

ESCOLA
SUPERIOR DE
TECNOLOGIA DA
SAÚDE DE
COIMBRA



Instituto Superior de
Engenharia de Coimbra

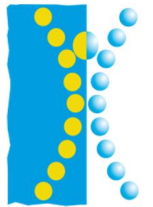
Instituto Politécnico de Coimbra
Instituto Superior de Engenharia de Coimbra
Escola Superior de Tecnologia da Saúde Coimbra

Sistema de Informação para a realização de testes de Audição Dicótica e Interação Binaural

João Ferreira

Mestrado em Sistemas e Tecnologias da Informação para a Saúde

Coimbra, Dezembro, 2012



ESCOLA
SUPERIOR DE
TECNOLOGIA DA
SAÚDE DE
COIMBRA



Instituto Superior de
Engenharia de Coimbr

Instituto Politécnico de Coimbra
Instituto Superior de Engenharia de Coimbra
Escola Superior de Tecnologia da Saúde Coimbra

Mestrado em Sistemas e Tecnologias da Informação para a Saúde

Projecto/Estágio I e Projecto/Estágio II

Sistema de Informação para a realização de testes de Audição Dicótica e Interacção Binaural

João Ferreira

Orientador:

António Carvalho
ESTESC

Co- Orientador:

Cláudia Reis
ESTESC

Coimbra, Dezembro, 2012

Agradecimentos

Aos orientadores António Carvalho e Cláudia Reis agradeço a orientação prestada.

A todos os meus colegas que me ajudaram a ultrapassar pequenos problemas durante o desenvolvimento com novas ideias para a implementação.

A toda comunidade *on-line* que através de fóruns, num espírito de partilha de informação, me ajudaram a esclarecer dúvidas e ultrapassar alguns problemas com que me fui deparando.

Em último, um obrigado à minha família pelo apoio e incentivo dados, para a realização deste projecto.

Resumo

Nas últimas duas décadas a evolução das Tecnologias de Informação (TI), mudou a forma como o mundo lida com a informação trazendo inúmeras vantagens competitivas, obrigou a uma grande reestruturação de sistemas e procedimentos existentes.

No caso específico da saúde as TI disseminaram-se rapidamente, e continuam em constante desenvolvimento, por forma a lidarem de forma eficiente com armazenamento, recuperação e gestão de informação clínica, para auxílio dos profissionais clínicos nas suas tomadas de decisão.

Embora as TI se encontrem disseminadas na saúde, ainda existem algumas áreas que não fazem uso dos potenciais da tecnologia e que continuam a utilizar Sistemas de Informação (SI) ultrapassados e pouco eficientes.

Esta dissertação enquadra-se no Grupo de Investigação em Bioacústica e Sistemas (GIBS), que pretende modernizar a área da Audiologia em Portugal que actualmente utiliza procedimentos ultrapassados e pouco eficientes.

Tem como objectivo propor um sistema de informação na área da Audiologia, que possa ser utilizado de forma consistente para a realização de testes e gestão de informação clínica.

Abstract

In the last two decades the evolution of Information Technologies (IT) has changed the way the world deals with information bringing numerous competitive advantages, forced a major restructuring of existing systems and procedures.

In the specific case of health IT have spread rapidly, and remain in constant development in order to deal efficiently with the storage, retrieval and management of clinical information in order to assist clinicians in their decisions.

Although IT are disseminated on health, there are still some areas that do not make use of the potential of technology and continue to use Information Systems (IS) outdated and inefficient. This work is part of the Grupo de Investigação em Bioacústica e Sistemas (GIBS), which aims to modernize the area of Audiology in Portugal which currently uses outdated and inefficient procedures.

It aims to propose an information system in the field of audiology to be used consistently to do testing and management of clinical data.

Índice

1.	Introdução	1
1.1.	Âmbito do projecto.....	1
1.2.	Objectivos e motivação.....	1
1.3.	Organização da dissertação.....	1
2.	Estado da Arte	3
2.1.	Sistemas de informação.....	3
2.2.	Sistemas de informação na Saúde.....	4
2.3.	Registos clínicos.....	6
2.4.	Processamento auditivo.....	7
2.4.1.	Testes de avaliação do processamento auditivo.....	7
2.4.2.	Testes dicóticos	8
2.4.2.1.	Teste dicótico de dígitos	8
2.4.2.2.	SSW ou testes de palavras espondeicas alternadas	9
2.5.	Aplicação dos testes nos dias de hoje.....	9
2.6.	Conclusões.....	10
3.	Metodologia de Desenvolvimento do Sistema de Informação.....	11
3.1.	Processo de desenvolvimento de SI	11
3.2.	Metodologias de desenvolvimento	11
3.3.	Prototipagem.....	15
4.	Análise e desenvolvimento	17
4.1.	Requisitos.....	17
4.2.	Tecnologia utilizada no desenvolvimento da solução	19
4.2.1.	Java	19
4.2.2.	IDE's de desenvolvimento.....	20
4.2.3.	Postgres SQL.....	20
4.2.4.	Cartão do Cidadão	20
4.2.5.	Dispositivo com Android	20
4.3.	Interface Gráfica	21
4.3.1.	Definição da interface gráfica.....	22
4.4.	Descrição conceptual da solução implementada	25
4.4.1.	Sistema.....	26
4.4.2.	Perfil de Administrador	28

4.4.2.1.	Gerir Protocolos Dicótico de Dígitos.....	31
4.4.2.2.	Gerir Protocolos SSW.....	33
4.4.2.3.	Gerir Sons Dicóticos de Dígitos / SSW.....	34
4.4.3.	Perfil de Técnico	35
4.4.3.1.	Testes	36
4.4.4.	Perfil de Utilizador	38
4.5.	Estrutura de dados.....	40
4.6.	Aspectos relevantes da implementação	42
4.6.1.	Cartão do cidadão Português	42
4.6.2.	Realização de testes verbais.....	44
4.6.3.	Realização de testes não-verbais.....	46
4.6.4.	Reprodução de sons.....	47
4.6.5.	Adição de sons ao sistema	48
4.6.6.	Gestão de sons no sistema.....	49
4.6.7.	Adição de protocolos ao sistema.....	50
4.6.8.	Formulário de realização de um teste	52
4.6.9.	Criação de relatórios.....	52
4.6.10.	Gestão de relatórios	54
4.6.11.	Gestão do interface gráfico	54
5.	Avaliações do Sistema de Informação	55
5.1.	Compatibilidade da aplicação	55
5.2.	Requisitos de Sistema	58
6.	Conclusões e Trabalhos Futuros.....	60
7.	Bibliografia	63

Índice de figuras

Figura 1 - Representação esquemática da definição de O'Brien.....	4
Figura 2 – Representação gráfica do desenvolvimento em cascata.....	12
Figura 3 - Representação gráfica do desenvolvimento iterativo e incremental.....	14
Figura 4 - Representação esquemática das metodologias ágeis.....	15
Figura 5 – Representação gráfica da metodologia de prototipagem.....	16
Figura 6 - Protótipo inicial do esquema de navegação.....	16
Figura 7 - Metro UI do novo Windows 8.....	22
Figura 8 – Definição da interface gráfica.....	23
Figura 9 - Painel de tarefa.....	24
Figura 10 - Botões da aplicação.....	24
Figura 11 - Janela de aviso.....	25
Figura 12 – Diagrama de casos de uso do SI.....	27
Figura 13 - Ecrãs de login da aplicação.....	27
Figura 14 - Menu de escolha de perfil.....	28
Figura 15 - Diagrama de casos de uso do perfil de Administrador.....	29
Figura 16 - Registo de um novo utilizador.....	30
Figura 17 - Diagrama de casos de uso da gestão de protocolos dicóticos de dígitos.....	31
Figura 18 - Construção de um protocolo de dígitos.....	32
Figura 19 - Diagrama de casos de uso da gestão de protocolos SSW.....	33
Figura 20 - Diagrama de casos de uso da gestão de sons.....	34
Figura 21 - Diagrama de casos de uso do perfil de técnico.....	35
Figura 22 -Diagrama de casos de uso dos testes.....	36
Figura 23 - Formulário de início de teste.....	37
Figura 24 - Diagrama de casos de uso do perfil de Utilizador.....	38
Figura 25 - Formulário de registo de um novo Paciente.....	39
Figura 26 - Modelo físico da base de dados.....	41
Figura 27 - Leitor do cartão do cidadão.....	42
Figura 28 - Janela de pin da aplicação do cartão do cidadão.....	43
Figura 29 - Leitor do cartão do cidadão com <i>pin pad</i>	43
Figura 30 - Realização de testes verbais.....	45
Figura 31 - Protocolo de comunicação.....	46

Figura 32 - Aplicação para dispositivo Android	47
Figura 33 - Inserção de sons no sistema	49
Figura 34 - Adição de novo protocolo de dígitos	51
Figura 35 - Formulário de realização de um teste	52
Figura 36 - Relatório de resultados de teste dicótico de dígitos	53
Figura 37 - Aplicação em funcionamento em Windows XP, Vista, 7 e 8.....	55
Figura 38 - Teste em Mac OS 10.5 Leopard com erro.....	56
Figura 39 - Teste em Mac OS 10.8 Mountain Lion com falha.....	57
Figura 40 - Teste em Linux 11.10 com falha	57
Figura 41 - Compatibilidade por Sistema Operativo.....	58

Índice de tabelas

Tabela 1 - Requisitos do SI.....	19
Tabela 2 - Leitura de dados do cartão do cidadão.....	43
Tabela 3 - Requisitos mínimos de sistema Java.....	58
Tabela 4 - Requisitos mínimos de sistema Postgres.....	58
Tabela 5 - Requisitos mínimos de sistema CC.....	59
Tabela 6 -Requisitos mínimos de sistema SI com base de dados local.....	59
Tabela 7 - Requisitos mínimos de sistema SI com base de dados externo.....	59

Lista de anexos

Anexo A –Descrição das tabelas da base de dados	1
---	---

Lista de acrónimos

GIBS – Grupo de Investigação em Bioacústica e Sistemas

CC – Cartão do Cidadão

SI – Sistema de Informação

SIBC - Sistemas de Informação Baseados em Computadores

TI – Tecnologias de Informação

PA – Processamento Auditivo

PPA – Perturbação do Processamento Auditivo

SSW – *Staggered Spondaic Word*

PDSI – Processo de Desenvolvimento de um Sistema de Informação

ES – Engenharia de Software

SAM – Sistema de apoio ao médico

SAPE – Sistema de apoio à prática de Enfermagem

1. Introdução

1.1. Âmbito do projecto

A presente dissertação enquadra-se no Grupo de Investigação em Bioacústica e Sistemas (GIBS) e descreve o trabalho realizado durante as cadeiras de Projecto 1 e Projecto 2 do Mestrado de Sistemas e Tecnologias de Informação para a Saúde.

O objectivo do projecto é desenvolver um Sistema de Informação para a realização de testes de Audição Dicótica e Binaural.

1.2. Objectivos e motivação

O objectivo principal do projecto passa por implementar um sistema de informação que permita que sejam realizados testes dicóticos e de interacção binaural recorrendo às tecnologias de informação.

Este é um projecto inovador no sentido que não existe nenhum SI no mercado Português com as características específicas que permitam a realização dos testes.

Actualmente todos os testes são realizados recorrendo a leitores de CD e folhas de papel, embora seja um sistema funcional, recorre a procedimentos antiquados e que facilmente poderão introduzir erros nos resultados.

O SI foi desenvolvido a nível académico, mas sempre com a motivação de poder vir a ser utilizado de forma consistente num ambiente real.

1.3. Organização da dissertação

Esta dissertação encontra-se dividida em seis capítulos.

O segundo capítulo é composto por um estado da arte em SI, sendo descrita a sua evolução no tempo e os seus principais recursos. Ainda neste capítulo é feita uma descrição dos SI na saúde, do processamento auditivo e dos tipos de testes implementados.

No terceiro capítulo é feita uma introdução ao processo de desenvolvimento de SI, e um enquadramento das metodologias de desenvolvimento de Software que ajuda a enquadrar a metodologia escolhida.

No quarto capítulo é efectuada a análise de requisitos do SI, apresentada a tecnologia utilizada e são descritos os pontos mais relevantes da implementação, a estrutura de dados e o SI implementado na forma conceptual.

No quinto capítulo é efectuada uma avaliação do SI desenvolvido, quanto às suas características e aos seus requisitos.

No sexto capítulo são apresentadas as conclusões e as melhorias futuras a adicionar ao SI.

2. Estado da Arte

2.1. Sistemas de informação

Sistema de informação é a expressão utilizada para denominar as actividades de aquisição, armazenamento, processamento, manipulação e análise de informação. (Filomena Castro Lopes, 2009)

Informação por si só contém vários significados, o termo associado aos SI, pode ser definido como um conjunto de dados organizados de forma compreensível e que permita uma fácil interpretação.

Por outro lado dados representam um conjunto de variáveis independentes que não tem utilidade até serem analisados convenientemente. Através da análise e caso os dados sejam relevantes, são então convertidos em informação. Os dados só são efectivamente úteis a partir do momento que é possível obter um tipo de informação. (West, 1996)

Os SI evoluíram naturalmente ao longo dos anos associados à evolução da tecnologia, antes da era dos computadores os SI eram baseados em grandes arquivos onde eram armazenadas grandes quantidades de informação registada em papel.

A funcionalidade dos primeiros SI era reduzida, serviam apenas para armazenar e consultar informação, sendo a consulta uma tarefa que podia apresentar alguma dificuldade caso o arquivo não estivesse convenientemente organizado. (The Computer Society (IEEE-CS), 2005)

Com o evoluir das tecnologias de informação (TI) e o aparecimento dos computadores iniciou-se o abandono dos SI em formato de papel e surgiram os sistemas de informação baseados em computadores (SIBC)¹. (Filomena Castro Lopes, 2009)

Os SI agregados às TI tornaram-se mais funcionais e a aquisição, armazenamento, recuperação, manipulação e transmissão de informação passou a ser feita de forma mais simples e mais rápida com inúmeras vantagens.

A optimização do fluxo da informação permite uma clara redução de custos operacionais e administrativos, aumentando-se a produtividade e a integridade da informação.

As tomadas de decisão e análises ou cruzamento de dados tornam-se mais fáceis e podem até ser efectuadas directamente pelo SI. (Varajão, 2010)

O autor O'Brien definiu os recursos que se relacionam num SI (Figura 1):

¹ Apesar de existir uma diferença clara entre SI e SIBC, ao longo da dissertação o SIBC desenvolvido será apenas referenciado como SI, por uma questão de simplificação e por hoje em dia ser facilmente feita a associação de um SI à utilização de TI.

- **Pessoas** – As pessoas que interagem com um SI, podem ser especialistas tais como gestores de bases de dados, administradores de rede ou simples utilizadores responsáveis pela introdução e manipulação de dados no SI.
- **Hardware** – Todos os componentes físicos necessários para o correcto funcionamento do SI, podem ser computadores, periféricos de entrada e periféricos de saída, aparelhos de ar condicionados para correcta refrigeração dos servidores, entre outros.
- **Software** – Todo o software utilizado no SI.
- **Base de dados** – Local de armazenamento dos dados, geralmente de forma relacional.
- **Rede** – Sistema que permite que a informação seja partilhada entre vários equipamentos do SI. (O'Brien, 2003)

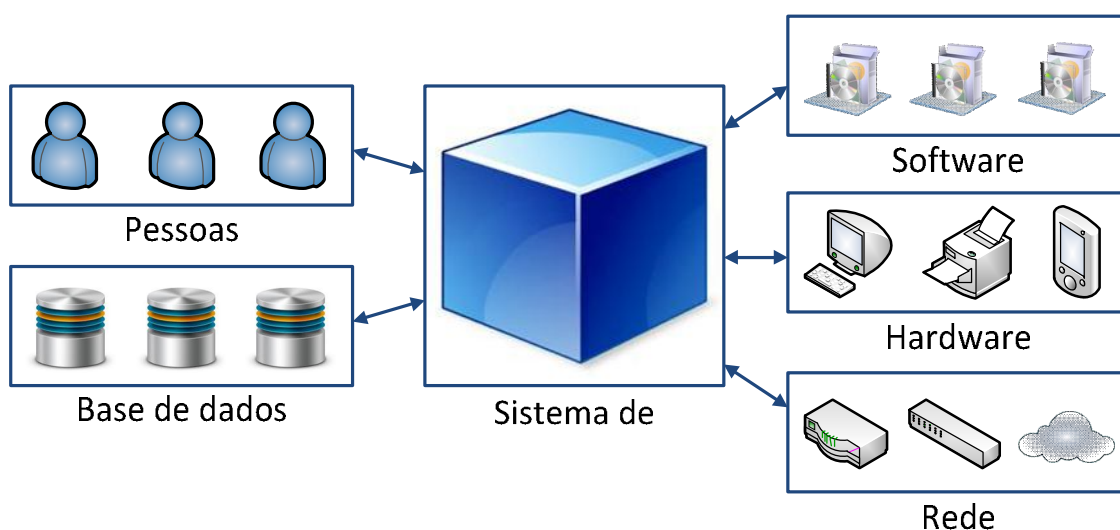


Figura 1 - Representação esquemática da definição de O'Brien

2.2. Sistemas de informação na Saúde

Os sistemas de informação na Saúde são um instrumento de importância crítica para o apoio à decisão clínica e para uma correcta e eficiente gestão de dados clínicos.

A introdução dos SI nos hospitais portugueses ocorreu em 1994, por via de software que visava a contabilização da produtividade.

Ao longo dos últimos 18 anos muitas soluções foram implementadas por forma a agilizar e melhorar os serviços médicos, contudo a grande dimensão da área da Saúde obrigou a que o SI fosse dividido em subsistemas.

Esta subdivisão nem sempre teve os resultados esperados e software como o SONHO e o SINUS revelaram-se desajustados perante a dimensão do Sistema Nacional de Saúde Português. A inexistência de um *datacenter* agregador de informação, associada à dificuldade de normalização de dados levou a que ocorram falhas em sistemas de importância vital.

Detectados os pontos fracos e tendo em mente que os SI poderiam ser aproveitados de forma mais eficiente para registo de dados clínicos nasceram o Sistema de apoio ao médico (SAM) e o Sistema de apoio à prática de Enfermagem (SAPE).

A aplicação SAM tem por objectivo informatizar o registo e consulta das actividades diárias das equipas de médicos. Através do SAM a equipa médica pode, nomeadamente:

- Efectuar prescrições de medicamentos;
- Requisitar exames complementares de diagnóstico e terapêutica;
- Prescrever baixas médicas;
- Registar/consultar informação clínica recolhida nas consultas, quer seja de carácter geral ou especificamente de um dos programas de saúde definidos pela Direcção Geral de Saúde;
- Consultar o histórico clínico do utente, incluindo as prescrições, consultas e baixas que lhe estejam associadas. (Paulo Gomes, 2009)

Além da informação clínica o SAM dispõe ainda de informação administrativa, nomeadamente no que respeita à gestão de consultas.

O SAPE visa o tratamento e organização da informação processada nos actos de enfermagem. Nesta aplicação o profissional de enfermagem pode:

- Consultar o plano de trabalho para a intervenção prevista num determinado contacto incluída no programa das equipas de enfermagem;
- Registar/consultar os sintomas apresentados pelo utente;
- Registar/consultar as intervenções de enfermagem com base no diagnóstico efectuado;
- Consultar/registar o plano de trabalho elaborado pelo sistema com base na informação clínica nele inserido;
- Consultar as tabelas de parametrização e codificação da actividade de enfermagem. (Paulo Gomes, 2009)

Os SI na saúde evoluíram para um funcionamento em rede, tornando-se mais flexíveis às mudanças, permitindo controlar informação essencial que pode evitar a duplicação de exames.

O desenvolvimento de SI associados à Saúde é um processo que evolui a par com a tecnologia, actualmente os sistemas deixaram de estar centralizados e disponíveis apenas em instituições de Saúde, através da internet o utente pode ter acesso a alguma da sua informação clínica.

Um exemplo desta evolução é a e-Agenda que permite a marcação de consultas sem necessidade de o utente se deslocar a um centro clínico. (Ministério da Saúde, 2010)

Com um percurso atribulado e sem estar isento de problemas os SI foram adaptados à saúde em Portugal, permitindo e contribuindo para um maior envolvimento e autonomia dos utentes face aos profissionais de saúde. (Sistemas de Informação em Saúde e Saúde Online, 2010)

Já existe muito trabalho realizado nesta área de reconhecido valor, no entanto ainda existem áreas do Sistema de Saúde que ainda não beneficiaram dos SI.

2.3. Registos clínicos

O registo clínico de um paciente é na sua forma tradicional um conjunto de registos que são arquivados ao longo do tempo e permitem manter um histórico dos processos de acompanhamento e tratamento dos pacientes.

Um registo clínico contém um conjunto de dados relacionados com o paciente que estão divididos em dados clínicos (resultantes de observações, tratamentos e diagnósticos) e dados administrativos (relacionados com o processo burocrático do paciente).

Nas instituições de saúde a partilha de registos do paciente é feita em torno dos dados clínicos, assegurando um processo de comunicação eficaz entre diversos prestadores de cuidados.

