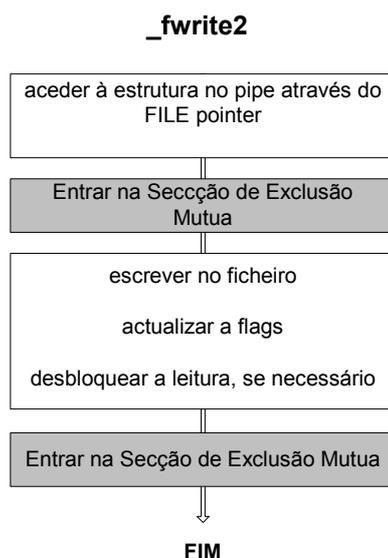


Figura 28 – Funcionamento da função *Fwrite2*

Por último resta a função de leitura no pipe. O algoritmo de leitura é apresentado esquematicamente na figura 7. A função de leitura recebe a informação do descritor de ficheiro de onde se irá ler os dados. Para realizar a leitura do ficheiro é necessário entrar numa zona de exclusão mútua e bloquear os acessos de escrita.

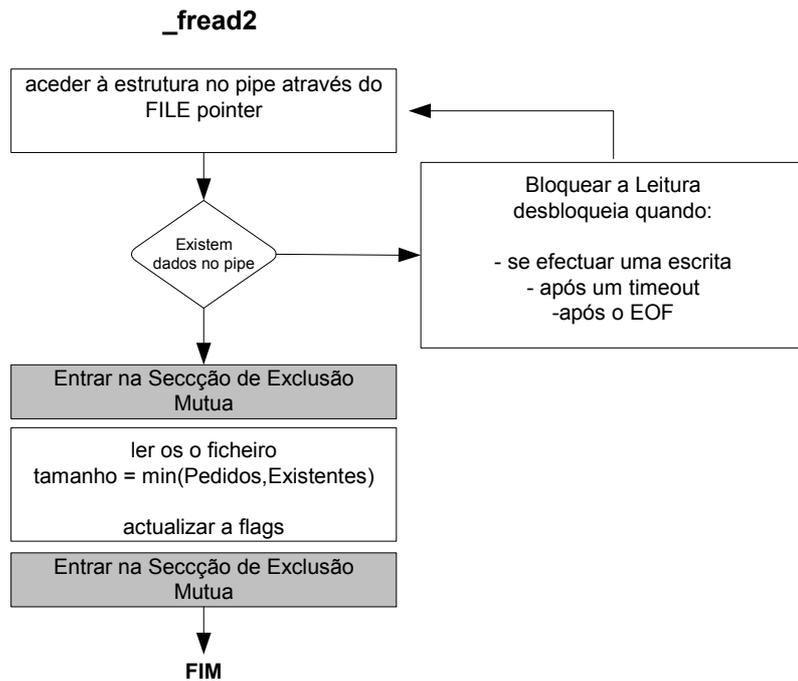


Figura 29 – Funcionamento da função *Fread2*

Anexo IV

Especificação do PDA

A aplicação do PDA pode ser executada em qualquer PDA que possua um processador StrongARM e utilize como sistema operativo o Pocket PC. O PDA utilizado na realização deste trabalho foi um Compaq iPAQ H3870. Este modelo do iPAQ ficou conhecido por ser o primeiro a trazer suporte Bluetooth incorporado.



Figura 30 – Fotografias do PDA utilizado neste trabalho

Dimensões	8.4 x 13.3 x 1.6 cm
Peso	190 g
Processador	Intel StrongARM SA-1110; 206 MHz
Memória RAM	64 MB
Ecrã	3.8 in; 240 x 320 pixels; 16 bits
Outros	Backlit display; Microfone; Altifalante; Entrada para Auscultadores; Porta de Infra-vermelhos; Slot SD Card; Bluetooth
Bateria	Lítio; 10 Horas de autonomia.

Placa de Rede e Microfone:

Foi adicionado um módulo de expansão, colocado por trás do iPAQ, para utilizar a placa de rede *wireless*, o que tornou o iPAQ mais pesado e menos cómodo. Futuramente com o desenvolvimento das placas de rede *wireless* para o *slot* SD, que está incorporado no PDA, não será necessário utilizar o módulo de expansão.



Figura 31 – PDA com o módulo de expansão

O iPAQ tem um microfone incorporado mas é muito susceptível ao ruído. Através da tecnologia Bluetooth, é possível utilizar um microfone com melhor qualidade. O que permite melhorar o reconhecimento de fala em ambiente com muito ruído de fundo.