Segundo (Teixeira, 2008) podemos encontrar quatro tipologias diferentes de organização da informação clínica:

- **Registo clínico de paciente orientado ao tempo** – *“(time-oriented patient record) – tipologia de organização dos dados de forma cronológica, identificando cada episódio pela respectiva data. Os dados resultantes do estudo do paciente, radiologia ou outro tipo de exame, são igualmente registados pelos clínicos de acordo com um eixo temporal.”*
- **Registo clínico de paciente orientado à fonte** – *“(source-oriented patient record) – das primeiras tipologias de organização de dados clínicos a aparecer, estando estes organizados de acordo com a sua fonte de origem. Portanto a proveniência dos dados determina a catalogação dos dados e conseqüente registo. A ordem é ditada pela origem da informação (histórica clínica; exames físicos, resultados dos meios complementares de diagnóstico, etc.).”*
- **Registo clínico de paciente orientado ao protocolo** – *“(protocol-oriented record) – os dados são organizados de acordo com um protocolo. Por exemplo, quando o paciente está a ser submetido a um tratamento específico (diabetes, asma, etc.) são utilizados templates padronizados (ou protocolos) no registo dos dados. Esses templates têm um formato pré-estruturado e estipulam que dados específicos têm que ser obtidos e registados pelos clínicos e que plano de tratamento irá ser efectuado ao paciente.”*
- **Registo de paciente orientado ao problema** – *“(problem-oriented record) – nesta tipologia os dados são organizados por problema. A cada doente é atribuído um ou mais problemas, sendo que, para cada problema os dados clínicos são organizados de acordo com a estrutura SOAP:*
 - **Subjective observations** – *observações subjectivas (dados) da história clínica do paciente;*
 - **Objective observations** – *observações objectivas (dados) identificados pelo clínico e resultantes dos exames, observações e testes;*
 - **Assessment** – *conclusões em termos de diagnóstico;*
 - **Plan** - *plano médico, tratamento ou atitude para resolver o problema.”*
(Teixeira, 2008)

No SI desenvolvido o registo clínico do paciente é orientado ao protocolo, por ser um sistema que tem como objectivo a realização de dois tipos de testes audiológicos.

2.4. Processamento auditivo

Por definição processamento auditivo (PA) é um conjunto de capacidades que o indivíduo possui para interpretar o que ouve.

Através do PA é possível identificar a localização dos sons, efectuar discriminação auditiva, reconhecer padrões auditivos e aspectos temporais da audição.

O PA é realizado através de um conjunto de aptidões necessárias para interpretar o que o indivíduo ouve. (Warren, 2010)

- **Detecção do som** – *“Identificação da presença/ausência de som. É uma capacidade do sistema nervoso periférico.”*
- **Sensação sonora** – *“É uma experiência que está directamente relacionada com intensidades, frequências e qualidade do estímulo sonoro.”*
- **Localização** – *“É a capacidade de identificar o local de origem do som, sendo indispensável uma audição binaural.”*
- **Reconhecimento** – *“É um processo que exige uma experiência anterior, ou seja, é a identificação correcta de um estímulo por meio de conhecimento previamente adquirido.”*
- **Discriminação** – *“Consiste na capacidade de detectar diferenças entre padrões sonoros.”*
- **Atenção** – *“É a capacidade de identificar a mensagem primária na presença de sons competitivos.”*
- **Memória** – *“Capacidade de armazenar as informações acústicas e poder recuperá-las posteriormente, quando necessário.”*
- **Compreensão** – *“Capacidade de interpretar correctamente o significado da informação acústica.”*

A realização de todas estas etapas só é possível quando um indivíduo não possui qualquer perturbação do processamento auditivo (PPA).

Uma PPA é por definição uma perda ou um atraso de em alguma das etapas do PA, que inviabiliza uma ou mais aptidões do PA. (Martins, 2008)

2.4.1. Testes de avaliação do processamento auditivo

Por forma a avaliar a ausência ou presença de PPA existem vários tipos de testes sonoros, verbais e não-verbais. (Martins, 2008)

- **Processamento auditivo temporal** – *“Avaliam a capacidade de analisar os aspectos temporais da audição, nomeadamente, resolução temporal, mascaramento temporal, integração temporal e ordenação temporal.”*
- **Dicóticos** – Reproduzem um estímulo principal num ouvido e simultaneamente reproduzem no outro ouvido um estímulo competitivo. (Pfeiffer, 2007) Avaliam a capacidade de separar ou integrar estímulos auditivos diferentes.
- **Monoaurais de baixa redundância** – *“Avaliam a capacidade de reconhecer um estímulo degradado ou modificado apresentado em cada ouvido separadamente.”*
- **Interação Binaural (dicótico)** – *“Avaliam a capacidade do sistema auditivo central processar a informação recebida em ambos os ouvidos ao mesmo tempo (testes dicóticos).”*

2.4.2. Testes dicóticos

O trabalho realizado e descrito nesta dissertação apenas incidiu sobre os testes dicóticos e os testes de interação binaural, cada elemento do GIBS trabalhou sobre um teste ou um grupo de testes específico.

Nos subcapítulos abaixo será feito enquadramento dos testes e uma breve descrição do seu funcionamento, sem entrar em detalhes relativos às características dos sons reproduzidos e a características audiológicas.

2.4.2.1. Teste dicótico de dígitos

O teste dicótico de dígitos começou por ser usado por Kimura para testar indivíduos com lesões no lobo temporal, os testes antecessores apresentavam dois pares de palavras em simultâneo nos dois ouvidos, e era pedido ao paciente para repetir o maior número de palavras possível. (Kimura, 1967)

No teste dicótico de dígitos são reproduzidos dois dígitos em cada ouvido em simultâneo e é pedido ao indivíduo testado que repita os números ouvidos. (Lemos, 2008)

Os dígitos são combinados dois a dois, de forma a eliminar dígitos iguais. A ordenação dos pares é feita de forma aleatória e assim é construída uma lista de reprodução de pares de dígitos. (Lemos, 2008)

No teste original são apresentados 20 estímulos distintos, ou seja, 80 dígitos no total (40 para cada ouvido). O resultado consiste em contar o número de dígitos repetidos correctamente para cada ouvido em separado. (Lemos, 2008)

Existem algumas versões alternativas deste teste, se um paciente não poder responder oralmente, pode escrever os dígitos ou apontar numa placa a resposta, caso o paciente seja incapaz de repetir os quatro dígitos, existe uma versão que consiste em apresentar apenas um dígito em cada ouvido.

Este teste destina-se a avaliar lesões do tronco cerebral, córtex e inter-hemisféricas. (Timothy N. Welsh, 2000)

2.4.2.2. SSW ou testes de palavras espondáicas alternadas

O teste SSW foi um dos primeiros testes de avaliação do processamento auditivo central empregue por Audiologistas, na actualidade continua a ser utilizado para avaliar o processamento auditivo central, o teste é frequentemente realizado a crianças com dificuldades na linguagem e na aprendizagem. (PubMed.gov, 1983)

No teste SSW, os estímulos são formados por palavras com duas sílabas longas alternadas. Cada estímulo é composto por quatro palavras, as primeiras duas palavras são apresentadas isoladamente designando-se por não competitivos, a terceira e quarta são apresentadas em simultâneo nos dois ouvidos designando-se por competitivo. (Martins, 2007)

O teste inicia-se sempre no ouvido direito e vai alternado entre ouvido direito e ouvido esquerdo, desta forma metade dos estímulos tem início no ouvido direito e outra metade tem início no ouvido esquerdo. (Martins, 2007)

O teste na sua versão portuguesa é composto por 40 itens cada um formado por quatro palavras, totalizando um total de 160 vocábulos. (Martins, 2007)

A interpretação dos resultados deste teste é feita com base nos resultados obtidos na parte competitiva e na parte não competitiva. (PubMed.gov, 1983)

O indivíduo deve repetir o que ouviu obedecendo à ordem de apresentação das palavras. (Martins, 2007)

2.5. Aplicação dos testes nos dias de hoje

Apesar de existirem diferenças entre os dois tipos de testes, a sua aplicação é feita da mesma forma.

Actualmente os teste são administrados de forma manual, podendo de uma forma geral dizer-se que são administrados recorrendo a um SI que quase não aproveita as vantagens das TI.

A administração dos testes é feita recorrendo a um cd que contém os protocolos² gravados, associado a este cd existem folhas de resposta em papel que são preenchidas pelo Audiologista manualmente com as respostas do Paciente. (Lemos, 2008)

² Define-se como protocolo o conjunto de sequências ordenadas de um determinado tipo de teste.

A análise das respostas é efectuada manualmente e apenas no final da administração dos testes.

2.6. Conclusões

Apesar de a administração dos testes já ser feita recorrendo a um SI, verifica-se que é um SI que não faz uso das mais recentes tecnologias de informação e que apresenta várias desvantagens:

- A criação de um novo protocolo é uma tarefa que requer a gravação de um novo cd e a criação de uma nova folha de respostas.
- A existência de vários protocolos distintos obriga a que existam vários cd's e várias folhas de resposta.
- A análise de resultados é uma etapa que pode ser considerada crítica pois não é possível eliminar o factor de erro humano.

Com as TI existentes nos dias de hoje é possível construir de raiz um SI que optimize a administração de testes, análise de respostas e construção de protocolos.

A criação de raiz de um novo SI de informação devidamente planeado e estruturado assente nas novas tecnologias permite, colmatar as falhas apresentadas simplificando os seguintes processos:

- Gestão da informação clínica;
- Criação de protocolos de teste;
- Administração de testes;
- Análise de resultados;

3. Metodologia de Desenvolvimento do Sistema de Informação

3.1. Processo de desenvolvimento de SI

Segundo (Teixeira, 2008) o processo de desenvolvimento de um sistema de informação (PDSI) pode ser iniciado de duas formas distintas:

- Com base na tecnologia;
- Com base no problema;

No primeiro caso procura-se uma solução com base numa tecnologia específica independentemente do problema inicial.

No segundo caso procura-se a melhor forma de solucionar o problema inicial independentemente da tecnologia escolhida para atingir o objectivo.

O PDSI deve ser efectuado com algumas precauções, devem ser ouvidos todas as pessoas responsáveis pela organização onde vai ser implementado o SI, bem como todas as pessoas que directa ou indirectamente vão ser afectadas com a introdução do novo SI e finalmente adoptar metodologias que proporcionem o correcto levantamento de requisitos e que levem à correcta solução do problema. Por último é necessário fazer o levantamento tecnológico e enquadrá-lo na solução pretendida por forma a controlar as despesas associadas às TI.

Uma falha numa das etapas do PDSI pode originar um SI que não soluciona o problema existente, que em alguns casos pode piorar a situação existente, ou causar resistência por parte dos utilizadores, que continuaram a utilizar os métodos existentes antes do SI.

No SI desenvolvido a abordagem utilizada foi motivada pelo problema, significa que o SI pretende solucionar um problema já existente, no entanto e devido a restrições económicas foi necessário também fazer a abordagem com base na tecnologia. Desta forma tentou-se encontrar o ponto médio que satisfaz as duas abordagens sem prejuízo da solução final.

3.2. Metodologias de desenvolvimento

Engenharia de Software (ES) é por definição a disciplina que aborda de forma sistemática as operações de desenvolvimento e manutenção de *software*, desde as fases preliminares de levantamento de requisitos e desenvolvimento do sistema, até à fase de manutenção.

A origem da Engenharia de Software deveu-se ao grande crescimento de aplicações informáticas que se realizou nos últimos anos. Associado a este crescimento está o aumento de complexidade das mesmas que fez com que fosse necessário criar metodologias que permitissem de forma eficiente controlar todas as fases de um projecto.

Segundo (Teixeira, 2008), PDSI não é uma tarefa linear e acarreta alguns riscos tais como:

- Gastos superiores ao orçamento inicial;
- Tempo de desenvolvimento superior ao tempo estipulado inicial;
- Baixa qualidade e flexibilidade dos SI desenvolvidos;
- Excesso ou défice de funcionalidades que não correspondem à solução do problema;

Para contornar estes riscos à medida que a indústria do *software* evoluiu foram desenvolvidas metodologias e técnicas que visam tornar o PDSI uma actividade mais produtiva e eficiente.

Metodologia na ES é um processo com uma série de etapas recomendadas para a construção de um *software*.

Ao longo dos anos foram surgindo várias metodologias de desenvolvimento de *software*, a escolha de uma delas deve ser feita de forma ponderada, por forma a ser compatível com o problema e com os recursos existentes.

Os SI começaram por ser desenvolvidos de forma sequencial no que se denominou por metodologia em Cascata representada esquematicamente na Figura 2, esta no entanto começou a ser abandonada em favor de metodologias mais eficazes, por ser um modelo de risco e propício a falhas. (Selecting a Development Approach, 2008)

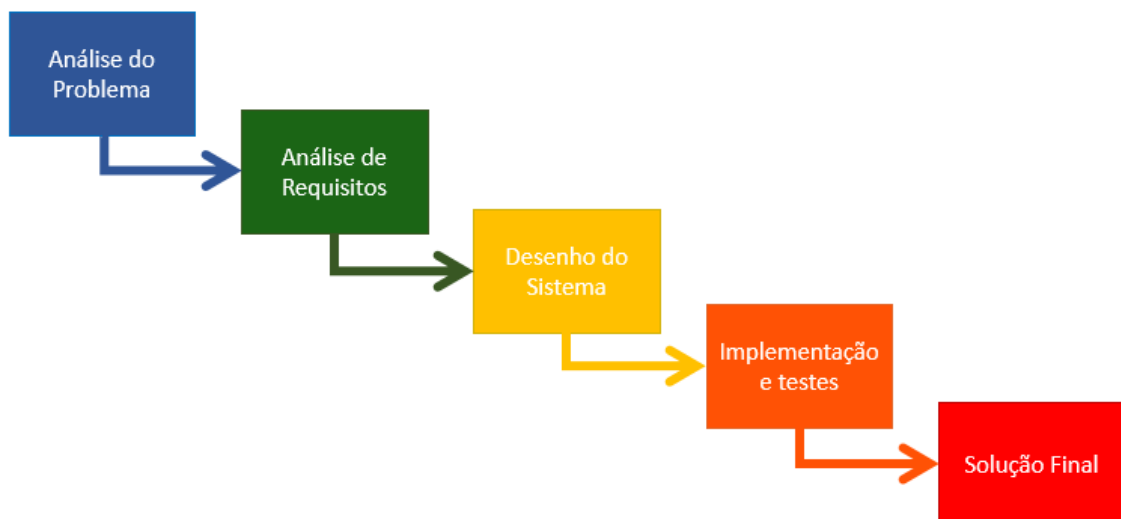


Figura 2 – Representação gráfica do desenvolvimento em cascata

Nesta metodologia o início da etapa seguinte depende sempre do término da anterior, podendo ser feita uma analogia com uma linha de montagem industrial sequencial. Esta abordagem é muito pouco flexível no que diz respeito à introdução de alterações numa das etapas anteriores, caso esta situação aconteça devido a uma alteração de requisitos numa das etapas iniciais todas as etapas seguintes estão condicionadas e poderão ter de ser refeitas. (Filomena Castro Lopes, 2009)

Outro aspecto negativo do desenvolvimento em cascata é o facto de não existir *feedback* ao longo do desenvolvimento, aumentando a probabilidade de se alcançar uma solução final que não cumpra ou não satisfaça na íntegra os resultados esperados. (Selecting a Development Approach, 2008)

Como o SI apenas será testado nas últimas etapas da metodologia quanto mais cedo for cometido um erro mais tarde será detectado e por consequência mais morosa será a sua correcção.

Actualmente e aliado à rápida evolução da tecnologia e ao facto de as organizações terem contextos de rápida mudança torna-se impraticável definir um documento final de requisitos nas primeiras etapas do processo. (Teixeira, 2008)

Todos estes factores negativos levaram a que a indústria de ES procura-se métodos mais flexíveis e que adaptem a requisitos dinâmicos.

Para colmatar as falhas do desenvolvimento em cascata surgiram as metodologias iterativas e incrementais.

A ideia de desenvolvimento de *software* a partir destas metodologias passa por trabalhar inicialmente com um conjunto reduzido de requisitos que permitam construir versões intermédias, versões estas que são analisadas levando a um novo levantamento de requisitos e construção de uma nova versão, este processo é repetido até ser atingida a versão final. (Videira, 2005)

Segundo (Videira, 2005) o desenvolvimento iterativo apresenta uma série de vantagens face ao desenvolvimento em cascata:

- Os riscos e dúvidas com maior importância são identificados no início do projecto, nas iterações iniciais onde ainda é possível tomar medidas para os corrigir sem grandes custos associados;
- Os utilizadores têm uma participação activa, o que permite identificar os requisitos reais do sistema;
- A execução de testes contínuos ao longo das várias versões permite uma avaliação objectiva do estado do projecto;

O desenvolvimento incremental permite aumentar a dimensão do projecto pouco-a-pouco ao longo das várias versões.

Combinando o desenvolvimento iterativo e incremental (Figura 3), a equipa responsável pelo projecto vai alargar pouco-a-pouco a qualidade, detalhe e âmbito do sistema desenvolvido.



Figura 3 - Representação gráfica do desenvolvimento iterativo e incremental

As metodologias incrementais e iterativas têm como desvantagem a burocracia e o volume excessivo de documentação e formalismos que são geradas ao longo das várias iterações.

Em 2001 uma equipa de Engenheiros de Software reuniu-se nos Estados Unidos da América para discutir e desenvolver metodologias alternativas que colmassem as falhas dos métodos existentes até então.

Desta reunião surgiram as metodologias ágeis como resposta ao facto das metodologias já não conseguirem lidar com o ritmo de mudança e grau de incerteza que o mercado actual, dinâmico e competitivo requer. (Teixeira, 2008)

Em contraponto com os métodos mais antigos as metodologias ágeis valorizam um processo de resposta rápido através do envolvimento dos desenvolvedores em tempo real com os clientes obtendo o seu feedback através da apresentação de *software* funcional numa estrutura altamente flexível à mudança.

Os métodos ágeis tentam minimizar os riscos de falhas através do desenvolvimento em curtos períodos de tempo, denominados de iterações.

Cada iteração funciona como o desenvolvimento de um pequeno projecto, onde são implementadas algumas das funcionalidades.

Ao contrário das metodologias anteriores onde as iterações não estão necessariamente orientadas para a adição de novas funcionalidades, uma metodologia ágil (Figura 4) procura implementar uma nova versão ao fim de cada iteração. (Teixeira, 2008)

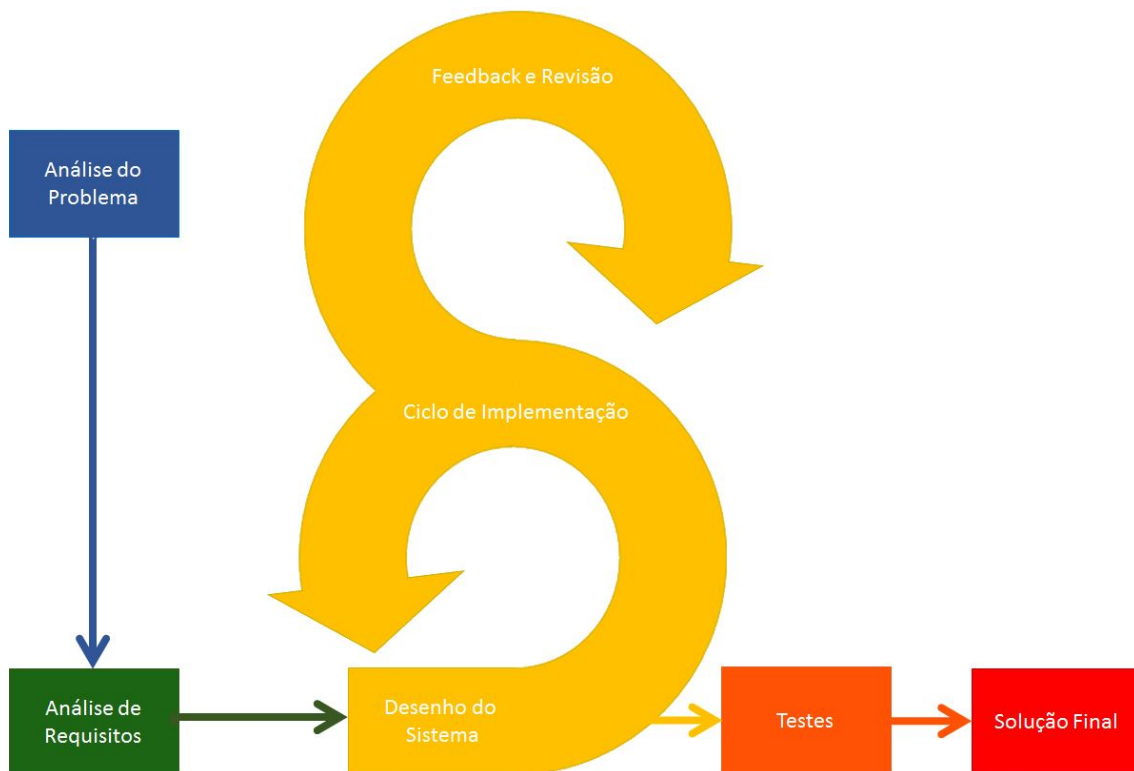


Figura 4 - Representação esquemática das metodologias ágeis

3.3. Prototipagem

Na fase inicial do projecto verificou-se que existiam alguns requisitos que não eram consistentes e que poderiam vir a sofrer alterações. Como este é um sistema que está a ser desenvolvido pela primeira vez e como não existem sistemas que sirvam de base de comparação o design e o interface gráfico levantavam algumas dúvidas de funcionalidade.

Para fazer face às condicionantes enumeradas em cima optou-se por uma metodologia evolutiva, em que o sistema é construído em várias etapas. Cada uma das etapas é testada pelo cliente, com a finalidade de avaliar os requisitos implementados e descobrir novos requisitos para a etapa seguinte. Este processo é repetido até se chegar a uma versão final onde sejam cumpridos todos os requisitos.

Das metodologias evolutivas optou-se pela prototipagem (Figura 5), esta metodologia tem por base a construção de protótipos visuais, que podem ou não ser funcionais. Com esta metodologia o cliente consegue ter uma melhor percepção da aplicação e das suas funcionalidades.

Esta é uma metodologia bastante flexível no que diz respeito à mudança de requisitos e é a mais indicada a utilizar quando os requisitos não estão bem definidos ou ainda não são totalmente conhecidos. (Filomena Castro Lopes, 2009)

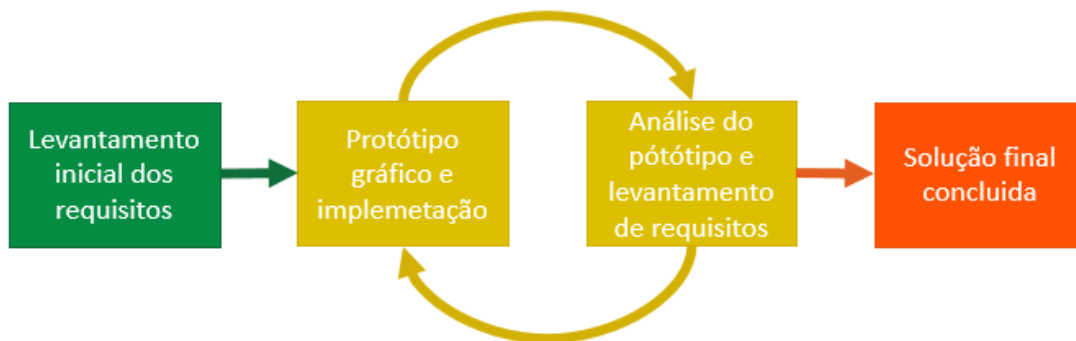


Figura 5 – Representação gráfica da metodologia de prototipagem

Ao longo do desenvolvimento do SI foram construídos inúmeros protótipos, nem todos funcionais, mas todos gráficos, por forma a avaliar a melhor forma de interacção.

Juntamente com estes protótipos foi elaborada documentação que regista as evoluções, alterações efectuadas e os requisitos das versões seguintes.

A Figura 6 representa um dos primeiros protótipos não funcionais em que é definido o esquema de navegação, concluída a validação por parte da equipa do GIBS passou-se à implementação do protótipo.

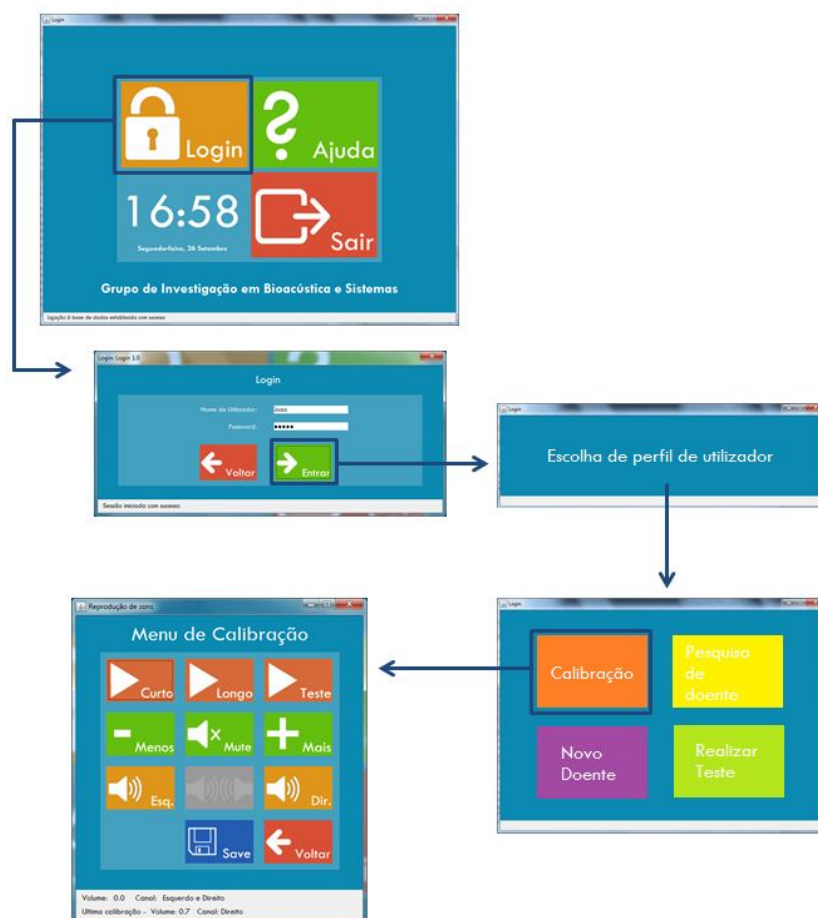


Figura 6 - Protótipo inicial do esquema de navegação

4. Análise e desenvolvimento

4.1. Requisitos

O sucesso ou insucesso de um *software* ou SI depende da forma como é efectuado o levantamento de requisitos e da forma como cumpre as expectativas dos utilizadores.

O processo de levantamento de requisitos não é uma tarefa simples e deve ser encarada com toda a seriedade pois é o ponto-chave de todo o *software* uma falha nesta fase poderá deitar por terra todo o trabalho realizado ao longo de meses.

Os requisitos devem começar a ser definidos depois de definidas as restrições do projecto, quer sejam estas económicas ou tecnológicas, por forma a fazer o real enquadramento da solução proposta e a satisfazer os objectivos da organização onde se insere. (Filomena Castro Lopes, 2009)

Ao longo dos vários protótipos por que passou o SI aqui apresentado, foi construída uma lista de requisitos, em conjunto com os orientadores do projecto e futuros utilizadores da aplicação. Estes requisitos foram identificados como funcionais e não funcionais.

Requisitos funcionais são aqueles que descrevem os serviços que o sistema deverá oferecer, de uma forma genérica são todos aqueles que respondem à pergunta “O que faz o sistema?”. (Teixeira, 2008)

Os requisitos não funcionais são aqueles que descrevem as restrições na implementação do sistema são expressos em termos de custos, desempenho, segurança, fiabilidade, portabilidade e usabilidade. É esta classe de requisitos que limita muitas das vezes a forma como os requisitos funcionais são implementados. (Teixeira, 2008)

Da análise efectuada ao longo do projecto definiu-se que o SI deveria cumprir os requisitos apresentados na Tabela 1.

Descrição	Tipo
Deve ser desenvolvido com o mínimo custo associado	Não funcional
Deve garantir a segurança de dados	Não funcional
Deve ser multiplataforma, independente do sistema operativo	Não Funcional
Deve ser multiposto	Não funcional
Deve ter capacidade de funcionar com bases de dados locais, ou com bases de dados externas alojadas nos servidores do GIBS.	Não Funcional
A interface gráfica deve ser compatível com a utilização de ecrãs tácteis.	Não Funcional
Deve ter um ecrã de Login para o utilizador efectuar a autenticação, através da introdução do nome de utilizador e password.	Funcional
Deve permitir três tipos distintos de perfis de utilizador: 1- Administrador; 2- Técnico; 3- Utilizador;	Funcional

Um utilizador pode ter mais do que um perfil.	Funcional
O Administrador deve ter privilégios para gerir utilizadores: 1- Criar novos Administradores, Técnicos ou Utilizadores; 2- Activar/Desactivar Administradores, Técnicos ou Utilizadores;	Funcional
O Administrador deve poder alterar os dados pessoais dos Pacientes.	Funcional
O Administrador deve poder gerir os testes dicóticos de dígitos e SSW: 1- Criar novos protocolos; 2- Activar/Desactivar protocolos; 3- Editar protocolos;	Funcional
A criação de protocolos é associada ao posto.	Funcional
A criação de um protocolo SSW deve ser feita apenas de forma manual.	Funcional
A criação de um protocolo dicótico de dígitos deve ser feita de quatro formas distintas: 1- Manual; 2- Manual com repetição; 3- Semiautomática; 4- Automática;	Funcional
Os protocolos dicóticos de dígitos e SSW devem poder ser criados para efeitos de demonstração.	Funcional
O Administrador deve poder gerir os sons do SI: 1- Adicionar novo som; 2- Activar/Desactivar som; 3- Editar Som;	Funcional
O Administrador deve poder gerir os sons da calibração.	Funcional
Os sons adicionados na aplicação devem ficar associados ao posto.	Funcional
O Administrador deve ter acesso às opções globais do SI.	Funcional
O Técnico deve poder efectuar a calibração de som da aplicação.	Funcional
Devem existir três sons de calibração distintos.	Funcional
Os sons de calibração devem ter identificação das suas características visíveis.	Funcional
A calibração deve ser guardada associada ao posto.	Funcional
Deve ser possível alterar os parâmetros da calibração enquanto se reproduzem os sons de calibração.	Funcional
No caso de já existir uma calibração guardada, a calibração deve ser carregada no início da aplicação.	Funcional
O Audiologista ao alterar a calibração deverá ter informação relativa à calibração actual e à calibração guardada.	Funcional
A escala de som no painel de calibração deverá ser representada em percentagem entre 0% e 100%.	Funcional
O Técnico deve poder pesquisar pacientes.	Funcional
O Técnico deve poder consultar o histórico dos pacientes.	Funcional
O Técnico deve poder gerir a sua agenda pessoal: 1- Consultar agenda; 2- Efectuar marcações;	Funcional
O Técnico deve poder realizar testes dicóticos de dígitos e SSW.	Funcional
O Técnico deve ter acesso a uma breve descrição do funcionamento dos testes.	Funcional
O Técnico deve ter opção de efectuar um teste em modo de demonstração.	Funcional
Durante a realização de um teste o técnico só deve poder avançar para o	Funcional

som seguinte no final da reprodução do som anterior.	
No final de um teste o técnico deve poder descartar ou guardar na base de dados os resultados do teste.	Funcional
O Utilizador deve poder gerir a agenda de todos os Técnicos: 1- Consultar agendas; 2- Efectuar marcações;	Funcional
O Utilizador deve poder pesquisar Pacientes.	Funcional
O Utilizador deve poder consultar o histórico dos Pacientes.	Funcional
O Utilizador deve poder registar novos Pacientes.	Funcional
Todos os utilizadores da aplicação devem poder alterar a sua password.	Funcional
O SI deve ser compatível com o cartão do cidadão português.	Funcional
Um utilizador deve poder efectuar login na aplicação utilizando o cartão do cidadão português.	Funcional
Deve ser possível preencher os formulários de registo de Pacientes, Técnicos e Utilizadores, automaticamente através da leitura dos dados do cartão do cidadão.	Funcional
Deve ser possível pesquisar pacientes através do cartão do cidadão.	Funcional
Deve ser possível imprimir em formato de papel ou PDF o relatório de um determinado teste de um Paciente com as respostas certas e erradas.	Funcional
Um relatório deve ter identificação do posto onde foi realizado o teste e a identificação do Técnico que realizou o teste.	Funcional
Deve ser gerado um número identificador único para cada posto automaticamente.	Funcional
Deve ser desenvolvida uma aplicação compatível com um dispositivo móvel e que permita a realização de testes dicóticos de dígitos a pessoas que não consigam falar.	Funcional

Tabela 1 - Requisitos do SI

4.2. Tecnologia utilizada no desenvolvimento da solução

Para o desenvolvimento do SI de forma a cumprir os requisitos não funcionais, foram utilizadas tecnologias *open source* e multiplataforma, desta forma foi possível desenvolver o SI com o mínimo de custos associados garantindo que a aplicação é independente do sistema operativo em que é executada.

Em baixo é enumerada toda a tecnologia que foi utilizada para a construção do Sistema de Informação.

4.2.1. Java

A escolha da linguagem de desenvolvimento do SI recaiu sobre o Java por esta ser gratuita, estável, orientada a objectos, e pelas soluções de portabilidade que oferece, cumprindo assim o requisito não funcional de ser multiplataforma estabelecido inicialmente. (Programming, 2012)

Foi utilizada a versão 7, a mais recente existente no início da implementação do SI, esta versão poderá causar alguns problemas de compatibilidade em computadores com arquitecturas mais antigas e que já não estão disponíveis no mercado actualmente.

4.2.2. IDE's de desenvolvimento

Para o desenvolvimento foram utilizados dois IDE's distintos, para a aplicação principal a escolha recaiu sobre o NetBeans, por ser um IDE que permite a construção de ambientes gráficos de forma bastante eficiente, gerando sempre o código associado a cada elemento, garantindo que os requisitos não funcionais associados aos custos continuam a ser cumpridos.

O segundo IDE foi utilizado para desenvolver uma aplicação móvel para Android, a escolha recaiu sobre o Eclipse, por ser o IDE oficial de desenvolvimento para esta plataforma, e porque à semelhança do NetBeans garante que os requisitos não funcionais são cumpridos. (Cinar, 2012)

4.2.3. Postgres SQL

O Postgres SQL foi o motor de base de dados escolhido, por ser gratuito e por ter um conjunto de funcionalidades que permitem de forma fácil e intuitiva gerir os dados associados ao SI.

Das funcionalidades que fizeram a escolha recair sobre o Postgres destaco os tipos de dados flexíveis que permitem por exemplo criar uma coluna com dados armazenados num vector. (Postgres, 2012)

4.2.4. Cartão do Cidadão

O SI foi desenvolvido de forma a assegurar a compatibilidade com o cartão do cidadão português, por considerar que este é uma mais-valia a qualquer sistema de informação recente.

Para implementação desta funcionalidade foi utilizado o SDK fornecido pela identidade gestora do CC, e um leitor de cartões simples.

4.2.5. Dispositivo com Android

O Android foi escolhido para o desenvolvimento de uma aplicação móvel compatível com a aplicação principal, por ser um sistema operativo livre e por existirem diversos equipamentos de baixo custo no mercado com este sistema operativo.

Outro factor considerado na escolha foi o facto de a linguagem de programação utilizada ser o Java.

4.3. Interface Gráfica

A interface gráfica que permite a interacção homem máquina é umas das componentes mais importantes num SI, uma interface usável e agradável pode ser favorável à decisão de aquisição ou utilização de um SI, uma interface sem estas características pode deitar por terra todo um projecto por muito bem implementado que esteja.

Uma interface gráfica bem definida deve garantir facilidade de interacção por parte do utilizador, não deve exigir um tempo de aprendizagem muito grande sobre pena de redução de produtividade e introdução de erros inaceitáveis no SI, o que levará ao seu fracasso. (Teixeira, 2008)

Um dos requisitos não funcionais relativos à interface gráfica prevê que a aplicação seja compatível com ecrãs tácteis, optou-se por uma abordagem diferente das aplicações tradicionais baseadas em barras de menus, por forma a facilitar a interacção por parte do utilizador sem utilização do rato.

Todo o ambiente gráfico tem como inspiração a linguagem de design da Microsoft, Metro UI.

A equipa da Microsoft que desenvolveu esta linguagem inspirou-se nos sinais do Metro de Seattle onde a companhia tem a sua sede e no estilo de design Suíço desenvolvido na Suíça em 1950. (Burns, 2012)

O estilo de design Suíço define a utilização dos seguintes elementos: tipografia, iconografia e organização de elementos de forma proporcional.

- Tipografia – A fonte utilizada para o texto deve ser limpa e simples sem adornos desnecessários que distraiam o destinatário da mensagem. (Burns, 2012)
- Iconografia – O design Suíço faz uso de ícones para reforçar a mensagem que se pretende passar ou quando a mensagem é dirigida a um público internacional que facilmente pode interpretar o significado através do ícone. Os ícones em conjunto com a tipografia foram utilizados em grande escala pela primeira vez nos Jogos Olímpicos de 1972 em Munique. O estilo de design Suíço e a utilização de ícones familiares ajudou a organização a comunicar com os espectadores internacionais presentes nos jogos. (Burns, 2012)

- Organização proporcional dos elementos – De forma a manter a organização e a simplicidade para que nada desvie a atenção do destinatário da mensagem o conteúdo é organizado em formas geométricas iguais ou que permitam uma visualização com espaçamento igual entre os elementos, o tamanho da fonte utilizada no texto e os ícones é igual em todos os objectos. (Burns, 2012)

Um dos pontos-chave do Metro UI (Figura 7) é uma melhor objectividade do conteúdo das aplicações, recorrendo a conteúdos gráficos coloridos e expressivos das funcionalidades é possível melhorar a experiencia de utilização. (Schooley, 2012)

A linguagem de design foi desenvolvida de forma a agrupar tarefas semelhantes, e a eliminar menus e outros aspectos gráficos supérfluos, o que melhora a velocidade de utilização. (Schooley, 2012)



Figura 7 - Metro UI do novo Windows 8

4.3.1. Definição da interface gráfica

A interface gráfica teve como inspiração o Metro UI, no sentido em que a interacção do utilizador com a aplicação é feita através de grandes botões coloridos, tudo o resto foi desenvolvido e planeado de raiz.

A interface gráfica criada tem uma identidade própria pois está adaptada ao máximo às funcionalidades requeridas, foi um dos grandes desafios no desenvolvimento do SI, e que cumpre na íntegra o requisito não funcional de que a aplicação deverá ser compatível com ecrãs tácteis.

É expectável que o primeiro contacto do utilizador com este interface cause alguma estranheza, é necessário alguma habituação ao novo conceito, contudo a aprendizagem deverá ser feita de forma rápida e intuitiva.

Caso o utilizador já esteja familiarizado com os novos sistemas da Microsoft que recorrem ao Metro UI a adaptação à aplicação será feita de forma natural.

O aspecto negativo desta escolha recai sobre os utilizadores que não estão familiarizados com produtos Microsoft, pois a aplicação quando executada em sistemas operativos que não o Windows continuará a ter o mesmo aspecto gráfico, para estes utilizadores a curva de aprendizagem poderá ser um pouco superior.

Para uma melhor compreensão da interface desenvolvida segue-se uma explicação detalhada de todas as suas características e funcionalidades (Figura 8).



Figura 8 – Definição da interface gráfica

- 1- **Título** – Identifica o painel onde se encontra o utilizador.
- 2- **Painel de agrupamento** – Os painéis de agrupamento têm no máximo quatro botões visíveis todos coloridos, não tendo as cores dos botões um significado específico atribuído.
- 3- **Botão de continuação** – Botão apenas visível se o painel tiver continuação. Um painel tem continuação quando existem mais de 4 botões, no painel de agrupamento.
- 4- **Barra de estados** – Esta barra é constante ao longo de toda a aplicação, é aqui que são apresentadas instruções ao utilizador e onde são exibidas mensagens geradas pela própria aplicação.

A barra pode apresentar 4 cores, que tem significados distintos:

- Cinzento – Para informações sobre o funcionamento da aplicação;
- Verde – Para operações realizadas com sucesso.
- Amarelo – Para erros não críticos, ou consultas à base de dados sem retorno de resultados.
- Vermelho – Para erros críticos.

- 5- **Botão de retroceder** – Este botão permite navegar um nível para trás nos painéis, até ao painel de perfis. No painel de perfis o botão é substituído pelo botão de *logout*.

Figura 9 - Painel de tarefa

- 6- **Painel de tarefa** – (Figura 9) Painéis onde o utilizador realiza as tarefas da aplicação.



Figura 10 - Botões da aplicação

- 7- **Botões** – (Figura 10) Conjunto de botões utilizados nos painéis. Todos os botões são apresentados numa cor viva.

7.1 – **Botão de agrupamento** – Os botões de agrupamento são representados com texto no canto inferior direito e com uma imagem ilustrativa no lado esquerdo. Os botões de agrupamento dão acesso a novos painéis de agrupamento ou a um painel de tarefa.

7.2 – **Botão de tarefa** – Os botões de tarefa seguem a mesma representação gráfica dos botões de agrupamento, embora sejam mais pequenos. Estes botões apenas aparecem nos painéis de tarefa, e executam a tarefa descrita no texto.

7.3 – **Botão de tarefa simples** – Estes botões apenas aparecem nos painéis de tarefa durante a realização de um teste, não possuem texto e têm como única função marcar uma resposta certa ou errada.

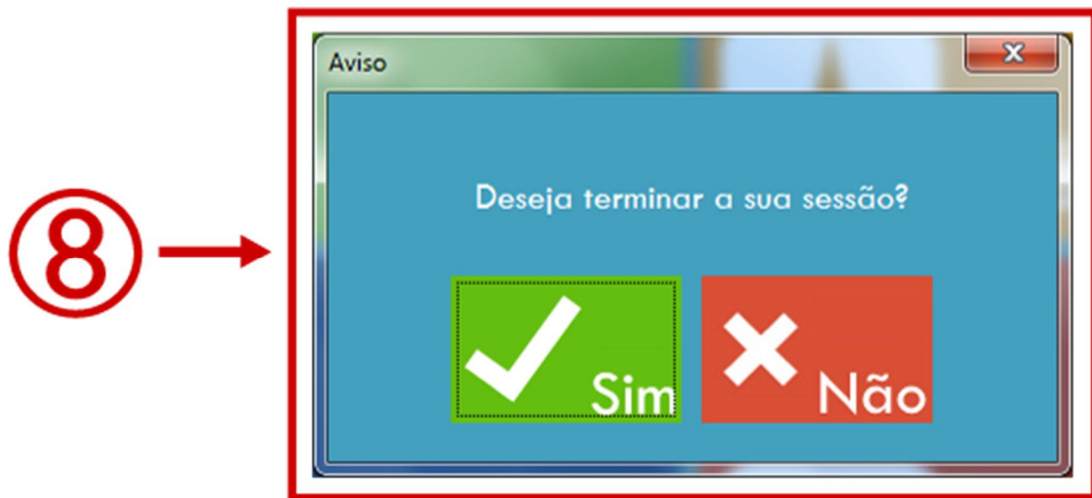


Figura 11 - Janela de aviso

- 8- **Janela de aviso** – (Figura 11) Esta janela foi desenvolvida de origem para se enquadrar com o estilo da aplicação, é apresentada ao utilizador sempre que pode ocorrer alguma alteração no estado da aplicação ou na estrutura de dados. É composta por uma mensagem de aviso e por dois botões de tarefa.

4.4. Descrição conceptual da solução implementada

A descrição da aplicação desenvolvida é apresentada de forma conceptual através de diagramas de casos de uso por forma a tornar mais fácil a leitura de todas as funcionalidades disponíveis e por quem podem ser utilizadas.

Um diagrama de casos de uso descreve as funcionalidades de um sistema, por outras palavras pode afirma-se que é um conjunto de diagramas que descreve a sequência de eventos que um actor utiliza para completar um processo. (Rosenberg, 2007)

Um caso de uso representa uma interacção entre um actor e o sistema e tem uma descrição da funcionalidade.