Vista detalhada do PDA:



Figura 32 – Vista frontal do PDA

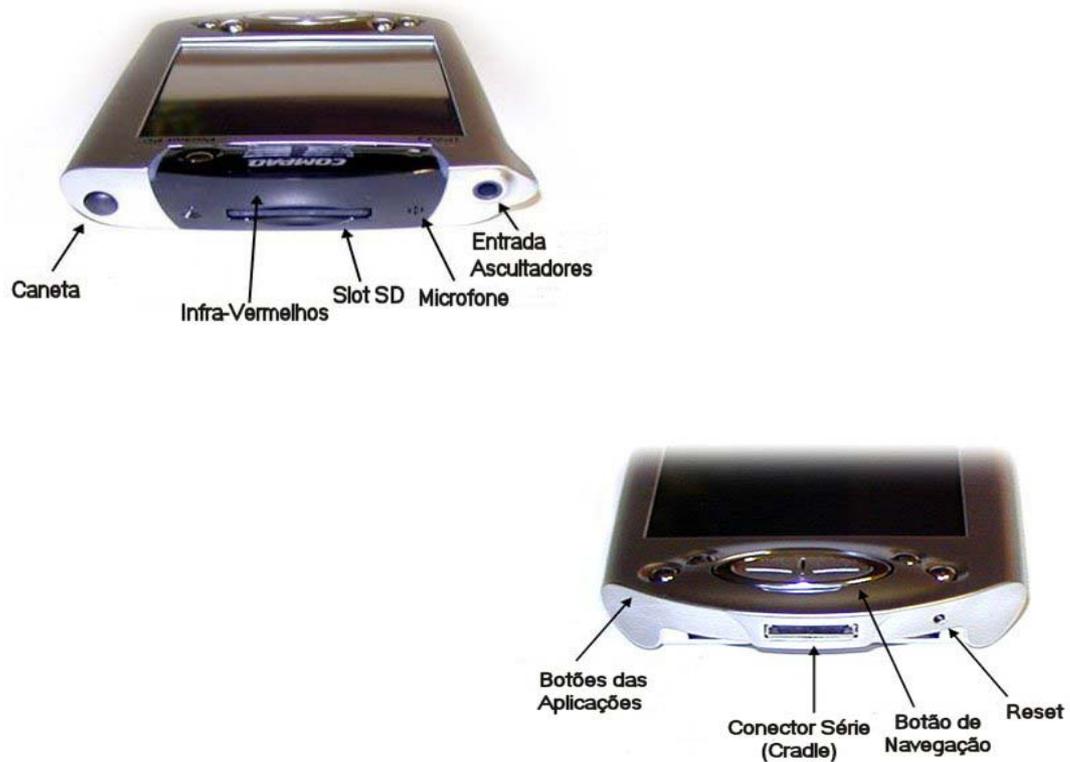


Figura 33 – Vista da parte cima e de baixo do PDA

Anexo V

Testes de Desempenho, PC vs PDA

De forma a determinar as diferenças de desempenho entre as arquitecturas PC e PDA em relação ao reconhecimento de fala, foram realizados testes de desempenho. O objectivo dos testes é avaliar a capacidade de processamento do sistema de ficheiros, ao nível da realização de cálculos matemáticos, escrita e leitura de ficheiros.

Estes testes foram baseado em operações muito comuns no reconhecimento de fala e que representam uma grande fatia do tempo de total de processamento.

Na tabela abaixo são apresentadas algumas especificações da arquitectura:

	PC	PDA
Processador	Pentium III	StrongArm SA1110
Velocidade de Processamento.	1100 MHz	206 MHz
Cache L1	16 Kb Instruções 16 Kb Dados	16 Kb Instruções 8 Kb Dados
Cache L2	256 Kb	-
Memória Primária	256 Mb	64 Mb
Tipo	SDRAM	SDRAM
Memória Secundaria	10 Gb	-

Foram realizados os seguintes testes:

- A.** Multiplicação de duas matrizes (16x16) de números inteiros, com o resultado a ser guardado numa matriz (16x16). Operação repetida um milhão de vezes.
- B.** Divisão de duas matrizes (16x16) de números com virgula flutuante, com o resultados a ser guardado numa matriz (16x16). Operação repetida um milhão de vezes.
- C.** Escrita de um Megabyte (1024^2 bytes) num ficheiro e respectiva leitura.
- D.** Escrita de 10 Megabyte (10×1024^2 bytes) num ficheiro e respectiva leitura.

Os resultados dos testes são apresentados na tabela abaixo:

<i>Tempo Médio (ms)</i>	A	B	C	D
PC	7591	12407	230	2174
PDA	31319	241777	12633	128319
Speedup PDA/PC	4	19	55	59

Através da análise dos resultados é possível chegar às seguintes conclusões:

- A performance do PC é muito superior à do PDA, especialmente a nível do armazenamento de ficheiros, o que é normal comparando as duas especificações.
- Em termos de cálculos numéricos, a má prestação do PDA tem como origem o processador. O StrongARM foi desenhado para sistemas embebidos e não para efectuar cálculos aritméticos complexos.
- A grande disparidade de resultados em relação à escrita e leitura de ficheiros pode ser justificada pela frequência interna do processador do PDA ser muito inferior à do PC, o que torna o acesso à memória RAM mais demorado.