Um actor é um humano ou uma máquina que interage directamente com o sistema para executar tarefas, é sempre externo ao sistema e está exclusivamente ligado a casos de uso.

Um caso de uso é uma especificação de um conjunto de acção que o sistema executa e que produzem um resultado que pode ou não ser visível pelo utilizador. Os casos de uso são representados por uma elipse, com o nome no seu interior.

Os casos de uso podem ser relacionados entre si de três formas distintas, através de uma ligação normal, através de um *include* ou através de *extend*.

As ligações normais representam o fluxo de continuidade da aplicação, as ligações marcadas com um *include* representam casos que ocorrem obrigatoriamente, as ligações com *extend* representam casos de uso opcionais, ou seja que nem sempre ocorrem.

As ligações normais são representadas por setas e as ligações com *include* ou *extend* são representadas por setas que contém a descrição da sua função. (Rosenberg, 2007)

Nos casos de uso serão apresentados 4 actores distintos:

- Administrador
- Técnico
- Utilizador
- Paciente

A representação dos vários actores no diagrama será dividida por módulos, por forma a manter o nível de complexidade do diagrama baixo, sempre que se justificar serão incluídas como forma de complemento na descrição dos casos de uso, imagens reais da aplicação.

4.4.1. Sistema

Este é o módulo que representa o início do sistema, o utilizador efectua a sua autenticação e é redireccionado para a sua área de trabalho mediante o seu perfil (Figura 12).

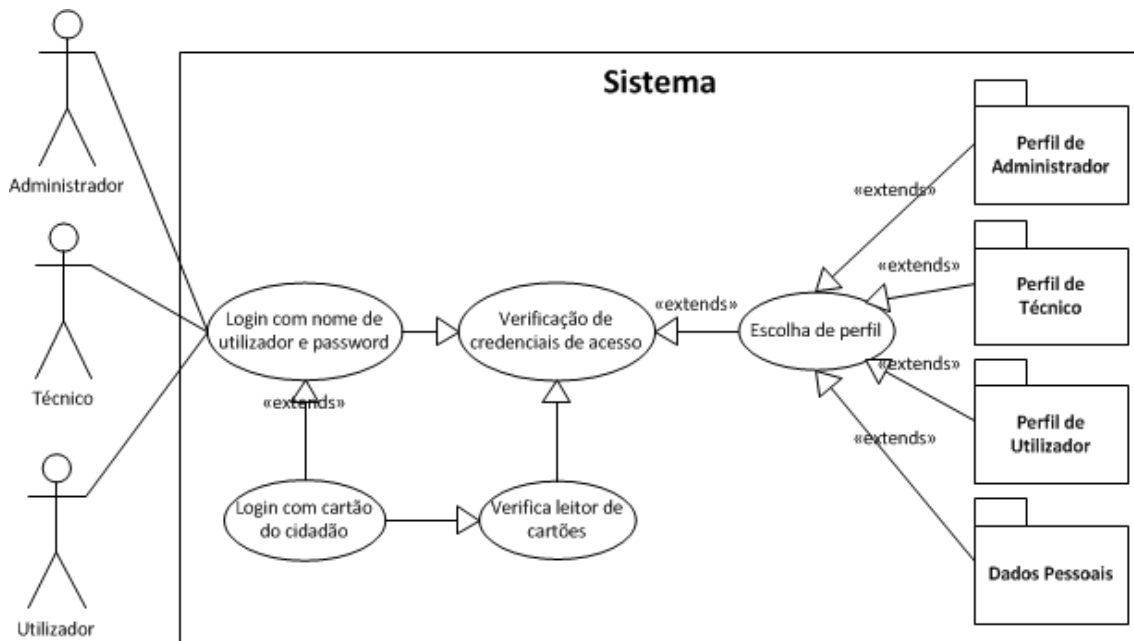


Figura 12 – Diagrama de casos de uso do SI

Descrição dos casos de uso

Login com nome de utilizador e password – (Figura 13) O utilizador efectua a autenticação na aplicação introduzindo as suas credenciais, nome de utilizador e password.

Login com cartão do cidadão – (Figura 13) O Utilizador depois de passar pelo modo de autenticação com nome de utilizador e password pode escolher autenticar-se com CC mediante a introdução de um código pin.

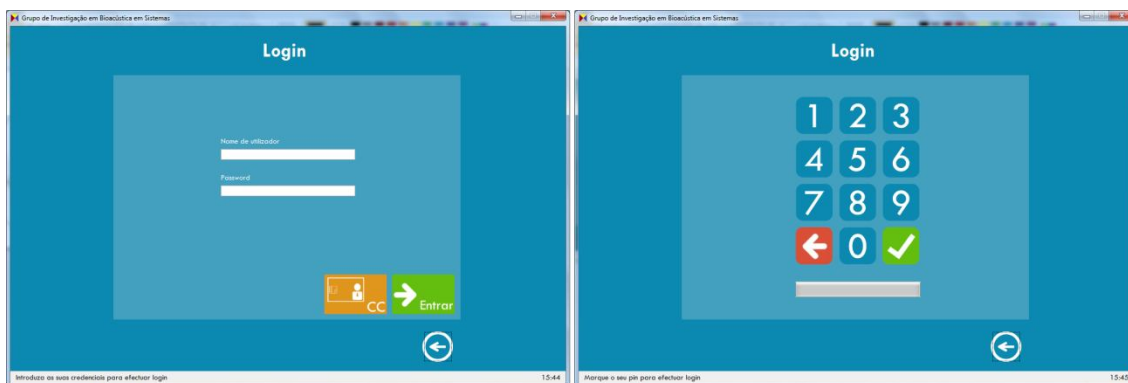


Figura 13 - Ecrãs de login da aplicação

Verifica leitor de cartões – Ao escolher o método de autenticação com CC a aplicação verifica a existência do *hardware* necessário para efectuar a operação.

Verifica credenciais de acesso – A aplicação acede à base de dados e verifica se as credenciais introduzidas estão correctas.

Escolha de perfil – (Figura 14) Caso a verificação das credenciais tenha sido efectuada com sucesso o utilizador é reencaminhado para um painel de escolha de perfil, e que dará acesso a

todos os perfis que tem associados à sua conta. Os casos de uso representados no diagrama em forma de agregação correspondem a cada tipo de perfil e serão explicados em detalhe.



Figura 14 - Menu de escolha de perfil

4.4.2. Perfil de Administrador

Este é o módulo do Perfil de Administrador (Figura 15) onde são detalhadas todas as funções administrativas que é possível processar na aplicação.

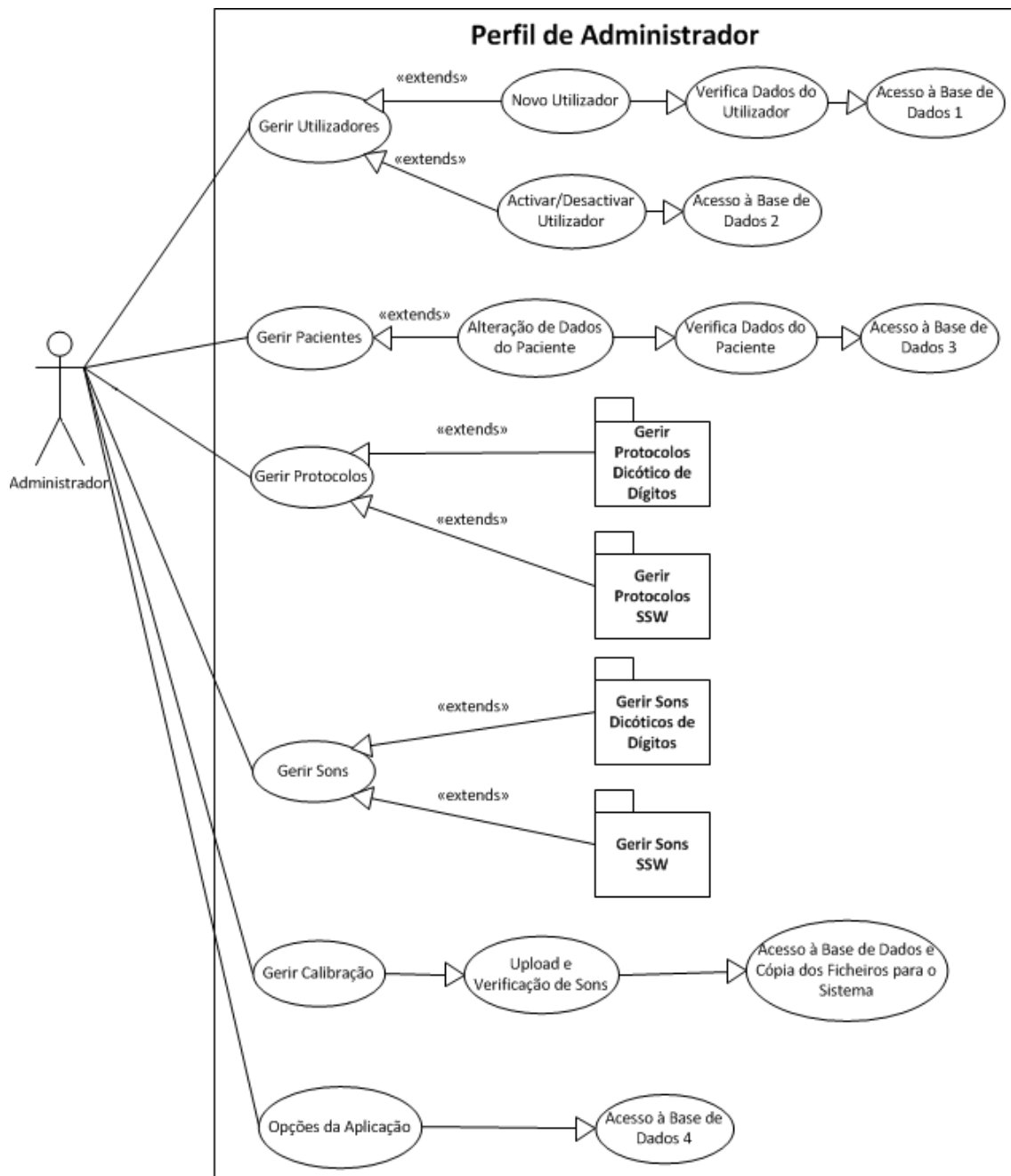


Figura 15 - Diagrama de casos de uso do perfil de Administrador

Descrição dos casos de uso

Gerir Utilizadores – Acesso às opções de gestão de todos os utilizadores do sistema.

Novo Utilizador – Preenchimento do formulário (Figura 16) de um novo utilizador indicando os seus dados pessoais, permissões e credenciais de acesso. A autenticação com cartão do cidadão é opcional, caso o utilizador não seja possuidor de um, os campos respectivos devem ficar em branco. As credenciais de acesso são definidas inicialmente pelo Administrador, devendo o utilizador alterar os códigos de acesso na primeira vez que acede à aplicação.

Figura 16 - Registo de um novo utilizador

Verifica Dados do Utilizador – É efectuada uma verificação dos dados inseridos no formulário.

Acesso à Base de Dados 1 – Caso a verificação anterior seja realizada com sucesso é efectuado um acesso à base de dados, primeiro para verificar se o novo utilizador não é coincidente com um já registado e posteriormente é efectuado o registo.

Activar/Desactivar Utilizador – Permite ao Administrador desactivar um utilizador do sistema, uma vez desactivado o utilizador deixa de se conseguir autenticar na aplicação.

Acesso à Base de Dados 2 – É feito um acesso à base de dados para actualizar o estado do utilizador.

Gerir Pacientes – É efectuado um acesso à base de dados que permite ao Administrador visualizar todos os pacientes registados na aplicação.

Alteração de Dados do Paciente – O Administrador pode alterar os dados do Paciente caso tenham sido introduzidos de forma incorrecta.

Verifica Dados do Paciente – É efectuada uma verificação à integridade dos dados alterados pelo Administrador, por forma a verificar se não violam nenhuma das regras estabelecidas.

Acesso à Base de Dados 3 – Caso a verificação anterior tenha sido efectuada com sucesso é efectuado um acesso à base de dados para alterar o registo do Paciente.

Gerir Protocolos – Acesso às opções de gestão de protocolos.

Gerir Sons – Acesso às opções de gestão de sons.

Gerir Calibração – Acesso ao menu de gestão de sons da calibração.

Upload e Verificação de Sons – O Administrador efectua o *upload* dos ficheiros de calibração para o sistema, é feita uma verificação ao tipo de ficheiro, por forma a garantir que o som está no formato WAV a 16 bits.

Acesso à Base de Dados e Cópia dos Ficheiros para o Sistema – Caso a verificação anterior tenha sido realizada com sucesso os ficheiros são copiados para a pasta de sistema correspondente e é efectuado o registo do novo som na base de dados.

Opções da Aplicação – Acesso às opções da aplicação

Acesso à Base de Dados 4 – É efectuado o acesso à base de dados para actualizar as opções da aplicação.

4.4.2.1. Gerir Protocolos Dicótico de Dígitos

Este é o módulo onde é feita toda a gestão dos protocolos dicóticos de dígitos (Figura 17).

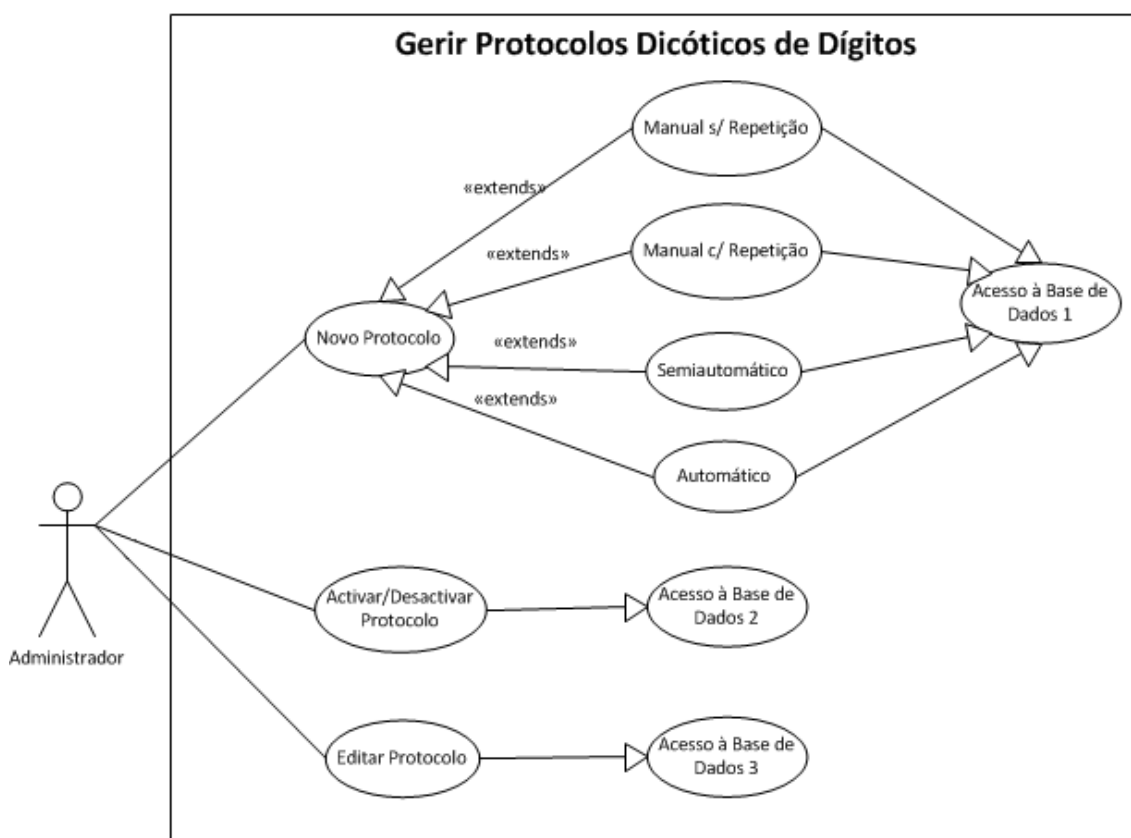


Figura 17 - Diagrama de casos de uso da gestão de protocolos dicóticos de dígitos

Descrição dos casos de uso

Novo Protocolo – Acesso ao menu de escolha do modo de criação de um novo protocolo.

Manual s/ Repetição – (Figura 18) Criação de um protocolo de forma manual, em que o utilizador escolhe todas as sequências que pretende, neste modo é garantido que uma sequência já introduzida no protocolo não pode voltar a ser repetida.

Manual c/ Repetição – Criação de um protocolo de forma manual, em que o utilizador escolhe todas as sequências que pretende, neste modo é possível repetir as sequências no protocolo, a única restrição é não poder repetir duas vezes seguidas a mesma sequência.

Semiautomático – Modo que permite criar protocolos que repetem padrões de sequências definidas manualmente pelo Administrador, é necessário criar o primeiro padrão e introduzir o número de repetições do mesmo.

Automático – Modo que cria protocolos de forma autónoma, o Administrador apenas tem de introduzir o número de sequências que pretende.



Figura 18 - Construção de um protocolo de dígitos

Acesso à Base de Dados 1 – É efectuado um acesso à base de dados para guardar os protocolos grades.

Activar/Desactivar Protocolo – Permite ao Administrador desactivar um protocolo do sistema, uma vez desactivado o protocolo deixa de poder ser administrado.

Acesso à Base de Dados 2 – É efectuado um acesso à base de dados para actualizar o estado do registo do protocolo.

Editar Protocolo – Permite ao Administrador editar os dados identificativos dos protocolos.

Acesso à Base de Dados 3 – É efectuado um acesso à base de dados para actualizar os dados do registo do protocolo.

4.4.2.2. Gerir Protocolos SSW

Este é o módulo onde é feita toda a gestão dos protocolos dicóticos de dígitos (Figura 19).

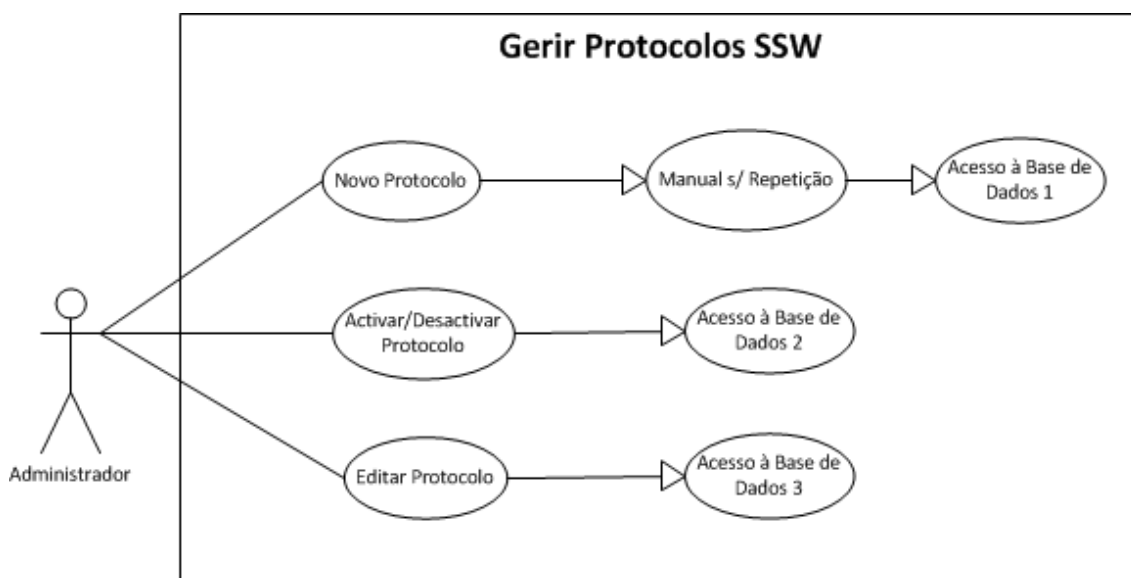


Figura 19 - Diagrama de casos de uso da gestão de protocolos SSW

Descrição dos casos de uso

Novo Protocolo – Acesso ao menu de escolha do modo de criação de um novo protocolo.

Manual s/ Repetição – Devido às características especiais do teste SSW apenas é possível criar protocolos de forma Manual e sem repetição, o utilizador escolhe as sequências que pretende introduzir no protocolo e é garantido que estas nunca se repetem.

Acesso à Base de Dados 1 – É efectuado um acesso à base de dados para inserir os dados relativos ao novo protocolo.

Activar/Desactivar Protocolo – Permite ao Administrador desactivar um protocolo do sistema, uma vez desactivado o protocolo deixa de poder ser administrado.

Acesso à Base de Dados 2 – É efectuado um acesso à base de dados para actualizar o estado do registo do protocolo.

Editar Protocolo – Permite ao Administrador editar os dados identificativos dos protocolos.

Acesso à Base de Dados 3 – É efectuado um acesso à base de dados para actualizar os dados do registo do protocolo.

4.4.2.3. Gerir Sons Dicóticos de Dígitos / SSW

A gestão de sons dicóticos de dígitos e de sons SSW está agrupada e é representada neste diagrama, devido aos casos de uso serem exactamente os mesmos (Figura 20).

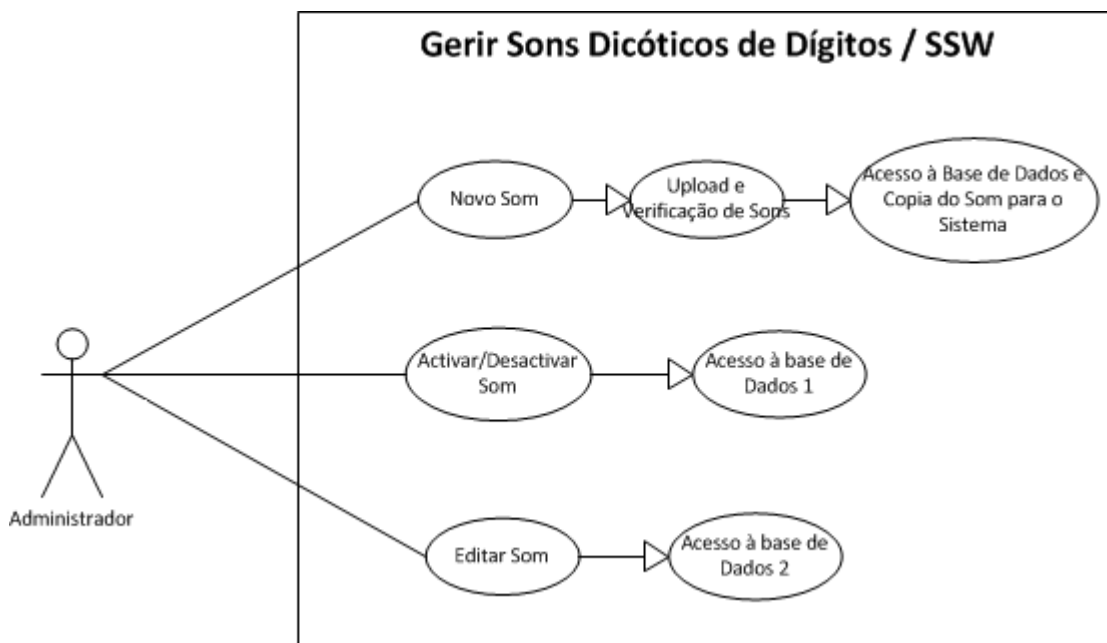


Figura 20 - Diagrama de casos de uso da gestão de sons

Descrição dos casos de uso

Novo Som – Acesso ao painel onde é possível efectuar a adição de um novo som ao sistema.

Upload e Verificação de Sons – O Administrador efectua o *upload* dos ficheiros de som para o sistema, é feita uma verificação ao tipo de ficheiro, por forma a garantir que o som está no formato WAV a 16 bits.

Acesso à Base de Dados e Cópia do Som para o Sistema – Caso a verificação anterior tenha sido realizada com sucesso os ficheiros são copiados para a pasta de sistema correspondente e é efectuado o registo do novo som na base de dados.

Activar/Desactivar Som – Permite ao Administrador desactivar um som do sistema, uma vez desactivado o protocolo deixa de poder ser administrado.

Acesso à Base de Dados 1 – É efectuado um acesso à base de dados para alterar o estado do registo do som, e posteriormente é efectuado outro acesso para alterar o estado de todos os protocolos que utilizem este som.

Editar Som – Permite ao Administrador editar os dados identificativos dos som.

Acesso à Base de Dados 3 – É efectuado um acesso à base de dados para actualizar os dados do registo do som.

4.4.3. Perfil de Técnico

Este é o módulo de técnico onde são detalhadas todas as opções que um Técnico Audiologista tem na aplicação (Figura 21).

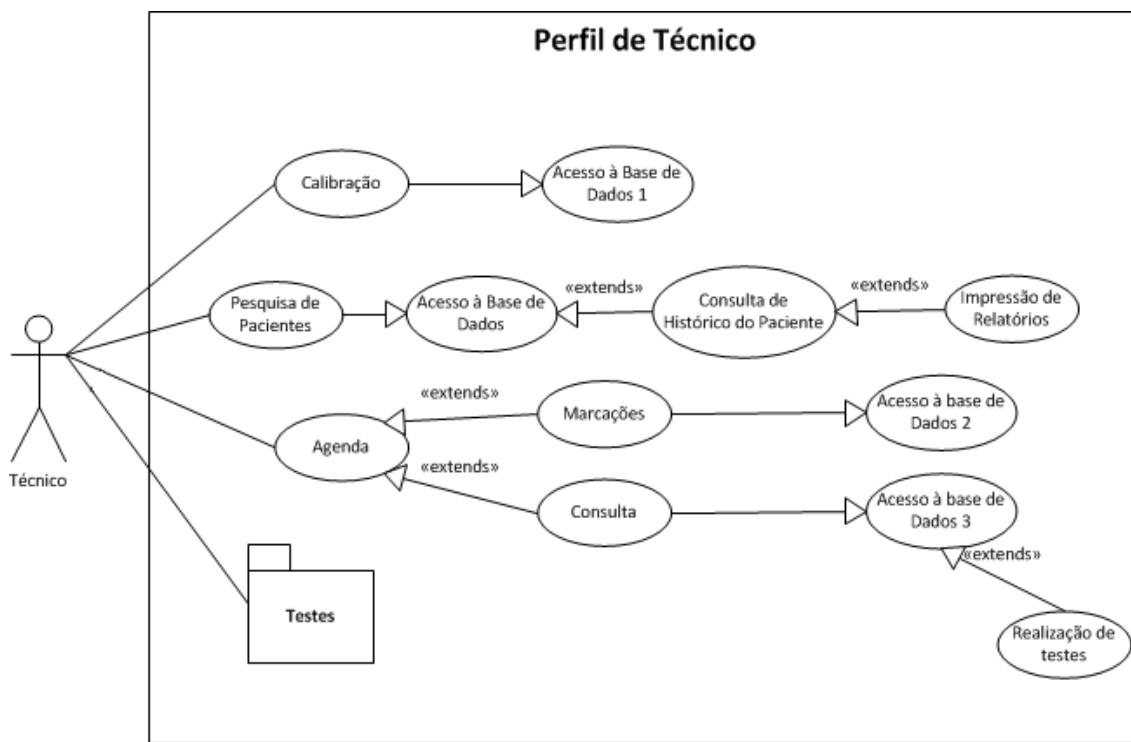


Figura 21 - Diagrama de casos de uso do perfil de técnico

Descrição dos casos de uso

Calibração – Acesso ao painel de calibração, que permite configurar o volume de som com que os testes são reproduzidos.

Acesso à Base de Dados 1 – Caso tenha sido feita alguma alteração é efectuado um novo acesso à base de dados para actualizar os valores correspondentes à calibração de som.

Pesquisa de Pacientes – Acesso ao painel de pesquisa de Pacientes

Acesso à Base de Dados – É efectuado um acesso à base de dados com base nos parâmetros introduzidos pelo Técnico.

Consulta de Histórico de Paciente – Depois de efectuada a pesquisa de Paciente com sucesso é possível ao Técnico consultar todo o histórico de testes do Paciente.

Impressão de Relatórios – Caso seja necessário o técnico pode imprimir um relatório com os resultados do teste em PDF.

Agenda – Acesso ao painel de agenda pessoal do técnico.

Marcações – É possível ao técnico efectuar a marcação de testes na sua agenda pessoal.

Acesso à Base de Dados 2 – É efectuado um acesso à base de dados para registar a marcação do novo teste.

Consulta – O técnico pode a qualquer momento consultar a sua agenda pessoal.

Acesso à Base de Dados 3 – Na consulta à agenda de um Técnico é efectuado um acesso à base de dados que devolve todos os testes agendados, com os parâmetros seleccionados.

Realização de Testes – Caso o Técnico pretenda pode iniciar directamente a realização de um teste a partir da agenda, uma vez que os dados do Paciente já estão gravados no sistema.

4.4.3.1. Testes

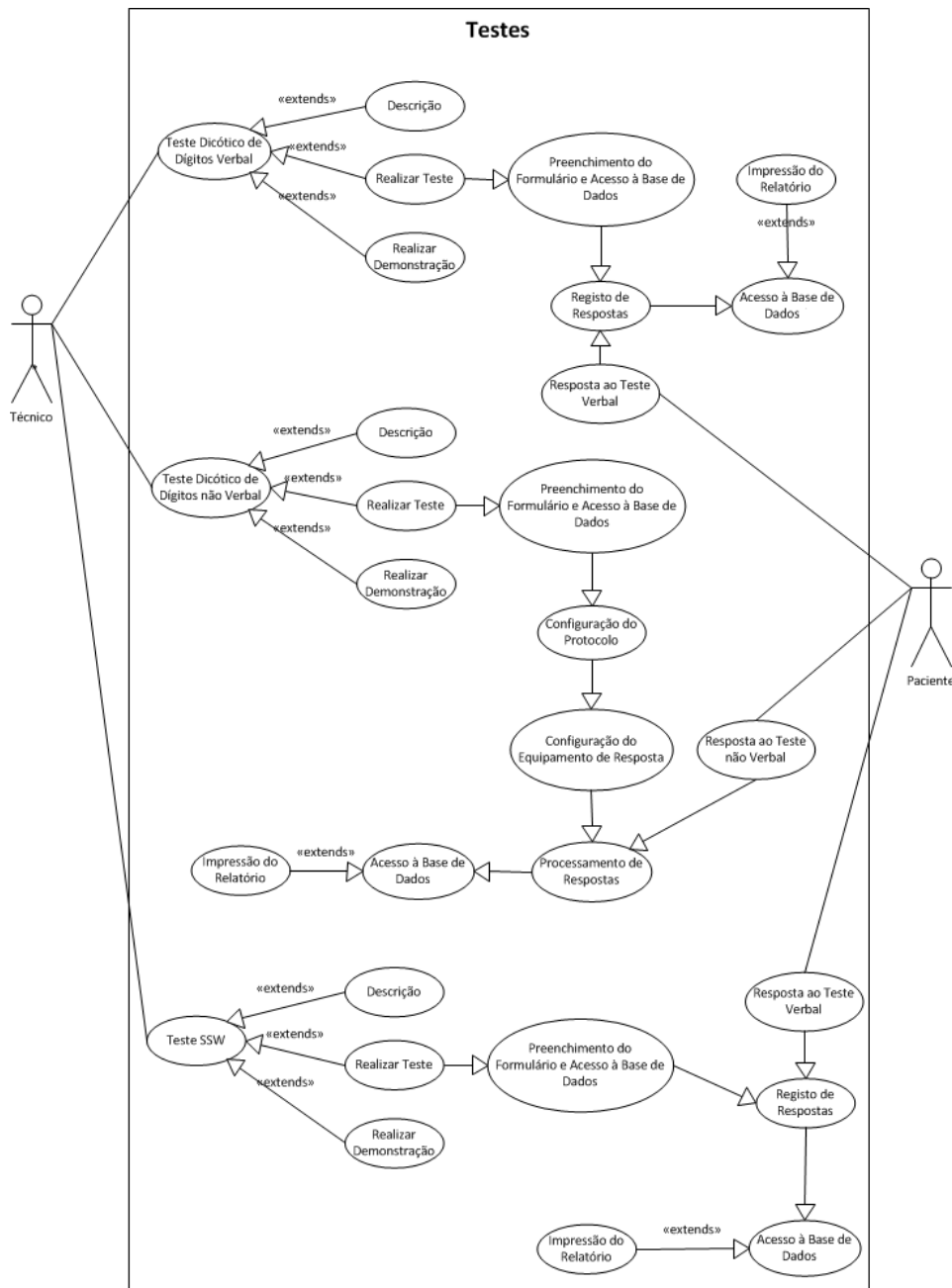


Figura 22 -Diagrama de casos de uso dos testes

Descrição dos casos de uso

Teste Dicótico de Dígitos Verbal – (Figura 22) Acesso ao painel com as opções de um teste dicótico de dígitos realizado verbalmente.

Descrição – Acesso à descrição sumária do funcionamento do teste.

Realizar Teste – Início do procedimento de realização de um teste.

Preenchimento do Formulário e Acesso à Base de Dados – (Figura 23) Inicialmente é efectuado um acesso à base de dados por forma a localizar o paciente que vai realizar o teste. Uma vez localizado é efectuado um acesso ao seu histórico para identificar todos os protocolos realizados. Toda a informação é apresentada ao técnico que escolhe o protocolo a aplicar ao Paciente. Após esta operação é efectuado novo acesso à base de dados para armazenar localmente de forma temporária o protocolo a ser aplicado.

The image shows a software interface window titled "Grupo de Investigação em Bioacústica em Sistemas". The main content area has a blue background and is titled "Formulário do Teste de Dicótico de Dígitos". It contains a form with the following fields and values:

- Cartão do Cidadão: [Dropdown menu]
- Nome: JOÃO CARLOS OLIVEIRA FERREIRA
- Data de Nascimento: 1987-03-24
- Protocolos: surval sa
- Protocolos realizados: [Dropdown menu]

At the bottom right of the form area, there are two buttons: "Pesquisar" (with a magnifying glass icon) and "Ok" (with a checkmark icon). Below these is a circular back arrow button. The status bar at the bottom of the window displays "Resultados de pesquisa" on the left and "18:50" on the right.

Figura 23 - Formulário de início de teste

Registo de Resposta – Marcação das respostas correctas e erradas dadas pelo Paciente.

Resposta Verbal ao Teste – Resposta à sequência ouvida

Acesso à Base de Dados 1 – No fim do teste é realizado um acesso à base de dados para adicionar os resultados ao histórico do paciente.

Impressão do Relatório – Caso o Técnico pretenda depois de concluído o teste pode imprimir um relatório em formato PDF com os dados do teste e com o respectivo resultado.

Realizar Demonstração – Reprodução de um teste sem carácter de avaliação, serve apenas para treinar o Paciente para a realização do teste.

Configuração do Protocolo – Configuração do protocolo de comunicação TCP/IP para realização de um teste dicótico de dígitos não verbal, a aplicação actua como servidor.

Configuração do Equipamento de Resposta – Configuração do protocolo de comunicação TCP/IP no equipamento com Android, que actua como cliente.

Processamento de Respostas – Processamento das respostas correctas e erradas de forma automática sem intervenção do Técnico.

Resposta ao Teste não Verbal – Resposta do Paciente no dispositivo com Android ao estímulo que ouviu.

Teste SSW – Acesso ao painel com as opções de um teste SSW.

4.4.4. Perfil de Utilizador

Este é o módulo que controla todas as funcionalidades de um Utilizador (Figura 24).

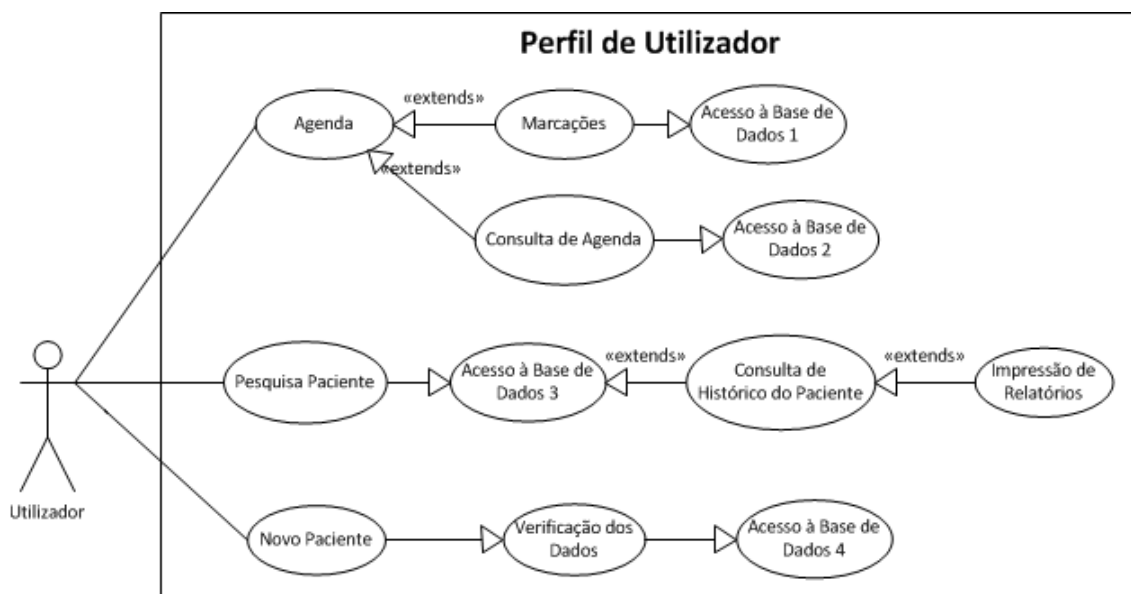


Figura 24 - Diagrama de casos de uso do perfil de Utilizador

Descrição dos casos de uso

Agenda – Acesso ao painel com as opções de agenda.

Marcações – O Utilizador pode efectuar marcações nas agendas de todos os técnicos.

Acesso à Base de Dados 1 – É efectuado um acesso à base de dados para registar a marcação do novo teste.

Consulta de Agenda – O Utilizador pode consultar a agenda de todos os Técnicos.

Acesso à Base de Dados 2 – É efectuado um acesso à base de dados que devolve a agenda pretendida pelo Utilizador.

Pesquisa de Paciente – O Utilizador introduz os dados dos Paciente que pretende pesquisar.

Acesso à Base de Dados 3 – É efectuado um acesso à base de dados por forma a localizar o Paciente com base nos dados introduzidos pelo Utilizador.

Consulta de Histórico de Paciente – Caso a pesquisa retorne resultado é possível ao Utilizador aceder ao histórico completo do Paciente.

Impressão de Relatórios Caso seja necessário o técnico pode imprimir um relatório com os resultados do teste em PDF.

Novo Paciente – (Figura 25) Preenchimento do formulário para adição de um novo Paciente ao sistema, o formulário pode ser preenchido de forma manual, ou com recurso ao cartão do cidadão.

Field	Value
Nome *	JOÃO CARLOS OLIVEIRA FERREIRA
Morada *	
Localidade *	
Código Postal *	0000-000
Data de Nascimento *	24-03-1987
Nº S.N.S. *	2839261
Nº BI / Nº CC *	132183
Contacto Telefónico *	
E-Mail	
Sexo *	Masculino

Figura 25 - Formulário de registo de um novo Paciente

Verificação de Dados – É efectuada uma verificação à integridade dos dados do novo paciente.

Acesso à Base de Dados – Caso a verificação anterior tenha sido concluída com sucesso é feito um acesso à base de dados para verificar se o Paciente já se encontra registado. Caso o resultado seja negativo é feito um novo acesso para gravar os dados do novo paciente.

4.5. Estrutura de dados

Durante os vários protótipos do SI foram definidos os dados que deveriam ser armazenados de forma permanente.

Definiu-se que seria necessário armazenar os dados pessoais de um Paciente, dados identificativos dos testes e dos respectivos tipos, histórico de testes realizados, utilizadores com permissões de acesso à aplicação, tipos de perfil de utilizadores, marcações de testes, definições de funcionamento da aplicação e parâmetros de calibração de som.

Para manipular e armazenar os dados gerados no SI optou-se por um modelo de dados relacional, em que todos os dados estão armazenados em tabelas e relacionados entre si.

Os dados foram normalizados até à terceira forma normal, por forma a permitir um armazenamento consistente e um acesso eficiente à base de dados, assim garante-se a diminuição de redundância e a inconsistência de dados.

Na definição do modelo de dados foi definido que protocolo e teste têm o mesmo significado.

Depois de várias interações obteve-se modelo físico da base de dados (Figura 26).

A representação do modelo está dividida por cores de forma a separar os grupos de dados.

As tabelas limitadas pela região verde armazenam os dados relativos aos utilizadores com acesso à aplicação, as tabelas limitadas pela região vermelha armazenam os dados relativos aos testes e à realização dos mesmos, as tabelas limitadas pela região amarela armazenam os dados relativos às configurações da aplicação.

A descrição detalhada das tabelas está disponível no Anexo A.

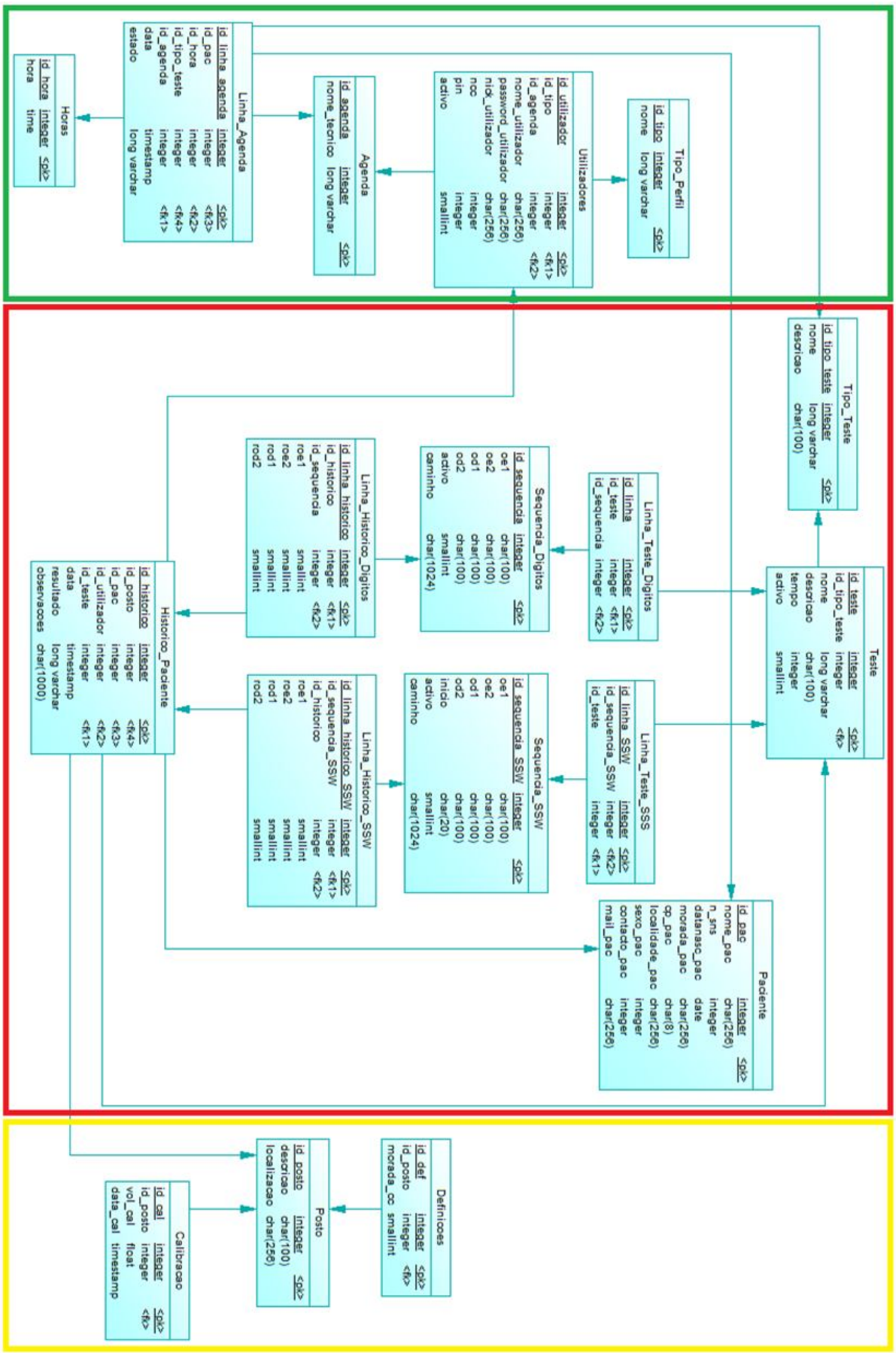


Figura 26 - Modelo físico da base de dados

4.6. Aspectos relevantes da implementação

4.6.1. Cartão do cidadão Português

O CC foi introduzido no SI por dois motivos, o primeiro prende-se com o facto de se eliminar o factor humano de erro no preenchimento de formulários, uma vez que os dados presentes no cartão são fidedignos e são os dados reais do paciente, desta forma garante-se que a base de dados possui dados consistentes identificativos do paciente. O segundo motivo prende-se com a rapidez de utilização e automatização de processos, com o CC é possível pesquisar directamente o paciente na base de dados, uma vez que o número do cartão é único é garantido que apenas é devolvido um resultado, no caso o resultado correcto sendo assim eliminado o tempo de filtragem numa lista de resultados.

O funcionamento do CC está sujeito à instalação do *software* do cartão e à utilização de leitores de cartões compatíveis, o requisito não funcional de desenvolver o SI com o mínimo custo associado manteve-se uma vez que não foi necessário realizar investimento na aquisição dos leitores de cartões.

Utilizou-se no desenvolvimento um modelo simples de leitor (Figura 27) *PC Twin Reader*.



Figura 27 - Leitor do cartão do cidadão

A implementação das funcionalidades de leitura de dados foi feita recorrendo à biblioteca *pteidlib.jar* disponível no SDK do CC.

O SDK destina-se a aplicações que utilizam o cartão do cidadão com o objectivo de ler apenas os dados identificativos.

São considerados dados identificativos os dados que estão impressos no cartão, juntamente com a morada. (Cidadão, 2010)

É possível aceder aos dados que estão impressos no CC de forma simples, no entanto para aceder à morada é necessário introduzir o PIN correspondente, este PIN é fornecido pela identidade gestora do CC.

Para contornar esta questão foram desenvolvidas duas formas de leitura de dados, uma simples e outra completa com morada incluída (Tabela 2).

Leitura simples	Leitura completa
<ul style="list-style-type: none">• Nome completo• Data de nascimento• N° S.N.S.• N° CC• Sexo	<ul style="list-style-type: none">• Nome completo• Data de nascimento• N° S.N.S.• N° CC• Sexo• Morada• Código Postal• Localidade

Tabela 2 - Leitura de dados do cartão do cidadão

A leitura completa dos dados é apenas utilizada no registo de um novo paciente, em todas as outras tarefas que envolvam o CC é utilizada a leitura simples.

Ao efectuar a leitura completa é aberta uma janela (Figura 28) para o portador do CC efectuar a introdução do seu PIN de morada, janela esta que faz parte da aplicação do CC.

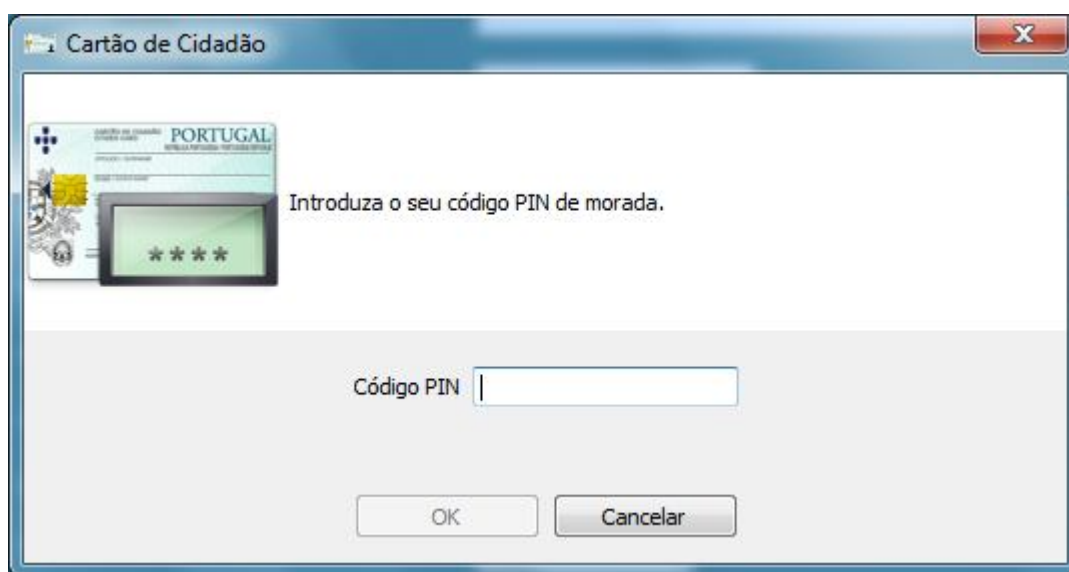


Figura 28 - Janela de pin da aplicação do cartão do cidadão

Por uma questão de segurança deve ser utilizado um leitor de cartões com teclado incorporado, como representado na Figura 29.



Figura 29 - Leitor do cartão do cidadão com pin pad

Caso o Paciente não saiba o PIN os dados serão lidos em modo simples.

Foi implementada uma funcionalidade a que apenas o administrador da aplicação tem acesso e que permite activar ou desactivar a leitura completa dos dados do CC.

Antes de cada leitura de dados do cartão é feita uma verificação para saber se o cartão se encontra no leitor, em caso de erro é apresentada uma mensagem de erro não crítica ao utilizador.

4.6.2. Realização de testes verbais

A realização de um teste implica várias consultas à base de dados, e respectivo armazenamento dos dados devolvidos de forma temporária na aplicação.

Antes de um teste ser iniciado é feito um acesso à base de dados, para armazenar em *Array Lists* temporários os sons a reproduzir e as respectivas posições de reprodução, desta forma não é necessário fazer acessos à base de dados durante a realização do teste.

Um *Array List* é uma estrutura dinâmica de dados, isto significa que é ideal para ser utilizada quando não se sabe na altura da implementação quantos elementos vão fazer parte da estrutura, associado contém um conjunto de métodos definidos pelo Java que permite em qualquer altura adicionar ou remover elementos da sem que o *Array* perca a sua consistência ou funcionalidade. (Coelho, 2010)

A solução foi implementada desta forma para evitar que um protocolo tenha de ser cancelado a meio por uma falha momentânea da base de dados, estando os sons guardados localmente no computador onde o teste é realizado, garante-se em caso de falha da base de dados que é possível levar o teste até ao fim, os *Array List* foram utilizados devido ao carácter dinâmico dos testes que permitem existência de protocolos com diferente numero de elementos.

Para a realização de testes verbais (dicótico de dígitos e SSW) foi desenvolvido um painel de tarefa que facilita ao máximo a interacção do audiologista para registar as respostas (Figura 30).

Foram definidos alguns pressupostos que se mantêm constantes ao longo de todo o teste.

O botão de tarefa simples representa a acção que se vai realizar sobre uma resposta, este botão alterna entre vermelho e verde, o vermelho corresponde à acção de marcar uma resposta errada e o verde corresponde à acção de marcar uma resposta correcta.

A cor em que são representados os números ou das palavras da sequência, representa o estado da resposta, vermelho significa que a resposta está errada e branco significa que a resposta está correcta.



Figura 30 - Realização de testes verbais

- 1- Respostas erradas
- 2- Respostas correctas

Inicialmente foi definido que todas as respostas se encontravam correctas no início da reprodução de uma sequência, tendo o audiologista apenas de marcar as respostas erradas.

Esta parecia ser a forma mais correcta de efectuar o registo, mas durante a utilização de um dos protótipos verificou-se que seria mais fácil inverter a lógica e considerar todas as respostas erradas no início da reprodução, tendo o audiologista de assinalar as respostas certas.

Esta alteração deveu-se ao facto de no primeiro caso ser necessário memorizar as respostas do Paciente para encontrar as respostas erradas, o que podia introduzir erros no resultado do final teste.

No segundo caso o audiologista não necessita de memorizar a sequência de respostas, apenas tem de marcar as respostas certas. Esta solução pode significar mais trabalho para o audiologista no caso de existirem mais respostas certas do que erradas, mas diminui a possibilidade de ocorrerem erros devido à memorização.

As respostas de um Paciente durante a realização do teste são armazenadas em *Array Lists* temporários.

Durante a realização de testes é ainda possível ao audiologista corrigir uma resposta que tenha assinalado mal inadvertidamente, ou repetir o som caso o Paciente solicite.

No caso de o Paciente abandonar o teste, é apresentado o resultado até ao momento do abandono.

Após a conclusão de um teste é feito um acesso à base de dados onde são armazenadas as respostas e o resultado do mesmo.

Caso a inserção dos dados seja feita com sucesso todos os dados temporários armazenados na aplicação são eliminados.

4.6.3. Realização de testes não-verbais

A realização de um teste não-verbal é feita com o auxílio de um equipamento móvel com Android e ecrã táctil.

Este foi um requisito funcional introduzido na aplicação depois de analisar algumas variantes do teste dicótico de dígitos, nomeadamente versões para pessoas que não conseguem falar.

Para a implementação de uma solução compatível com o SI, recorreu-se ao Protocolo TCP-IP e definiu-se o esquema de comunicação representado na Figura 31.

O TCP/IP é o principal protocolo de envio e recebimento de dados na Internet de forma simples é a linguagem utilizada para que dois equipamentos comuniquem entre si. É composto por várias camadas que tem funções específicas para manter o correcto endereçamento e para garantir a integridade dos dados durante a comunicação. (Boavida, 2010)



Figura 31 - Protocolo de comunicação

A ligação é criada pelo Técnico, que apenas necessita de configurar a porta de comunicação na aplicação, ficando esta em espera até a configuração no dispositivo móvel estar concluída.

A configuração do dispositivo móvel deve ser feita pelo técnico com o IP apresentado no SI e com a porta que definiu.

A aplicação móvel (Figura 32) contém um teclado numérico de grandes dimensões para o paciente efectuar as respostas e vibra cada vez que uma resposta é registada.

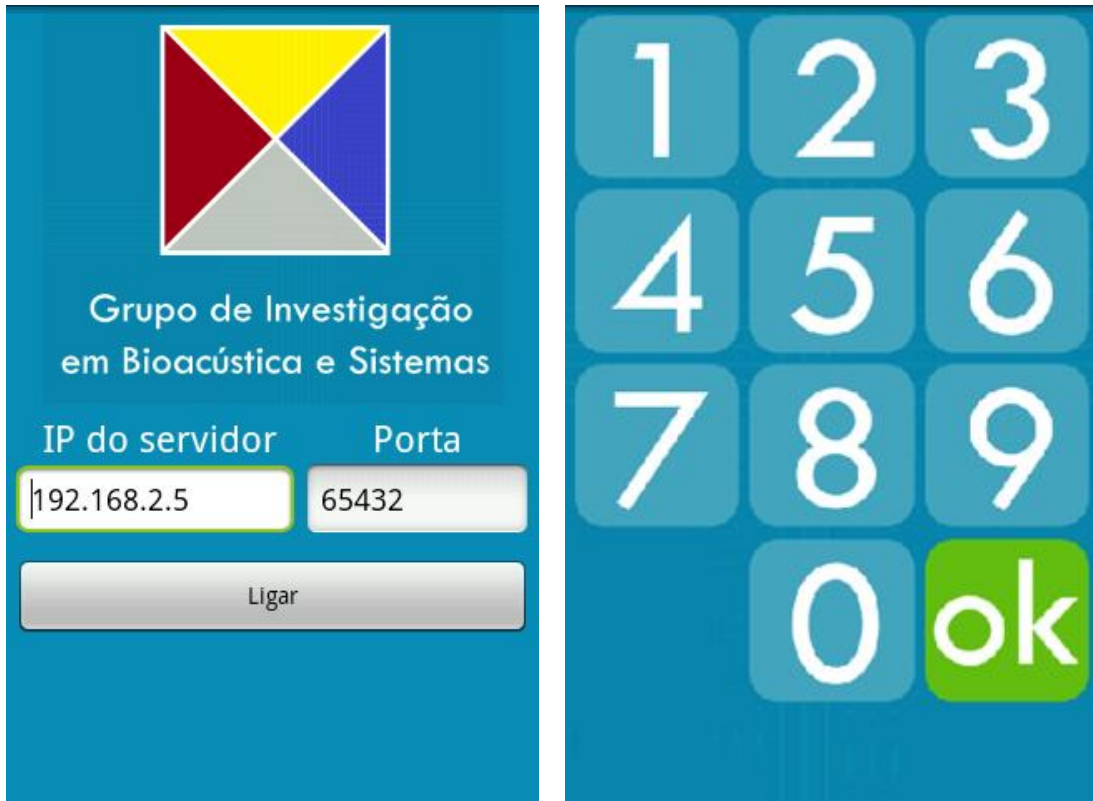


Figura 32 - Aplicação para dispositivo Android

Durante a realização de um teste o técnico não tem qualquer tipo de controlo sobre a realização do mesmo.

Todo o processamento dos dados recebidos da aplicação móvel é feito de forma automática.

A aplicação apenas foi desenvolvida para a realização de testes dicóticos de dígitos.

4.6.4. Reprodução de sons

A reprodução de sons no SI é feita recorrendo à API Java Sound, que possibilita o controlo dos ficheiros WAV reproduzidos.

A API é bastante completa, permitindo reprodução e captura de som em Midi e WAV a 16 bits.

A limitação de 16 bits que não é possível de ser contornada, causou alguns problemas durante o desenvolvimento, porque os sons de teste estavam a ser gravados a 24 bits. (Oracle, 2012)

No SI apenas são utilizadas as funcionalidades de reprodução de ficheiros WAV.

Todas as reproduções de som são feitas em *threads*, para evitar o bloqueio da aplicação, uma vez iniciada a reprodução de um ficheiro WAV a *thread* que está a efectuar a reprodução bloqueia a reprodução de outros ficheiros, por forma a evitar a reprodução de sons sobrepostos.

Uma *thread* é uma divisão do processo principal de uma aplicação para que várias tarefas possam ser executadas de forma concorrente. (Lea, 1999)

4.6.5. Adição de sons ao sistema

A aplicação permite a adição de novos ficheiros WAV ao sistema, para implementar esta funcionalidade foram definidos alguns protocolos por forma a tornar este procedimento o mais automático possível.

Para um preenchimento automático das características do ficheiro WAV, o nome do mesmo deve seguir os seguintes protocolos:

- 1- OE1_OE2_OD1_OD2.wav
- 2- E_OE1_OE2_OD1_OD2.wav
- 3- D_OD1_OD2_OE1_OE2.wav

O primeiro protocolo é utilizado para os sons do teste dicótico de dígitos, *OE1* e *OE2* representam por ordem os dígitos reproduzidos no ouvido esquerdo, *OD1* e *OD2* representam por ordem os dígitos reproduzidos no ouvido direito.

O segundo protocolo é utilizado para os sons do teste dicótico de dígitos, *E* significa que a sequência inicia a reprodução no ouvido esquerdo, *OE1* e *OE2* representam por ordem as palavras reproduzidas no ouvido esquerdo, *OD1* e *OD2* representam por ordem as palavras reproduzidas no ouvido direito.

O terceiro protocolo é utilizado também para os sons do teste dicótico de dígitos, *D* significa que a sequência inicia a reprodução no ouvido direito, *OD1* e *OD2* representam por ordem as palavras reproduzidas no ouvido direito, *OE1* e *OE2* representam por ordem as palavras reproduzidas no ouvido esquerdo.

O segundo e o terceiro protocolo são semelhantes mas a ordem de representação das palavras no ouvido esquerdo e no ouvido direito varia consoante o início da sequência.

Em todos os protocolos as características do ficheiro são separadas pelo carácter *underscore* (*_*).

Caso o nome dos ficheiros cumpra o protocolo estabelecido o preenchimento do formulário de um novo som é feito de forma automática, sendo que características do som são lidas directamente do nome.

No caso de o ficheiro não cumprir o protocolo estabelecido, o preenchimento do formulário tem de ser feito de forma manual.

Na inserção de um novo som (Figura 33) é sempre possível ao Administrador reproduzir o som para verificar de forma auditiva as suas características.



Figura 33 - Inserção de sons no sistema

- 1- Nome do ficheiro, com o formato do terceiro protocolo.
- 2- Características preenchidas de forma automática.

4.6.6. Gestão de sons no sistema

Ao adicionar um novo som ao sistema este fica associado ao posto e o ficheiro WAV é copiado para uma pasta local.

Na base de dados apenas ficam armazenadas as características do ficheiro e a sua localização.

Optou-se por esta solução para não sobrecarregar a base de dados e a rede no caso de estar a utilizar a base de dados externa, com os ficheiros WAV, desta forma garante-se a redução dos tempos de espera no início de um teste pois não é necessário efectuar o *download* dos sons.

Para evitar erros de execução durante um teste é feita uma verificação da existência dos ficheiros antes do início do mesmo.

Em caso de erro é apresentada uma mensagem de erro crítico na barra de estados e o protocolo e o som em falta são marcados na base de dados com inactivos, é também efectuada uma pesquisa de forma automática de todos os protocolos que utilizem o som e são também marcados como inactivos.

4.6.7. Adição de protocolos ao sistema

A adição de novos protocolos de dígitos ao SI pode ser feita de várias formas, foram definidos quatro modos para a construção de um novo protocolo.

- 1- Manual sem repetição
- 2- Manual com repetição
- 3- Semiautomático
- 4- Automático

No início da construção de um novo protocolo são lidos da base de dados todos os sons activos existentes no sistema e são armazenados temporariamente em *array lists*.

Manual sem repetição – Neste modo o Administrador pode construir um protocolo sem repetição das sequências de dígitos. Sempre que é adicionada uma sequência ao protocolo esta é removida do *array list* temporário e adicionada ao *array list* que armazena o novo protocolo antes de ser inserido na base de dados.

O método de construção de testes implementado permite que a qualquer momento, se possa remover uma sequência do *array list* do novo protocolo, sendo esta novamente adicionada ao *array list* temporário.

Manual com repetição – Neste modo o utilizador pode repetir as sequências no mesmo protocolo, tendo com única limitação o facto de não poder repetir duas vezes seguidas a mesma sequência. Desta forma as sequências nunca são removidas do *array list* temporário, apenas são adicionadas ou removidas do *array list* que armazena o novo protocolo.

Caso o utilizador tente adicionar duas sequências iguais seguidas, é apresentada uma mensagem de erro não crítico na barra de estados.

Semiautomático – Este modo segue as mesmas regras de construção que o modo manual com repetição.

É denominado por Semiautomático porque permite ao Administrador construir protocolos com padrões de sequências que podem ser repetidas. Apenas é necessário construir o padrão de sequências e introduzir o número de repetições pretendidas.

Os três primeiros modos de construção de protocolos partilham a mesma interface gráfica (Figura 34) que é composta por duas listas onde são apresentados os sons existentes no SI e os sons adicionados ao novo protocolo.

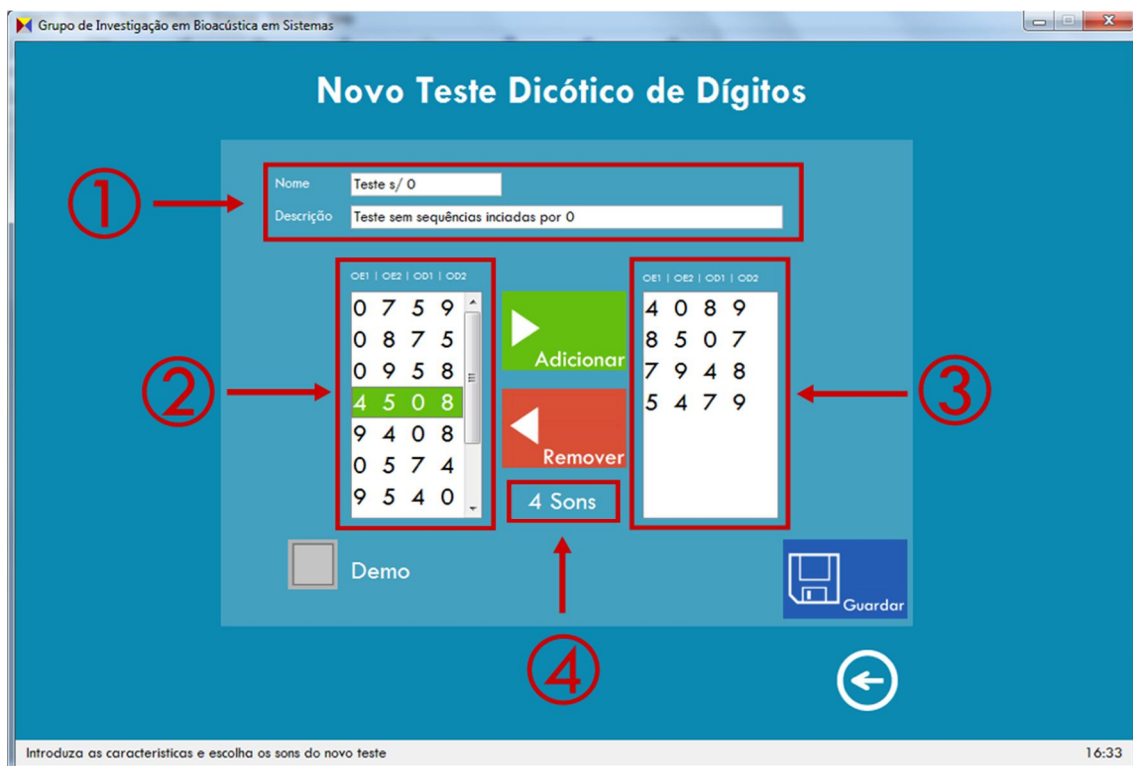


Figura 34 - Adição de novo protocolo de dígitos

- 1- Informação descritiva do novo protocolo;
- 2- Lista dos sons existentes no sistema;
- 3- Lista dos sons adicionados ao novo protocolo;
- 4- Número de sons adicionados ao novo protocolo;

Automático – Neste modo o Administrador apenas tem de introduzir o número de sequências e o SI gera de forma automática o protocolo.

Os protocolos gerados automaticamente ficam armazenados na base de dados com um nome na forma *Auto + n^o do protocolo*.

O limite máximo de sequências para um protocolo dicótico de dígitos independentemente da forma de construção é 30.

Qualquer protocolo pode ser construído como protocolo de demonstração, esta opção serve para definir protocolos que são utilizados para treinar os Pacientes antes do início de um teste real.

A adição de protocolos SSW é feita apenas de forma manual e sem repetição, optou-se apenas por implementar esta opção por o teste ter regras menos flexíveis que o teste dicótico de dígitos.

Para a construção de um novo protocolo SSW são utilizados três *array lists*, a leitura dos sons disponíveis no sistema é feita para dois *array lists* temporários, por forma a separar os sons que tem início no ouvido direito dos sons que tem início no ouvido esquerdo.

A lista dos sons existentes no sistema vai alternado o seu preenchimento conforme os sons vão sendo adicionados ao *array list* que armazena o novo protocolo.

A criação de um novo protocolo é concluída com a introdução do mesmo na base de dados, caso esta operação seja efectuada com sucesso todos os dados temporários são eliminados.

4.6.8. Formulário de realização de um teste

A realização de um teste inicia-se com o preenchimento de um formulário (Figura 35), que serve para identificar o Paciente e o teste que vai realizar.

Para efectuar o preenchimento deste formulário são feitas várias consultas à base de dados, por forma a identificar o Paciente, os protocolos que já realizou, e todos os protocolos do tipo de teste escolhido.

Os dados destas consultas ficam armazenados na aplicação até um teste ser terminado e o seu resultado ser introduzido na base de dados, sendo depois eliminados.

The screenshot shows a software application window titled "Grupo de Investigação em Bioacústica em Sistemas". The main content area has a blue background and is titled "Formulário do Teste de Dicótico de Dígitos". The form contains several input fields and buttons:

- A dropdown menu labeled "Nome" with a downward arrow.
- A text input field containing "João Carlos".
- A text label "Nome" followed by the value "JOÃO CARLOS OLIVEIRA FERREIRA".
- A text label "Data de Nascimento" followed by the value "1987-03-24".
- A dropdown menu labeled "Protocolos" with the value "AUTO 2" and a downward arrow.
- A dropdown menu labeled "Protocolos realizados" with a downward arrow.
- An orange button with a magnifying glass icon and the text "Pesquisar".
- A green button with a checkmark icon and the text "Ok".
- A white circular button with a left-pointing arrow.

At the bottom of the window, there is a status bar with the text "Resultados de pesquisa" on the left and "18:41" on the right.

Figura 35 - Formulário de realização de um teste

4.6.9. Criação de relatórios

Os relatórios são utilizados para apresentar os resultados da realização de um teste. No caso do teste dicótico de dígitos servem também para apresentar as respostas certas e erradas.

Para a implementação desta funcionalidade foi utilizada a biblioteca *itextpdf.jar*.

O *iText* é uma biblioteca que permite a criação de *PDF's* de forma automática, a partir de componentes gráficos do Java.

O *iText* é um software livre e *open source*, é possível ser utilizado desde que se cumpram as regras do *Affero General Public License* (*iText - Normas de utilização*), neste momento todas as regras são cumpridas e o uso da biblioteca é legal, caso existam alterações na forma com o SI é distribuído, deverá ser equacionada a remoção desta funcionalidade da aplicação ou a aquisição de uma licença do *iText* que liberta o SI das regras do *Affero General Puublic License*.

Os relatórios (Figura 36) são construídos a partir de um *JPanel* que contem vários *JLabels* onde é colocada a informação, sendo depois convertido pelo *iText* para *PDF*.

Os componentes *JPanel* e *JLabel* foram introduzidos no Java na versão 1.2 e fazem parte do swing. O swing é o agregador de todos os componentes gráficos que são apresentados no ecrã de uma aplicação Java. O *Jlabel* é o elemento nativo que permite mostrar texto e o *Jpanel* é o elemento agregador que pode conter vários elementos de outros tipos. (Coelho, 2010)

Grupo de Investigação em Bioacústica e Sistemas

Data: 12-07-2012 - 11:43
Nome: João Carlos Oliveira Ferreira
Dicótico de Dígitos
Teste: curval sa

Observações: Teste verbal
Resultado: Erros OE - 6/8 | Erros OD - 4/8

0 7 5 9
0 8 7 5
0 7 5 9
0 8 7 5

Teste realizado por: João Ferreira
Posto: 1

1
2
3

Figura 36 - Relatório de resultados de teste dicótico de dígitos

- 1- Informação do Paciente e do teste
- 2- Resultados e respostas do teste
- 3- Informação do posto e do Técnico

Este processo é feito de forma transparente para o utilizador da aplicação que nunca tem acesso ao *JPanel* de construção do relatório.

Esta funcionalidade implica que esteja instalado no sistema um *software* leitor de *PDF's* por forma a ser possível a visualização do documento.

Um relatório é sempre aberto no *software* instalado de leitura de *PDF's* após ordem de impressão.

4.6.10. Gestão de relatórios

Todos os relatórios impressos em PDF são armazenados de forma temporária no terminal onde foi dada ordem de impressão.

Um relatório é gerado com um nome único, para gerar este nome recorre-se à data actual e ao relógio do sistema.

Uma vez que não existe necessidade de manter estes ficheiros armazenados, sempre que a aplicação é encerrada, os relatórios existentes na pasta temporária são apagados de forma automática.

4.6.11. Gestão do interface gráfico

Existe uma classe na aplicação responsável por gerir o interface gráfico, o esquema de navegação é feito em profundidade. À medida que o utilizador avança na aplicação vai aumentando o nível de profundidade de navegação.

Uma classe em programação é uma estrutura responsável por agregar vários objectos semelhantes, neste caso os painéis gráficos. Uma classe é responsável ainda por definir um conjunto de métodos capazes de efectuar as operações sobre os objectos, neste caso a classe é responsável por criar, armazenar e eliminar o painel gráfico e ainda capaz de gerir a estrutura de navegação. (Martins, 2003)

Em cada nível de profundidade é invocado o método de criação para adição de um novo painel gráfico.

Quando o utilizador retrocede um nível na profundidade de navegação o painel do nível superior é destruído por forma a libertar memória.

5. Avaliações do Sistema de Informação

5.1. Compatibilidade da aplicação

A aplicação desenvolvida cumpre o requisito não funcional de ser independente do sistema operativo utilizado. Foram realizados testes nos principais sistemas operativos utilizados a nível mundial, Windows XP, Windows Vista, Windows 7, Windows 8, Mac OS X e Linux Ubuntu 11.10. (Stats, 2012)

Os testes realizados incidiram apenas sobre a interface gráfica e a usabilidade da aplicação.

Nos sistemas Windows a compatibilidade da aplicação é de 100%, não tendo sido detectado nenhum problema de funcionamento desde o Windows XP ao Windows 8.

Na Figura 37 - Aplicação em funcionamento em Windows XP, Vista, 7 e 8 é apresentada a aplicação em funcionamento nas várias versões do Windows, no canto superior esquerdo Windows XP, no canto superior direito Windows Vista, no canto inferior esquerdo Windows 7, no canto inferior direito Windows 8.

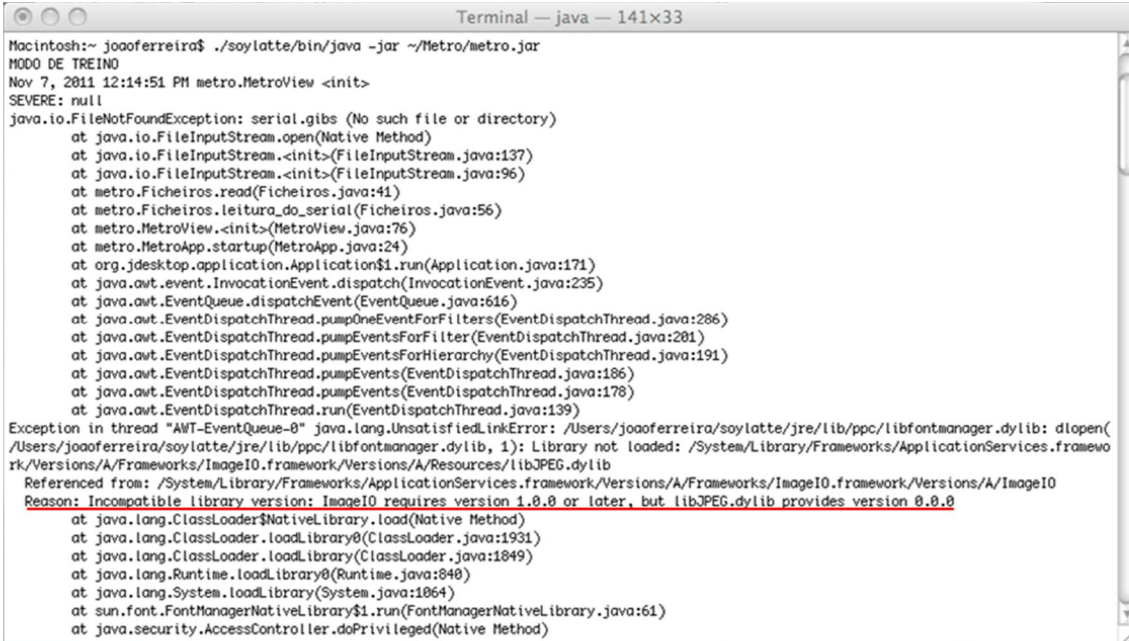


Figura 37 - Aplicação em funcionamento em Windows XP, Vista, 7 e 8

Em Mac OS X o funcionamento da aplicação está dependente do processador do equipamento, em equipamentos produzidos até 2005 com processadores *Power PC* a aplicação não corre

devido à falta de actualização do Java. Uma vez que foi utilizado o Java 7 para desenvolvimento a aplicação apenas funciona em equipamentos posteriores a 2005 e com processador intel x86 para os quais foi desenvolvida a mais recente versão do Java. Desta forma a aplicação é apenas compatível com Mac OS 10.6 *Snow Leopard*, Mac OS 10.7 *Lion* e Mac OS 10.8 *Mountain Lion*.

Apenas foi possível efectuar testes na versão de Mac OS 10.5 *Leopard* (Figura 38 - Teste em Mac OS 10.5 Leopard com erro) e Mac OS 10.8 *Mountain Lion* (Figura 39), na versão 10.5 não foi possível executar a aplicação, na versão 10.8 a aplicação funcionou normalmente apenas foram detectadas pequenas falhas no alinhamento dos botões.



```
Terminal — java — 141x33
Macintosh:~ jooferreira$ ./soylatte/bin/java -jar ~/Metro/metro.jar
MODO DE TREINO
Nov 7, 2011 12:14:51 PM metro.MetroView <init>
SEVERE: null
java.io.FileNotFoundException: serial.gibs (No such file or directory)
  at java.io.FileInputStream.open(Native Method)
  at java.io.FileInputStream.<init>(FileInputStream.java:137)
  at java.io.FileInputStream.<init>(FileInputStream.java:96)
  at metro.Ficheiros.read(Ficheiros.java:41)
  at metro.Ficheiros.leitura_do_serial(Ficheiros.java:56)
  at metro.MetroView.<init>(MetroView.java:76)
  at metro.MetroApp.startup(MetroApp.java:24)
  at org.jdesktop.application.Application$1.run(Application.java:171)
  at java.awt.event.InvocationEvent.dispatch(InvocationEvent.java:235)
  at java.awt.EventQueue.dispatchEvent(EventQueue.java:616)
  at java.awt.EventDispatchThread.pumpOneEventForFilters(EventDispatchThread.java:286)
  at java.awt.EventDispatchThread.pumpEventsForFilter(EventDispatchThread.java:281)
  at java.awt.EventDispatchThread.pumpEventsForHierarchy(EventDispatchThread.java:191)
  at java.awt.EventDispatchThread.pumpEvents(EventDispatchThread.java:186)
  at java.awt.EventDispatchThread.pumpEvents(EventDispatchThread.java:178)
  at java.awt.EventDispatchThread.run(EventDispatchThread.java:139)
Exception in thread "AWT-EventQueue-0" java.lang.UnsatisfiedLinkError: /Users/jooferreira/soylatte/jre/lib/ppc/libfontmanager.dylib: dlopen(
/Users/jooferreira/soylatte/jre/lib/ppc/libfontmanager.dylib, 1): Library not loaded: /System/Library/Frameworks/ApplicationServices.framework
/Versions/A/Frameworks/ImageIO.framework/Versions/A/Resources/libJPEG.dylib
Referenced from: /System/Library/Frameworks/ApplicationServices.framework/Versions/A/Frameworks/ImageIO.framework/Versions/A/ImageIO
Reason: Incompatible library version: ImageIO requires version 1.0.0 or later, but libJPEG.dylib provides version 0.0.0
  at java.lang.ClassLoader$NativeLibrary.load(Native Method)
  at java.lang.ClassLoader.loadLibrary0(ClassLoader.java:1931)
  at java.lang.ClassLoader.loadLibrary(ClassLoader.java:1849)
  at java.lang.Runtime.loadLibrary0(Runtime.java:848)
  at java.lang.System.loadLibrary(System.java:1864)
  at sun.font.FontManagerNativeLibrary$1.run(FontManagerNativeLibrary.java:61)
  at java.security.AccessController.doPrivileged(Native Method)
```

Figura 38 - Teste em Mac OS 10.5 Leopard com erro

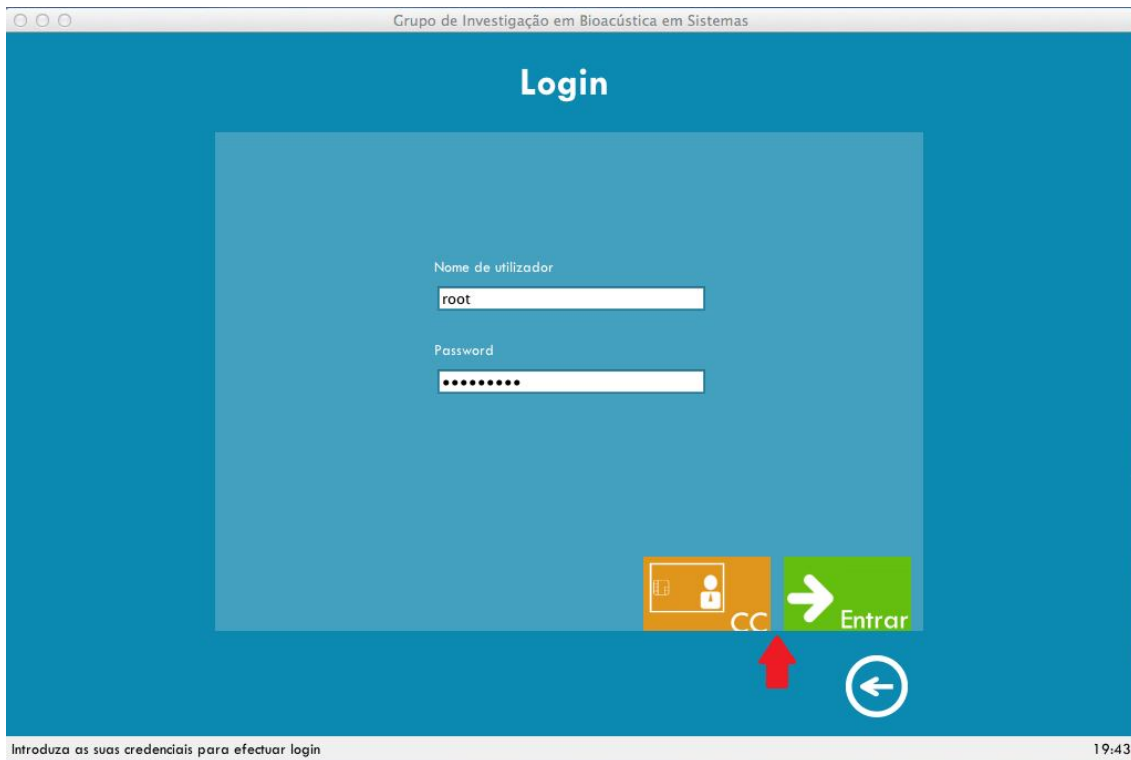


Figura 39 - Teste em Mac OS 10.8 Mountain Lion com falha

Os testes em Linux foram realizados na distribuição Ubuntu versão 11.10, a aplicação funciona sem problemas e apenas foram detectados pequenos desalinhamentos dos botões à semelhança do que acontece em Mac OS (Figura 40).

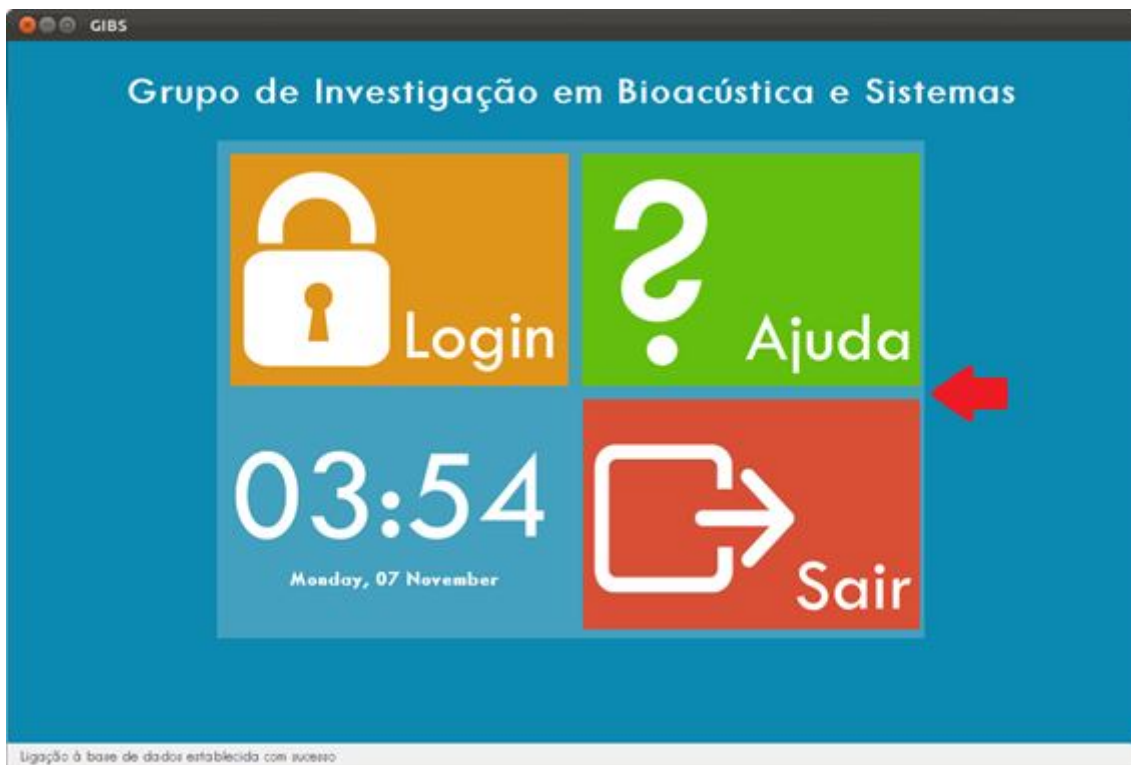


Figura 40 - Teste em Linux 11.10 com falha

Através dos testes efectuados é possível afirmar que a aplicação é compatível com sistemas operativos Windows, em sistemas com base Unix como o caso do Mac OS e do Linux Ubuntu, serão necessários fazer alguns ajustes gráficos. A Figura 41 representa a compatibilidade da aplicação por sistema operativo.



Figura 41 - Compatibilidade por Sistema Operativo

5.2. Requisitos de Sistema

Os requisitos mínimos da aplicação foram definidos com base nos requisitos mínimos do Software que suporta o funcionamento do SI.

Para execução do Java são necessários os seguintes requisitos mínimos (Tabela 3): (Java, 2012)

Processador	Sistema Operativo	Memória RAM	Espaço em Disco
X86 ou X64 (32 ou 64 bits)	Windows XP ou superior	128 MB	124MB
	Mac OS 10.5.8 ou superior	64MB	58MB
	Linux (versões posteriores a 2007)	64MB	68MB

Tabela 3 - Requisitos mínimos de sistema Java

Para execução do motor de base de dados Postgres SQL são necessários os seguintes requisitos mínimos (Tabela 4): (Postgres Install Requirements, 2011)

Processador	Sistema Operativo	Memória RAM	Espaço em Disco
X64 (64 bits) Dual Core	Windows XP ou superior	2GB	100MB
	Linux		

Tabela 4 - Requisitos mínimos de sistema Postgres

Para utilização do cartão do cidadão português são necessários os seguintes requisitos mínimos (Tabela 5): (Cidadão, 2012)

Processador	Sistema Operativo	Memória RAM	Espaço em Disco
X86 ou X64 (32 ou 64)	Windows XP ou	128 MB	33MB

bits)	superior		
	Mac OS 10.5.8 ou superior	128MB	33MB
	Linux (versões posteriores a 2007)	128MB	33MB

Tabela 5 - Requisitos mínimos de sistema CC

Para utilização da aplicação móvel é necessário um dispositivo com Android 2.1 e 1MB de espaço livre no equipamento, é também necessária uma rede wireless, pois é por este meio que é efectuada a comunicação de dados.

Com base nos dados apresentados a tabela seguinte representa os requisitos mínimos para o funcionamento do sistema. O SI pode ser instalado em duas configurações distintas como tal existem dois requisitos mínimos distintos.

Para sistemas que utilizem a base de dados local os requisitos mínimos estão apresentados na Tabela 6.

Processador	x64 (64 bits) Dual-Core
Sistema Operativo	Windows XP ou Superior
	Linux (Distribuição posterior a 2007)
Memória Ram	2GB
Espaço em Disco	120GB
Ligação à Internet	Opcional
Ligação a rede Wireless	Opcional
Equipamento com Android	Opcional
Leitor de Carões	Opcional

Tabela 6 -Requisitos mínimos de sistema SI com base de dados local

Para SI que apenas utilizem a base de dados externa os requisitos mínimos são apresentados na Tabela 7.

Processador	X86 ou x64 (32 ou 64 bits) Dual-Core
Sistema Operativo	Windows XP ou Superior
	Mac OS 10.5.8 ou Superior
	Linux (Distribuição posterior a 2007)
Memória Ram	2GB
Espaço em Disco	120GB
Ligação à Internet	Obrigatório
Ligação a rede Wireless	Opcional
Equipamento com Android	Opcional
Leitor de Carões	Opcional

Tabela 7 - Requisitos mínimos de sistema SI com base de dados externo

6. Conclusões e Trabalhos Futuros

Ao desenvolver um SI para realizar testes audiológicos foi possível perceber que ainda existe muito trabalho a desenvolver para adaptar as TI a esta área da Saúde, para que possam ser substituídos os processos manuais.

O desenvolvimento de um SI para a realização de testes audiológicos introduz uma melhoria prática clínica na Audiologia, a falta de soluções deste género em Portugal levou o GIBS a realizar este trabalho que se conclui com a versão actual do sistema apresentada ao longo da dissertação.

A metodologia de desenvolvimento escolhida revelou-se a mais adequada ao longo do processo pois permitiu testar várias soluções distintas e efectuar as correcções necessárias aos requisitos funcionais.

Os objectivos propostos inicialmente foram concretizados, os requisitos foram cumpridos e foi possível desenvolver um SI funcional.

Este trabalho foi um grande desafio pois foi necessário enquadrar várias tecnologias diferentes no mesmo SI e fazer com que todas comunicassem entre si de forma segura sem introduzirem erros na aplicação.

O SI desenvolvido encontra-se num estado de maturação que permite a sua utilização para a realização de testes contudo o carácter vinculativo dos mesmos não deve ser levado em linha de conta pois o software em questão não foi sujeito a nenhuma avaliação de desempenho e de qualidade que permita garantir a qualidade do teste audiológico.

Recorrendo à terminologia utilizada pela IBM nos anos 50 para classificar o estado de maturação dos seus produtos, e que se estendeu a toda a indústria de software a versão actual encontra-se em Pré-alpha. (Patton, 2005)

Significa que todas as actividades de análise de requisitos, desenvolvimento de software e teste dos componentes desenvolvidos foram concluídas, estado pronto para se submeter a uma bateria intensiva de testes que possa identificar as falhas para que a aplicação possa chegar à fase Alpha.

O passo seguinte e que se enquadra nos desenvolvimentos futuros deste projecto é iniciar a validação do conteúdo produzido através de testes de caixa branca e caixa preta.

Os testes de caixa branca focam-se no código fonte da aplicação e são orientados à lógica pretendem avaliar o fluxo de dados, condições lógicas, ciclos e encontrar código fonte que devido a lógica errada nunca é executado. (Patton, 2005)

Para utilização desta técnica poderá ser utilizada a ferramenta gratuita JUnit que permite desenvolver cenários de teste, para avaliar o código fonte produzido em Java. (JUnit, 2013)

Os testes de caixa preta avaliam o comportamento externo do sistema sem preocupação da forma como este foi implementado. Um exemplo típico deste teste é fornecer dados de entrada à aplicação e comparar os resultados de saída com resultados previamente conhecidos, quanto mais entradas forem fornecidas e quanto maior for o número de resultados conhecidos mais eficaz será o teste, sendo o cenário ideal testar todas as combinações possíveis. (Patton, 2005)

O teste de caixa preta tem como base os requisitos funcionais definidos na especificação inicial.

Concluídos estes testes o software deverá atingir o estado beta. Esta fase inicia-se quando todos os requisitos estão implementados e testados. O software em estado beta é software ainda com anomalias e deverá ser sujeito a um conjunto de testes que permita reduzir um possível impacto negativo nos utilizadores.

Os testes nesta fase são desenvolvidos por pessoas externas ao projecto e que pela sua actividade virão a ser os utilizadores finais da aplicação.

Tipicamente nestes testes são identificadas falhas a nível gráfico que dificultam ou impossibilitam a utilização do software.

No caso específico deste software deverá ser distribuído a vários audiologistas a versão beta da aplicação para que possa ser testada, juntamente com a aplicação deverão ser fornecidas as ferramentas necessários para que eventuais falhas possam ser reportadas.

Após as correcções de eventuais falhas detectadas na fase beta de testes o Software passará à versão RTM que significa *release to manufacturing*, esta será a versão destinada a uma utilização profissional.

Findos os testes e atingida a maturidade necessária para uma utilização isenta de falhas deverá ser feito o empacotamento da solução e de todos o software externo que utiliza, para que possa ser instalada recorrendo a instalador automático sem necessidade de configurações manuais, no estado actual em que o software se encontra todo o processo de instalação é manual. (Wright, 2011)

A Audiologia não se esgota nos tipos de testes implementados neste software e a longo prazo o SI poderá ganhar mais valor se forem adicionados novos tipos de testes audiológicos. Antes desta introdução será importante chegar a uma versão final das especificações actuais antes da inclusão de novos recursos na aplicação por forma a evitar que esta entre em ciclo e nunca passe do estado de beta devido à necessidade constante de efectuar testes de validação.

Durante o tempo de vida útil do SI deverá ser mantido o suporte ao mesmo por forma a garantir que tudo continua funcional e para efectuar as alterações necessárias causadas pela evolução das TI.

Foi para mim muito gratificante e enriquecedor a realização deste trabalho, quer a nível de conhecimento ganho na área da Audiologia, quer a nível técnico que me permitiu as capacidades de desenvolvimento em Java.

Resta-me concluir que mantenho toda a disponibilidade para ajudar a levar este projecto avante, para que possa chegar a uma primeira versão de produção que possa ser utilizada em todo o país.

7. Bibliografia

- Boavida, Fernando. 2010.** *TCP/IP - Teoria e Prática*. s.l. : FCA, 2010. 978-972-722-745-7.
- Burns, Kyle. 2012.** *Beginning Windows 8 Application Development – XAML Edition*. s.l. : Apress, 2012. 978-1430245667.
- Cidadão, Cartão do. 2012.** Cartão do Cidadão - Software do Cartão de Cidadão . [Online] 2012. [Citação: 04 de Dezembro de 2012.]
http://www.cartaodecidadao.pt/index.php?option=com_content&task=view&id=102&Itemid=44&lang=pt.
- Cidadão, Portal do. 2010.** Cidadão, Documentação do Cartão do Cidadão. 2010.
- Cinar, Onur. 2012.** *Android Apps with Eclipse*. s.l. : Apress, 2012. 978-1430244349.
- Coelho, Pedro. 2010.** *Programação em Java*. s.l. : FCA, 2010. 978-972-722-361-9.
- Filomena Castro Lopes, Maria Paula Morais, Armando Jorge Carvalho. 2009.** *Desenvolvimento de Sistemas de Informação*. Portugal : FCA, 2009.
- iText - Normas de utilização.** iText - Normas de utilização. *iText - Normas de utilização*. [Online] iText. [Citação: 12 de Julho de 2012.] <http://itextpdf.com/terms-of-use/agpl.php>.
- Java. 2012.** Java - What are the system requirements for Java 7? . *Java*. [Online] 2012. [Citação: 04 de Dezembro de 2012.]
- JUnit. 2013.** JUnit. *JUnit*. [Online] JUnit, 2013. [Citação: 01 de 10 de 2013.] <http://junit.org/>.
- Kimura, Doreen. 1967.** *Functional Asymmetry of the Brain in Dichotic Listening*. s.l. : Elsevier, 1967.
- Lea, Doug. 1999.** *Concurrent Programming in Java: Design Principles and Pattern*. s.l. : Sun, 1999. 978-0201310092.
- Lemos, Isabel Cristina Cavalcanti. 2008.** Teste Dicótico de Dígitos (etapa de escuta direcionada) em crianças com fissura labiopalatina. *Revista Brasileira de Otorrinolaringologia*. 74, 2008, Vol. 5.
- Martins, Elsa Milena do Carmo. 2008.** *Criação de um conjunto de testes para a avaliação do processamento auditivo*. Aveiro : UA, 2008.
- Martins, F. Mário. 2003.** *Programação Orientada aos Objectos em JAVA*. s.l. : FCA, 2003. 978-972-722-196-7.
- Martins, Ferreira. 2007.** *Testes de avaliação do processamento auditivo central*. Aveiro : Universidade de Aveiro, 2007.

- Ministério da Saúde. 2010.** Portal da Saúde. [Online] 2010. [Citação: 12 de Novembro de 2012.] <http://www.min-saude.pt/portal/conteudos/servicos+online/eagenda.htm>.
- Murphy, Mark. 2009.** *Beginning Android*. s.l. : Apress, 2009. 978-1-4302-2419-8 .
- O'Brien, James. 2003.** *Introduction to Information Systems: Essentials for the Internetworked e-Business Enterprise*. s.l. : McGraw Hill, 2003.
- Oracle. 2012.** Java Sound API. *Java Sound API*. [Online] Oracle, 2012. <http://www.oracle.com/technetwork/java/index-jsp-140234.html>.
- Patton, Ron. 2005.** *Software Testing*. s.l. : SAMS, 2005. 978-067232798.
- Paulo Gomes, Nuno Paiva e Bernardo Simões. 2009.** *Análise da Viabilidade Económica das aplicações SAM e SAPE*. s.l. : Universidade Nova de Lisboa, 2009.
- Pfeiffer, Marcela. 2007.** *Processamento Auditivo e Potenciais Evocados de Tronco Cerebral* . Rio de Janeiro : Universidade Veiga de Almeida, 2007.
- Postgres Install Requirements. 2011.** Postgres Install Requirements. *Postgres Install Requirements*. [Online] 2011. [Citação: 2012 de Dezembro de 04.] <http://www.postgresql.org/download/>.
- Postgres. 2012.** Postgres. [Online] 2012. [Citação: 22 de Agosto de 2012.]
- PubMed.gov. 1983.** PubMed.gov. *PubMed.gov*. [Online] Keith RW, 1983. <http://www.ncbi.nlm.nih.gov/pubmed/6653932>.
- Rosenberg, Doug. 2007.** *Use Case Driven Object Modeling with UML Theory and Practice*. s.l. : Apress, 2007. 978-1590597743.
- Schooley, Brent. 2012.** *Designing for Windows 8: Fundamentals of Great Design in Windows Store Apps*. s.l. : Apress, 2012. 1430249595.
- Selecting a Development Approach*. **Office of Information Services. 2008.** 2008.
- Sistemas de Informação em Saúde e Saúde Online*. **Espanha, Rita. 2010.** 2010.
- Stats, Global. 2012.** Global Stats OS. [Online] 2012. [Citação: 2012 de 12 de 02.] <http://gs.statcounter.com/#os-ww-monthly-201111-201211>.
- Teixeira, Leonor da Conceição. 2008.** *Contribuições para o Desenvolvimento de Sistemas de Informação na Saúde: Aplicação na área da Hemofilia* . Aveiro : Universidade de Aveiro, 2008.
- The Computer Society (IEEE-CS). 2005.** *Computing Curricula 2005*. 2005. 1-59593-359-X.
- Timothy N. Welsh, Digby Elliott. 2000.** *Gender differences in a dichotic listening and movement task*. s.l. : Department of Kinesiology, McMaster University, 2000.
- Varajão, Luís Amaral e João. 2010.** Planeamento de Sistemas de Informação. *Planeamento de Sistemas de Informação*. s.l. : FCA, 2010.

Videira, Alberto Silva e Carlos. 2005. *UML Metodologias e Ferramentas CASE Vol.1.* s.l. : Centro Atlantico, 2005.

Warren, Richard M. 2010. *Auditory Perception An Analysis and Synthesis.* s.l. : Cambridge University Press, 2010. 9780511754777.

West, Matthew. 1996. *Developing High Quality Data Models.* London : Julian Fowler, 1996.

Wright, David. 2011. *Software Lifecycle Management Standards: Real-World Scenarios & Solutions for Savings.* s.l. : IT Governance , 2011. 978-1849282048.

Anexo A – Descrição das tabelas da base de dados

Um utilizador é caracterizado por:

- id_utilizador – identificador único utilizado como chave primária, obrigatório;
- id_tipo – tipo de perfil associado ao utilizador, obrigatório;
- nome_utilizador – nome completo do utilizador, obrigatório;
- password_utilizador – password de acesso à aplicação, obrigatório;
- nick_utilizador – nome de utilizador utilizado para autenticação na aplicação, obrigatório;
- ncc – número do CC utilizado para autenticação na aplicação, opcional;
- pin – código de quatro dígitos para autenticação na aplicação, no modo de autenticação com CC, opcional;
- activo – estado do perfil, apenas pode ter dois valores, verdadeiro ou falso que identificam se o utilizador se encontra activo ou inactivo, obrigatório;

Um tipo de perfil é caracterizado por:

- id_tipo – identificador único utilizado como chave primária, obrigatório;
- nome – descrição do tipo de perfil, obrigatório;

Uma agenda é caracterizada por:

- id_agenda – identificador único utilizado como chave primária, obrigatório;
- id_utilizador – identificador do utilizador a quem está associada a agenda, obrigatório;
- nome_utilizador – nome dado à agenda, obrigatório;

Cada marcação de uma agenda é armazenado na tabela linha_agenda que é caracterizada por:

- id_linha_agenda – identificador único utilizado como chave primária, obrigatório;
- id_pac – identificador do paciente, para o qual é efectuada a marcação, obrigatório;
- id_hora – identificador da hora para a qual é marcado o teste, obrigatório;
- id_tipo_teste – identificador do tipo de teste a marcar, obrigatório;
- id_agenda – identificador da agenda que está associado a um determinado Técnico, obrigatório;
- data – data da marcação do teste, obrigatório;
- estado – estado da marcação, pode assumir os valores, activo, cancelado ou realizado, obrigatório;

Os dados de cada paciente registado no SI, são armazenados na tabela paciente, caracterizada por:

- id_pac – identificador único utilizado como chave primária, obrigatório;

- nome_pac – nome completo do paciente, obrigatório;
- n_sns – número do serviço nacional de saúde (apenas para pacientes possuidores de um número do serviço nacional de saúde), opcional;
- n_cc – número do bilhete de identidade ou cartão do cidadão (apenas para pacientes portugueses possuidores de documento de identificação), opcional;
- data_nasc_pac – data de nascimento do paciente, obrigatório;
- morada_pac – morada do paciente, obrigatório;
- cp_pac – código postal do paciente, obrigatório;
- localidade_pac – localidade do paciente, obrigatório;
- sexo_pac – sexo do paciente, obrigatório;
- contacto_pac – contacto telefónico do paciente, opcional;
- mail_pac – endereço de correio electrónico, opcional;

A estrutura de dados foi pensada de forma a permitir a adição de novos tipos de testes no futuro, por forma a tornar esta adição mais simples foi criada a tabela tipo_teste que é caracterizada por:

- id_tipo_teste – identificador único utilizado como chave primária, obrigatório;
- nome – nome do tipo de teste, obrigatório;
- descrição – breve descrição do tipo de teste, obrigatório;

Um teste é armazenado na tabela teste e é caracterizado por:

- id_teste – identificador único utilizado como chave primária, obrigatório;
- id_tipo_teste – identificador do tipo de teste, obrigatório;
- nome – nome do teste, obrigatório;
- descrição – breve descrição do teste, obrigatório;
- tempo – intervalo de tempo existente entre a reprodução das sequências, obrigatório;
- activo – estado do teste, apenas pode ter dois valores, verdadeiro ou falso que identificam se o utilizador se encontra activo ou inactivo, obrigatório;

Cada sequencia de dígitos adicionada ao SI é armazenada na tabela sequencia_digitos, caracterizada por:

- id_sequencia – identificador único utilizado como chave primária, obrigatório;
- oe1 – dígito reproduzido em primeiro lugar no ouvido esquerdo, obrigatório;
- oe2 – dígito reproduzido em segundo lugar no ouvido direito, obrigatório;
- od1 – dígito reproduzido em primeiro lugar no ouvido direito, obrigatório;
- od2 – dígito reproduzido em segundo lugar no ouvido direito, obrigatório;
- caminho – localização física do ficheiro WAV que contem a sequencia, obrigatório;
- activo – estado da sequencia, apenas pode ter dois valores, verdadeiro ou falso que identificam se a sequência se encontra activa ou inactiva, obrigatório;

Um teste dicótico de dígitos além das suas características, que são armazenadas na tabela teste, contém sequências de dígitos, as sequências associadas aos testes são armazenadas na tabela sequencia_digitos, caracterizada por:

- id_linha – identificador único utilizado como chave primária, obrigatório;
- id_teste – identificador do teste a que a sequência se encontra associada, obrigatório;
- id_sequencia – identificador da sequência associada ao teste, obrigatório;

Cada sequência SSW adicionada ao SI é armazenada na tabela sequencia_SSW, caracterizada por:

- id_sequencia – identificador único utilizado como chave primária, obrigatório;
- oe1 – palavra reproduzido em primeiro lugar no ouvido esquerdo, obrigatório;
- oe2 – palavra reproduzido em segundo lugar no ouvido direito, obrigatório;
- od1 – palavra reproduzido em primeiro lugar no ouvido direito, obrigatório;
- od2 – palavra reproduzido em segundo lugar no ouvido direito, obrigatório;
- inicio – identificador do lado de reprodução da sequência, que pode tomar os valores esquerdo ou direito, obrigatório;
- caminho – localização física do ficheiro WAV que contém a sequência, obrigatório;
- activo – estado da sequência, apenas pode ter dois valores, verdadeiro ou falso que identificam se a sequência se encontra activa ou inactiva, obrigatório;

Um teste SSW além das suas características, que são armazenadas na tabela teste, contém sequências SSW, as sequências associadas aos testes são armazenadas na tabela sequencia_digitos, caracterizada por:

- id_linha – identificador único, utilizado como chave primária, obrigatório;
- id_teste – identificador do teste a que a sequência se encontra associada, obrigatório;
- id_sequencia_ssw – identificador da sequência associada ao teste, obrigatório;

Um dos requisitos funcionais do SI é a capacidade de armazenar o histórico de testes realizados pelos pacientes, estes dados são armazenados na tabela histórico_paciente que é caracterizada por:

- id_historico – identificador único, utilizado como chave primária, obrigatório;
- id_posto – identificador do posto onde foi realizado o teste, obrigatório;
- id_pac – identificador do paciente que realizou o teste, obrigatório;
- id_utilizador – identificador do utilizador que realizou o teste, obrigatório;
- id_teste – identificador do teste realizado pelo paciente, obrigatório;
- data – data da realização do teste, obrigatório;
- resultado – resultado do teste, obrigatório;

As respostas do teste dicótico de dígitos são armazenadas na tabela linha_historico_digitos que é caracterizada por:

- id_linha_historico – identificador único, utilizado como chave primária, obrigatório;
- id_historico – identificador do histórico associado à resposta, obrigatório;
- id_sequencia – identificador da sequência a que o Paciente respondeu, obrigatório;
- roe1 – estado da resposta do Paciente ao estímulo reproduzido em primeiro lugar no ouvido esquerdo, opcional;

- roe2 – estado da resposta do Paciente ao estímulo reproduzido em segundo lugar no ouvido esquerdo, opcional;
- rod1 – estado da resposta do Paciente ao estímulo reproduzido em primeiro lugar no ouvido direito, opcional;
- rod2 – estado da resposta do Paciente ao estímulo reproduzido em segundo lugar no ouvido direito, opcional;

As respostas do teste dicótico de dígitos são armazenadas na tabela *linha_histórico_SSW* que é caracterizada por:

- *id_linha_historico_SSW* – identificador único, utilizado como chave primária, obrigatório;
- *id_historico* – identificador do histórico associado à resposta, obrigatório;
- *id_sequenciaSSW* – identificador da sequência a que o Paciente respondeu, obrigatório;
- roe1 – estado da resposta do Paciente ao estímulo reproduzido em primeiro lugar no ouvido esquerdo, opcional;
- roe2 – estado da resposta do Paciente ao estímulo reproduzido em segundo lugar no ouvido esquerdo, opcional;
- rod1 – estado da resposta do Paciente ao estímulo reproduzido em primeiro lugar no ouvido direito, opcional;
- rod2 – estado da resposta do Paciente ao estímulo reproduzido em segundo lugar no ouvido direito, opcional;

O registo de um posto é efectuado na tabela *posto* que é caracterizada por:

- *id_posto* – identificador único utilizado como chave primária e como *serial number*, obrigatório;
- *descricao* – descrição do posto onde está instalada a aplicação, obrigatório;
- *localizacao* – localização do posto, opcional;

A calibração associada a cada posto é registada na tabela *calibração*, que é caracterizada por:

- *id_cal* – identificador único utilizado como chave primária, obrigatório;
- *id_posto* – identificador do posto, associado à calibração, obrigatório;
- *vol_cal* – volume de calibração, obrigatório;
- *data_cal* – data da calibração, obrigatório;